

R. Mikut, M. Reischl (Hrsg.)

Institut für Angewandte Informatik /  
Automatisierungstechnik



## **Proceedings**

# **15. Workshop Computational Intelligence**

Dortmund,  
16.-18. November 2005





R. Mikut, M. Reischl (Hrsg.)

**Proceedings 15. Workshop Computational Intelligence**

Dortmund, 16.-18. November 2005

Schriftenreihe des

Instituts für Angewandte Informatik / Automatisierungstechnik

an der Universität Karlsruhe (TH)

Band 9

# **Proceedings**

## **15. Workshop Computational Intelligence**

Dortmund, 16. -18. November 2005

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 2005  
Print on Demand

ISSN 1614-5267  
ISBN 3-937300-77-5

## VORWORT

Dieser Tagungsband enthält die Beiträge des 15. 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 16.-18. November 2005 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), 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

<b>R. Seising</b> <i>Medizinische Universität Wien:</i> 1965 - Zadeh Begins: „A new view on system theory“. Ein Beitrag zum 40. Jubiläum der Fuzzy Set Theorie	1
<b>Th. A. Runkler</b> <i>Siemens AG, München:</i> Clustering power law data	20
<b>F. Rehm, F. Klawonn, R. Kruse</b> <i>DLR Braunschweig, FH Braunschweig/Wolfenbüttel, Universität Magdeburg:</i> Visualizing single fuzzy c-means clusters	36
<b>Ch. Katz, Th. A. Runkler, K. Heesche</b> <i>Siemens AG, München:</i> Fuzzy clustering using similarity measures: Clustering relational data by object data methods	46
<b>N. Bauersfeld, K.-D. Kramer</b> <i>Hochschule Harz:</i> Methoden der Clusteranalyse für die Auswertung von Impedanzspek- tren eines selektiven Gassensors für Hochtemperaturanwendungen bis 900°C	59
<b>M. Hanss</b> <i>Universität Stuttgart:</i> Fuzzy arithmetic - a new approach to modeling and simulation of uncertain systems	74
<b>M. Sternke, P. Krause, H. Kiendl</b> <i>Universität Dortmund, Nutech Solutions:</i> Nutzung negativer Regeln in TSK-Fuzzy-Systemen	89
<b>T. Stolze, S. Braune, K.-D. Kramer</b> <i>Hochschule Harz:</i> Mikrocontrollerbasiertes Regelungssystem für Ventilaktuatoren mit Fuzzy-Control	104
<b>F. Rügheimer, R. Kruse</b> <i>Universität Magdeburg:</i> Datenanalyse-Plattform InformationMiner	117
<b>O. Burmeister, M. Reischl, R. Mikut</b> <i>Forschungszentrum Karlsruhe:</i> Zeitvariante Klassifikatoren für Brain Machine Interfaces und Neuroprothesen	129

<b>V. Gördes, K. Henning, M. Hühne, U. Lehmann, M. Pack</b> <i>FH Südwestfalen, Iserlohn:</i> SPAM-Abwehr auf Basis von Computational Intelligence (KNN und Bayes-Filter)	144
<b>H. Schulte</b> <i>Universität Kassel:</i> On fuzzy-descriptor systems and nonlinear model following control of two-link robot manipulators	146
<b>A. Lehmann, R. Mikut, T. Asfour</b> <i>Forschungszentrum Karlsruhe, Universität Karlsruhe:</i> Petri-Netze zur Aufgabenüberwachung in humanoiden Robotern	157
<b>H. Sommer</b> <i>Universität Kassel:</i> Zur Modellbildung kontextabhängiger Systeme	172
<b>J. Till, G. Sand, S. Engell</b> <i>Universität Dortmund:</i> Applied stochastic integer programming: hybrid evolution strategies vs. exact algorithms	182
<b>F. Hoffmann, D. Schauten</b> <i>Universität Dortmund:</i> Strukturelle Evolution von Fuzzy-Reglern am Beispiel des inversen Rotationspendels	197
<b>W. Jakob</b> <i>Forschungszentrum Karlsruhe:</i> Auf dem Weg zum industrietauglichen evolutionären Algorithmus	212
<b>R. Krohling, M. Sterneke, D. Schauten, F. Hoffmann</b> <i>Universität Dortmund:</i> Partikel Schwarm Optimierung eines Regelungssystems für ein industrielles Hydraulikventil	227
<b>T. Schmidt, D. Henrich</b> <i>Universität Bayreuth:</i> Inferenz mit Fuzzy-Zeit-Termen	241
<b>T. Förster, W. Kästner, R. Hampel, U.-S. Altmann, S. Grusla, M. Rubik, D. Haake, T. Brunne</b> <i>Hochschule Zittau/Görlitz (FH), Vattenfall Europe Powerconsultant GmbH, Vattenfall Europe Generation AG &amp; Co. KG:</i> Modellierung hochdimensionaler Zusammenhänge in einer Dampferzeugerfeuerung mittels künstlicher neuronaler Netze	253

# 1965 - Zadeh Begins: „A New View on System Theory“ Ein Beitrag zum 40. Jubiläum der Fuzzy Set Theorie.

**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  
e-Mail: Rudolf.Seising@meduniwien.ac.at

## 1 Vorbemerkungen

*Summer 2005 – Batman Begins:* Im vergangenen Sommer kam der Film *Batman Begins* in die Kinos. Nachdem in den 1990er Jahren einige schlechte Filme das Batman-Thema lächerlich gemacht hatten, erzählt *Batman Begins* wie es dazu kam, dass ein junger Mann Batman wurde.



*Summer 1965 – Zadeh Begins* soll auf die Entstehungsgeschichte der Fuzzy Set Theorie verweisen. Lotfi Zadeh war mit über 40 Jahren kein junger Mann mehr, als er damals die Theorie der Fuzzy Sets begründete, doch diese Theorie war noch sehr jung, als sie sich belächelt und bekämpft wurde. Heute gilt die Theorie der Fuzzy Sets als Theorie „normaler Wissenschaft“, die viele Anwendungsgebiete gefunden hat.

„A New View on System Theory“ war der Titel eines Vortrags, den Zadeh 1965 auf einem Symposium zur Systemtheorie hielt. Den Text für den später erschienen Konferenzband hatte er dann schon anders genannt: „Fuzzy Sets and Systems“. Der Titel des hier vorliegenden Beitrags soll betonen, dass Zadeh die Theorie der Fuzzy Sets und Fuzzy Systeme als eine verallgemeinerte Systemtheorie begründete.

## 2 Einführung

“Fuzzy Sets” war der erste Artikel über Fuzzy Sets, den Lotfi Zadeh in einer wissenschaftlichen Zeitschrift veröffentlichte. Er erschien in *Information and Control*, einer Zeitschrift, deren Herausbergremium Zadeh angehörte. Dies wirkte sich offenbar nicht verzögernd auf den Begutachtungsprozeß des Manuskripts aus, denn der am 30. November 1964 eingereichte Text wurde schon im Juni-Heft des Jahres 1965 publiziert. Zadeh präsentierte hier die Definition der Fuzzy Sets und ihre algebraischen Eigenschaften, er führte den Begriff der Fuzzy-Relation ein und in Erweiterung des klassischen Trennungstheorems für gewöhnliche Mengen eines für konvexe Fuzzy Sets.



Abb. 1: Lotfi A. Zadeh (geb. 1921) in Berkeley in den 1960-er Jahren

40 Jahre nach diesem völlig unerwarteten Auftritt ist es höchste Zeit, die Geschichte der Fuzzy Set Theorie nachzuvollziehen und zu interpretieren. Fuzzy Sets, Fuzzy Logik und ihre Anwendungen in Wissenschaft, Technik, Wirtschaft und Medizin sind historisch interessante Forschungsthemen geworden, und als solche sind ihre Entstehung und Entwicklung in die Wissenschafts- und Technikgeschichte des 20. Jahrhunderts einzubetten.

Die aus meiner Münchner Habilitationsschrift hervorgegangene Geschichte der Fuzzy Set Theorie ist vor wenigen Monaten erschienen [1]. Basierend auf einer Fülle Quellenmaterials wie Zeitschriftenartikeln, Konferenzbeiträgen, Briefen und vor allem vielen Interviews mit Pionieren der Fuzzy Set Theorie entstand eine verdichtete Geschichte der Fuzzy Set Theorie und ihrer ersten Anwendungen [2-6].

Die Theorie der Fuzzy Sets ist kein Resultat von Grundlagenforschungen in Mengentheorie, Logik oder Mathematikphilosophie, sondern das Ergebnis der Überlegungen eines mathematisch orientiert arbeitenden US-amerikanischen Elektroingenieurs: Lotfi Zadeh sah sich zur Mitte des 20. Jahrhunderts mit neuen Forschungsthemen für die Elektrotechnik konfrontiert: Informationstheorie, Kybernetik und Systemtheorie. In der vorliegenden Darstellung wird die Entstehung der Fuzzy Set Theorie und der Fuzzy Systeme aus dieser Sicht gedeutet.

### 3 Elektrotechnik, Systemtheorie, Informationstheorie – Aufbrüche und Differenzen in den 1950er Jahren

Lotfi Zadeh wurde 1921 als Sohn eines aserbaidischen Geschäftsmannes und Zeitungskorrespondenten und einer russischen Ärztin in Baku, der Hauptstadt der damaligen Sowjetrepublik Aserbaidschan geboren. Er ging in der iranischen Hauptstadt Teheran auf eine presbyterianische Schule, wo er auch englisch lernte, danach studierte er in Teheran Elektrotechnik, wo er 1942 den *Bachelor*-Grad (BA) erhielt. Da er für sich keine Möglichkeit sah, im Iran als Forscher zu arbeiten, emigrierte er 1944, nach einjähriger Tätigkeit für die United States Army Forces im Iran, in die Vereinigten Staaten, wo er einige Zeit bei den *International Electronics Laboratories* in New York City tätig war, dann aber sein Studium am Massachusetts Institute of Technology (MIT) in Boston fortsetzen konnte. Er besuchte Vorlesungen von Norbert Wiener, Robert Fano und Ernst Adolf Guillemin und schloß 1946 mit dem *Scientiae Magister*-Grad (SM) in Electrical Engineering das Studium ab.



Abb. 2: Lotfi A. Zadeh mit seinen Eltern in New York.

Als 1945 seine Eltern in die USA nach New York auswanderten, bemühte sich Zadeh um eine dortige Anstellung und wurde 1946 *Instructor* mit Lehrverpflichtung im *Department of Electrical Engineering* der Columbia University. Supervisor für sein Dissertationsvorhaben war der damalige Associate Professor John Ralph Ragazzini, und

1949 erhielt Zadeh den Ph. D. von der Columbia University für seine Arbeit *Frequency Analysis of Variable Networks*. [7]

Mit Ernst A. Guillemin, der ein begnadeter Lehrer für Elektrotechnik war, verband Zadeh nach seinem Wechsel vom MIT zur Columbia University das gemeinsame Engagement für eine intensivere mathematische Grundausbildung der US-amerikanischen Elektrotechnikstudenten. Beide gehörten auch dem Herausbergremium der *IRE Transactions on Circuit Theory* an, und als im September 1955 ein Themenheft über die Theorie der Fourier-Integrale erschien [8], hatte Guillemin der sich schon länger dafür eingesetzt hatte die Fourier-Mathematik im Lehrplan für Elektroingenieure festzuschreiben, eine Einführung in diese Thematik geschrieben [9], und Zadeh folgte mit einer Arbeit zur Verallgemeinerung des Fourier Integrals, die er gemeinsam mit dem Mathematiker Kenneth S. Miller verfasst hatte [10].



Ernst Adolph Guillemin

Völlig unterschiedlich waren allerdings die Ansichten von Professor Guillemin und Student Zadeh über das Verhältnis von Theorie und Praxis in der elektrotechnischen Netzwerktheorie, und schon diese Differenzen werfen ein frühes Licht auf Zadehs spätere Begründung der Fuzzy Set Theorie. Guillemin selbst hat diese Problematik sehr schön im Vorwort zu seinem 1952 erschienenen Lehrbuch *Introductory Circuit Theory* thematisiert, wobei er seine Leser mit Robert Louis Stevensons Roman *The Strange Case of Dr. Jekyll and Mr. Hyde* aus dem Jahre 1886 konfrontierte, denn Guillemin verdeutlichte anhand von „Dr. Jekyll und Mr. Hyde“ den „dualen Charakter der Netzwerktheorie“. Eine knappe Inhaltsangabe des Romans sei der Argumentation Guillemins vorausgeschickt:

Dr. Jekyll ist ein angesehener Arzt, der meint, „dass der Mensch in Wahrheit nicht eins sondern wahrlich zwei ist“. In seinem chemischen Versuchslabor entschließt er sich zum Selbstversuch, um seine beiden Naturen voneinander zu trennen: „Lange stand meine Tinktur bereit. Ich kaufte einst von einer chemischen Fabrik eine erhebliche Quantität eines besonderen Salzes, das, wie ich nach meinen Experimenten wußte, das letzte erforderliche Ingrediens darstellte, und spät, in einer verfluchten Nacht, mischte ich diese Elemente, überwachte ihr Kochen und Brodeln in der Retorte, und als das Aufwallen nachgelassen hatte, nahm ich meinen ganzen Mut zusammen und stürzte die Arznei hinunter.“ So entsteht sein alle schlechten Charaktereigenschaften verneinendes Alter Ego namens Edward Hyde, eine nächtens mordende Kreatur, von der Dr. Jekyll tagsüber bald zu ahnen beginnt, dass dies seine zweite Natur ist. Nach einiger Zeit geht die für die Rückverwandlung in Dr. Jekyll notwendige Medizin zur Neige: „Mein Vorrat an dem Salz, der seit der Zeit des ersten Versuches nie erneuert worden war, begann zu schwinden. Ich sandte nach frischem Ersatz und mischte den Trank. Das Aufwallen erfolgte und auch die erste Änderung der Färbung, doch nicht die zweite. Ich trank ihn, aber die Wirkung blieb aus.“ Nun gewinnt Mr. Hyde nach und nach der Oberhand über Dr. Jekyll und so bleibt diesem keine andere Lösung als der Selbstmord. In einem letzten Brief berichtet Dr. Jekyll über alle diese Geschehnisse und äußert darin auch die Vermutung, „daß mein ursprünglicher Vorrat unrein war, und daß es gerade diese unbekannte Verunreinigung war, die dem Trank seine Wirksamkeit verlieh.“[11] Für die exakt erforderliche Menge des Salzes konnte Dr. Jekyll weder Maß noch Zahl angeben, da er sie nie wußte, sondern hat mit unscharfen Mengen gerechnet und hantiert hatte. – Nun folgt die hier anknüpfende Passage in Guillemins Vorwort zu seinem Lehrbuch:

„One final point. In teaching of this subject I regard it as important to remind the student frequently that network theory has a dual character (no connection with the principle of duality); it is a Dr. Jekyll-Mr. Hyde sort of thing; it is two-faced, if you please. There are two aspects to this subject: the physical and the theoretical. The physical aspects are represented by Mr. Hyde – a smooth character who isn't what he seems to be and can't be trusted. The mathematical aspects are represented by Dr. Jekyll – a dependable, extremely precise individual who always responds according to established custom.

Dr. Jekyll is the network theory that we work with on paper, involving only pure elements and only the ones specifically included. Mr Hyde is the network theory we meet in the laboratory or in the field. He is always hiding parasitic elements under his jacket and pulling them out to spoil our fun at the wrong time. We can learn all about Dr. Jekyll's orderly habits in a reasonable period, but Mr. Hyde will continue to fool and confound us until the end of time. In order to be able to tackle him at all, we must first become well acquainted with Dr. Jekyll and his orderly ways. This book is almost wholly concerned with latter. I am content to leave Mr. Hyde to the boys in the laboratory.“[12]<sup>1</sup>

Somit war die von Guillemin gelehrte Netzwerktheorie eine Theorie perfekter Objekte, die sich von den Objekten der Wirklichkeit gravierend unterschied. Zadeh erinnerte sich an diese strikte, einseitige Haltung, die er nicht teilen mochte und konnte:

„Everything was idealized: resistors, capacitors, inductors, they were all perfect elements. So even at that point I had some discussions with him. I said this is unrealistic. The real world is not like that. I mean resistors are not pure resistors, capacitors are not pure capacitors, and so forth and I said, I told him that I think, that at some point in the future circuits will be designed and analysed using computers even when I was a student at MIT” [13]

Zwischen den Ansichten Guillemins und Zadehs darüber, was Netzwerktheorie sein und leisten sollte, bestanden unüberbrückbare Unterschiede. Sie sahen verschiedene Welten:“ ... because his world, it was not a world of mathematics, it was a world of real things, but it was an idealized world” [13] und Zadeh resümiert viel später:

„He was happy with that world ... He constructed the world by himself. It was a perfect world. Everything was perfect in that. He was happy and so

---

<sup>1</sup> „Ein letzter Punkt. Als Lehrender betrachte ich es als wichtig, die Studenten häufig daran zu erinnern, daß die Netzwerktheorie einen dualen Charakter hat. Es handelt sich um eine Art Dr. Jekyll-Mr. Hyde Sache, sie hat zwei Gesichter, wenn Sie wollen. Es gibt zwei Aspekte, den physikalischen und den theoretischen: der physikalische Aspekt wird von Mr. Hyde repräsentiert, ein glatter Charakter, der nicht ist, was er zu sein scheint und dem man nicht trauen kann. Die mathematischen Aspekte werden von Dr. Jekyll repräsentiert – ein vertrauenswürdiger, extrem präziser Mensch, der immer der Situation entsprechend reagiert.

Dr. Jekyll ist die Netzwerktheorie, die wir auf dem Papier erarbeiten, die nur reine Elemente berücksichtigt, und nur diese spezifisch enthält. Mr. Hyde ist die Netzwerktheorie, die wir im Labor oder draußen im Feld antreffen (Anwendungsbereichen). Er hält immer parasitäre Elemente unter seinem Jacket verborgen, die er hervorholt, um uns zur falschen Zeit den Spaß zu verderben. Wir können alles über Dr. Jekylls gesittete Gewohnheiten in einer sinnvollen Zeitspanne lernen, aber Mr. Hyde wird es ad absurdum führen und uns am Ende verwirren. Um in der Lage zu sein, ihn dennoch zu fassen, müssen wir uns zunächst mit Dr. Jekyll und seinen ordentlichen Wegen vertraut machen. Dieses Buch befaßt sich fast gänzlich damit. Ich bin gewillt, Mr. Hyde den „Jungs“ im Labor zur überlassen.“ (Übersetzung R.S.)

he never considered noise, he never considered non-linearities, he never considered imprecision; he never considered those things, so it was an idealized world. He was happy.” [13]

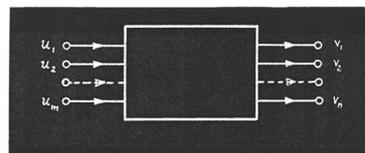
Es gab Guillemins Welt der realen jedoch idealisierten Kondensatoren, Spulen und Widerstände nicht wirklich, und die „Jungs im Labor“ mussten damit zurecht kommen, dass die von ihnen geschalteten Systeme Nichtlinearitäten, Ungenauigkeiten und Rauschen zeigten. In dieser Welt war Zadeh zu Hause; er hatte sich insbesondere mit nicht-linearen Systemen in der Elektrotechnik beschäftigt, und als in den 50-er Jahren die Systemtheorie als eine neue wissenschaftliche Disziplin entstand, in der es um die allgemeinen Prinzipien zur Charakterisierung von Input-Output-Beziehungen ging, wurde Zadeh einer ihrer Pioniere. Für die New Yorker Studentenzeitschrift *Columbia Engineering Quarterly* schrieb er 1954 den Beitrag *System Theory*, mit dem er eine leicht verständliche Einführung geben wollte:

„If you have never heard of system theory, you need not feel like an ignoramus. It is not one of the well-established branches of science. In fact, it has not yet been officially recognized as a scientific discipline. It does not appear on programmes of meetings of scientific societies nor in indices to scientific publications. It does not have well-defined boundaries, nor does it have settled objectives.“ [14]

Ein System sei – und hier zitierte er *Webster’s Dictionary* - „an aggregation or assemblage of objects united by some form of interaction or interdependence.“ [14] Als Beispiele für solche Systeme nannte er Körper, die sich gegenseitig anziehen, eine Gruppe von Menschen, die eine Gesellschaft bilden, einen Komplex miteinander verwobener Industriezweige, elektrische Netzwerke und Digitalcomputer.

Gleich zu Beginn des Artikels prangt zwischen Überschrift und Textbeginn die Darstellung eines Systems als “Blackbox” mit Inputs und Outputs (Abb.3), und im Text betonte Zadeh, dass sich im Falle der Darstellungsmöglichkeit von Inputs und Outputs als zeitabhängige Funktionen  $u_1, \dots, u_m$ , bzw.  $v_1, \dots, v_n$ , ( $m, n \in \mathbb{N}$ ) das dynamische Verhalten des Systems aus der Input-Output-Beziehung  $(v_1, \dots, v_n) = f(u_1, \dots, u_m)$  ergibt. Auf diese Weise wurde die klassische Theorie der elektrischen Schaltungen und Netzwerke in die Systemtheorie eingebettet.

## System Theory



L. A. Zadeh  
Associate Professor  
Electrical Engineering

Abb.3: Illustration zum Zadehs Artikel von 1954 [14].

In New York wurde Zadeh bald mit den im Krieg entwickelten Digitalcomputern und mit der Informationstheorie konfrontiert. Schon als er 1946 an die Columbia University gekommen war, hatte er die Gelegenheiten Claude Elwood Shannon und Norbert Wiener über ihre im Zweiten Weltkrieg der Geheimhaltung unterlegenen Forschungsergebnisse vortragen zu hören, die sie später ungeheuer erfolgreich publizierten [15, 16]. Zadeh war von der neuen Informationstheorie begeistert, moderierte 1949 an der Columbia University die weltweit erste Debatte über Digitalcomputer, an der neben Shannon auch Edmund C. Berkeley, der Autor des Buchs *Giant Brains* [17] teilnahm, und als sich in den 50er Jahren innerhalb des traditionsreichen I.R.E. (*Institute of Radio Engineers*) „Professional Groups“ mit eigenen „Transactions“ bildeten, gehörte Zadeh neben Shan-

non, Wiener und anderen zu jener Gruppe, die ab 1953 die *I.R.E. Transactions on Information Theory* herausgab.

1956 schrieben Shannon und Wiener in ihrer Eigenschaft als Mitglied des editorial board der *IRE Transactions on Information Theory* jeweils ein Editorial. Shannon begann im März-Heft mit einem Aufruf, unter der Überschrift *The Bandwagon*, nach aller Popularität, die der Informationstheorie in den vergangenen Jahren zukam, nun wieder zu ernsthafter Forschung und Entwicklung auf höchster wissenschaftlicher Ebene zurückzukehren. Der große Erfolg der Informationstheorie auf so vielen Gebieten – informationstheoretische Begriffe wie Information, Entropie und Redundanz wurden mittlerweile auch in Psychologie, Wirtschafts- und Sozialwissenschaften verwendet – sei erfreulich, er verwische aber ihre abstrakte Bedeutung: „Indeed, the hard core of information theory is, essentially, a branch of mathematics, a strictly deductive system.“ Die potentiellen Autoren der Zeitschrift animierte er daher, nur die klarsten und besten Leistungen zu veröffentlichen: „Research rather than exposition is the keynote, and our critical thresholds should be raised.“ [18]



Abb. 4.: Drei Mitglieder des Editorial Board der *IRE Transactions on Information Theory*: Claude E. Shannon, Norbert Wiener, Lotfi A. Zadeh.

Die interdisziplinäre Ausweitung seiner mathematischen Theorie der Kommunikation, wie Shannon die Informationstheorie zunächst genannt hatte, gefiel ihm nicht; und damit wandte er sich auch offen gegen Wiener, der sich im Übrigen ebenfalls als Begründer der Informationstheorie verstand. Wiener nutzte im Juni 1956 die Gelegenheit, Shannons Angriff zu kontern. Auch er plädierte für ein „Zurück zu den Wurzeln!“, doch er interpretierte dies völlig anders! *What is Information Theory?* hatte er sein Editorial überschrieben, und nachdem er den Leser darüber informiert hatte, dass Shannon und er die Begründer der Informationstheorie sind, stimmte er Shannon zu, dass die Informationstheorie kein „magic key“ sei, doch dann schrieb er:

„I am pleading in this editorial that Information Theory go[es] back of its slogans and return to the point of view from which it originated: that of the general statistical concept of communication.“ [19]

Mit Hinweisen auf den statistischen Charakter der Mechanik nach Willard Gibbs und der die gesamte Physik dominierenden Quantentheorie stellte Wiener auch die Informationstheorie in diesen Gesamtzusammenhang:

„What I am here entreating is that communication theory be studied as one item in an entire context of related theories of a statistical nature, and that it should not lose its integrity by becoming a special vested interest attached to a certain set of slogans and chilchés.“ [19]

Schließlich sprach sich Wiener ganz deutlich gegen die drei Monate zuvor von Shannon geäußerte Ansicht aus:

„I hope that these TRANSACTIONS may encourage this integrated view of communication theory by extending its hospitality to papers which, why they bear on communication theory, cross its boundaries, and have a scope covering the related statistical theories. In my opinion we are in a dangerous age of overspecialization. [19]

Die von Shannon geforderte Besinnung auf den Kern der Informationstheorie führe in eine wissenschaftliche Enge; es sei wünschenswerter, die statistisch fundierte Informationstheorie möglichst vielen wissenschaftlichen Bereichen zugrunde zu legen.

Als „step in the direction of a broad theory of communication, as contemplated by Wiener“ bezeichneten die bei der RAND Corporation in Santa Monica beschäftigten Mathematiker Richard E. Bellman und seinen Mitarbeiter Robert E. Kalaba [20] ihren Artikel *On the Role of Dynamic Programming in Statistical Communication Theory*, der im darauf folgenden Jahr in derselben Zeitschrift erschien: Sie verwiesen explizit auf Wieners Editorial, als sie ein einfaches Modell des allgemeinen Kommunikationsproblems betrachteten, in dem die Funktion des Kommunikationskanals als mathematische Transformation dargestellt wurde:

Eine Quelle  $S$  produziert zu diskreten Zeitpunkten Folgeglieder reiner Signale  $x$  mit (stochastischem oder deterministischem) Rauschen  $r$ . Das kombinierte Signal  $x' = F(x, r)$  ist der Input in den Kommunikations-

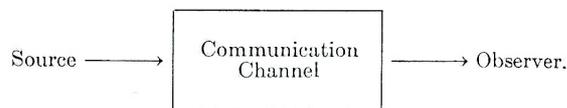


Abb. 5: Blackbox Kommunikationssystem Bellman und Kalaba [20].

kanal, der ansonsten als Blackbox anzusehen ist, und von dem ein Output-Signal  $y$  emittiert wird (Abb. 5). Von der Beobachtung dieses Outputs ist dann auf die Eigenschaften des ursprünglich reinen Signals  $x$  zu schließen.

Die Transformation  $T$  repräsentiert die Übertragung des Inputsignals  $x'$  durch den Kommunikationskanal, es gilt also:  $y = T(x') = T(F(x, r))$ .

Die Menge aller Kommunikationssysteme entspricht somit einer entsprechenden Menge von Transformationen  $T$ , und führt man eine Ordnung auf dieser Menge ein, so lassen sich mit deren Hilfe Kommunikationssysteme vergleichen bzw. ihre Leistung bewerten, und so sollten dann auch optimale Kommunikationssysteme gefunden werden.

#### 4 Superiority, Inferiority, Optimality, Noninferiority

*What Is Optimal?* war die Überschrift des Editorials von Lotfi Zadeh, das im März 1958 in den *IRE Transactions on Information Theory* erschien und zu Beginn fragte er die Leser, wie vernünftig es eigentlich sei, dass wir optimale Lösungen des Systemverhaltens suchen. Erst seit Wieners Arbeiten über elektrische Filter und seiner Vorhersage-theorie habe sich dieses Streben etabliert. Vielleicht sei man nun aber auf dem besten Wege, „to make a fetish of optimality. If a system is not “best” in one sense or another, we do not feel satisfied. Indeed, we are apt to place too much confidence in a system that is, in effect, optimal by definition.“ [21]

Ein optimales System zu finden, das bedeute zunächst ein Leistungskriterium zu wählen, dann eine Klasse akzeptabler Systeme entsprechend verschiedener Bedingungen hinsichtlich des Designs, der Kosten, etc. zu spezifizieren und schließlich eines dieser Systeme aus der spezifizierten Klasse als das „beste“ hinsichtlich dieser Kriterien anzunehmen. Zadeh bezweifelte, dass dieser Weg vernünftiger sei als der „relatively un-

sophisticated approach of the pre-Wiener era“: [21] Durch die Wahl eines einzigen Leistungskriteriums blieben alle anderen Kriterien, die ebenfalls einen Beitrag zur Leistungsbewertung liefern könnten, unberücksichtigt. Das Problem werde zu sehr vereinfacht, wenn skalare Funktionen zur Darstellung der Systemleistung dienten, vektorwertige Funktionen wären dem Problem angemessener allerdings dann auch schwieriger zu handhaben.

Auf allgemeiner Stufe kritisierte Zadeh das Vorgehen zur rationalen Wahl von Entscheidungsfunktionen bei Unsicherheit: „Was ist zu tun, wenn die Wahrscheinlichkeiten für den „Zustand der Natur“, der ein Problem kennzeichnet, nicht bekannt sind?“ – Zadeh verwarf hier die üblichen auf der Stochastik bzw. der Spieltheorie gründenden Lösungsmethoden:

„At present no completely satisfactory rule for selecting decision functions is available, and it is not very likely that one will be found in the foreseeable future. Perhaps all that we can reasonably expect is a rule which, in a somewhat equivocal manner, would delimit a set of “god” designs for a system. In any case, neither Wiener’s theory nor the more sophisticated approaches of decision theory have resolved the basic problem of how to find a “best” or even a “good” system under uncertainty.“ [21]

Dieses Plädoyer ist ein frühes Zeugnis für Zadehs Zweifel an der traditionellen Mathematik, die bisher doch weitestgehend unangefochten als Werkzeug zur Erfassung realer Systeme angesehen wurde. Noch zögerte Zadeh allerdings, sich vollends abzuwenden, noch suchte er nach Auswegen innerhalb dieser traditionellen Mathematik, doch das waren Rückzugsgefechte!

1963 erschien ein Korrespondenzbeitrag von Zadeh in den *IEEE Transactions on Automatic Control*. Er trägt den Titel *Optimality and Non-Scalar-Valued Performance Criteria* [20] und er schließt nahtlos an *What is Optimal?* an, denn Zadeh kritisierte wieder die skalarwertigen Systemleistungsgrößen. Es sei eine ernsthafte Schwäche damaliger Theorien zur *Optimalen Steuerung*, dass sie von der Annahme ausgingen, die Leistung eines Systems  $S$  könne durch eine einzige Zahl, den reellwertigen Leistungsindex  $P(S)$ , gemessen werden, und ein System  $S_0$  heiße dann optimal in einer Klasse (Menge)  $\Sigma$  von Systemen, wenn gilt:  $P(S_0) \geq P(S)$  für alle  $S$  in  $\Sigma$ .

Da es im Allgemeinen mehr als eine Betrachtungsweise gebe, die Leistung eines Systems  $S$  zu bewerten und in den meisten Fällen diese Betrachtungsweisen nicht zu einem einzigen skalarwertigen Kriterium zusammengefasst werden können, begannen damit die Schwierigkeiten, den Optimalitätsbegriff vernünftig zu fassen. Ein System  $S$  könne ja in einer Hinsicht durchaus besser (*superior*) gegenüber einem System  $S'$  sein, in einer anderen Hinsicht allerdings schlechter (*inferior*). Die Menge  $\Sigma$  aller betrachteten Systeme sei daher zwar partiell, nicht aber vollständig geordnet. Zadeh schlug nun vor, die Optimalität von einer anderen Systemeigenschaft zu unterscheiden, die er *Noninferiority* nannte.

Eine Teilmenge (*constraint set*)  $C$  von  $\Sigma$  sei durch Einschränkungen an das System  $S$  definiert, und auf  $\Sigma$  sei eine partielle Ordnung „ $\succeq$ “ definiert, wodurch jedem System  $S$  in  $\Sigma$  die folgenden drei disjunkten Teilmengen von  $\Sigma$  zugeordnet werden können:

- 1) Die Teilmenge  $\Sigma_{\succ}(S)$  aller Systeme, die besser als  $S$  sind (*superior*).
- 2) Die Teilmenge  $\Sigma_{\preceq}(S)$  aller Systeme, die schlechter oder gleich  $S$  sind (*inferior*).
- 3) Die Teilmenge  $\Sigma_{\sim}(S)$  aller Systeme, die mit  $S$  nicht vergleichbar sind.

Da jedes System ein Element in  $\Sigma$  ist, fällt jedes System in eine dieser Kategorien und die Vereinigung von  $\Sigma_>(S)$ ,  $\Sigma_<(S)$  und  $\Sigma_=(S)$  ist  $\Sigma$ , die Menge aller betrachteten Systeme. Mit Hilfe dieser Teilmengen von  $\Sigma$  definiert Zadeh die Systemeigenschaften *Nichtinferiorität* und *Optimalität*:

*Definition 1:* Ein System  $S_0$  ist in  $C$  *nichtinferior*, wenn die Schnittmenge von  $C$  und  $\Sigma_>(S_0)$  leer ist. Es gilt also:  $C \cap \Sigma_>(S_0) = \emptyset$

Es gibt somit kein System in  $C$ , das besser als  $S_0$  ist. Äquivalent dazu kann man auch sagen:  $S_0$  ist nicht schlechter (*inferior*) als die anderen Systeme in  $C$ .

*Definition 2:* Ein System  $S_0$  ist in  $C$  *optimal*, wenn  $C$  in  $\Sigma_<(S_0)$  enthalten ist. Es gilt also:  $C \subseteq \Sigma_<(S_0)$

Jedes System in  $C$  ist somit schlechter (*inferior*) als  $S_0$  oder gleich  $S_0$ .

Die Definitionen der Mengen  $\Sigma_>(S_0)$  und  $\Sigma_<(S_0)$  machen unmittelbar einsichtig, dass ein optimales  $S_0$  notwendigerweise *nichtinferior* ist, die Umkehrung aber nicht gelten muss. Bei einer vollständigen Ordnung der Menge  $\Sigma$  aller betrachteten Systeme durch ein skalares Kriterium ist  $\Sigma_=(S_0)$  die leere Menge, während  $\Sigma_>(S_0)$  und  $\Sigma_<(S_0)$  zueinander komplementäre Mengen sind. Wenn der Durchschnitt von  $C$  und  $\Sigma_>(S_0)$  leer ist, enthält  $\Sigma_<(S_0)$  die Menge  $C$  sicher. Somit sind *Nichtinferiorität* und *Optimalität* in diesem Falle äquivalent und der Unterschied zwischen diesen Begriffen ist bei skalarwertigen Kriterien nicht erkennbar.

Zadeh schlug nun eine partielle Ordnung der Menge  $\Sigma$  vor, die durch ein vektorwertiges Leistungskriterium berücksichtigt wird: Das System  $S$  sei durch einen Vektor  $x = (x_1, \dots, x_n)$  charakterisiert, dessen reellwertige Komponenten z. B. die Werte von  $n$  veränderlichen Parametern des Systems  $S$  sind, und  $C$  sei eine Teilmenge des  $n$ -dimensionalen Euklidischen Raumes. Die Leistung des Systems  $S$  werde durch einen  $m$ -dimensionalen Vektor  $p(x) = [p_1(x), \dots, p_m(x)]$  gemessen, wobei jedes  $p_i(x)$ ,  $i = 1, \dots, m$ , eine reellwertige Funktion von  $x$  ist.

Es gilt nun  $S \geq S'$  genau dann, wenn  $p(x) \geq p(x')$ . D. h. also:  $p_i(x) \geq p_i(x')$ ,  $i = 1, \dots, m$ .

Zur Illustration einiger Aussagen über nichtinferiore Systeme betrachtete Zadeh den Fall, in dem  $\Sigma_>(S)$  oder äquivalent dazu  $\Sigma_>(x)$  ein fester Kegel mit Scheitelpunkt bei  $x$  und die Constraint-Menge  $C$  eine abgeschlossene beschränkte Teilmenge im Vektorraum ist (Abb. 6).

Dieser Fall tritt z. B. dann ein, wenn  $p_i(x)$ ,  $i = 1, \dots, m$  von der Form ist:  $p_1(x) = a_1^i x_1 + \dots + a_n x_n^i$ , wobei  $a^i = (a_1^i, \dots, a_n^i)$  ein konstanter Vektor, nämlich der Gradient von  $p_i(x)$  ist, d.h.  $a^i = \text{grad } p_i(x)$ . In diesem Falle ist  $\Sigma_>(x)$  der Polarkegel des Kegels, der durch die  $a^i$  aufgespannt wird.

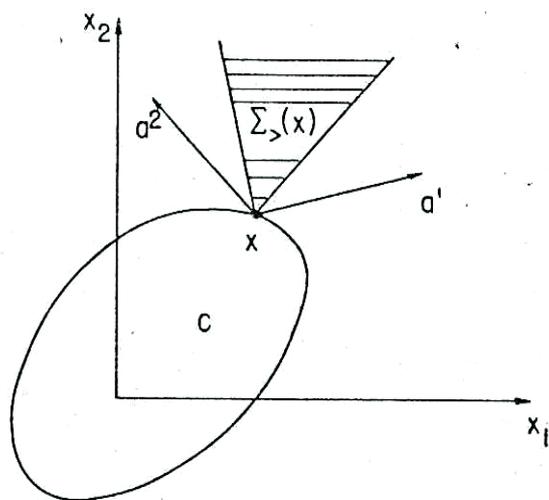


Abb. 6: Illustration der Bedeutungen der beiden Mengen  $C$  und  $\Sigma_>(x)$ . ([22], p. 59).

Nach Definition 1 kann dann ein *nichtinferiorer* Punkt kein Punkt im Inneren der Menge  $C$  sein kann. Wenn  $C$  darüber hinaus eine *konvexe* Menge ist, dann ist die Menge aller nichtinferioren Punkte auf dem Rand von  $C$  die Menge  $\Gamma$  aller Punkte  $x_0$ , durch die Hyperebenen verlaufen können, die Menge  $C$  und Menge  $\Sigma_{>}(x_0)$  voneinander trennen, separieren. (In Abb. 7 ist die Menge  $\Gamma$  die stark gezeichnete Linie auf dem Rand von  $C$ .) Ist  $x_0$  ein solcher Punkt und ist  $\gamma$  die (vom Inneren der Menge  $C$  wegzeigende) Normale auf der Hyperebene in diesem Punkt  $x_0$ , dann gehört  $\gamma$  zum Polarkegel der Menge  $\Sigma_{>}(x_0)$ , denn  $\gamma$  bildet keinen stumpfen Winkel mit den Vektoren in  $\Sigma_{>}(x_0)$ .<sup>2</sup>

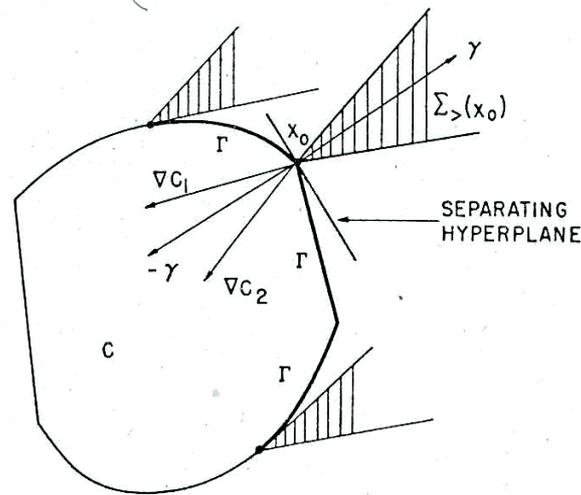


Abb. 7: Menge der nichtinferioren Punkte auf dem Rand der Menge  $C$  ([22], p. 59).

## 5 Mustererkennung

Zadeh hatte bei der im letzten Abschnitt referierten Untersuchung nicht ohne Grund die Konvexität der Menge  $C$  gefordert, denn für konvexe Mengen  $A_1$  und  $A_2$  gilt, dass wenn sie disjunkt sind, sie sich durch eine Hyperebene – im zweidimensionalen Fall ist dies eine Gerade (siehe Abb. 6b) – trennen lassen. Zu jedem Punkt lässt sich somit eindeutig festlegen, ob er zu Menge  $A_1$  oder  $A_2$  gehört. Die Problematik dieser „linearen Separierbarkeit“ von Mengen war in den 1950-er und 1960-er Jahren von großem Interesse für die Forschung zur Musterklassifikation (pattern classification), denn mathematisch kann jedes Muster durch eine Punktmenge im  $n$ -dimensionalen reellen Vektorraum  $\mathbf{R}^n$  repräsentiert werden. Lässt sich dieser Raum  $\mathbf{R}^n$  in Mengen unterteilen, von denen jeder nur einen Punkt einer bestimmten Mustermenge enthält und keinen einer anderen Mustermenge, dann ist das Problem der Musterklassifikation gelöst, denn jeder seiner Punkte lässt sich einer Menge in  $\mathbf{R}^n$  eindeutig zuordnen.

Als der Elektroingenieur George Nagy 1968 in den *Proceedings of the IEEE* auf 26 Seiten den *State of the Art in Pattern Recognition* beschrieb [21], charakterisierte er sie als ein Sammelbecken ganz verschiedenartiger Probleme, mit denen sich Wissenschaftler unterschiedlicher Disziplinen beschäftigten. Dazu zählten statistische Entscheidungstheorie, Schaltungstheorie, Automatentheorie, Mengentheorie, Steuerungs- und Regelungstheorie, Sprachanalyse, Informationstheorie, Mathematische Programmierung

<sup>2</sup> Interessant ist eine Fußnote zu diesem Text ([22], Fn. 2), in der Zadeh seinen Begriff der *Nichtinferiorität* erläutert: Es gebe den ganz ähnlichen Begriff der „admissibility“ (Zulässigkeit) in der Statistik, der in Anlehnung an Wald [23] geprägt wurde, aber „We use the term „noninferior“ because the terms admissible and efficient have become too fuzzy through varying usage in different contexts. (An equivalent mathematical term is „maximal“.)“ Das von Zadeh hier benutzte Wort „fuzzy“ verweist zwar nicht direkt auf die von ihm zwei Jahre später eingeführte Theorie der Fuzzy Sets, ist aber dennoch beachtenswert. „You remind me something I forgot. ... But I used that word in a nontechnical sense.“ sagte Zadeh, als er im Interview damit konfrontiert wurde [12].

und auch die „nerve net studies“ von Frank Rosenblatt, mit dem Nagy 1962 und 1963 an einem Audi-Perceptron gearbeitet hatte. Ein Perceptron sollte die Aufgabe der Klassifizierung von Merkmalsmustern (*pattern classification*), wie z. B. handgeschriebenen Schriftzeichen meistern. Muster (*pattern*) repräsentieren Objekte der Realität, meist visuelle Bildpunkte; es konnten aber auch Frequenzmuster, die ein sprachliches Zeichen, einen Laut repräsentieren, als Muster bezeichnet werden. Das erste Perceptron – eine Maschine, die visuelle Muster klassifizieren konnte – hatte Rosenblatt fünf Jahre früher mit Charles Wightman an der *Cornell University* entwickelt. Dieses *Mark I Perceptron* genannte erste künstliche neuronale Netzwerk hatte einen 20×20 Pixel großen Bildsensor, konnte einfache Ziffern erkennen und hatte 512 mit einem Motor angetriebene Potentiometer, die für jedes der variablen Verbindungsgewichte zuständig waren. Rosenblatt beschrieb das Perceptron 1958 in einem Aufsatz für den *Psychological Review* [25] als das erste lernfähige Modell eines neuronalen Netzes, von dem gezeigt werden konnte, dass der vorgeschlagene Lernalgorithmus stets zum Erfolg führte, wenn das Problem überhaupt eine Lösung hatte. Zwar wiesen Marvin Minsky und Seymour Papert 1969 nach, dass Rosenblatts Perceptron das mathematische Problem der linearen Separierbarkeit prinzipiell nicht lösen konnte [26], doch wurde bis zu diesem Zeitpunkt fieberhaft zur maschinellen Mustererkennung geforscht, und in diesem Rahmen ist auch Zadehs Engagement zu sehen.

## 6 Von Systemen und ihren Zuständen zu Fuzzy Systemen

Schon seit 1958 wollte Zadeh ein Buch über lineare Systemtheorie schreiben, ein Vorhaben, das er dann erst 1963 gemeinsam mit seinem Kollegen Charles A. Desoer in Berkeley verwirklichte [27]: *Linear System Theory: The State Space Approach* heißt dieses Buch, für das Zadeh in den ersten vier Kapiteln eine Einführung in den auch im Buchtitel genannten *State Space Approach* gab, für die er auf das Konzept der Turingmaschine zurückgriff:

„Roughly speaking, a Turing machine is a discrete time ( $t = 0, 1, 2, \dots$ ) system with a finite number of states or internal configurations, which is subjected to an input having the form of a sequence of symbols (drawn from a finite alphabet) printed on a tape which can move in both directions along its length. The output of the machine at time  $t$  is an instruction to print a particular symbol in the square scanned by the machine at time  $t$  and to move in one or the other direction by one square. A key feature of the machine is that the output at time  $t+1$  and the state at time  $t+1$  are determined by the state and the input at time  $t$ . ([28], S. 858.)

Mit den Bezeichnungen  $s_t$  für den Zustand (state),  $u_t$  für den Input und  $y_t$  für den Output jeweils zum Zeitpunkt  $t$  lassen sich die Operationen einer Turingmaschine dann folgendermaßen schreiben:

$$s_{t+1} = f(s_t, u_t), \quad y_t = g(s_t, u_t), \quad t = 0, 1, 2, \dots,$$

wobei  $f$  und  $g$  Funktionen von Paaren der Variablen  $s_t$  und  $u_t$  sind.

Zadeh verwies darauf, dass diese „Turing-Darstellung“ bzw. „Zustandsgleichungen eines Systems“ schon von Shannon in seinem „epoch-making paper on the mathematical theory of communication“ verwendet wurden [15], als dieser Kanäle mit Rauschen darstellte. Dort waren aber mit  $s_t$  und  $u_t$  nicht  $s_{t+1}$  und  $u_{t+1}$  sondern deren Wahrscheinlichkeitsverteilungen festgelegt worden. Dann nannte Zadeh die Arbeiten John von Neumanns zur Automatentheorie, und er brachte diesen Zustandsbegriff in das Gebiet der

Differentialgleichungssysteme ein, wo die Zustandsgleichungen folgende Form annehmen:

$$\dot{s}(t) = f(s(t), u(t)) \quad \text{und} \quad y(t) = g(s(t), u(t)), \quad \text{mit} \quad \dot{s}(t) = d/dt s(t).$$

$s(t)$ ,  $y(t)$ ,  $u(t)$  sind Vektoren, die Zustand, Input und Output des Systems zum Zeitpunkt  $t$  darstellen. Diese Begriffe seien in den USA erst seit der Publikation von Richard Bellmans *Dynamic Programming* [29] bekannt, schrieb Zadeh, in der Sowjetunion allerdings schon seit den 1940er Jahren im Bereich der *Automatischen Steuerung* benutzt worden.<sup>3</sup>

Im April 1963 trug Zadeh beim *Second Systems Symposium* am *Case Institute of Technology* vor mehr als 200 Teilnehmern über seinen State-Space-Approach vor; es handelte sich um eine interdisziplinäre Veranstaltung, bei der etwa die Hälfte der Vortragenden ausgewiesene Vertreter aus der *Kybernetik* bzw. der *Allgemeinen Systemtheorie* kamen.<sup>4</sup> Zu den Pionieren dieser *Allgemeinen Systemtheorie* gehörte Kenneth E. Boulding<sup>5</sup>, der sich während mancher Präsentation zu kleinen Gedichten inspirieren ließ, die im Konferenzband, der im darauffolgenden Jahr unter dem Titel *Views on General Systems Theory* publiziert wurde, die Beiträge einleitend abgedruckt wurden. Zadehs Beitrag *The Concept of State in System Theory* wurde von Boulding folgendermaßen charakterisiert:

A System is a big black box  
Of which we can't unlock the locks,  
And all we can find out about  
Is what goes in and what goes out.  
Perceiving input-output pairs,  
Related by parameters,

Permits us, sometimes, to relate  
An input, output, and a state.  
If this relation's good and stable  
Then to predict we may be able,  
But if this fails us – heaven forbid!  
We'll be compelled to force the lid!

K. B.

Als Zadeh 1962 für die Jubiläumsausgabe der *Proceedings of the IRE* zum fünfzigjährigen Bestehen des Institute of Radio Engineers den Artikel „From Circuit Theory to System Theory“ [28] schrieb, verarbeitete er auch einige Absätze seines früheren, in der Studentenzeitung erschienenen Artikels [14]. Außerdem behandelte er Probleme und Anwendungen der Systemtheorie und ihrer Beziehungen zur Informations- und Regelungstheorie, vor allem aber betonte er, dass derselbe abstrakte Systembegriff in vielerlei Gestalten in der Wissenschaft Verwendung finde.

“Thus, whether a system is electrical, mechanical or chemical in nature does not matter to a system theorist. What matters are the mathematical relations between the variables in terms of which the behaviour of the system is described” ([28], S. 856).

Er hob die großen Fortschritte der Systemtheorie hervor, die einerseits auf den wissenschaftlich-technischen Erfolgen hochkomplexer Systeme in den Gebieten automatischer Regelung, Mustererkennung, Datenverarbeitung, Kommunikation und maschinelles

<sup>3</sup> Zadeh nannte Anatolij I. Lur'e, Mark A. Aizerman, Alexandr M. Letov, Nikolai N. Krasovskii, Iol G. Malkin und Lev S. Pontryagin.

<sup>4</sup> Neben Zadeh trugen hier vor: Mihaljo D. Mesarović, Kenneth E. Boulding, Russell L. Ackoff, Abraham Charnes und William W. Cooper, Hilary Putnam, John Myhill, R.W. Gerard, William K. Linvill, Robert Kalaba, Rudolf F. Drenick, W. Ross Ashby, Anatol Rapoport und C. West Churchman. [30].

<sup>5</sup> Der Wirtschaftswissenschaftler Kenneth Boulding war im Jahre 1954 eines der Gründungsmitglieder der *Society for General Systems Research*.

Rechnen basierten und andererseits in den Ansätzen zur quantitativen Analyse der extrem komplexen lebendigen sowie der Mensch-Maschine-Systeme, die in den Lebenswissenschaften, dem Operations Research und anderen Disziplinen zu sehen seien. ([28], S. 856f.). Er führte auch seinen neuen Zugang zur Systemtheorie über den Begriff des Zustands aus. Doch auch in anderer Hinsicht bot dieser Text Neues: Zadeh betonte ausdrücklich, die Nähe seiner systemtheoretischen Überlegungen zu den Jahrzehnte älteren Betrachtungen von Ludwig von Bertalanffy, dem Begründer der *Allgemeinen Systemtheorie*. Dann lenkte er die Aufmerksamkeit der Leser auf Unterschiede zwischen mathematischer und nichtmathematischer Systembehandlung und zwischen belebten und nichtbelebten Systemen, und in diesem Zusammenhang benutzte er erstmals das Wort “fuzzy”, als vage Kennzeichnung einer “neuen Mathematik”:

“In fact, there is a fairly wide gap between what might be regarded as “animate” system theorists and “inanimate” system theorists at the present time, and it is not at all certain that this gap will be narrowed, much less closed, in the near future. There are some who feel that this gap reflects the fundamental inadequacy of the conventional mathematics – the mathematics of precisely-defined points, functions, sets, probability measures, etc. – for coping with the analysis of biological systems, and that to deal effectively with such systems, which are generally orders of magnitude more complex than man-made systems, we need a radically different kind of mathematics, the mathematics of fuzzy or cloudy quantities which are not describable in terms of probability distributions.“ ([28], S. 857).

Vom 20. April bis zum 22. April 1965 fand das Symposium on System Theory am Polytechnic Institute in Brooklyn statt, und hier sprach Zadeh über “A New View on System Theory”. Diese „neue Sicht auf die Systemtheorie“ war allerdings von völlig anderer Qualität als die des State-Space-Approach, denn sie enthielt tatsächlich den Vorschlag einer „neuen Mathematik“, die Mathematik der Fuzzy Sets und Fuzzy Systeme, “which provide a way of treating fuzziness in a quantitative manner” ([31], S. 29). Erstmals sprach Zadeh vor einem großen Auditorium über seine neue Theorie der Fuzzy Sets, und in dem später herausgegebenen Tagungsband trägt der gedruckte Beitrag dann auch einen anderen Titel: „Fuzzy Sets and Systems“.

Zadehs Definition eines Fuzzy Systems war denkbar einfach: Ein System  $S$  mit Input  $u(t)$ , Output  $y(t)$  und Zustand  $x(t)$  ist ein *Fuzzy System*, wenn  $u(t)$  oder  $y(t)$  oder  $x(t)$  oder irgendeine Kombination davon sich auf Fuzzy Sets erstreckt. Beispielsweise sei ein *Input* von  $S$  zum Zeitpunkt  $t$  spezifiziert als „erheblich mehr als 5“, dann ist der *Output* ein Fuzzy Set, und ein System, das auf solch ungenau definierte Inputs agieren kann, ist ein Fuzzy System. Systeme, deren Zustände durch *fuzzy* Adjektive wie *leicht*, *schwer*, *nicht sehr schwer*, *sehr leicht* etc. beschrieben werden, seien Fuzzy Systeme. Zadeh führte aus, dass sich diese Begriffe auf Situationen beziehen „in which the source of imprecision is not a random variable or a stochastic process but rather a class or classes which do not possess sharply defined boundaries.“ Dies seien keine Klassen oder Mengen im üblichen Sinne, “since they do not dichotomize all objects into those that belong to the class and those that do not.” Fuzzy Sets führte er in diesem Text folgendermaßen ein: “the concept of a fuzzy set”, that is a class in which there may be a continuous infinity of grades of membership, with the grade of membership of an object  $x$  in a fuzzy set  $A$  represented by a number  $\mu_A(x)$  in the interval  $[0,1]$ .” Damit sei ein bequemer Weg gefunden, um den Vorgang der Abstraktion zu definieren, “– a process which plays a basic role in human thinking and communication.” Das dann folgende Beispiel bezieht sich wieder auf die Musterklassifikation: “For example, suppose that we are concerned

with devising a test for differentiating between handwritten letters  $O$  and  $D$ . One approach to this problem would be to give a set of handwritten letters and indicate their grades of membership in the fuzzy sets  $O$  and  $D$ . On performing abstraction on these samples, one obtains the estimates  $\tilde{\mu}_O$  and  $\tilde{\mu}_D$  of  $\mu_O$  and  $\mu_D$  respectively. Then given a letter  $x$  which is not one of the given samples, one can calculate its grades of membership in  $O$  and  $D$ , and, if  $O$  and  $D$  have no overlap, classify  $x$  in  $O$  or  $D$ ." ([31], S. 29)

Damit diese Abstraktion mathematisch sinnvoll wird, sei genügend a priori Information über die Zugehörigkeitsfunktion  $\mu_A$  erforderlich, um Fehlergrenzen des Schätzers  $\mu_A$  zu benennen. ([31], S. 30). Wir Menschen können hervorragend abstrahieren, auch dann, wenn es um mathematisch schlecht formalisierte Probleme geht, aber wir verstünden nicht, auf welche Weise uns dieser Abstraktionsprozeß gelingt, und deshalb können wir ihn auch nicht in Maschinenbefehle umsetzen.

## 7 „Fuzzy Sets“

Als Lotfi Zadeh im April 1965 seinen Vortrag über "A New View on System Theory" hielt, war der Artikel "Fuzzy Sets" längst im Druck. Somit muß nach den Quellen zur „ersten Idee“ der Fuzzy Sets früher gesucht werden, und in der Tat ist bis zum Sommer 1964 zurück zu gehen! Damals hielt Zadeh einen Vortrag über Mustererkennung bei einer Veranstaltung der Wright-Patterson Air Force Base in Dayton, Ohio. Danach plante er einige Wochen in Santa Monica mit Richard Bellman zu arbeiten, mit dem ihn seit den späten 1950er Jahren eine enge Freundschaft verband. Während seiner Reise stellte Zadeh Überlegungen darüber an, eine „graduelle Zugehörigkeit“ von Punkten für das Problem der Mustertrennung mathematisch zu formalisieren. Ob er diese Idee schon in seinen Vortrag in Ohio eingebaut hatte, ist offenbar nicht mehr zu klären: Es gibt kein Textmanuskript und Lotfi Zadeh selbst vermag sich nicht mehr genau zu erinnern. Kurz darauf, in Santa Monica, diskutierte er seine Ideen allerdings schon eingehend mit Bellman, der ihn ermunterte sie weiter zu verfolgen. Zadeh schickte ihm dann bald eine Ausarbeitung dieser Diskussionen mit dem Titel "Abstraction and Pattern Classification". Bellman, der damals das *Journal of Mathematical Analysis and Applications* herausgab, antwortete, dass er den Text publizieren wollte, dies geschah dann aber erst 1966 [32], und als Autoren sind hier Richard Bellman, Robert Kalaba und Lotfi Zadeh aufgeführt. Dieser Artikel ist in Titel und Text gleich dem im Oktober 1964 erschienenen Memorandum RM-4307-PR für die RAND-Corporation [33]. Dies ist offensichtlich der erste Text, in dem das Konzept der Fuzzy Sets eingeführt wurde, "a notion which extends the concept of membership in a set to situations in which there are many, possibly a continuum of, grades of membership." ([33], S. 1) Das Manuskript für den Artikel "Fuzzy Sets" entstand später, denn Zadeh sandte es erst im November 1964 an die Zeitschrift *Information and Control*. In diesem November erschien eine textgleiche Version als Bericht des Electronics Research Laboratory (ERL) der University of California in Berkeley [34].



Abb. 8: Richard Ernest Bellman

„Fuzzy Sets“ beginnt mit dem Satz: „More often than not, the classes of objects encountered in the real physical world do not have precisely defined criteria of membership.“ ([33], S. 338) – Schon der erste Satz dieser Einleitung bringt Zadehs Überzeu-

gung auf den Punkt, zu der er nach jahrzehntelanger Bearbeitung realer physikalischer Objekte mit mathematisch-präzisen Methoden gekommen war. Nun wartete er aber mit Beispielen auf, die nicht erkennen lassen, wie diese Theorie entstand: Die Klasse der Tiere enthalte z.B. ganz klar Hunde, Pferde, Vögel etc. und selbstverständlich gehörten Berge, Flüssigkeiten und Pflanzen sicherlich nicht dazu. Es gebe aber Objekte, die diesbezüglich einen unklaren Status hätten: Seesterne, Bakterien usw. Dieselbe Art von Unklarheit trete in dem Falle einer Zahl wie 10 bezüglich der „Klasse“ aller reellen Zahlen, die viel größer als 1 sind, auf. Natürlich seien die „Klasse aller reellen Zahlen, die viel größer als 1 sind“, oder die „Klasse aller schönen Frauen“ oder die „Klasse aller großen Männer“ keine Klassen oder Mengen in der gewöhnlichen mathematischen Bedeutung dieser Bezeichnungen. Dennoch sei es eine Tatsache, dass derart ungenau definierte „Klassen“ eine wichtige Rolle im menschlichen Denken spielen, und dann nennt er neben dem Bereich der Kommunikation von Information die Abstraktion, und insbesondere bei der Mustererkennung.

Zadehs Definition der Fuzzy Sets ist bekannt: „A fuzzy set (class)  $A$  in  $X$  is characterized by a membership (characteristic function) function  $f_A(x)$  which associates with each point in  $X$  a real number in the interval  $[0,1]$ , with the value of  $f_A(x)$  at  $x$  representing the ‚grade of membership‘ of  $x$  in  $A$ .“ ([35], S. 339)<sup>6</sup>

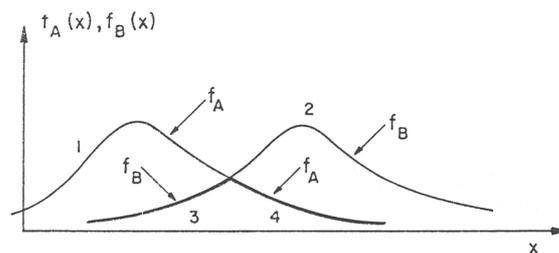


Abbildung 9: Illustration der Vereinigung und des Durchschnitts von Fuzzy Sets in  $\mathbb{R}^1$ . [35]

Dann folgend definierte Zadeh das *leere Fuzzy Set*,<sup>7</sup> die *Gleichheit* von Fuzzy Sets, das *Komplement* eines Fuzzy Sets, *Fuzzy-Teilmengen*, *Vereinigung* und *Durchschnittsbildung* für Fuzzy Sets (siehe Abb. 9).<sup>8</sup> Für die Behauptung, dass alle Fuzzy Sets einer Grundmenge  $X$  einen distributiven Verband mit 0 und 1 bilden, verwies er auf die *Verbandstheorie* von Garrett Birkhoff [34] und erwähnte lediglich, dass die Gesetze von De Morgan und die Distributivgesetze für die Fuzzy Sets in  $X$  mit den definierten Operationen Geltung haben. Schließlich definierte Zadeh *Konvexität* und *Beschränktheit* von Fuzzy Sets:

*Definition 3:* Ein Fuzzy Set  $A$  ist *konvex*  $\Leftrightarrow \Gamma_\alpha = \{x \mid \mu_A(x) \geq \alpha\}$  ist konvex für alle  $\alpha \in (0,1]$ .

*Definition 4:* Ein Fuzzy Set  $A$  ist *beschränkt*  $\Leftrightarrow \Gamma_\alpha = \{x \mid \mu_A(x) \geq \alpha\}$  ist beschränkt  $\forall \alpha > 0$ , und das heißt:  $\forall \alpha > 0 \exists$  ein endliches  $R(\alpha)$  so, dass  $\|x\| \leq R(\alpha) \forall x \in \Gamma_\alpha$ .

<sup>6</sup> In diesem Text bezeichnete Zadeh die Zugehörigkeitsfunktion durch den Buchstaben  $f$ . Im Folgenden wird der seit Jahren übliche griechische Buchstabe  $\mu$  zur Bezeichnung der Zugehörigkeitsfunktion verwendet. Da die Abbildungen 9 und 10 dem Artikel „Fuzzy Sets“ [35] entnommen wurden, werden Zugehörigkeitsfunktionen dort mit dem Buchstaben  $f$  bezeichnet.

<sup>7</sup> Zwei Fuzzy Sets  $A$  und  $B$  sind gleich,  $A = B \Leftrightarrow \mu_A(x) = \mu_B(x)$  für alle  $x \in X$ .

<sup>8</sup> Zadeh definierte die *Vereinigung* zweier Fuzzy Sets  $A$  und  $B$  als ein Fuzzy Set  $C = A \cup B$  mit der Zugehörigkeitsfunktion  $f_C(x) = \text{Max} [\mu_A(x), \mu_B(x)]$ , er bewies auch, dass dies das kleinste sowohl  $A$  als auch  $B$  enthaltende Fuzzy Set ist. Entsprechend definiert er den *Durchschnitt* zweier Fuzzy Sets  $A$  und  $B$  als Fuzzy Set  $C = A \cap B$  mit der  $\mu_C(x) = \text{Min} [\mu_A(x), \mu_B(x)]$ , das ist das größte Fuzzy Set, das sowohl in  $A$  als auch in  $B$  enthalten ist.

Beide Definitionen waren für die in dem Artikel folgenden Untersuchungen wichtig, denn nun befasste sich Zadeh auch hier mit dem Problem aus der Musterklassifikation, Mengen voneinander zu trennen. Gewöhnliche konvexe Mengen  $A$  und  $B$  lassen sich leicht separieren, wenn sie disjunkt sind, denn dann existiert nach dem Trennungstheorem eine separierende Hyperebene  $H$ , auf deren einen Seite die Menge  $A$ , auf deren anderer Seite  $B$  liegt. Zadeh erweiterte dieses Theorem auf den Fall *beschränkter* Fuzzy Sets  $A$  und  $B$ , wobei er darauf verzichtete, die Disjunktheit zu fordern, da dies für den Fall der Fuzzy Sets zu einschränkend ist.

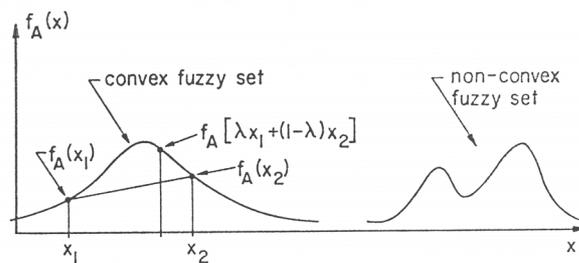


Abb. 10: Konvexe und nicht konvexe Fuzzy Sets in  $\mathbf{E}^1$ . [35]

**THEOREM.** Es seien  $A$  und  $B$  beschränkte konvexe Fuzzy Sets in  $\mathbf{E}^n$  mit maximalen Zugehörigkeitsgraden  $M_A$  bzw.  $M_B$  [ $M_A = \text{Sup}_x \mu_A(x)$ ,  $M_B = \text{Sup}_x \mu_B(x)$ ]. Es sei  $M$  der maximale Zugehörigkeitsgrad für den Durchschnitt  $A \cap B$  ( $M = \text{Sup}_x \text{Min} [\mu_A(x), \mu_B(x)]$ ). Dann ist  $D = 1 - M$ . ([35], p. 352).

Für den Beweis ging Zadeh von zwei beschränkten Fuzzy Sets  $A$  und  $B$  sowie einer durch die Gleichung  $h(x) = 0$  definierten Hyperebene  $H$  im euklidischen Raum  $\mathbf{E}^n$  aus, so dass alle  $x$ , für die  $h(x) \geq 0$  gilt, auf der einen Seite von  $H$  liegen und alle  $x$ , für die  $h(x) \leq 0$  gilt, auf deren anderer Seite. Für jede Hyperebene  $H$  existiert eine Zahl  $K_H$ , so dass auf der einen Seite dieser von  $H$  gilt:  $f_A(x) \leq K_H$  und auf der anderen Seite von  $H$ :  $f_A(x) \geq K_H$ . Das Infimum dieser beiden Zahlen bezeichnete Zadeh mit  $M_H$  und dessen Komplement  $D_H = 1 - M_H$  als den *Grad der Separation* von  $A$  und  $B$  durch  $H$ .

Da im Allgemeinen eine ganze Hyperebenen-schar gegeben ist, formulierte Zadeh die Trennungsaufgabe folgendermaßen: Finde diejenige Hyperebene aus einer Hyperebenenfamilie  $\{H_\lambda\}$ , die den höchstmöglichen Grad der Separation realisiert. Für den Spezialfall, dass die Hyperebenen  $H_\lambda$  im euklidischen Raum  $\mathbf{E}^n$  liegen und  $\lambda$  Laufvariable in  $\mathbf{E}^n$  ist, definiert Zadeh den Separationsgrad von  $A$  und  $B$  über folgende Relation, den Index  $\lambda$  der Einfachheit halber weglassend:  $D = 1 - M$ , wobei  $M = \text{Inf}_H M_H$  das Infimum von  $M_H$  aller Hyperebenen  $H$  ist.

Zum Ende des Artikels bewies Zadeh, dass der höchste mittels einer Hyperebene im euklidischen Raum  $\mathbf{E}^n$  erreichbare Separationsgrad zweier konvexer und beschränkter Fuzzy Sets  $A$  und  $B$  das Komplement des maximalen Grades im Durchschnitt  $A \cap B$  ist. (Für den Fall  $n = 1$  ist dies in Abb. 11 gezeigt.)

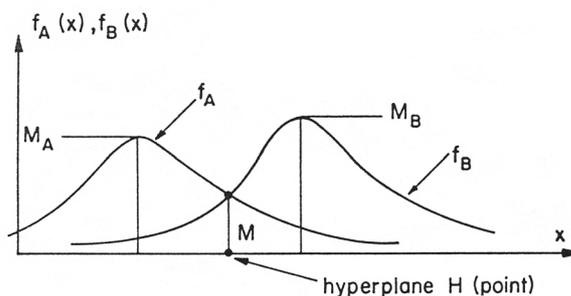


Abb. 11: Illustration zum Trennungstheorem für konvexe Fuzzy Sets im Euklidischen Raum  $\mathbf{E}^1$ , [35].

## 8 Ausblick

Die Theorie der Fuzzy Sets hat sich völlig anders entwickelt, als Lotfi Zadeh in den Jahren nach 1965 gedacht hatte, und noch 1969 erwartete er, dass vor allem die nicht-exakten Wissenschaften diese Fuzzy Sets aufgreifen würden.

“What we still lack, and lack rather acutely, are methods for dealing with systems which are too complex or too ill-defined to admit of precise analysis. Such systems pervade life sciences, social sciences, philosophy, economics, psychology and many other “soft” fields.”[37]

In den 1990er Jahren antwortete er auf die Frage, wo er damals Anwendungsmöglichkeiten seiner Theorie gesehe habe:

“In many, many fields. I expected people in the social sciences-economics, psychology, philosophy, linguistics, politics, sociology, religion and numerous other areas to pick up on it. It’s been somewhat of a mystery to me why even to this day, so few social scientists have discovered how useful it could be. Instead, Fuzzy Logic was first embraced by engineers and used in industrial process controls and in “smart” consumer products such as hand-held camcorders that cancel out jittering and microwaves that cook your food perfectly at the touch of a single button. I didn’t expect it to play out this way back in 1965.” [38]



Abb. 12: Lotfi Zadeh bei der Jubiläumsfeier „40 Jahre Fuzzy Sets“ am 30. Juli 2005 während des IFSA World Congress in Peking.

Heute ist die Theorie der Fuzzy Sets inmitten des „Soft Computing“ genannten Forschungsprogramms fest verankert, und Zadeh treibt die theoretische Entwicklung noch immer weiter an, wie seine zahlreichen Vorträge auf vielen Kongressen im Bereich des Soft Computing belegen.

Als im Juli 2005 in Peking der Weltkongress der International Fuzzy Systems Association (IFSA) stattfand, feierte man dort – mit Lotfi Zadeh als Ehrengast - zwei runde Jubiläen: „20 Jahre IFSA“ und „40 Jahre Fuzzy Set Theorie“ (Abb. 12).

### Dank

Ich danke Lotfi Zadeh für seine jahrelange großartige und Unterstützung meines Projekts zur Geschichte der Fuzzy Set Theorie und ihrer Anwendungen.

## 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 (Boethius Bd. 54) 2005.
- [2] Seising, Rudolf: The 40th Anniversary of Fuzzy Sets – A New View on System Theory. *Proceedings of the North American Fuzzy Information Processing Society (NAFIPS) Annual Conference Soft Computing for Real World Applications*, 22-25 June, 2005, Ann Arbor, Michigan, USA. Editors: Hao Ying, Dimitar Filev, CD: Catalog-Nr. 05TH8815C, ISBN: 0-7803-9188-8, pp. 92-97.
- [3] Seising, Rudolf: 1965 – “Fuzzy Sets” appear – A Contribution to the 40th Anniversary. *Proceedings of the Conference FUZZ-IEEE 2005*, Reno, Nevada, May 22-25, 2005. (CD)
- [4] Seising, Rudolf: 40 years ago: „Fuzzy Sets“ is going to be published, *Proceedings of the 2004 IEEE International Joint Conference on Neural Networks / International Conference on Fuzzy Systems (FUZZ-IEEE 2004)*, July 25-29, 2004, Budapest, Hungary (Auch als CD: ISBN: 0-7803-8354-0).
- [5] Seising, Rudolf: Noninferiority, Adaptivity, Fuzziness in Pattern Separation: Remarks on the Genesis of Fuzzy Sets. *Proceedings of the Annual Meeting of the North American Fuzzy Information Processing Society*, Banff, Alberta, Canada: Fuzzy Sets in the Hearth of the Canadian Rockies 2004 (NAFIPS 2004), June 27-30, 2004, Banff, Alberta, Canada. Edited by Scott Dick, Lukasz Kurgan, Petr Musilek, Witold Pedrycz, and Marek Reformat, S. 2002-2007. (Auch als CD.)
- [6] 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 in Bommerholz bei Dortmund, Gästehaus der Universität Dortmund „Haus Bommerholz“.*
- [7] Zadeh, Lotfi A.: Frequency Analysis of Variable Networks. *Proceedings of the IRE*, Vol. 38, pp. 291-299, March 1950.
- [8] *IRE Transactions On Circuit Theory*. Published by the Professional Group on Circuit Theory under the aegis of the Institute of Radio Engineers, Vol. CT-2, September, 1955, Nr.3.
- [9] Guillemin, Ernst A.: The Fourier Integral – A Basic Introduction. *IRE Transactions on Circuit Theory*, Vol. CT-2, Nr.3, pp. 227-230, September, 1955.
- [10] Miller, Kenneth S. and Lotfi A. Zadeh: Generalization of the Fourier Integrals. *IRE Transactions on Circuit Theory*, Vol. CT-2, Nr. 3, S. 256-260, 1955.
- [11] Stevenson, Robert L.: *Dr. Jekyll und Mr. Hyde*. Frankfurt am Main: Insel, 2001.
- [12] Guillemin, Ernst A.: *Introductory Circuit Theory*. New York, London, Sydney, 1953, S. xx.
- [13] Zadeh, Lotfi A. Interview mit Rudolf Seising (16. Juni 2001, Berkeley, USA).
- [14] Zadeh, L. A.: System Theory, *Columbia Engineering Quarterly*, November (1954), S. 16-19, 34: 16.
- [15] Shannon, Claude E.: A Mathematical Theory of Communication, *Bell System Technical Journal*, Vol. 27, S. 379-423, S. 623-656, July and October, 1948.
- [16] Wiener, Norbert: *Cybernetics or Control and Communication in the Animal and the Machine*, Cambridge-Mass., New York: Hermann & Cie., the Technology Press, and John Wiley & Sons, 1948. Berkeley, Edmund Callis: *Giant Brains or*

- Machines that Think*. New York, London: John Wiley & Sons, Chapman & Hall, 1949.
- [17] Shannon, Claude E.: The Bandwagon. Editorial. *IRE Transactions on Information Theory*, March 1956.
- [18] Wiener, Norbert: What is Information Theory? Editorial, *IRE Transactions on Information Theory*, June 1956.
- [19] Bellman, Richard E., Robert Kalaba: On the Role of Dynamic Programming in Statistical Communication Theory. *IRE Transactions on Information Theory*, S. 197-203, September 1957.
- [20] Zadeh, L. A.: What is Optimal? Editorial, *IRE Transactions on Information Theory*, Vol. IT-4, Nr. 1, 1958.
- [21] Zadeh, Lotfi A.: Optimality and Non-Scalar-Valued Performance Criteria. *IEEE Transactions on Automatic Control*, Correspondence, S. 59f, January 1963.
- [22] Wald, Abraham: *Statistical Decision Functions*. New York: John Wiley and Sons Inc., 1950.
- [23] Nagy, George: State of the Art in Pattern Recognition. *Proceedings of the IEEE*, Vol. 56, No. 5, S. 836-863, May 1968.
- [24] Rosenblatt, Frank: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, Vol. 65, No. 6, 1958, S. 386-408.
- [25] Minsky, Marvin and Seymour Papert: *Perceptrons*. Cambridge, Mass.: MIT Press, 1969.
- [26] Zadeh, L. A., Desoer, C. A.: *Linear System Theory: The State Space Approach*. New York 1963.
- [27] Zadeh, L. A.: From Circuit Theory to System Theory. *Proceedings of the IRE*, Vol. 50 (1962), S. 856-865.
- [28] Bellman, Richard E.: *Dynamic Programming*. Princeton, NJ: University Press, 1957.
- [29] Mesarovic, M. D.: Views on General Systems Theory. Proceedings of the Second Systems Symposium at Case Institute of Technology. Huntington, New York: Robert E. Krieger Publ. Comp. 1964.
- [30] Zadeh, L. A.: Fuzzy Sets and Systems. In: Fox, J. (ed.): System Theory. Microwave Research Institute Symposia Series XV, Brooklyn, New York: Polytechnic Press 1965, p. 29-37.
- [31] Bellman, R. E., Kalaba, R., Zadeh, L. A.: Abstraction and Pattern Classification. *Journal of Mathematical Analysis and Applications* 13, 1966, S. 1-7.
- [32] Bellman, R. E., Kalaba, R., Zadeh, L. A.: Abstraction and Pattern Classification. Memorandum RM- 4307-PR, The RAND Corporation, October 1964.
- [33] Zadeh, L. A.: Fuzzy Sets, Electronic Research Laboratory (ERL), University of California Berkeley, Report No. 64-44, November 16, 1964.
- [34] Zadeh, L. A.: Fuzzy Sets, Information and Control, 8, 1965, pp. 338-353.
- [35] Birkhoff, Garrett: Lattice Theory. (= American Mathematical Society Colloquium Publications, Vol. 25). New York, 1948.
- [36] Zadeh, Lotfi A.: *Toward a Theory of Fuzzy Systems*. Electronic Research Laboratory, College of Engineering, University of California, Berkeley 94720, Report No. ERL-69-2, June 1969.
- [38] Interview with Lotfi Zadeh, Creator of Fuzzy Logic, by Betty Blair, *Azerbaijan International*, Winter 1994 (2.4):  
[http://www.azer.com/aiweb/categories/magazine/24\\_folder/24\\_articles/24\\_fuzzylogic.html](http://www.azer.com/aiweb/categories/magazine/24_folder/24_articles/24_fuzzylogic.html) (2. Mai 2003).

# Clustering Power Law Data

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

Many data sets containing frequencies or counts approximately exhibit power law (more precisely Zipf's law) distributions. For the example of German and international city inhabitant data sets we show that such data sets can be modeled as fuzzy combinations of local power law distributions. These fuzzy combinations can be represented as first order Takagi–Sugeno fuzzy models with inverse rank inputs. For building these models we consider four different approaches: Gustafson–Kessel clustering in the input–output product space, fuzzy  $c$ –regression models, error minimization with  $c$ –means partition functions, and error minimization with a new kind of membership functions that we call *local  $c$ –means* partition functions. The experiments show that the new proposed approach with local  $c$ –means partition functions produces rules with intuitive, smooth, and convex membership functions and at the same time yields the lowest model errors.

## 1 Introduction: Power law data sets

Data from man made or naturally occurring phenomena often exhibit the property that small values occur very often, but high values only very rarely. Examples for this type of data distributions include sizes of cities or countries, frequencies of words, and usage counts in telecommunication or data networks. Mathematically, these data can often be (at least approximately) described by *power law* distributions. There exist at least three different types of power law distributions, depending on the random variable considered:

1. Zipf's law [17] states that the data  $y \in Y = \{y_1, \dots, y_n\} \subset \mathbb{R}$  are inversely proportional to their ranks  $r$ ,

$$y \sim r^{-\alpha}, \quad (1)$$

with the exponent  $\alpha \approx 1$ . This means that the largest value of  $y$  is (approximately) twice as large as the second largest value, and the second largest value is (approximately) one and a half as large as the third largest value. More generally, if we sort the data in  $Y$  so that  $y_1 > y_2 > \dots > y_n$ , then we have

$$y_k \sim k^{-\alpha}, \quad (2)$$

and further

$$\frac{y_k}{y_{k+1}} \approx \frac{k+1}{k} = 1 + \frac{1}{k}. \quad (3)$$

In other words, the higher the rank  $k$ , the smaller the relative difference between adjacent values  $y_k$  and  $y_{k+1}$ . Examples for data that usually follow Zipf's law include frequencies of words.

2. Let the cumulative distribution function  $P$  denote the probability that  $y$  is *larger* than a threshold  $x \in \mathbb{R}$ . Pareto's law [10] states that this cumulative distribution function is

$$P(x) \sim x^{-\alpha}. \quad (4)$$

Examples for data that usually follow Pareto's law include people's incomes.

3. Let the probability *density* function  $p$  represent the probability that  $y$  is *exactly equal* to  $x \in \mathbb{R}$ . The corresponding version of Pareto's law states that this probability density function is

$$p(x) \sim x^{-\alpha}. \quad (5)$$

The two versions of Pareto's law are connected by the following theorem.

$$P(x) = \beta \cdot x^{-\alpha} \quad \Leftrightarrow \quad p(x) = \alpha \cdot \beta^\alpha \cdot x^{-(\alpha+1)} \quad (6)$$

This means that if the cumulative distribution function follows Pareto's law with an exponent  $\alpha$ , then the corresponding probability density function will also follow Pareto's law, but with an exponent  $\alpha + 1$ , and vice versa.

In this paper we are mainly interested in data *ranks*, so if we talk about power law here, then we will always mean Zipf's, not Pareto's law. As examples for data sets following Zipf's law we consider here the number of inhabitants of cities, in particular (1) all German cities with at least 20,000 inhabitants, and (2) all international cities with at least 1,000,000 inhabitants. The German city data set is taken from [3], available at <http://www.destatis.de/download/jahrbuch/bevj04.pdf>, and the international city data set [2] is available at <http://www.citypopulation.de>. Table 1 shows parts of the German and international city data sets. The German city data set contains 688 cities, and the international city data set contains 428 cities. Both lists are ordered by the number of inhabitants, so the rank  $k = 1$  corresponds to the city with the maximum number of inhabitants. Both lists show that the difference of inhabitants decreases with increasing rank, i.e. the differences in the  $y_k$  column get lower as we proceed down the list. Fig. 1 shows the number of inhabitants versus the inverse of the rank plotted for both data sets. Instead of the rank we display the *inverse* of the rank,  $x_k = 1/k$ ,  $k = 1, \dots, n$ , because data sets that *exactly* follow Zipf's law (1) would produce points on a straight line in such a diagram. In both diagrams we observe that the left part ( $x < 0.06$ ) is quite straight and thus follows Zipf's law very well. The German city data set shows a plateau at about  $0.06 \leq x < 0.2$ , and then a further increasing linear behavior for  $x > 0.2$ . In the international city data set the slope also decreases at about  $x = 0.06$ , then further decreases at about  $x = 0.15$ , and finally increases again at  $x = 0.5$ .

## 2 First order Takagi–Sugeno fuzzy modeling

Apparently, Zipf's law does not hold for the city data sets in total, but only for *subsets* in local intervals of  $x$ . The locally approximately linear parts in Fig. 1 are not separated

Table 1: The city data sets

(a) German cities			(b) international cities		
$k$		$y_k$	$k$		$y_k$
1	Berlin	3,392,400	1	Tokyo	34,000,000
2	Hamburg	1,728,800	2	Mexico City	22,350,000
3	München	1,234,700	3	Seoul	22,050,000
4	Köln	968,600	4	New York	21,800,000
5	Frankfurt am Main	643,700	5	Sao Paulo	20,000,000
6	Dortmund	590,800	6	Bombay	19,400,000
7	Stuttgart	588,500	7	Delhi	19,000,000
8	Essen	585,500	8	Los Angeles	17,750,000
9	Düsseldorf	571,900	9	Jakarta	16,850,000
10	Bremen	543,000	10	Osaka	16,750,000
11	Hannover	517,300	11	Calcutta	15,350,000
12	Duisburg	508,700	12	Cairo	15,250,000
13	Leipzig	494,800	13	Manila	14,550,000
14	Nürnberg	493,400	14	Karachi	13,800,000
15	Dresden	480,200	15	Moscow	13,650,000
16	Bochum	388,900	16	Shanghai	13,400,000
17	Wuppertal	363,500	17	Buenos Aires	13,350,000
18	Bielefeld	324,800	18	Dacca	12,750,000
19	Bonn	308,900	19	Rio de Janeiro	12,000,000
20	Mannheim	308,800	20	London	11,950,000
⋮	⋮		⋮	⋮	
688	Rastede	20,000	428	Shizuoka	1,000,000

crisply, but interpolated in a fuzzy way. Therefore, for modeling [12] these two data sets we do not fit (crisp) piecewise linear functions, but use first order [14] Takagi–Sugeno fuzzy systems [15, 16] of the form

$$R_1 : \text{ IF } \mu_1(x) \text{ THEN } y = b_{10} + b_{11} \cdot x \quad (7)$$

⋮

$$R_c : \text{ IF } \mu_c(x) \text{ THEN } y = b_{c0} + b_{c1} \cdot x \quad (8)$$

whose output  $y \in \mathbb{R}$  is computed for a given input  $x \in \mathbb{R}^p$  using the convex combination function

$$y = \frac{\sum_{i=1}^c \mu_i(x) \cdot (b_{i0} + b_{i1} \cdot x)}{\sum_{i=1}^c \mu_i(x)}, \quad (9)$$

where in some cases the denominator is always equal to one and can then be omitted. We call these models *fuzzy Zipf models*.

There are various approaches for the automatic extraction of first order Takagi–Sugeno systems from data [11]. A popular modeling approach uses clustering techniques [6]. The membership functions  $\mu_1, \dots, \mu_c$  are then computed from the cluster partition matrix, and the matrix  $B$  of coefficients of the conclusion functions is computed from the cluster prototypes. Another popular modeling approach explicitly minimizes the approximation error

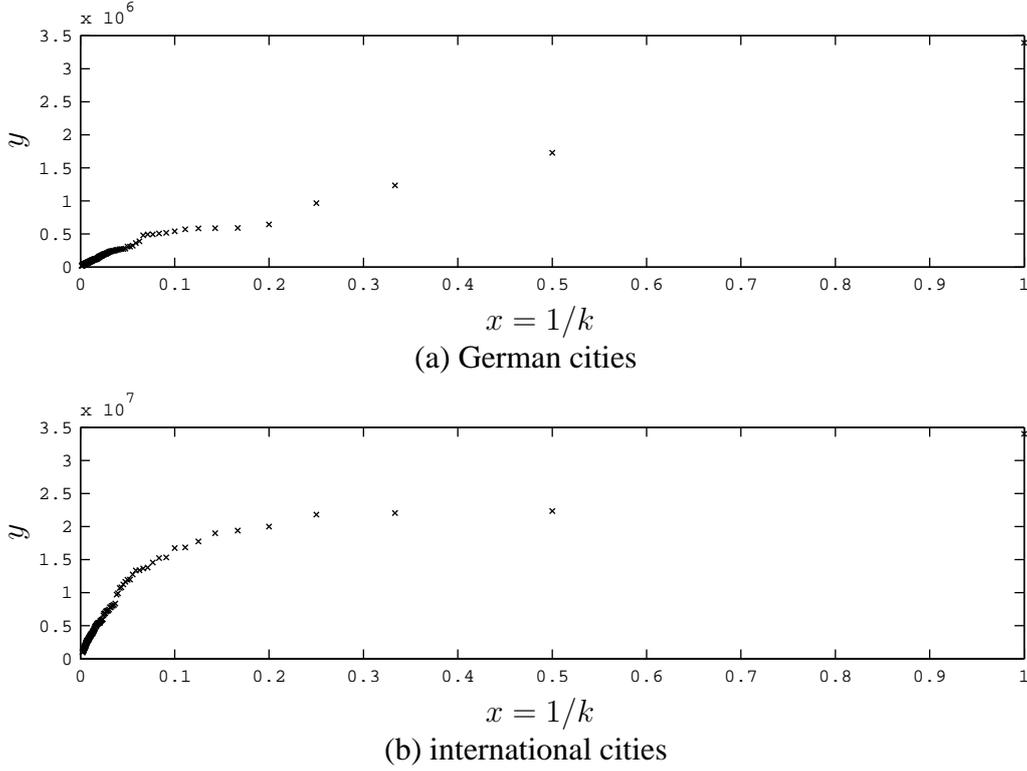


Figure 1: The city data sets.

of the Takagi–Sugeno system. In the following four sections we present three existing and one new approach for this modeling task: two clustering approaches using Gustafson and Kessel clustering and fuzzy  $c$ –regression models, and two error minimization approaches using global and local fuzzy input partitions.

### 3 Gustafson and Kessel clustering

The Takagi–Sugeno system to be generated for our city data sets uses one–dimensional inputs  $X$  and one–dimensional outputs  $Y$ . In this section, however, we concatenate the input and output data to a two–dimensional data set  $Z = \{(x_1, y_1), \dots, (x_n, y_n)\}$  for clustering, i.e. we perform clustering in the two–dimensional input–output product space [9]. One clustering model that is often used in the generation of first order Takagi–Sugeno fuzzy systems is the model by Gustafson and Kessel (GK) [4] which is an extension of the fuzzy  $c$ –means model [1] with local Mahalanobis distances [8]. Given an object data set  $Z = \{z_1, \dots, z_n\} \subset \mathbb{R}^p$ ,  $p \in \{1, 2, \dots\}$ , a number of clusters  $c \in \{2, \dots, n - 1\}$ , and a fuzzifier  $m > 1$ , the GK model can be formulated as the following problem: Minimize the objective function

$$J_{\text{GK}} = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m d_{ik}^2, \quad (10)$$

$d_{ik} = (v_i - z_k) \cdot A_i \cdot (v_i - z_k)^T$ ,  $i = 1, \dots, c$ ,  $k = 1, \dots, n$ , subject to the set of cluster centers  $V = \{v_1, \dots, v_c\} \subset \mathbb{R}^p$  and the partition matrix  $U \in M_{\text{fcn}}$ , where

$$M_{\text{fcn}} = \left\{ U \in [0, 1]^{c \times n} \mid \begin{aligned} &\sum_{i=1}^c u_{ik} = 1, k = 1, \dots, n, \\ &\sum_{k=1}^n u_{ik} > 0, i = 1, \dots, c \end{aligned} \right\}. \quad (11)$$

The local norm-inducing matrices  $A_i$  are computed by

$$A_i = (\rho_i \det S_i)^{1/p} \cdot (S_i^T \cdot S_i) \cdot S_i^T, \quad (12)$$

with the cluster volumes  $\rho_i > 0$ ,  $i = 1, \dots, c$ , and the local covariance matrices

$$S_i = \sum_{k=1}^n u_{ik}^m \cdot (v_i - z_k)^T \cdot (v_i - z_k), \quad (13)$$

$i = 1, \dots, c$ . For simplicity we set  $\rho_1 = \dots = \rho_c = 1$  here. The necessary conditions for extrema of  $J_{\text{GK}}$  are

$$u_{ik} = 1 \left/ \sum_{j=1}^c \left( \frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}} \right., \quad (14)$$

$$v_i = \frac{\sum_{k=1}^n u_{ik}^m z_k}{\sum_{k=1}^n u_{ik}^m}, \quad (15)$$

$i = 1, \dots, c$ ,  $k = 1, \dots, n$ . Optimization of the GK model can be done by randomly initializing  $V$ , initializing  $A_i$ ,  $i = 1, \dots, c$ , as  $p \times p$  unit matrices, and then perform  $t$  steps of alternatingly updating  $U$  by (14),  $V$  by (15), and  $A_1 \dots A_c$  by (12).

For the input parts  $X$  of the training data set  $Z$ , the values of the membership functions  $\mu_1, \dots, \mu_c$  can be immediately obtained from  $U$ . For generalization to  $x \in \mathbb{R}^p$  we extend expression (14) and obtain the membership functions

$$\mu_i(x) = 1 \left/ \sum_{j=1}^c \left( \frac{(v_i^{(x)} - x) \cdot A_i^{(x)} \cdot (v_i^{(x)} - x)^T}{(v_j^{(x)} - x) \cdot A_j^{(x)} \cdot (v_j^{(x)} - x)^T} \right)^{\frac{2}{m-1}} \right., \quad (16)$$

where  $v_i^{(x)}$  denotes the first  $p$  components of  $v_i$  and  $A_i^{(x)}$  denotes the upper left  $p \times p$  sub-matrix of  $A_i$ , referring to the input space,  $i = 1, \dots, c$ . Notice however, that for the evaluations presented here we consider the *training* data only and do not perform this generalization.

The matrix  $B$  of coefficients of the conclusion functions is obtained from the cluster centers  $V$  and the eigenvectors of the norm inducing matrices  $A_1, \dots, A_c$ . For  $p = q = 1$  these coefficients are

$$b_{i0} = v_i^{(y)} - v_i^{(x)} \cdot b_{i1}, \quad (17)$$

$$b_{i1} = e_i^{(y)} / e_i^{(x)}, \quad (18)$$

where  $v_i = (v_i^{(x)}, v_i^{(y)})$ , and  $e_i = (e_i^{(x)}, e_i^{(y)})$  is the eigenvector with the largest eigenvalue of  $A_i$ ,  $i = 1, \dots, c$ .

## 4 Fuzzy c–regression models

Another extension of fuzzy c–means is the fuzzy c–regression models (FCRM) approach [5]. In FCRM modeling we *do* distinguish between the inputs  $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^p$  and the corresponding outputs  $Y = \{y_1, \dots, y_n\} \subset \mathbb{R}^q$ , just as in the resulting Takagi–Sugeno system. We define the prototypical local functions  $f_i : \mathbb{R}^p \rightarrow \mathbb{R}^q$ ,  $i = 1, \dots, c$ , that are used in FCRM clustering and that also serve explicitly as the conclusion functions of the Takagi–Sugeno system. The FCRM model tries to fit the parameters of the local functions  $f$  and the memberships  $U$  so that the convex combination of the (quadratic) approximation errors of the Takagi–Sugeno system is minimized. To do so, the FCRM objective function is the same as the GK objective function (10), but now the “distance” represents the approximation error

$$d_{ik} = \|y_k - f_i(x_k)\|, \quad (19)$$

$i = 1, \dots, c$ ,  $k = 1, \dots, n$ , so the FCRM objective function explicitly reads

$$J_{\text{FCRM}} = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m \|y_k - f_i(x_k)\|^2, \quad (20)$$

The update of the parameters of the functions depends on the function prototypes used. For the functions

$$f_i = b_{i0} + b_{i1} \cdot x, \quad (21)$$

$i = 1, \dots, c$ , used in the Takagi–Sugeno system (7)–(8) in this paper, the optimal function parameters are

$$B_i = (X^T \Delta_i X)^{-1} X^T \Delta_i Y, \quad (22)$$

where

$$\Delta_i = \begin{pmatrix} u_{i1} & 0 & \dots & 0 \\ 0 & u_{i2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{in} \end{pmatrix}. \quad (23)$$

Optimization of the FCRM model can be done by randomly initializing  $U \in M_{fcn}$  (11), and then perform  $t$  steps of alternatingly updating  $B_1, \dots, B_c$  by (22) and  $U$  by (14) with  $D$  from (19).

Again, for the input parts  $X$  of the training data set  $Z$ , the values of the membership functions  $\mu_1, \dots, \mu_c$  can be immediately obtained from  $U$ . A generalization to  $x \in \mathbb{R}^p$  is not trivial and omitted here.

## 5 Global fuzzy input partitions

As we will see in the experiments later, a disadvantage of the FCRM model is that it produces highly non–convex partitions  $U$ , i.e. the memberships highly fluctuate as we move through the input space. To avoid this, we explicitly require the memberships to be of the form (14), where we use Euclidean distances  $d_{ik} = \|v_i^{(x)} - x_k\|$  for simplicity now. The only free parameters in this model are the cluster centers, more precisely the projections of the Gustafson Kessel cluster centers to the input space,  $V^{(x)}$ . The matrix  $B$

of parameters of the conclusion functions is computed by (17) and (18), with  $A_1, \dots, A_c$  from (12) and  $S_1, \dots, S_c$  from (13), just as in the Gustafson–Kessel model. We want to minimize the quadratic approximation error of the resulting Takagi–Sugeno system,

$$J_{\text{TS}} = \frac{1}{n} \sum_{k=1}^n \|y_k - \sum_{i=1}^c u_{ik} f_i(x_k)\|^2. \quad (24)$$

Notice that the definition of the membership functions by (14) guarantees that

$$\sum_{i=1}^c u_{ik} = 1, \quad (25)$$

so the denominator from (9) can be omitted here. Also notice the difference between  $J_{\text{TS}}$  (24) and  $J_{\text{FCRM}}$  (20) that reminds us that FCRM minimizes a *clustering criterion* but not the *quadratic approximation error* of the corresponding Takago–Sugeno system! To find the optimal cluster centers  $V^{(x)}$  for global input partitions, we have to optimize  $J_{\text{TS}}$  (24) with respect to  $V^{(x)}$ . This optimization can be done in many ways, e.g. using gradient descent, Newton’s method, or quasi–Newton optimization. Here, we use the Nelder–Mead simplex search method as described in [7] that was also used for solving the fuzzy assignment problem in [13]. With this optimization algorithm we always obtained satisfactory results.

## 6 Local fuzzy input partitions

Fig. 2a shows the global membership functions (14) for  $V^{(x)} = \{-0.8, 0, 0.1, 0.2\}$  and  $m = 2$  in  $X = [-1, 1]$ . The membership functions are smooth, but they are not convex. Consider in particular the membership function for the cluster at  $v_3^{(x)} = 0.1$  (dotted): Around  $x \approx -0.2$  it increases to about 0.2, then decreases to zero at  $x = 0$ , then increases to one at  $x = 0.1$ , then decreases to zero again at  $x = 0$ , and finally increases to  $u \rightarrow 0.25$  for  $x \rightarrow \infty$ . The reason for these non–convex shapes is that each membership value depends of all cluster centers  $V^{(x)}$ . To avoid these non–convex shapes we locally restrict the influence of the cluster centers, so that each membership value only depends of the immediately adjacent cluster centers. To do so, we sort the (one–dimensional) cluster centers, so that  $v_1^{(x)} < v_2^{(x)} < \dots < v_c^{(x)}$ . Data points  $x_k \leq v_1^{(x)}$  are then assigned to cluster 1,  $u_{1k} = 1$  and  $u_{2k} = \dots = u_{ck} = 0$ , and data points  $x_k \geq v_c^{(x)}$  are assigned to cluster  $c$ ,  $u_{1k} = \dots = u_{(c-1)k} = 0$  and  $u_{ck} = 1$ . This leads to so–called *shoulder* membership functions. For all other data points  $x_k$  with  $v_1^{(x)} < x_k < v_c^{(x)}$  we consider only the two neighboring cluster centers, say  $v_j^{(x)}$  and  $v_{j+1}^{(x)}$ , where  $v_j^{(x)} \leq x_k \leq v_{j+1}^{(x)}$ , and define

$$u_{jk} = 1 / \left( 1 + \left( \frac{d_{jk}}{d_{(j+1)k}} \right)^{\frac{2}{m-1}} \right), \quad (26)$$

$$u_{(j+1)k} = 1 / \left( 1 + \left( \frac{d_{(j+1)k}}{d_{jk}} \right)^{\frac{2}{m-1}} \right), \quad (27)$$

and  $u_{ik} = 0$  for  $i \notin \{j, j+1\}$ . Fig. 2b shows these local membership functions for  $V^{(x)} = \{-0.8, 0, 0.1, 0.2\}$  and  $m = 2$  in  $X = [-1, 1]$ . The local membership functions

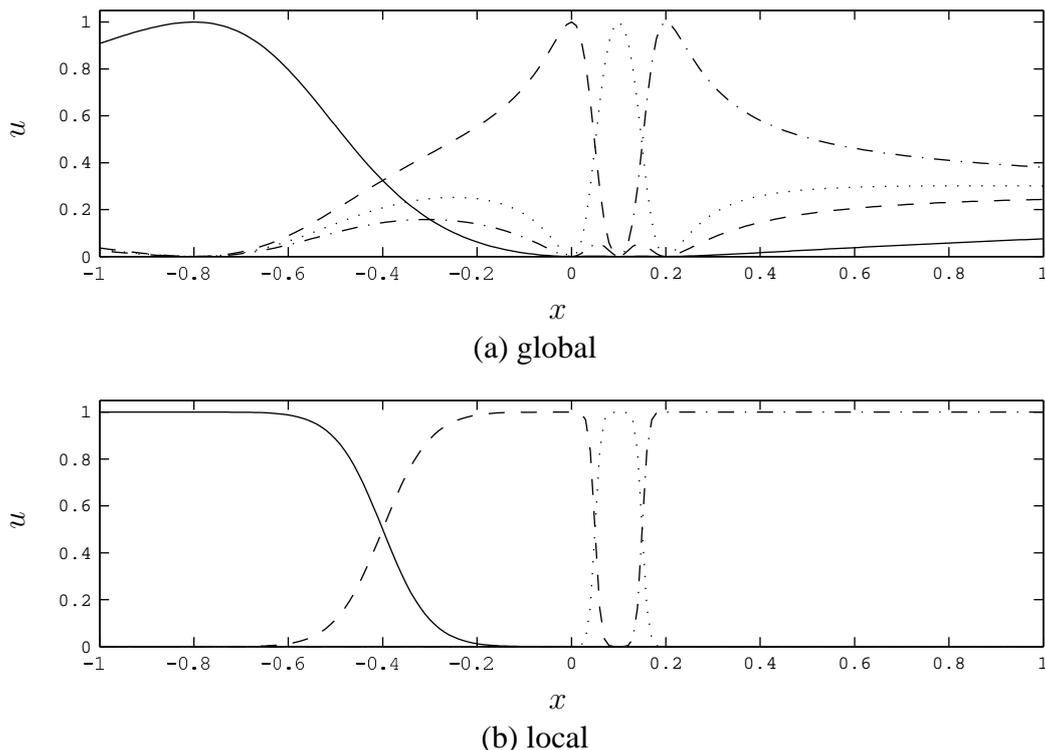
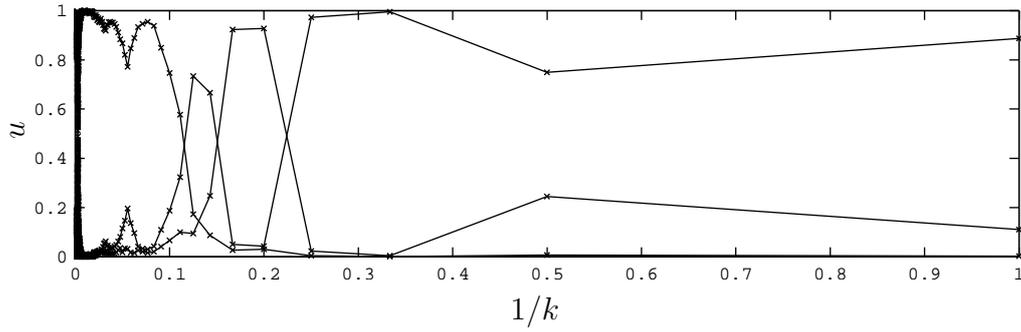


Figure 2: Global and local membership functions.

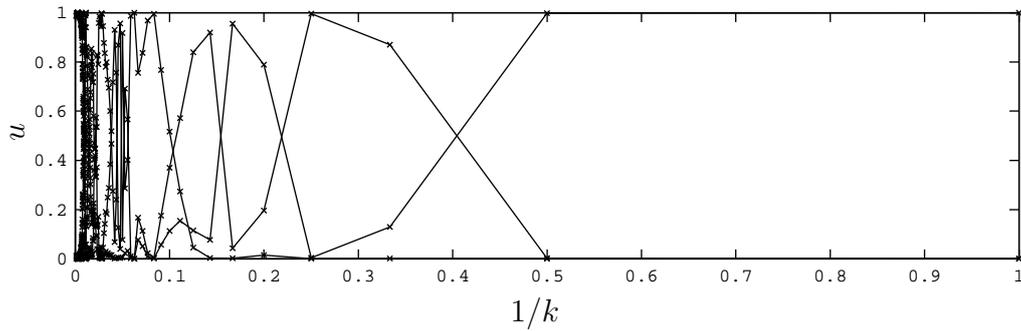
are smooth and convex, just as intended. Notice however, that the extension of this approach to multi-dimensional inputs is not trivial! Now we want to minimize the quadratic approximation error of the Takagi–Sugeno system (24) with *local* input partitions. This can be done in the same way as with the global fuzzy input partitions, again by using the Nelder–Mead simplex search method. The only difference is that the computation of the partition matrix  $U$  is done now by (26) and (27) instead of (14).

## 7 Experiments

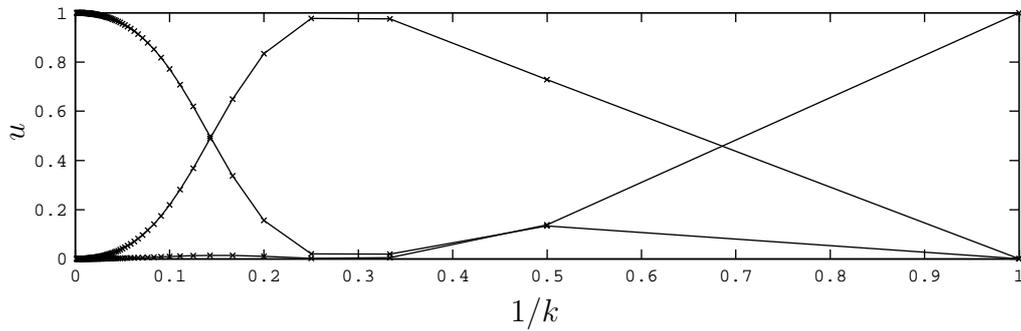
We applied all four fuzzy modeling methods to both the German and the international city data sets. For both clustering methods we ran  $t = 100$  iterations, and for the Nelder–Mead simplex search method we used an error threshold of  $10^{-4}$  for termination. Fig. 3 shows the membership functions obtained for the German city data set,  $c = 3$ . As pointed out above, the membership functions generated by clustering (a,b) show fluctuations, in particular the FCRM membership functions (b). Fig. 4 shows the (training) outputs of the Takagi–Sugeno systems for the German city data set. The original data are marked as  $\times$ , and the model outputs by the numbers 1, 2, and 3, indicating in which cluster the data point has its maximum membership, or which rule has the highest firing strength, respectively. Notice that in the sequences of cluster (or rule) assignments are monotonically increasing ( $1 \rightarrow 2 \rightarrow 3$ ) only in (c) and (d), while they toggle between 2 and 3 in (a) and (b). Also notice that the deviations between the data and the model outputs are quite high in (c). This model error is shown in Fig. 5 for all four cases. The lowest model errors are achieved in (a) and (d).



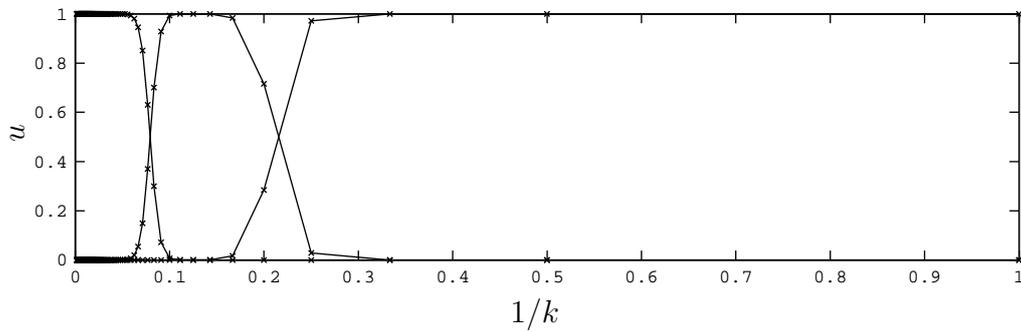
(a) Gustafson–Kessel



(b) fuzzy c–regression models

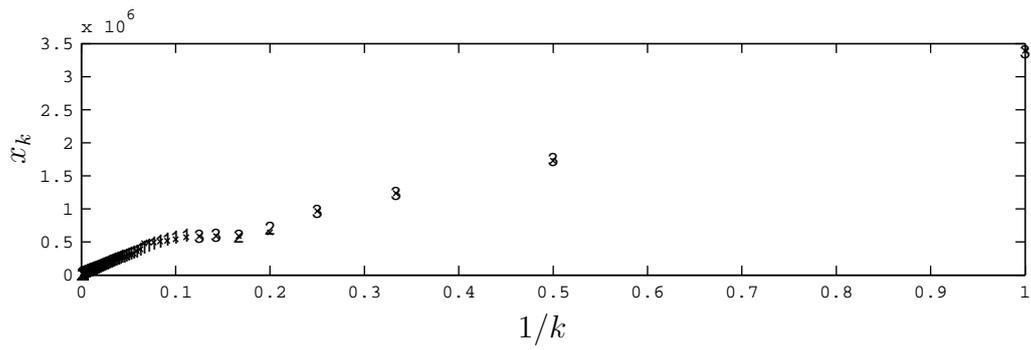


(c) global fuzzy input partition

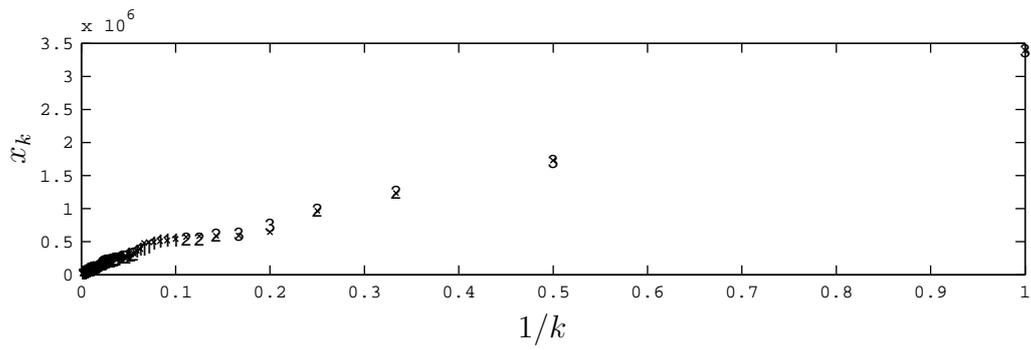


(d) local fuzzy input partition

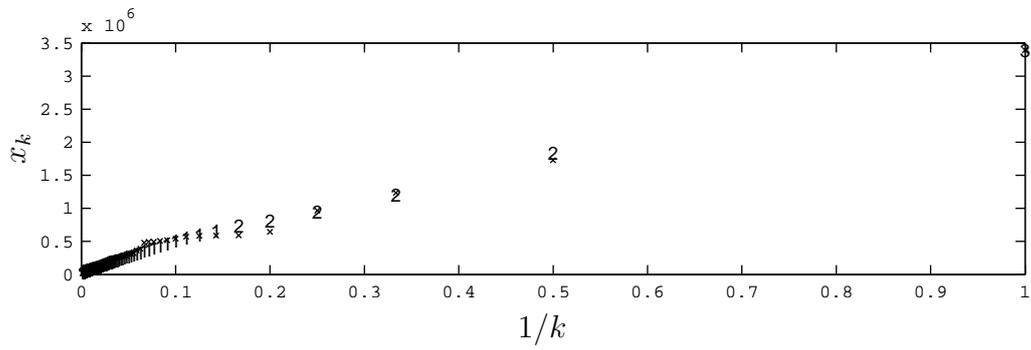
Figure 3: Membership functions (German cities).



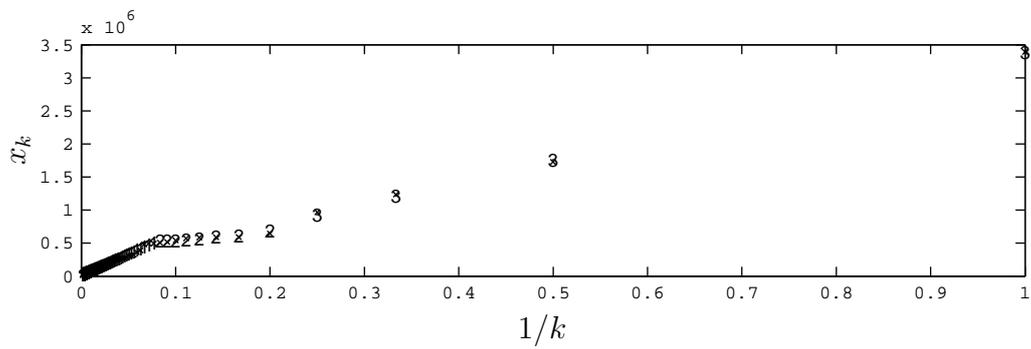
(a) Gustafson-Kessel



(b) fuzzy c-regression models

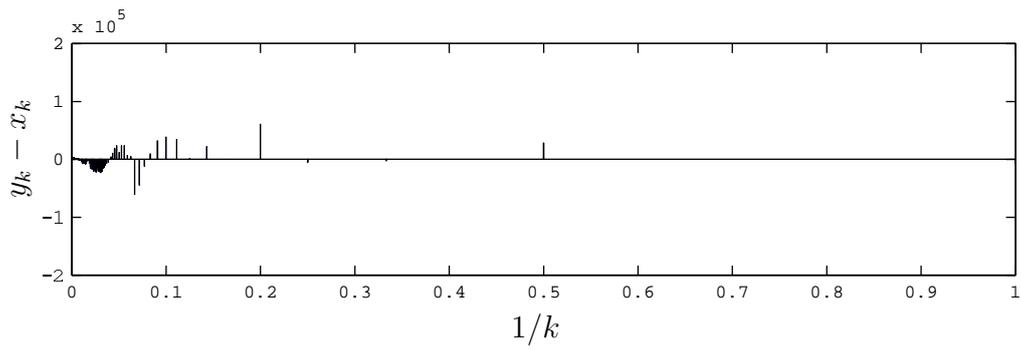


(c) global fuzzy input partition

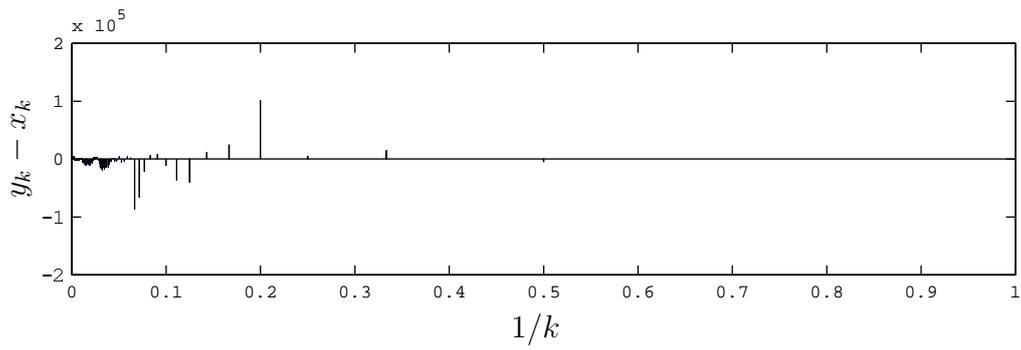


(d) local fuzzy input partition

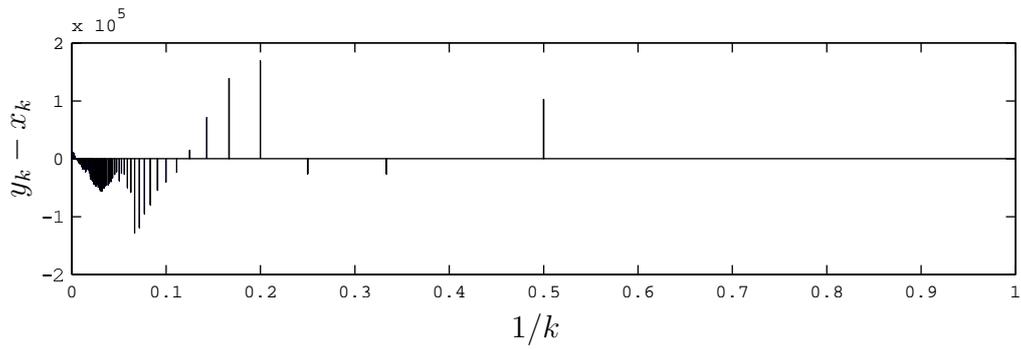
Figure 4: Model outputs (German cities).



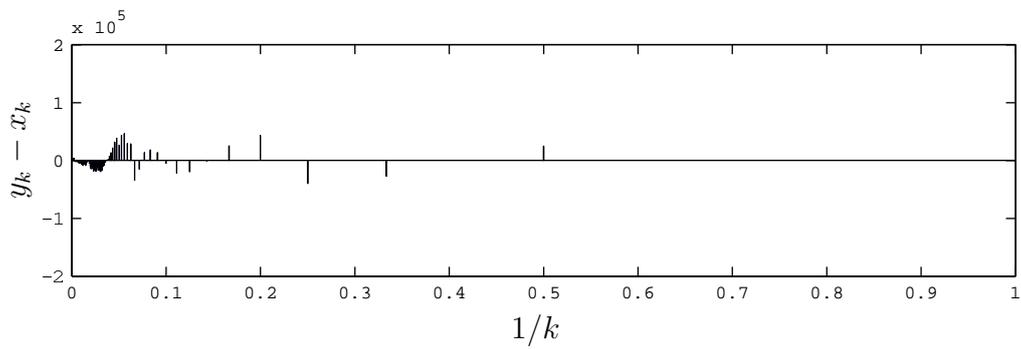
(a) Gustafson-Kessel



(b) fuzzy c-regression models

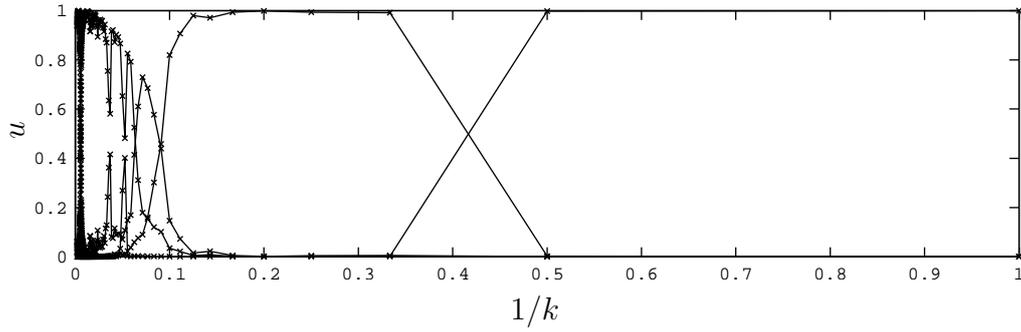


(c) global fuzzy input partition

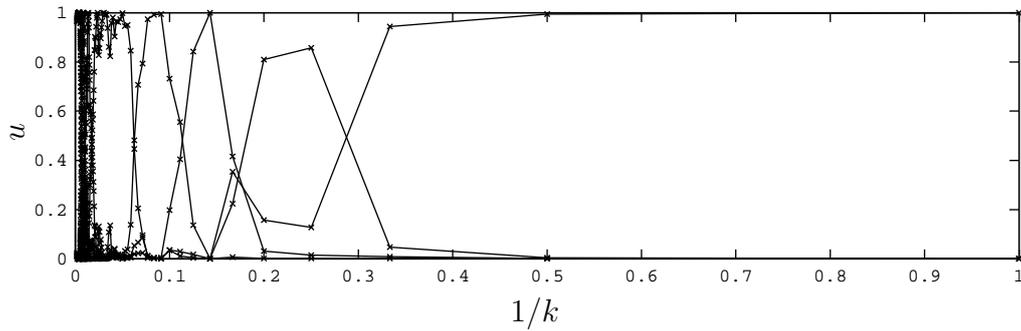


(d) local fuzzy input partition

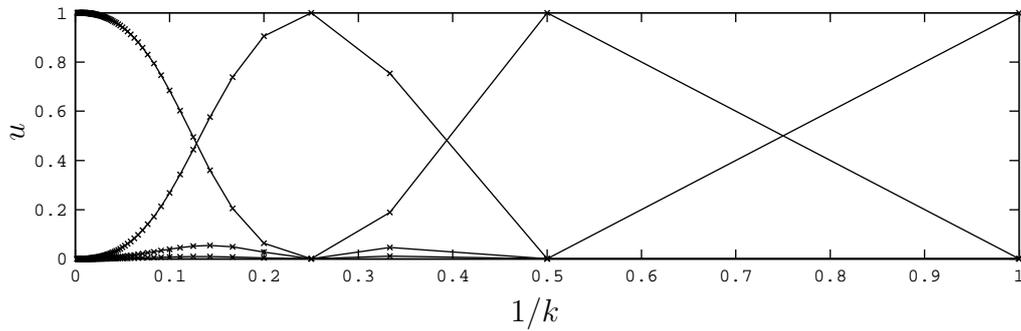
Figure 5: Model errors (German cities).



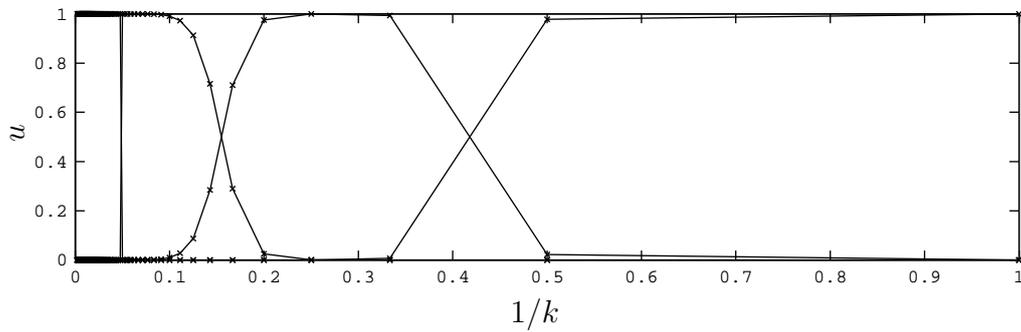
(a) Gustafson-Kessel



(b) fuzzy c-regression models



(c) global fuzzy input partition



(d) local fuzzy input partition

Figure 6: Membership functions (international cities).

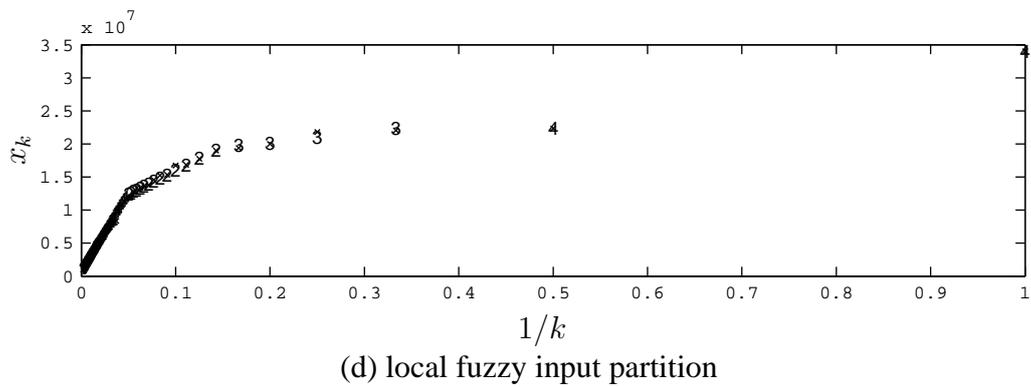
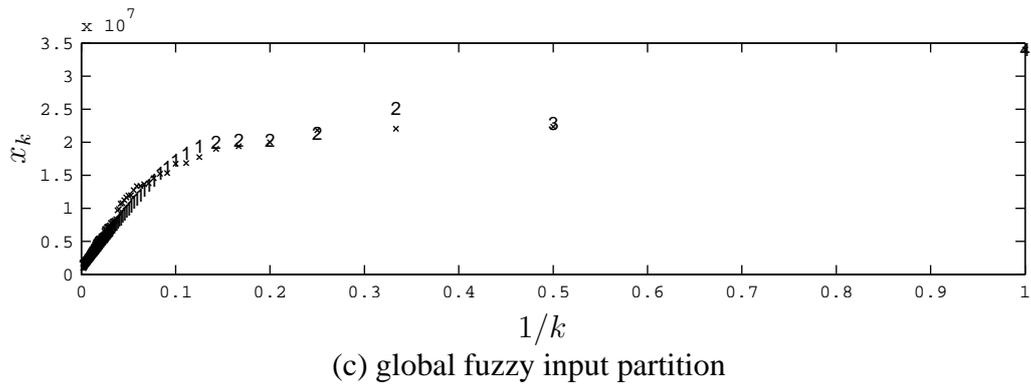
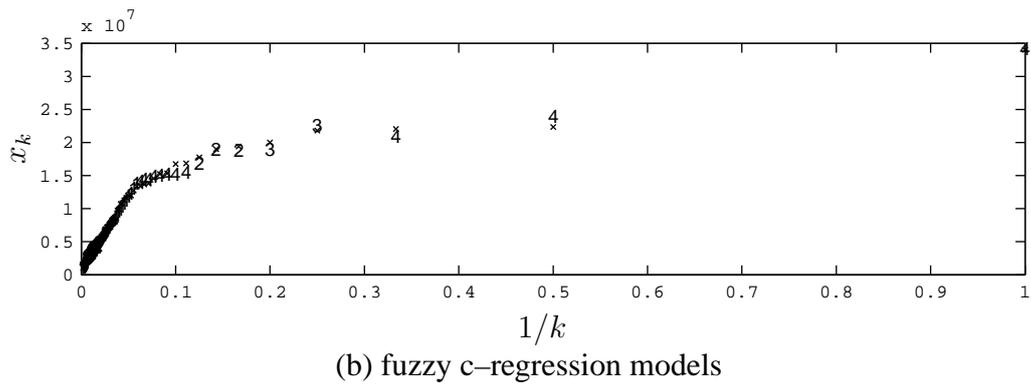
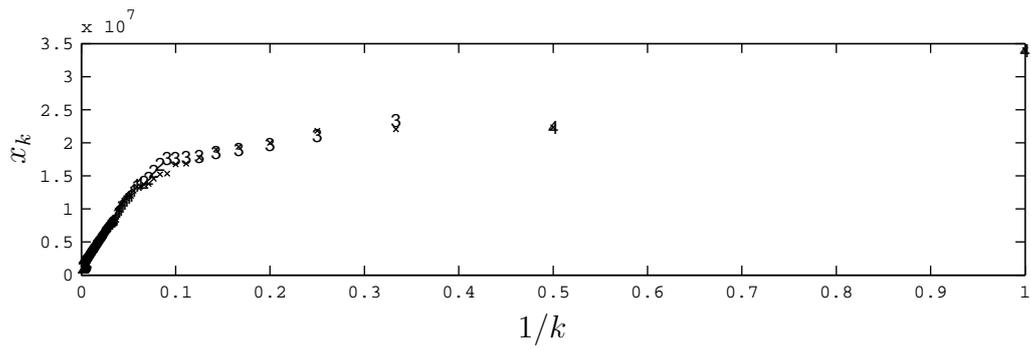


Figure 7: Model outputs (international cities).

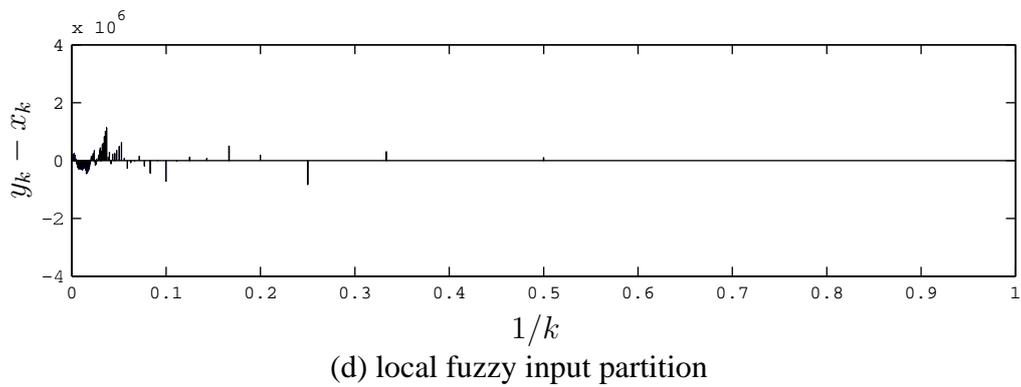
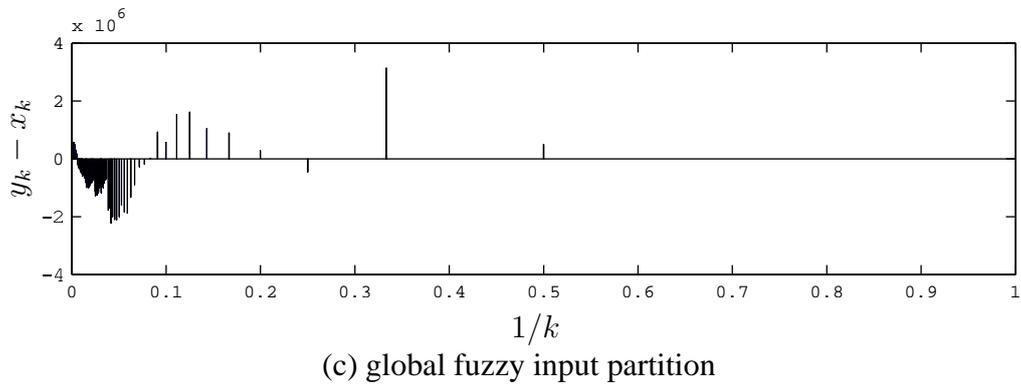
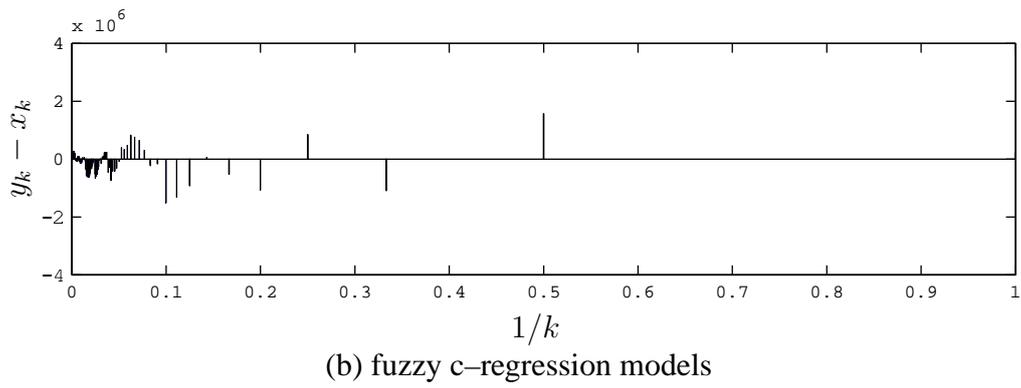
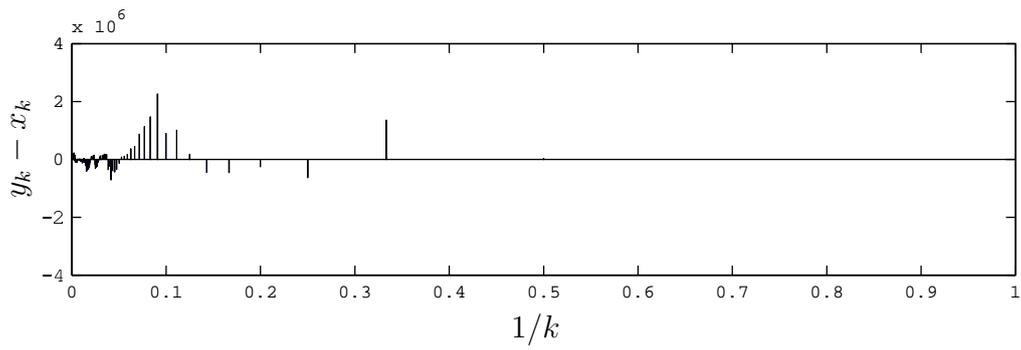


Figure 8: Model errors (international cities).

Figs. 6–8 show the corresponding results for the international city data set,  $c = 4$ . Fig. 6 shows that also for this data set (a) and (b) produce fluctuating membership functions. Fig. 7 shows again that the sequence of cluster (or rule) assignments is only monotonic in (c) and (d), and that the model error is highest in (c). Fig. 8 confirms this, and shows that the lowest model error is obtained in (d).

## 8 Conclusions

The main task of this paper was to model power law distributed data sets. As example data sets we considered the population numbers of German and international cities. Although these data sets are suspected to follow Zipf's law, they do not appear as a straight line in a diagram displaying the size against the inverse rank. Instead, we have shown that they can be accurately modeled by mixtures of multiple power law distributions with different exponents that appear as (fuzzy) combinations of piecewise linear functions in the size–inverse–rank diagram. These piecewise linear functions imply a *clustering* of the power law data: points on the same line belong to the same cluster.

To find an appropriate assignment of the data to the line pieces we applied four different methods to build first order Takagi–Sugeno fuzzy models:

- (a) Gustafson–Kessel clustering in the input–output product space, projection of the memberships to the input space, and computation of the conclusion function parameters from the cluster eigenvectors,
- (b) fuzzy  $c$ –regression models clustering with linear function prototypes,
- (c) minimization of the training error of the Takagi–Sugeno system using linear conclusion functions and (global)  $c$ –means partition functions, and
- (d) minimization of the training error of the Takagi–Sugeno system using linear conclusion functions and a new kind of membership functions that we call *local  $c$ –means* partition functions.

The experiments have shown that (a) and (b) produce fluctuating and therefore not very intuitive membership functions, whereas the membership functions obtained with (c) and (d) are smooth, convex, and therefore much more suitable for this clustering task. Methods (a) and (b) yielded acceptable model errors, but the lowest model errors were obtained with (d). Thus, the proposed modeling approach (d) that minimizes the training error of a Takagi–Sugeno system with linear conclusion functions and local  $c$ –means partition functions is the best option concerning both aspects, convexity and accuracy. We are currently working on the application of this approach to a real–world application with power law distributed data sets.

## References

- [1] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

- [2] T. Brinkhoff. The principal agglomerations of the world, 2005–01–30. Available at <http://www.citypopulation.de>, January 2005.
- [3] Statistisches Bundesamt. *Statistisches Jahrbuch für die Bundesrepublik Deutschland, Kapitel Bevölkerung*. Wiesbaden, 2004.
- [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. Switching regression models and fuzzy clustering. *IEEE Transactions on Fuzzy Systems*, 1(3):195–204, August 1993.
- [6] F. Höppner, F. Klawonn, R. Kruse, and T. A. Runkler. *Fuzzy Cluster Analysis — Methods for Image Recognition, Classification, and Data Analysis*. Wiley, 1999.
- [7] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9(1):112–147, 1998.
- [8] P. C. Mahalanobis. On the generalized distance in statistics. *Proceedings of the National Institute of Sciences of India*, 12:49–55, 1936.
- [9] N. R. Pal, K. Pal, J. C. Bezdek, and T. A. Runkler. Some issues in system identification using clustering. In *IEEE International Conference on Neural Networks*, pages 2524–2529, Houston, June 1997.
- [10] V. Pareto. *Cours d'économie politique*. Droz, Geneva, 1896.
- [11] T. A. Runkler. Automatic generation of first order Takagi–Sugeno systems using fuzzy c–elliptotypes clustering. *Journal of Intelligent and Fuzzy Systems*, 6(4):435–445, 1998.
- [12] T. A. Runkler. Nonlinear system identification with global and local soft computing methods. In *GMA-Workshop Fuzzy Control*, pages 163–176, Dortmund, October 2000.
- [13] T. A. Runkler. The fuzzy component assignment problem. In *GMA/GI Workshop Fuzzy Systems and Computational Intelligence, Dortmund, Germany*, pages 149–162, November 2004.
- [14] T. A. Runkler and R. H. Palm. Identification of nonlinear systems using regular fuzzy c–elliptotype clustering. In *IEEE International Conference on Fuzzy Systems*, pages 1026–1030, New Orleans, September 1996.
- [15] M. Sugeno and T. Yasukawa. A fuzzy–logic–based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems*, 1(1):7–31, February 1993.
- [16] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, 1985.
- [17] G. K. Zipf. *Psycho-Biology of Languages*. MIT Press, 1965.

# Visualizing Single Fuzzy c-Means Clusters

**Frank Rehm**

German Aerospace Center  
Braunschweig  
frank.rehm@dlr.de

**Frank Klawonn**

University of Applied Sciences  
Braunschweig/Wolfenbüttel  
f.klawonn@fh-wolfenbuettel.de

**Rudolf Kruse**

University of Magdeburg  
kruse@iws.cs.uni-magdeburg.de

## Abstract

Evaluation of clustering partitions is a crucial step in data processing. A multitude of measures exists, which - unfortunately - give for one data set various results. In this paper we present a visualization technique to visualize single clusters of high-dimensional data. Our method maps a single cluster to the plane trying to preserve the membership degrees. The resulting scatter plot illustrates separation of the respecting cluster and the need of additional prototypes as well. Since clusters will be visualized individually, additional prototypes can be added locally where they are needed.

## 1 Introduction

Partitioning datasets is an important task in many domains such as costumer segmentation, organizing textual information or gene expression analysis.

The fuzzy c-means algorithm (FCM) is a widely-used clustering technique resulting in a set of prototypes which represent the respective clusters and a partition matrix which contains for each feature vector the membership degrees to every single cluster. As most of the prototype-based clustering algorithms, FCM minimizes an objective function [1]. As a matter of fact, clustering algorithms always fit the clusters to the data, even if the cluster structure is not adequate for the problem. Thus, the quality of a partition cannot be verified meaningfully by the value of the objective function. Therefore, many validity measures are developed to analyze the adequateness of clustering results [10, 11, 13, 14].

Most of these measures evaluate the partitioning by means of analyzing the partition matrix resulting in a single value, which is of course associated with some loss of information. Primarily, global validity measures cannot give hints which part of the data should be explored more in detail.

VAT, Visual Assessment of Cluster Tendency, is a tool to visualize pairwise dissimilarity information of objects  $X = \{x_1, \dots, x_n\}$  as a square image with  $n^2$  pixels. VAT reorders the data objects so that the image highlights potential cluster structures. As a modification of this, bigVAT allows the visualization for larger data sets [4]. VCV, Visual Cluster Validity, is related to VAT, but takes the inter-datum distances into account that come from partitioning the data set [3].

Recently, two visualization methods using the information of a fuzzy clustering partition were presented in [11]. One method arranges the membership degrees to the respective cluster over the distances to the according prototype vector. Each cluster is represented in a single plot. The other method represents a whole fuzzy partition by plotting the highest membership degree to each feature vector over the corresponding second highest membership degree. Such a plot gives an overall impression of a partition.

We propose in this paper a visualization technique to visualize single clusters of high-dimensional data. Our method maps a single cluster to the plane trying to preserve the fuzzy membership degrees.

Section 2 gives a brief description of fuzzy c-means clustering. Section 3 describes our approach followed by some implementation aspects in section 4. In Section 5 we show some results on a synthetic data set and a reference data set, respectively. Section 6 contains our conclusion.

## 2 Fuzzy Clustering

Generally, fuzzy clustering algorithms partition a data set into several clusters as minimizing an objective function  $J$  that describes the sum of weighted distances  $d_{ij}$  between  $c$  prototypes vectors  $v_i$  and  $n$  feature vectors  $x_j$  of the feature space  $\mathbb{R}^p$

$$J = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d_{ij}. \quad (1)$$

Prototype vectors represent the respecting clusters by their location in the cluster's centre. By means of the fuzzifier  $m \in (1, \infty]$  one can control how much the clusters overlap. Widely overlapping clusters, which can be obtained with high values for  $m$ , will be reflected by almost equal membership degrees  $u_{ij}$  to every cluster. Rather crisp partitions can be found with small values for  $m$ . Usually, the fuzzifier is set to  $m = 2$ . In order to avoid the trivial solution assigning no data to any cluster by setting all  $u_{ij}$  to zero and avoiding empty clusters, the following constraints are required:

$$u_{ij} \in [0, 1] \quad 1 \leq i \leq c, \quad 1 \leq j \leq n \quad (2)$$

$$\sum_{i=1}^c u_{ij} = 1 \quad 1 \leq j \leq n \quad (3)$$

$$0 < \sum_{j=1}^n u_{ij} < n \quad 1 \leq i \leq c. \quad (4)$$

When the Euclidian norm

$$d_{ij} = d^2(\vec{v}_i, \vec{x}_j) = (\vec{x}_j - \vec{v}_i)^T (\vec{x}_j - \vec{v}_i)$$

is used as distance measure for distances between prototype vectors  $v_i$  and feature vectors  $x_j$ , the fuzzy clustering algorithm is called fuzzy c-means algorithm. Other distance measures can be applied resulting in clustering techniques which can adopt

different cluster shapes [8, 9]. With the Euclidian distance measure the fuzzy c-means algorithm finds approximately equally sized spherical clusters.

The minimization of the functional (1) represents a nonlinear optimization problem that is usually solved by means of Lagrange multipliers, applying an alternating optimization scheme [2]. This optimization scheme considers alternatingly one of the parameter sets, either the membership degrees

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{1}{m-1}}} \quad (5)$$

or the prototype parameters

$$\vec{v}_i = \frac{\sum_{j=1}^n (u_{ij})^m \vec{x}_j}{\sum_{j=1}^n (u_{ij})^m} \quad (6)$$

as fixed, while the other parameter set is optimized according to equations (5) and (6), respectively, until the algorithm finally converges.

### 3 Visualizing Single Clusters

We propose in this paper to visualize single clusters by projection of the data points onto the plane under the constraint that the membership degrees are preserved. To achieve this objective, we adopt the noise distance aspect of the noise clustering technique [5]. Noise clustering is based on the introduction of an additional noise cluster, that is supposed to contain all feature vectors that are about a certain distance, the noise distance  $\delta$ , away from all other prototype vectors. This means, that the prototype for the noise cluster  $v_c$  has no parameters. The clustering scheme differs only in one point from FCM. When calculating the membership degrees according to equation 5 the distance to the noise cluster is the fixed constant value  $d_{cj} = \delta^2$ . The proper specification of  $\delta$  is discussed in [5, 12].

In the following we consider two different approaches to visualize single clusters. For our first single cluster visualization approach we assume that only one regular cluster exists - the one we intent to visualize. Thus, we consider only the membership degrees for the respecting cluster  $i$  obtained by the fuzzy c-means clustering with  $c$  prototypes and sum the remaining membership degrees and judge the resulting cluster as noise.

With the objective to place the cluster in the plane, we need for each data point two coordinates. Note, that the constraint for the projection is not to preserve the distances  $d_{ij}$  but the membership degrees  $u_{ij}$ . So we are free to define the noise distance as  $\delta = 1$ . We propose to place the desired cluster  $i$  around  $(0, 0)$  and, according to the defined noise distance we place the noise cluster at  $(1, 0)$ . The only problem is the placement of the data points in the plane. The idea is, to compute the distances to these clusters by means of the membership degrees. To achieve this we consider equation (5) and relate the membership degree to the respective cluster  $u_{ij}$  with the membership degree to the noise cluster. If we only have two

clusters than automatically the other cluster forms the noise cluster. Otherwise, the membership degrees of all other clusters will be summed to form the noise cluster

$$\frac{u_{ij}}{u_{noisej}} = \frac{\frac{1}{\sum_{k=1}^c \left(\frac{d_{ij}}{d_{kj}}\right)^{\frac{1}{m-1}}}}{\frac{1}{\sum_{k=1}^c \left(\frac{d_{noisej}}{d_{kj}}\right)^{\frac{1}{m-1}}}}. \quad (7)$$

Since we merged all clusters  $v_k \neq v_i$  to one noise cluster, the number of clusters is always  $c = 2$ . Therefore, we can simplify equation 7 as follows:

$$\frac{u_{ij}}{u_{noisej}} = \frac{\frac{1}{\left(\frac{d_{ij}}{d_{ij}}\right)^{\frac{1}{m-1}} + \left(\frac{d_{ij}}{d_{noisej}}\right)^{\frac{1}{m-1}}}}{\frac{1}{\left(\frac{d_{noisej}}{d_{ij}}\right)^{\frac{1}{m-1}} + \left(\frac{d_{noisej}}{d_{noisej}}\right)^{\frac{1}{m-1}}}} \quad (8)$$

$$\frac{u_{ij}}{u_{noisej}} = \frac{\frac{1}{1 + \left(\frac{d_{ij}}{d_{noisej}}\right)^{\frac{1}{m-1}}}}{\frac{1}{\left(\frac{d_{noisej}}{d_{ij}}\right)^{\frac{1}{m-1}} + 1}} \quad (9)$$

$$\frac{u_{ij}}{u_{noisej}} = \frac{1 + \left(\frac{d_{noisej}}{d_{ij}}\right)^{\frac{1}{m-1}}}{1 + \left(\frac{d_{ij}}{d_{noisej}}\right)^{\frac{1}{m-1}}} \quad (10)$$

$$\frac{u_{ij}}{u_{noisej}} = \left(\frac{d_{noisej}}{d_{ij}}\right)^{\frac{1}{m-1}}. \quad (11)$$

Setting  $d_{noisej} = \delta = 1$  and redefining  $d_{ij}$  by  $d_{0j}$  leads to

$$\frac{u_{ij}}{u_{noisej}} = \left(\frac{1}{d_{0j}}\right)^{\frac{1}{m-1}}. \quad (12)$$

Solving equation (12) for  $d_{0j}$  we end up with

$$d_{0j} = \left(\frac{u_{noisej}}{u_{ij}}\right)^{m-1}. \quad (13)$$

With equation (13) one can compute the distance of each feature vector  $x_j$  to the cluster  $i$ , so that it is possible to draw a circle around  $(0,0)$  as one hint for the feature vector's position in the plane. With the distance to the other cluster  $(1,0)$ , one could draw another circle around the cluster centre. The intersection point of these two circles would be the position of the new feature vector in the plane. This can be analogous done by changing the perspective. We apply equation (13) considering the summed noise membership degree as the degree to the  $(1,0)$ -cluster and  $u_{ij}$  for the  $(0,0)$ -cluster leading to

$$d_{noisej} = \left(\frac{u_{ij}}{u_{noisej}}\right)^{m-1}. \quad (14)$$

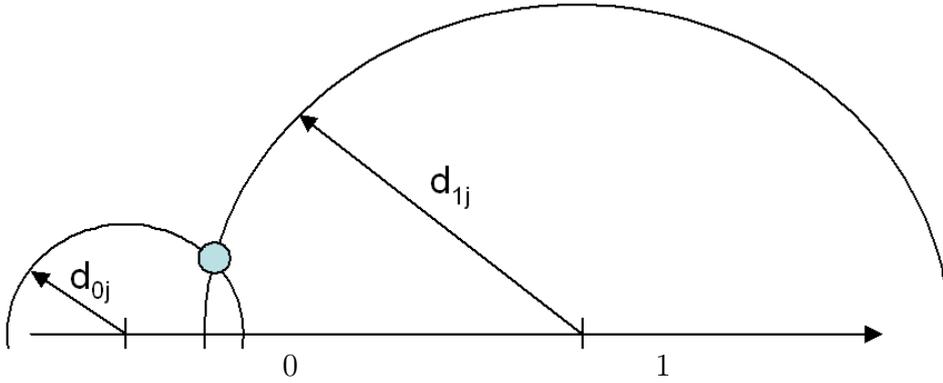


Figure 1: Placement of  $x_j$  in the Plane

Figure 1 illustrates this approach. The small circle, that represents the potential coordinates of  $x_j$ , can be drawn with distance  $d_{0j}$  obtained from equation (13). Analogous, the bigger circle can be drawn with  $d_{1j}$  that we get with equation (14). The intersection point of these two circles represents the feature vector  $x_j$  in the plane. The above approach leads to good results when visualizing single clusters of partitions with a small number of clusters  $c$ . Clustering with larger number of prototypes usually leads to a small number of high membership degrees and many border regions between two or more clusters. In these regions the prototypes compete for data points which results in smaller membership degrees. In most of the cases  $u_{noisej}$  will be higher than  $u_{ij}$  when clustering with many prototypes.

Much more meaningful might be a modification in that point, that cluster  $(1, 0)$  is not a noise cluster, but a virtual cluster that contains all feature vectors with the highest membership degree apart from  $u_{ij}$ . Let us denote the membership degree to the most competing cluster with  $u_{kj}$ . Applying equation (13) to compute  $d_{0j}$  we still have  $u_{noisej}$  which is now defined as  $u_{noisej} = 1 - u_{ij} - u_{kj}$ . For our second approach we replace equation (14) by the following

$$d_{1j} = \left( \frac{u_{kj}}{u_{noisej}} \right)^{m-1} \quad (15)$$

where  $u_{noisej} = 1 - u_{ij} - u_{kj}$  as mentioned above. This approach enables us to visualize some useful aspects: 1. which feature vectors can be assigned clearly to the respective cluster  $v_i$ , 2. if a feature vector can not be assigned to  $v_i$ , is there another cluster  $v_k$ , where the vector can be assigned to, 3. which feature vectors are near to one or more prototypes apart from  $v_i$  and  $v_k$ , 4. are there feature vectors that can not be assigned to any cluster clearly. Before section 5 illustrates these characteristics with some examples, we will make some implementation proposals in the following section.

## 4 Implementation Aspects

Apart from the clustering itself, which leads to the membership degrees  $u_{ij}$  another parameter affects the transformation, namely  $m$  (see equation 12, 13, 15). A priori, one would set  $m$  the same value as for the clustering. But it can be also useful to

modify this parameter. Practical tests have shown that in some cases, i.e. when a feature vector is very close to a prototype vector, no intersection point can be obtained in the plane and consequently the membership degrees to the respecting feature vector cannot be preserved exactly. The following rules handle such cases while trying to preserve membership degrees approximately:

```

if no intersection point then
   $\hat{x}_{2j} = 0$ 
  if  $\max(d_{0j}, d_{1j}) < 1$  then
     $\hat{x}_{1j} = d_{0j} / (d_{0j} + d_{1j})$ 
  else
    if  $d_{0j} > d_{1j}$  then
       $\hat{x}_{1j} = 1 + d_{1j}$ 
    else
       $\hat{x}_{1j} = -d_{0j}$ 
    end if
  end if
end if

```

Let us denote the transformed data set  $\hat{X}$ . With  $\hat{x}_{2j} = 0$  we define for a feature vector that is very close to one certain prototype vector a position on the  $x$ -axis on the plane. The rest of the rule tries to find the proper position for  $\hat{x}_j$  on the  $x$ -axis balancing the distances to cluster  $(0, 0)$  and cluster  $(1, 0)$ .

If the distance to both clusters is relatively small, say  $\max(d_{0j}, d_{1j}) < 1$ , than we compute a position between both clusters in relation to  $d_{0j}$  and  $d_{1j}$ . Otherwise, which means one or both clusters are about a distance of 1 or further away from the feature vector we distinguish whether cluster  $(0, 0)$  or cluster  $(1, 0)$  is nearer. If the distance of  $x_j$  to cluster  $(0, 0)$  is higher than the distance to cluster  $(1, 0)$  than  $\hat{x}_j$  will be placed to the right of cluster  $(1, 0)$  at  $\hat{x}_j = (1 + d_{1j}, 0)$ . If the distance  $d_{0j}$  to cluster  $(0, 0)$  is smaller than the distance  $d_{1j}$  to cluster  $(1, 0)$  than  $\hat{x}_j$  will be placed to the left of cluster  $(0, 0)$  at  $\hat{x}_j = (-d_{0j}, 0)$ . This concept enables us to place the data point at least accurate relative to the nearest cluster. However, it is not very interesting to know exactly how far away the feature vector is, since the distance is quite large in fact.

With these rules the membership degrees cannot be preserved exactly, but approximated intuitively. Alternatively, one can avoid this kind of approximation by modifying parameter  $m$  for the transformation process. Small values  $m \rightarrow 1$  prevent that no intersection point can be met. Otherwise, one can set higher values for  $m$  to force placements on the  $x$ -axis. Such transformations may be not that differentiated, but information can be reduced to some essential facts if needed. Generally, data points situated left from  $x_1 = 0.5$  can be assigned to cluster  $(0, 0)$ , while data points on the other side belong to another cluster.

## 5 Results

Let us first apply our visualization method to a synthetically data set. The cube data set (see figure 2(a)) consists of eight well separated clusters, which are in the

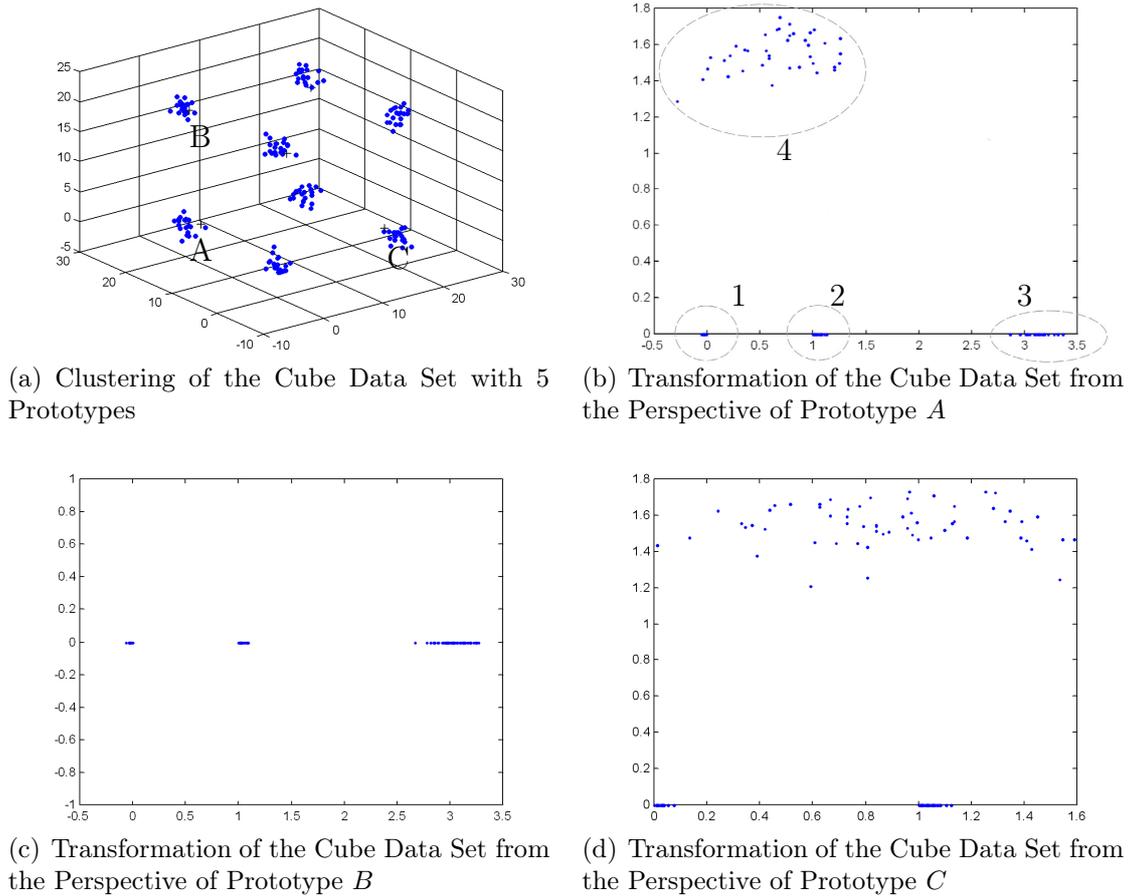


Figure 2: An Illustrative Example

corners of an imaginary 3-dimensional cube. A fuzzy c-means partition of the data set with five prototypes is shown in the figure. Of course, eight prototypes would be the best choice to partition the cube data set with. Thus, we can illustrate with this partition which information one can get from the visualization tool.

Figure 2(b) shows the transformation of the cube data set from the perspective of prototype A. Clearly four groups of data points can be observed (circled with a dashed line). The data points in group 1 are those, which can be clearly assigned to prototype A. Data points that are located in group 2 are those, which are not assigned to prototype A at all, but to any other prototype. Note, a transformation that only consists of these both groups is ideal. Group 3 stands for feature vectors, which are not assigned to prototype A and not to any other prototype. Instead, the data points have approximately the same membership degree to two or more prototype vectors (but not to prototype A). Group 4 represents feature vectors that have approximately the same membership degree to prototype A and another prototype.

Figure 2(c) shows the transformation of the cube data set from the perspective of prototype B. At first sight one can notice that group 4 is absent. That means in fact that no other prototype than prototype B has high membership degrees to the data points in group 1. A closer look reveals that the distance of prototype B to some misrepresented data is higher comparing to other prototypes, such as

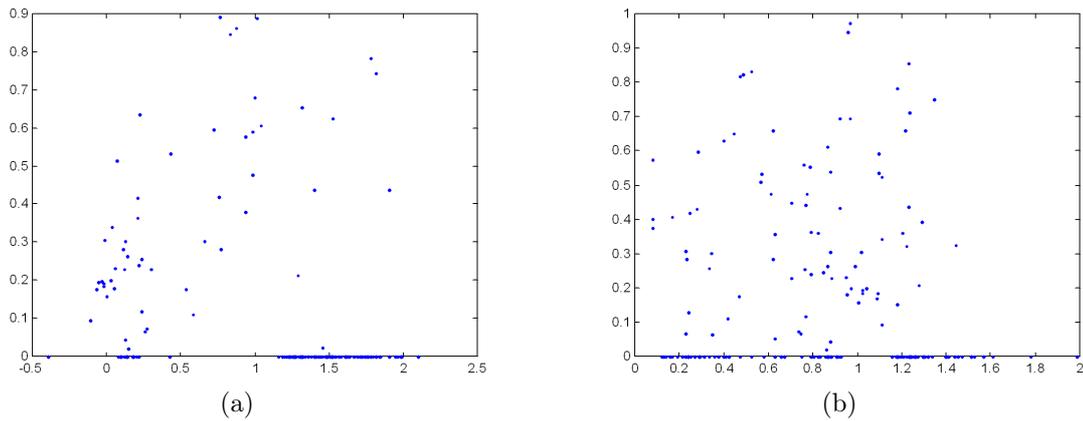


Figure 3: Some Transformations of Clusters of the Wine Data Set

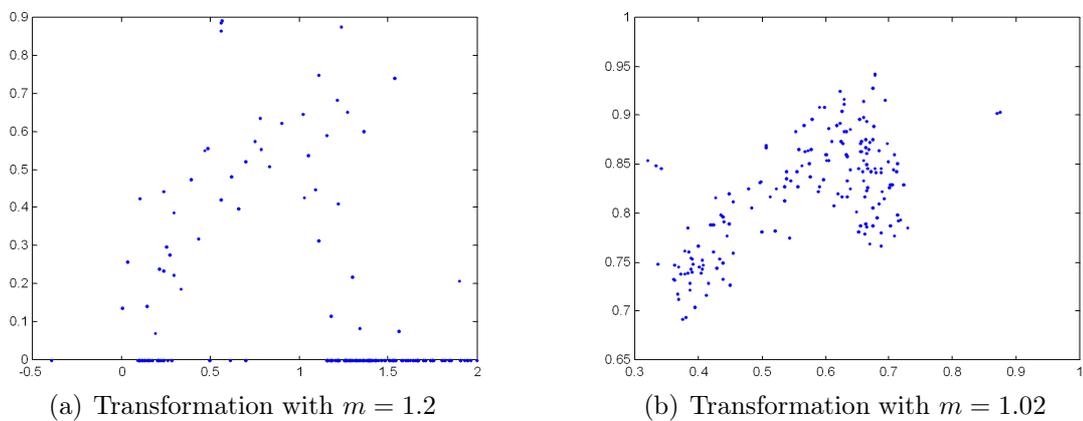


Figure 4: The Effect of Parameter  $m$

prototype  $A$  and  $C$ . All other data points that could contribute to a group 4 are clearly represented by some prototypes.

The transformation of the cube data set from the perspective of prototype  $C$  is shown in figure 2(d). Now group 3 is missing in the plot. This becomes evident, because all data points that are underrepresented are directly between prototype  $C$  and at least one other prototype. As we have discussed above, group 3 only occurs when data points have low membership degrees to the regarding prototype and approximately equal membership degrees to two or more other prototypes. Since prototype  $C$  is at least as near as other prototypes, group 3 cannot be formed.

Figure 3 and figure 4 show some results on the well known wine data set. The wine data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. Figure 3 shows exemplarily two clusters of the partitioning with four prototypes using a fuzzifier  $m = 1.2$ . The left one is a visualization of a quite compact cluster. Data points left from 0.5 on the  $x$ -axis whose component on the  $y$ -axis is greater than zero have only small membership degrees to the cluster  $(1, 0)$  even if their distance to cluster  $(0, 0)$  seems to be fear. This is due to the relatively small fuzzifier. The cluster shown in figure

3(b) is much more overlapping other clusters as the points on the  $x$ -axis fairly in the middle between both cluster indicate. As mentioned above, using small values for  $m$  leads to rather sensitive transformations. Even a relative small membership degree to a certain cluster attracts the data point in the transformation. To smooth this effect it is advisable to decrease  $m$  for the transformation or increase  $m$  for the clustering if possible. The effect of decreasing  $m$  for the transformation is shown in figure 4. While (b) shows the transformation of a cluster of the wine data set with  $m = 1.2$ , the figure (b) on the left shows the same cluster transformed with  $m = 1.02$ . The changeover from cluster  $(0, 0)$  to cluster  $(1, 0)$ , which is the imaginary line at 0.5 through the  $x$ -axis, is rather sparse. This fact indicates a compact cluster with only few feature vectors which cannot be assigned that clear to any cluster.

## 6 Conclusion

We presented in this paper a new method to visualize fuzzy clustering partitions on the plane. The visualization reveals whether a cluster is compact and if there is some data from the perspective of the respective cluster that is not well represented. Our results on two data sets are promising. Subject of future work will be the development of an appropriate evaluation method.

## References

- [1] Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York, 1981.
- [2] Bezdek, J.C.: A Convergence Theorem for the Fuzzy ISODATA Clustering Algorithms, IEEE Trans, PAMI, 2, pp. 1–8, 1980.
- [3] Hathaway, R.J., Bezdek, J.C.: Visual cluster validity for prototype generator clustering models. Pattern Recognition Letters, Volume 24, Issue 9-10, USA, 2003.
- [4] Huband, J.M., Bezdek, J.C., Hathaway, R.J.: bigVAT: Visual assessment of cluster tendency for large data sets. Pattern Recognition Letters, (in press).
- [5] Dave, R.N.: Characterization and detection of noise in clustering, Pattern Recognition Letters, 12, pp. 657-664, 1991.
- [6] Dave, R.N., Krishnapuram, R.: Robust Clustering Methods: A Unified View, IEEE Transactions on Fuzzy Systems, 5, pp. 270–293, 1997.
- [7] Dave, R.N., Sumit, S.: On Generalizing the Noise Clustering Algorithms. In Proceedings of the 7th Fuzzy Systems Association World Congress (IFSA'97), 3, pp. 205–210, 1997.
- [8] Gath, I., Geva, A.B.: Unsupervised optimal fuzzy clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11, pp. 773–781, 1989.

- [9] Gustafson, D.E., Kessel, W.C.: Fuzzy Clustering with a Fuzzy Covarianz Matrix. Proc. IEEE CDC, San Diego, pp. 761–766, 1979
- [10] Höppner, F., Klawonn, F., Kruse, R., Runkler, T.: Fuzzy Cluster Analysis, John Wiley & Sons, Chichester, 1999.
- [11] Klawonn, F., Chekhtman, V., Janz, E.: Visual Inspection of Fuzzy Clustering Results. In: Benitez, J., Cordon, O., Hoffmann, F., Roy, R. (eds.): Advances in Soft Computing: Engineering Design and Manufacturing. Springer, London , pp. 65–76, 2003.
- [12] Rehm, F., Klawonn, F., Kruse, R.: New approaches to noise clustering for detecting outliers. Soft computing for information mining (27th German conference on artificial intelligence - workshop proceedings), Ulm, 2004.
- [13] Rubens, M.: Fuzzy clustering algorithms and their cluster validity. European Journal of Operational Research 10, pp. 294–301, 1992.
- [14] Windham, M.P.: Cluster validity for fuzzy clustering algorithms. Fuzzy Sets and Systems 5, pp. 177–185, 1981.

# Fuzzy Clustering Using Similarity Measures: Clustering Relational Data by Object Data Methods

Christina Katz, Thomas A. Runkler, Kai Heesche  
Siemens AG Corporate Technology  
Information and Communications, CT IC 4  
81730 München  
Tel. 089/636-45372  
E-Mail Christina.Katz.ext@mchp.siemens.de

## Abstract

We distinguish between object data and relational data. Usually, object data are clustered by object data clustering methods such as fuzzy  $c$ -means, and relational data are clustered by relational clustering methods such as relational fuzzy  $c$ -means. We show that *relational* data can also be clustered by *object data* clustering methods, where the relation patterns serve as feature vectors. This is successfully shown in experiments with simple artificial and real world text data sets. These experiments also contain a study with different similarity measures and fuzziness indices that indicates a superiority of the cosine similarity and a low sensitivity for the fuzziness index.

## 1 Introduction: Object Data and Relational Clustering

We distinguish between *object* data sets  $X$  and *relational* data sets  $R$ . *Object* data sets contain a  $p$ -dimensional feature vector  $x_k \in \mathbb{R}^p$  for each object  $o_k$ ,  $p \in \mathbb{N}^+$ ,  $k = 1, \dots, n$ . *Relational* data sets contain a value  $r_{jk} \in \mathbb{R}$  quantifying the relation between each pair of objects  $(o_j, o_k)$ ,  $j, k = 1, \dots, n$ . Often we consider *positive* relations where  $r_{jk} \geq 0$  for all  $j, k = 1, \dots, n$ , and *symmetric* relations where  $r_{jk} = r_{kj}$  for all  $j, k = 1, \dots, n$ . We distinguish between *similarity* and *dissimilarity* relations, and require *similarity* relations to be *reflexive*,  $r_{kk} = 1$  for all  $k = 1, \dots, n$ , and *dissimilarity* relations to be *irreflexive*,  $r_{kk} = 0$  for all  $k = 1, \dots, n$ . Examples for dissimilarity relations include geometric (for example Euclidean) distances between feature vectors. Examples for similarity relations are given in Section 2.

Clustering [6] partitions a data set into  $c \in \{2, \dots, n-1\}$  subsets (clusters). *Object data clustering* clusters object data sets, and *Relational clustering* clusters relational data sets. One way to specify the clusters is to use a partition matrix  $U \in M$ , for example

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

*Hard clustering* produces *hard partition matrices*  $U \in M_{\text{hcn}}$ , and *fuzzy clustering* produces *fuzzy partition matrices*  $U \in M_{\text{fcn}}$ . In this paper we are dealing with fuzzy clustering only. A popular model for fuzzy object data clustering is *fuzzy  $c$ -means (FCM)* [1]

defined as the following problem: Minimize

$$J_{\text{FCM}} = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m d_{ik}^2, \quad (2)$$

subject to the set of cluster centers  $V = \{v_1, \dots, v_c\} \subset \mathbb{R}^p$  and the partition matrix  $U \in M_{\text{fcn}}$ , where

$$d_{ik} = \|v_i - x_k\|, \quad (3)$$

$i = 1, \dots, c, k = 1, \dots, n$ . The necessary conditions for extrema of  $J_{\text{FCM}}$  are

$$u_{ik} = 1 / \sum_{j=1}^c \left( \frac{d_{ik}}{d_{jk}} \right)^{\frac{2}{m-1}}, \quad (4)$$

$$v_i = \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m}, \quad (5)$$

$i = 1, \dots, c, k = 1, \dots, n$ . Optimization of the FCM model can be done by randomly initializing  $V$ , and then perform  $t$  steps of alternately updating  $U$  by (4) and  $V$  by (5). This algorithm is called *alternating optimization (AO)*, and we will denote AO the FCM model as FCM–AO. Notice the difference between distances  $d_{ik}$  between cluster centers and data points in (2) and the dissimilarities  $r_{ik}$  between pairs of data points in relational data sets!

The FCM model can be extended to relational data by inserting (5) into (2) [2] which yields the *relational fuzzy c-means (RFCM)* model [4] with the objective function

$$J_{\text{RFCM}} = \sum_{i=1}^c \frac{\sum_{j=1}^n \sum_{k=1}^n u_{ij}^m u_{ik}^m r_{jk}^2}{\sum_{j=1}^n u_{ij}^m}. \quad (6)$$

The necessary conditions for extrema of  $J_{\text{RFCM}}$  are

$$u_{ik} = 1 / \sum_{j=1}^n \frac{\sum_{s=1}^n \frac{u_{is}^m r_{sk}}{\sum_{r=1}^n u_{ir}^m} - \sum_{s=1}^n \sum_{t=1}^n \frac{u_{is}^m u_{it}^m r_{st}}{2 \left( \sum_{r=1}^n u_{ir}^m \right)^2}}{\sum_{s=1}^n \frac{u_{js}^m r_{sk}}{\sum_{r=1}^n u_{jr}^m} - \sum_{s=1}^n \sum_{t=1}^n \frac{u_{js}^m u_{jt}^m r_{st}}{2 \left( \sum_{r=1}^n u_{jr}^m \right)^2}}, \quad (7)$$

$i = 1, \dots, c, k = 1, \dots, n$ . Optimization of the RFCM model can be done by randomly initializing  $U$ , and then perform  $t$  steps of updating  $U$  by (7). Notice that if  $R$  is computed from an *object* data set  $X$  using the norm  $\|\cdot\|$  used in (3), then FCM and RFCM will produce the same results  $U$ . Extensions of RFCM include *Relational Alternating Cluster Estimation (RACE)* [10], *Relational Gustafson Kessel Clustering Using Medoids (RGKMdd)* [9], *non-Euclidean relational fuzzy c-means (NERFCM)* [3] and its *kernelized* version kNERFCM [5].

## 2 Clustering on similarities

The clustering models from the previous section are based on distances  $d_{ik}$  (2) between cluster centers and feature vectors or on dissimilarities  $r_{jk}$  (6) between pairs of objects. Instead of distances and dissimilarities we focus on *similarities* in this paper. In particular, we use the following four well-known similarity measures cosine, Dice, Jaccard and overlap:

- Cosine

$$s_C(a, b) = \frac{\sum_{i=1}^p a^{(i)}b^{(i)}}{\sqrt{\sum_{i=1}^p (a^{(i)})^2 \sum_{i=1}^p (b^{(i)})^2}} \quad (8)$$

The numerator indicates how good  $a^{(i)}$  and  $b^{(i)}$  correlate. The denominator is the geometric measure of the Euclidean length of both vectors.

- Dice

$$s_D(a, b) = \frac{2 \sum_{i=1}^p a^{(i)}b^{(i)}}{\sum_{i=1}^p (a^{(i)})^2 + \sum_{i=1}^p (b^{(i)})^2} \quad (9)$$

Here, the denominator is the sum of the squared feature vectors. To keep the similarity in  $[0,1]$  it has to be multiplied by two.

- Jaccard

$$s_J(a, b) = \frac{\sum_{i=1}^p a^{(i)}b^{(i)}}{\sum_{i=1}^p (a^{(i)})^2 + \sum_{i=1}^p (b^{(i)})^2 - \sum_{i=1}^p a^{(i)}b^{(i)}} \quad (10)$$

Here, the numerator describes the number of matching presences. The denominator is the number of attributes not involved in the matches [12]. The Jaccard measure is similar to the Dice measure, but yields higher (or the same) similarities. In general we have

$$s_D(a, b) = \frac{2s_J(a, b)}{1 + s_J(a, b)} \quad (11)$$

$$s_J(a, b) = \frac{s_D(a, b)}{2 - s_D(a, b)} \quad (12)$$

- Overlap

$$s_O(a, b) = \frac{\sum_{i=1}^p a^{(i)}b^{(i)}}{\min\left(\sum_{i=1}^p (a^{(i)})^2, \sum_{i=1}^p (b^{(i)})^2\right)} \quad (13)$$

Here, the denominator is the minimum of the squared feature vectors.

In all cases a higher value of the similarity  $s$  corresponds to a higher similarity of the two objects,  $s = 0$  means that the two objects have no common feature,  $s = 1$  means that the two documents have the same feature vectors. In particular, the similarity between a non-empty feature vector and itself is always one. In a first approach we simply replaced  $d_{ik}$  in (2) with one of the four similarity measures, where  $a = x_k$  and  $b = v_i$ , and used FCM-AO on *feature vectors*. This did not yield satisfactory results, because in most of the runs all objects were assigned to one cluster only. In a second approach we computed the relations  $r_{jk}$  from the feature vectors using one of the four similarity measures, where  $a = x_j$  and  $b = x_k$ , and then ran RFCM-AO. This did not yield satisfactory results either. In a third approach we then used the *relational* (similarity) data set and used the *object data model* FCM-AO. This is illustrated in the following section.

### 3 Relational Clustering by Object Data Clustering Algorithms

As pointed out in Section 1, object data clustering can be used to cluster object data sets, and relational clustering can be used to cluster relational data sets. In this paper, however, we demonstrate that *object* data clustering can also be used to cluster *relational* data sets. We first illustrate this with a very simple example. Consider a data set with five objects, where objects  $o_1, o_3$ , and  $o_4$  have similarity one (and dissimilarity zero) to each other, and the same for the remaining objects  $o_2$  and  $o_5$ . Obviously, this data set contains two natural clusters. We can summarize the similarities and dissimilarities in the following (crisp) relational data sets:

$$R_s = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad R_d = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}. \quad (14)$$

Imagine now that we interpret these *relational* data sets as *object* data sets  $X_s = R_s$  and  $X_d = R_d$ . Then each row in each data set is interpreted as a *feature vector*, where each feature  $x_{ki}$  represents the relation between object  $o_k$  and object  $o_i$ . The result of object data clustering is that data points with similar feature vectors are grouped together. In our approach this means that data points with similar *relation patterns* are grouped together. In our example, the data from the two clusters possess the relation patterns (feature vectors)  $v_1 = (01001)$  and  $v_2 = (10110)$ , respectively. Thus, object data clustering will yield the expected partitions

$$U_s = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}, \quad U_d = \begin{pmatrix} 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}, \quad (15)$$

just by using the relational data sets as if they were object data sets. The reader might easily verify that this simple example can be extended to  $c > 2$ .

In the following sections we will show that this approach does not only apply to our simple example data sets, but that it also works for real world data sets. As an example for these we picked *text data sets*.

## 4 Text Data Sets

From the internet many different text data sets are available, most of them contain news or newsgroup data, for example the Reuters data set or the 20 Newsgroups data set. The Reuters data set is a text categorization test collection which contains newswire articles and Reuters annotations. The collection consists of 22 data files where each of the first 21 files (reut2-000.sgm through reut2-020.sgm) contains 1,000 documents, while the last (reut2-021.sgm) contains 578 documents. The documents are related to five categories, which are “exchanges”, “orgs”, “people”, “places” and “topics”. The Reuters-21578 Distribution 1.0 test collection is available from <http://www.research.att.com/~lewis>.

The 20 Newsgroups data set is a collection of about 20,000 internet newsgroup (usenet) documents which are categorized into 20 different newsgroups: alt.atheism, comp.graphics, comp.os.ms-windows.misc, comp.sys.ibm.pc.hardware, comp.sys.mac.hardware, comp.windows.x, misc.forsale, rec.autos, rec.motorcycles, rec.sport.baseball, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, soc.religion.christian, talk.politics.guns, talk.politics.mideast, talk.politics.misc and talk.religion.misc. Every category contains about 1,000 texts. On average, each document contains about 2,357 characters and 279 words. The 20 Newsgroups data set is available from <http://people.csail.mit.edu/jrennie/20Newsgroups>.

Both the Reuters and the 20 Newsgroups data sets are particularly suitable for our experiments, since the documents are categorized which allows for an easy comparison between the clustering results and their “natural” classification. For our presentation here we chose the 20 Newsgroups data set, because each document is assigned exactly to only one category while in the Reuters data set each document may be assigned to many suitable categories, and in some cases to no category at all. This would make the comparison of the results more difficult. Because of the same reasons the 20 Newsgroups data set was also used in [11].

To keep computation time feasible, we applied our algorithm only to a *subset* of the 20 Newsgroups data set: We picked the first 50 texts from each comp.graphics and talk.politics.misc. These two newsgroups represent two very different semantic fields which makes clustering easier. Reading, preprocessing, and clustering these 100 texts with all 48 methods took about 12 minutes on an Intel Pentium III processor at 996 MHz with 256 MB physical memory.

## 5 Text Preprocessing

Fig. 1 shows an example of a raw data document to illustrate the following preprocessing steps. The first preprocessing step is the extraction of the actual text, i.e. the removal of all additional information like author, time, subject, etc. This is done as follows: The header information of each text begins in the first line and ends with at least one blank line, so the plain text begins after the last of these first blank lines.

The second preprocessing step is stemming. By stemming the number of different words that have to be considered for clustering is reduced to a smaller number of semantically equivalent terms. For example, *company* and *companies* are both transformed to the

Path: cantaloupe.srv.cs.cmu.edu[...]not-for-mail  
From: steveq@DIALix.oz.au (Steve Quartly)  
Newsgroups: comp.sys.amgia.graphics,comp.graphics  
Subject: WANTED: SIRD Alogorythmn  
Summary: WANTED: A Sird Alogorythmn  
Keywords: Sird  
Message-ID: <lpraaa\$pv@DIALix.oz.au>  
Date: 6 Apr 93 23:08:58 GMT  
Article-I.D.: DIALix.lpraaa\$pv  
Organization: DIALix Services, Perth, Western  
Australia  
Lines: 12  
NNTP-Posting-Host: localhost.dialix.oz.au  
X-Newsreader: NN version 6.4.19 #1

Hi,

I'm interested in writing a program to generate a SIRD picture, you know the stereogram where you cross your eyes and the picture becomes 3D.

Does anyone have one or know where I can get one?

Please e-mail to steveq@sndcrft.DIALix.oz.au with any replies.

Many thanks for your help.

Steve Q.

Figure 1: An example of a raw data document from the 20 Newsgroups data set

same term: company. Therefore stemming significantly reduces the complexity of the algorithm in terms of both computation speed as well as required storage capacity. A frequently used stemming algorithm is the so-called Porter stemmer [7], a string replacement algorithm with 62 replacement rules that are applied in a five step procedure. Here, we use a simpler stemming algorithm with only seven replacement rules that produces only slight modifications and keeps the readability of the word stems high. The rules of our simplified stemmer algorithm are:

1. ignore all words with more than 30 characters, since these words are considered as artifacts, e.g. aggregations of several single words
2. convert all words to lower case  
(America → america)
3. words which end with *-ies*: change the ending to *-y*  
(companies → company)
4. words which end with *-es*: cut off the *-es*  
(titles → titl)
5. words which end with *-s* : cut off the *-s*  
(players → player)
6. words which begin or end with a digit: cut off the digit  
(8bit → bit)
7. words which begin or end with a special character: cut off the special character  
(this? → this)

These actions are repeated until no replacement rule applies any more.

Here is an example of a word and its stemming-process:

"Stories"  $\xrightarrow{2}$  "stories"  $\xrightarrow{7}$  'stories'  $\xrightarrow{7}$  stories  $\xrightarrow{3}$  story

Before stemming we obtained a number of 7,032 words, after the stemming we have only 4,072 different word stems. To get substantial results we ignored non-specific terms and documents. We consider those terms non-specific which appear in more than 50% of the texts, so these words do not seem to contribute to the content of the texts. Examples for non-specific terms are and, is etc. Non-specific documents are those which do not contain any specific terms at all. The 100 documents we consider here contain one non-specific document, so we use only the remaining 99 documents which contain a total of 4,048 specific words.

## 6 Term counts

The basis for our clustering calculations is a term frequency matrix  $TF$ . Each entry  $TF_{ij}$  contains the number of times that term  $j$  appears in document  $i$ , so the document vectors are the rows and the term vectors are the columns of  $TF$ . To compute  $TF$ , we simply count the number of occurrences of each term in each document. Both, the column sums of the term frequencies, as well as the number of documents including term  $j$  in the 99

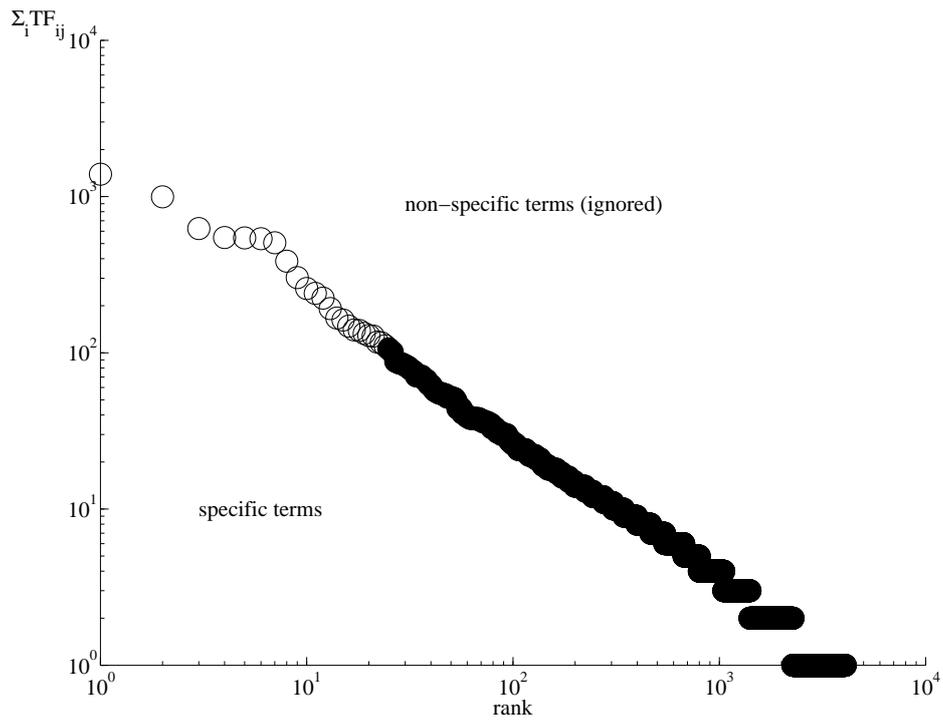


Figure 2: Term frequencies

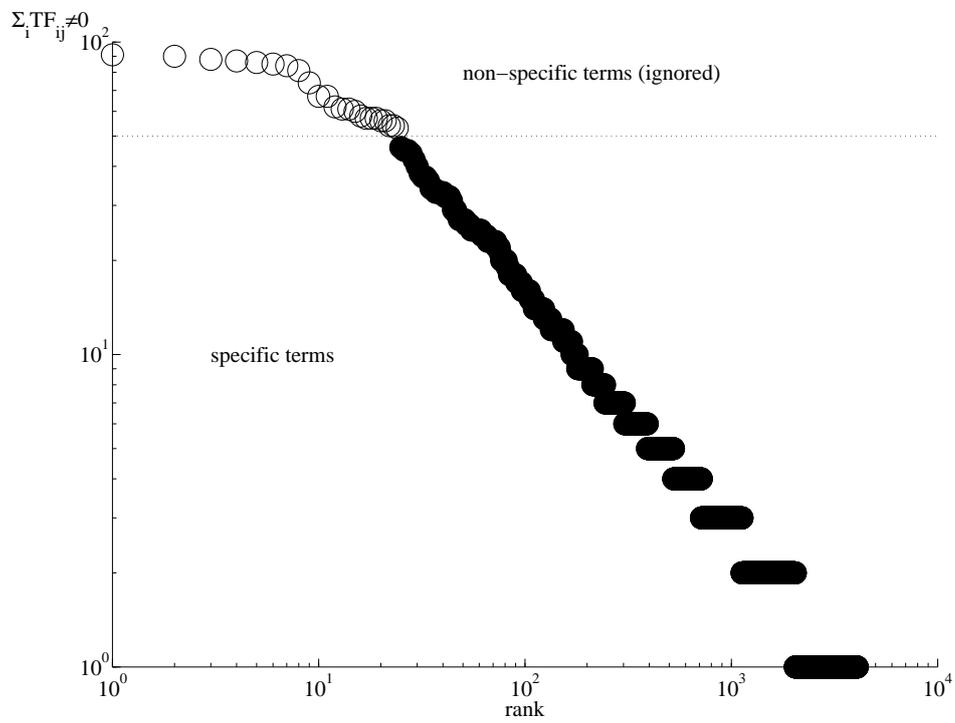


Figure 3: Document frequencies

considered documents roughly follow a power law distribution [8], as shown in Figs. 2 and 3. We consider a pair of documents similar if both documents contain many common terms, i.e. the documents are associated with similar  $TF$  rows. Since frequent terms are less specific than terms that appear less frequently, we multiply  $TF$  with the *inverse document frequency* ( $IDF$ ) defined as  $IDF_j = \log(N/n_j)$ , where  $N$  is the number of all documents and  $n_j$  is the number of documents containing term  $j$ . The logarithm in this expression compensates the exponential distribution of the terms shown in Figs. 2 and 3. The product of  $TF_i$  and  $IDF_j$  yields the so-called  $TF \cdot IDF$  count. Thus,  $TF \cdot IDF$  is high if a term appears often in a document but in only few other documents.

## 7 Document similarities

The  $TF \cdot IDF$  features allow to compute a data set containing the relations between each pair of documents. To generate these relational data sets we use the four similarity measures presented in Section 2 with the  $TF \cdot IDF$  rows of the two considered documents as the feature vectors  $a$  and  $b$ . Four similarity matrices  $S_C, S_D, S_J, S_O$  are generated by computing the similarities  $s_C, s_D, s_J, s_O$  for each pair of documents, so  $S_C, S_D, S_J, S_O \in [0, 1]^{100 \times 100}$ . We will later examine which similarity measure yields the best classification results.

## 8 Document clustering

For clustering the relational (similarity) data sets  $S_C, S_D, S_J$ , and  $S_O$  we applied FCM-AO with the distances  $d_{ik}$  replaced by one of the four similarity measures  $s_{ik}$ , as described in Section 2, with  $t = 100$  iterations,  $c = 2$  clusters, and the fuzziness indices  $m \in \{1.5, 2, 2.5\}$ . Notice again that we apply *object data* clustering to *relational* data sets here, as pointed out in Section 3. The combinations of one of the four similarity measures for generating the relational data set, one of the four similarity measures in clustering, and one of the three fuzziness indices yields at total  $4 \times 4 \times 3 = 48$  partition matrices  $U$ . We denote the similarity measures using the letters C, D, J, and O, and concatenate the three chosen options to a code string. For example, the code DJ2.5 means that  $s_D$  is used to generate the relational data set,  $s_J$  is used in clustering, and the fuzziness index is set to  $m = 2.5$ . The resulting 48 partition matrices are defuzzified, i.e. each document is (crisply) assigned to the cluster in which it has its maximum membership. The resulting crisp partitions are then compared with the original categorization of the documents.

Since the results of the fuzzy clustering depend on the random initialization of the partition matrices, we run the algorithm with ten different initializations and then pick the best result: We count how many documents belong to each defuzzified cluster and compute the mean value of these  $c$  counts. The best result out of the ten iterations is the one where this mean count is closest to the number of documents divided by the number of clusters.

## 9 Evaluation: Receiver operating characteristic

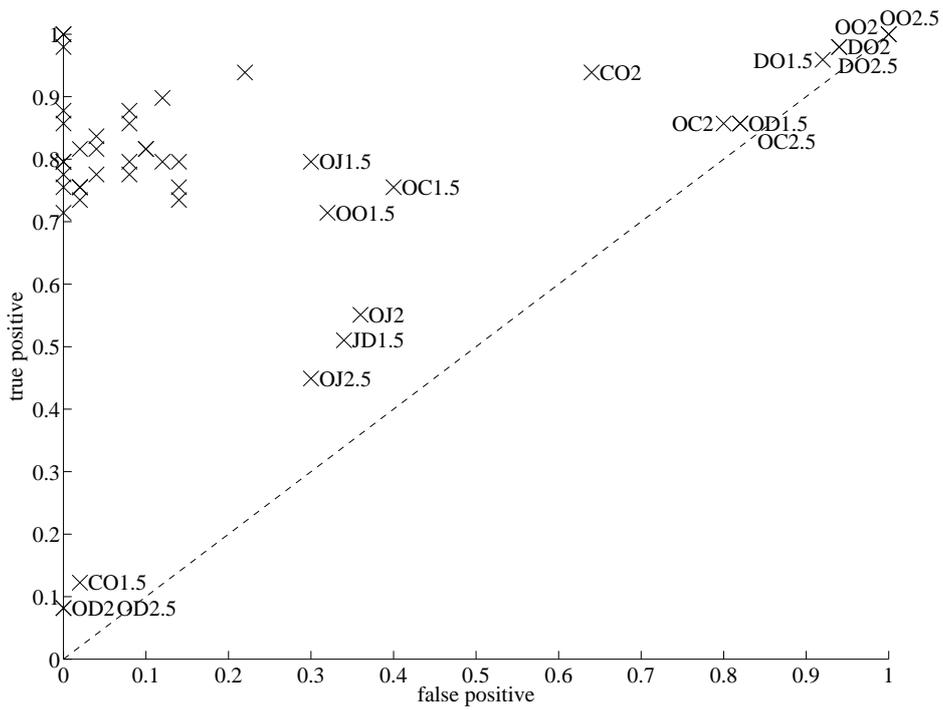
Receiver operating characteristics (ROC) are used to evaluate classifiers. The ROC is a plot of the number of true positive versus false positive classifications, where true positive

denotes that an object from class  $i$  is classified as  $i$  and false positive denotes that an object from class  $i$  is classified as  $j \neq i$ . An optimal classifier produces 100% true positives and 0% false positives, which yields a point at the top left corner in the ROC plot. Fig. 4 shows the ROC points for the class `comp.graphics` for the 48 cases using the code strings described above. Fig. 4a shows all the ROC points, where only the worst results are labeled, and Fig. 4b shows a zoom of only the best ROC points. The results show a clear superiority of the cosine similarity, whereas the overlap similarity performs worst. The sensitivity of the clustering approach to the fuzziness index  $m$  is relatively low. However, the best result is achieved for CC1.5 and CC2, i.e. for the cosine similarity in both cases and fuzziness indices of  $m = 1.5$  or  $m = 2$ . Notice that  $m = 2$  is a value of the fuzziness index that is also frequently used in other clustering applications. Also notice that for CC1.5 and CC2 the optimal classifications of 100% true positives and 0% false positives is achieved. However, this result is specific for the considered data sets and can of course not be generalized to all other data sets. Fig. 5 shows the (fuzzy) partition matrix for the case CC2. The solid curve shows the memberships for cluster one, and the dashed curve shows the memberships for cluster two. The larger the value the higher is the membership to the respective cluster. Clearly, both membership rows sum up to one. The data points are sorted, so that the left half represents the 49 documents labeled as `comp.graphics`, and right half represents the 50 documents labeled as `talk.politics.misc`. Apparently, cluster one can be associated with the class `comp.graphics`, and cluster two can be associated with the class `talk.politics.misc`. For the first 49 documents the solid curve is always above the dashed one, and for the following 50 documents it is always below. This confirms the 100% true positive and 0% false positive classification rate from Fig. 4a.

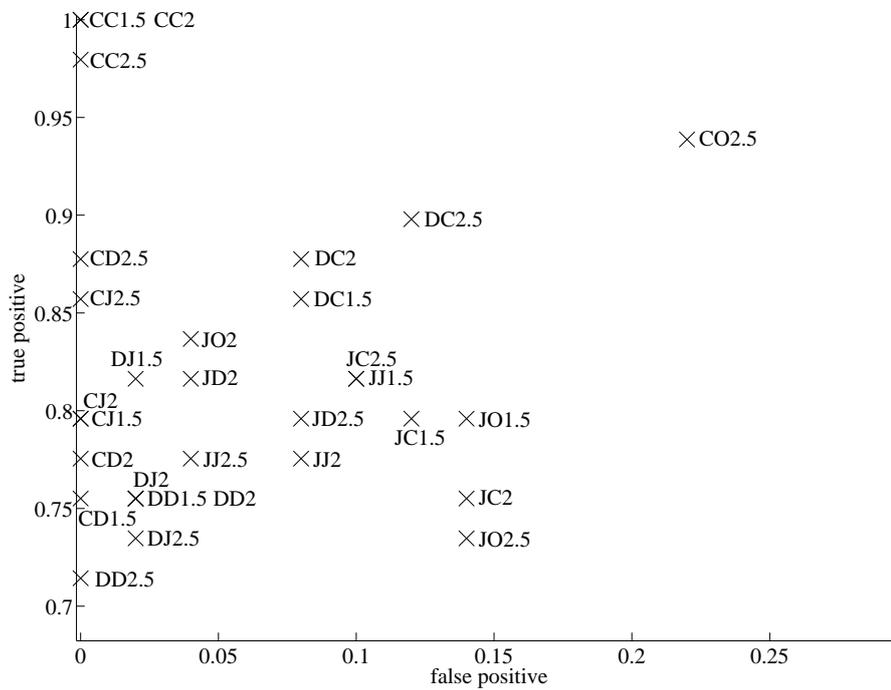
## 10 Conclusions

In this paper we have focused on the distinction between object data and relational data, and between object data clustering and relational clustering. Usually, object data are clustered using object data clustering methods, and relational data are clustered using relational clustering methods. In this paper, however, we have shown that *relational* data can also be clustered by *object data* clustering methods, where the relation patterns serve as feature vectors for each object. The applicability of an object data clustering method (fuzzy c-means) to relational data was illustrated by (i) a simple crisp example and (ii) by an extensive study with real world text data sets.

For our study of text data sets we considered the usenet documents from the 20 News-groups data set. From these we produced relational data sets using similarity measures. As pointed out above we then applied a modified version of FCM, an *object data* clustering model with similarity measures, to the resulting *relational* data sets. For the generation of the relational data sets as well as in the clustering model we used all possible combinations of four well-known similarity measures: cosine, Dice, Jaccard, and overlap. Moreover, we used three different fuzziness indices  $m \in \{1.5, 2, 2.5\}$ . In our experiments we generated and evaluated the corresponding partition matrices for all  $4 \times 4 \times 3 = 48$  cases. The experiments have shown that the best results are obtained by the cosine similarity, whereas the overlap similarity performs worst. The sensitivity to the fuzziness index  $m$  is low. In two experiments, a classification rate of 100% true positives and 0% false positives was achieved. Thus, our new approach of applying object data clustering to relational data proved to be very successful.



(a) Evaluation of all results (complete view, only worst results labeled)



(b) Evaluation of the best results (zoom of the complete view from (a))

Figure 4: Receiver operating characteristic for comp.graphics

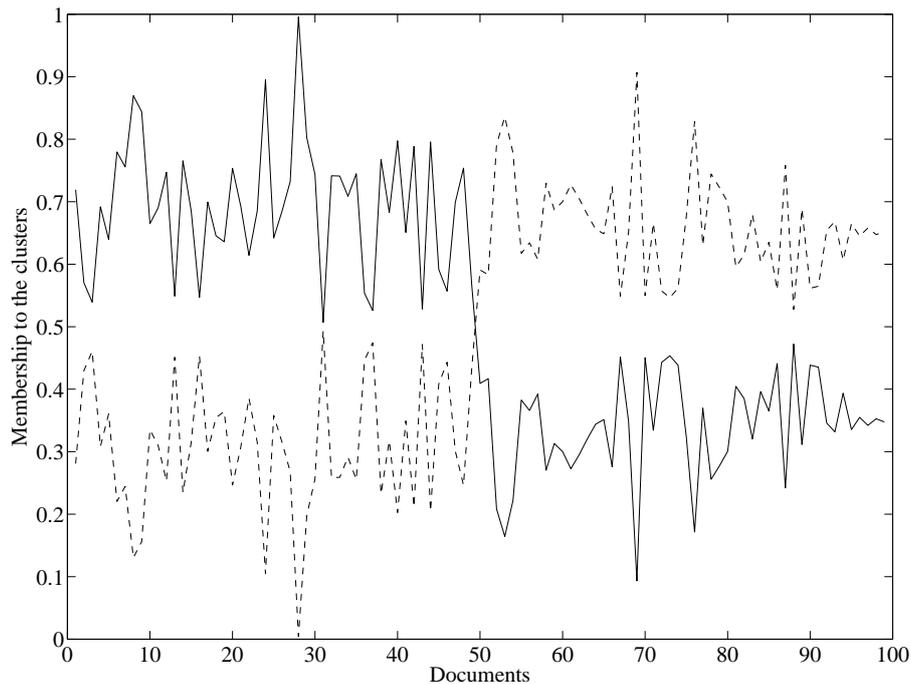


Figure 5: Partition matrix for CC2

## 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] R. J. Hathaway and J. C. Bezdek. NERF  $c$ -means: Non-Euclidean relational fuzzy clustering. *Pattern Recognition*, 27:429–437, 1994.
- [4] R. J. Hathaway, J. W. Davenport, and J. C. Bezdek. Relational duals of the  $c$ -means algorithms. *Pattern Recognition*, 22:205–212, 1989.
- [5] 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.
- [6] F. Höppner, F. Klawonn, R. Kruse, and T. A. Runkler. *Fuzzy Cluster Analysis — Methods for Image Recognition, Classification, and Data Analysis*. Wiley, 1999.
- [7] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, July 1980.
- [8] T. A. Runkler. Clustering power law data. In *GMA/GI Workshop Fuzzy Systems and Computational Intelligence, Dortmund, Germany*, November 2005.

- [9] T. A. Runkler. Relational Gustafson Kessel clustering using medoids and triangulation. In *IEEE International Conference on Fuzzy Systems*, pages 73–78, Reno, May 2005.
- [10] 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.
- [11] T. A. Runkler and J. C. Bezdek. Relational clustering for the analysis of internet newsgroups. In O. Opitz and M. Schwaiger, editors, *Exploratory Data Analysis in Empirical Research, Proceedings of the 25th Annual Conference of the German Classification Society*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 291–299. Springer, 2002.
- [12] P. Tan, M. Steinbach, and V. Kumar. *Introduction to data mining*. Addison Wesley, Boston, 2006.

# Methoden der Clusteranalyse für die Auswertung von Impedanzspektren eines selektiven Gassensors für Hochtemperaturanwendungen bis 900°C

**Norman Bauersfeld, Klaus-Dietrich Kramer**

Hochschule Harz, Fachbereich Automatisierung und Informatik

Friedrichstr. 57-59, 38855 Wernigerode

Tel.: (03943) 6259746

Fax: (03943) 6259747

E-Mail: {nbauersfeld,kkramer}@hs-harz.de

## Zusammenfassung

Im folgenden wird die Anwendung von Clusteranalysemethoden auf Daten von Hochtemperatursensoren (Impedanzspektren) beschrieben, um Gasspezies und deren Konzentrationen bestimmen zu können. Das resultierende Muster soll zur Gaserkennung und damit zur Prozesskontrolle und zum Prozessmonitoring eingesetzt werden. Anhand von Methoden zur Merkmalsgewinnung erfolgt die Evaluierung und Anwendung von Methoden der Daten- und Clusteranalyse auf, durch den Laborbetrieb gewonnenen Gasgemischdaten. Es werden Möglichkeiten der Reduktion der Datenmenge und der Merkmalsdimension mittels selbstlernender neuronaler Netze untersucht, um geeignete Clusterverfahren und deren Modifikationen herauszustellen und die Ergebnisse zu vergleichen. Damit wird die Klassifikation des reduzierten Datenraumes möglich. Die Untersuchungen und der Vergleich der Methoden erfolgen ausschließlich im Fokus der o. g. Anwendung zur Erarbeitung einer prototypischen (mobilen) Messplattform.

## 1 Einleitung

Die Prozesskontrolle und das Prozessmonitoring in Hochtemperatursystemen gewinnt zunehmend an Bedeutung. Herkömmliche Leitfähigkeitsgassensoren weisen nur eine eingeschränkte Selektivität bezüglich verschiedener Gasspezies in wasserstoffreduzierenden Atmosphären auf und sind nicht für Temperaturen bis 900°C ausgelegt. Ein neues Sensorkonzept zur Erkennung bereits kleiner Mengen Kohlenmonoxid (CO) soll zum Beispiel Katalysatorvergiftungen durch einen zu hohen CO-Anteil in *Proton Exchange Fuel Cells* (PEMFC) vermeiden. Die Online-Datenanalyse gelingt derzeit nur mit hochpreisigen optischen Messsystemen. Neue, weniger preisintensive Konzepte basieren auf Sensoren, deren Materialien und dessen Resonanzeigenschaften die Hochtemperaturanalyse ermöglichen (Lit. [1]) sowie einem neu entwickelten Messsystem für die mobile Online-Datenanalyse (Lit. [2]).

## 2 Sensordaten und Merkmalsextraktion

Die Impedanzspektren im Bereich der Resonanz der spezialisierten Gassensoren zeigen gasspezifische Veränderungen, die in Zusammenhang mit der Veränderung der

elektrischen und mechanischen Eigenschaften einer gassensitiven Schicht stehen. Die für die Datenanalyse verwendeten Sensoren bestehen aus einem Langsitresonator und einer sensitiven Schicht aus  $CeO_2$ . Durch mechanische und elektrische Änderungen können die Sensordaten in den zwei Modi *Quarzmikrowaage* und *Leitfähigkeitänderung* untersucht werden. Gemessen und analysiert wurden je Modi der Realteil der Admittanz für den ersten (Abb. 1) und dritten Oberton (Abb. 2) und deren Funktionsverlauf mittels einer modifizierten Lorentzfunktion (Gl. 1) angenähert (Marquardt-Approximation, Lit. [9]).

$$f(x, a..f) = \frac{a}{(1.0 + 4.0 * (\frac{x-b}{c})^2)^{-d}} + f * x + e \quad (1)$$

Die Lorentzfunktion definiert durch ihre Parameter die Merkmale  $a$  bis  $f$ . Ein Merkmal für die Clusteranalyse wird als geeignet eingestuft, wenn die Varianz genügend groß ist, um die Gaskonzentrationen unterscheiden zu können. Die durch die Funktion gewonnenen Fits liefern im Arbeitsbereich des Sensors von  $10^{-20}$  bis  $10^{-15}$  bar  $p(O_2)$  innerhalb der Toleranzgrenzen einen einheitlichen Approximationsfehler  $\chi^2$  (siehe Abb. 6, 1. Oberton, 0/100-CO/H<sub>2</sub>), der aus der Minimierung der Kostenfunktion (Gl. 2) ermittelt wird. Um Variationen durch kleinere Temperaturschwankungen ausschließen zu können wurden die gemessenen Parameter entsprechend temperaturkompensiert (Lit. [11]). Die primäre Untersuchung bezieht sich zunächst auf den Merkmalsraum der temperaturkompensierten Frequenzen im Bereich der Resonanz (Abb. 3, Abb. 4) aus Leitfähigkeits- und Masseffekt im Arbeitsbereich des Sensors. Die Frequenzen werden aus dem Parameter  $b$  (Gl. 1) der Lorentzfunktion, am Maximum der Amplitude, für beide Modi direkt gewonnen ( $b \equiv f(Y'_{max})\{A_{(1,3)}, B_{(1,3)}\}$ ). Die temperaturkompensierte Frequenz A im Bereich der Resonanz zeigt eine Gasgemischabhängigkeit durch Frequenzverschiebung, während die temperaturkompensierte Frequenz B neben einer CO/H<sub>2</sub>-Abhängigkeit auch  $p(O_2)$ -abhängig ist (Lit. [12]). Eine manu-visuelle Analyse des Frequenz-Frequenz-Diagramms (Abb. 5, standardisierte Daten) zeigt bereits Gruppierungen, deren Eigenschaften untersucht werden. Betrachtet werden Gruppen von 0/100, 25/75, 50/50 und 100/0%-CO/H<sub>2</sub>-Anteilen. Die aus der Datenvorlage durch Merkmalsextraktion gewonnenen Merkmalsvektoren spannen einen n-dimensionalen Merkmalsraum auf. Jede Dimension wurde einem Eignungstest unterzogen, d. h. auf ihren Beitrag zur Unterscheidung von Gasgemischgruppen untersucht. Für die folgenden Betrachtungen wird der 2-dimensionale Merkmalsraum als ausreichend angenommen.

### 3 Clusteranalyseverfahren

#### 3.1 Vorbetrachtungen

Der Clusteranalyse gehen zunächst einige Vorbetrachtungen für die Auswahl entsprechender Verfahren voraus.

- Wie viele Merkmale werden benötigt? Die Anzahl der Merkmale bestimmt die Dimension des zu analysierenden Merkmalsraumes und neben der damit verbundenen Rechengeschwindigkeit und -genauigkeit auch die Wahl geeigneter Darstellungsmethoden zur Begründung der Ergebnisse. Die Auswahl der oben

genannten Merkmale ermöglicht bereits eine ausreichende Datenanalyse der  $CO/H_2$ -Gemische.

- Welche Methode soll gewählt werden? Die Wahl der Methode ist abhängig von den statistischen Eigenschaften der Merkmale. Sind diese z. B. normalverteilt und in der Kalibrierung stabil, kann ein Bayes-Klassifikator eingesetzt werden. Nicht normalverteilte Merkmale und instabile Kalibrierdaten erfordern hingegen die Analyse per neuronalem Netz (Lit. [8]). Zunächst wurden Fuzzy-Clusterverfahren angewandt. Aufgrund fehlender (zeitlich bezogener) Datensätze kann bisher keine Aussage über die Stabilität des Klassifikators getroffen werden.
- Wie kann entschieden werden, ob ein Merkmal der Analyse hinderlich oder förderlich ist? Anhand physikalisch-chemischer Betrachtungen wurden die oben genannten Merkmale empfohlen (Lit. [1]).
- Welcher Grad an a-priori Wissen kann eingebracht werden? Die Sammlung von a-priori Wissen hängt im hohen Maße vom Verständnis der Vorgänge im Sensorsystem und der damit verbundenen Merkmalsgrößen zusammen, insofern zu diesen eine physikalische Beziehung herzustellen ist. Für das obige Beispiel der Frequenzen im Bereich der Resonanz und deren korrespondierenden Amplituden, ist ein Bezug herzustellen. Damit können auch die Clusteranalyseergebnisse einen wichtigen Beitrag zur Sensordimensionierung leisten.
- Wie gut ist das Klassifikationsproblem konditioniert? Je nach zu unterscheidender Genauigkeit der Anteile der Gasgemische müssen die Merkmale die Zielstellung unterstützen. Die Kondition des Klassifikators kann durch umfangreiche Arbeitszyklentests ermittelt werden. Unterstützt der Klassifikator nicht die erforderliche Trenngenauigkeit müssen die Methoden der Datenaufbereitung und -verarbeitung entsprechend optimiert werden.

### 3.2 Ansatz

Für die Auswahl eines geeigneten Analysealgorithmus können folgende Kriterien diskutiert werden:

- Wie groß ist der Grad der Zugehörigkeit der Daten zu den Clustern? Der Grad der Zugehörigkeit wird bei den Fuzzy-C-Means-Verfahren anhand der Berechnung der Zugehörigkeitsfunktion ermittelt.
- Welcher Art ist die Trennung der Cluster-Zentren? In Abhängigkeit des Clusterverfahrens können verschiedene Trennmöglichkeiten anhand der Datenverteilung und Dichte getestet werden.
- Welche Möglichkeiten gibt es, eine geeignete Clusteranzahl zu bestimmen? Bei den possibilistischen Clusterverfahren wird die Clusteranzahl zu Beginn anhand einer Hypothese festgelegt. Im Beispiel sind vier Gruppen zuzuordnen. Damit können bekannte Klassen bestätigt und manu-visuell plausibel bezeichnet werden.

- Besteht die Möglichkeit einer plausiblen Zuordnung einzelner Datensätze zu gefundenen Clustern?
- Können bekannte Klassen durch Cluster bestätigt werden?
- Wie hoch ist der Abstraktionsgrad des Klassifikators?
- Welcher Grad der Generalisierung des Klassifikators liegt vor?

Ziel der Kriterienfestlegung ist das Auffinden von ausgeprägten Gruppen, die durch Cluster beschrieben und mittels Klassen bezeichnet werden.

### 3.3 Problemstellung und allgemeiner possibilistischer Algorithmus

Die Clusteranalyse kann durch das Problem *Finden von Clustern* durch Lösen bzw. Minimieren eines Optimierungsproblems  $J$  beschrieben werden.

$$\chi^2 \equiv J_q(U, V) = \sum_{j=1}^N \sum_{c=1}^C (u_{j,c})^m d_c^2(x_j, v_c) \rightarrow \min_{U,V} \quad (2)$$

Eine Anzahl  $N$  an Daten wird dabei einem Cluster  $c$  mit  $u_{j,c}$  zugehörig erklärt, wobei  $d$  als Distanz zwischen einem Clusterzentrum  $v_c$  und einem Datum  $x_j$  passender Metrik berechnet wird.  $m$  wird als Fuzzifier bezeichnet. Je kleiner  $m$  gewählt wird, desto schärfer ist die Zugehörigkeit eines Datums zu einem Cluster. Als Nebenbedingungen werden festgelegt:

$$\sum_{c=1}^C u_{j,c} = 1, \forall j = 1, \dots, N, u_{j,c} \geq 0 \text{ und } \sum_{j=1}^N u_{j,c} > 0, \forall c = 1, \dots, C \quad (3)$$

D. h., jedes Datum wird mindestens einem Cluster mit einer von 0 größeren Zugehörigkeit zugeordnet, als die Summe der Zugehörigkeiten eines Datums zu allen Clustern 1 (entspricht 100%) ergeben muss. Die Zugehörigkeit  $u_{j,c}$  der Daten zu den Clusterzentren sowie die Zentren  $v_c$  werden iterativ durch Minimierung von  $J$  berechnet. Unter den genannten Bedingungen kann folgender Algorithmus (Lit. [7]) formuliert werden, der die Grundstruktur der possibilistischen Clusteranalyse wiedergibt.

- Jedes Cluster sei durch ein Element einer Menge  $C$  eindeutig charakterisierbar.
- Wähle die Anzahl  $c$  der Cluster,  $2 \leq c < n$ , ein  $m \in R_{>1}$ , die Abbruchgenauigkeit  $\epsilon$ .
- Initialisiere  $U_0$  und  $V_0$ , sowie  $i = 0$ .
- WIEDERHOLE  
 Erhöhe  $i$  um 1  
 Bestimme die optimale Clustermenge  $C_i$  und Clusterlage  $V$  zur Minimierung von  $J$  durch  $C_i$  und  $U_{i-1}$   
 Bestimme  $U_i$   
 BIS  $\| U_{i-1} - U_i \| \leq \epsilon$

Der Algorithmus bietet die Freiheitsgrade der Merkmalsauswahl (Vektoren  $x$ ), Distanzwahl, Festlegung der Clusteranzahl  $C$  sowie die Festlegung des Fuzzifizierungsgrades  $m$  an. Diese können das Clusteranalyseergebnis entscheidend beeinflussen und müssen am realen Beispiel getestet werden. In den folgenden Abschnitten werden die Ergebnisse aus der Anwendung possibilistischer Clusteranalysealgorithmen auf einen Beispieldatensatz (Abb. 5) vorgestellt.

### 3.4 Fuzzy-C-Means (FCM)

Die Datenanalyse wird zunächst mit dem FCM zur Erkennung von 4 Clustern gestartet. Der FCM erkennt kreisförmige Cluster im 2-dimensionalen Merkmalsraum. Unscharfe Zustandsübergänge können bereits mit Hilfe dieses Algorithmus abgebildet werden. Die Distanzberechnung zwischen einem Datensatz (Datum) und den Prototypen der Cluster erfolgt durch eine spezifische Metrik (Minkowski oder Mahalanobis repräsentiert durch die Matrix  $M$ ):

$$d^2 : D \times C \rightarrow Re, (x, v) \rightarrow (x - v)^T M (x - v) \quad (4)$$

Als Maß wird der euklidische Abstand und der Fuzzifizier 2 verwendet. Für die äußeren Gruppen (Abb. 7) kann damit bereits eine gute Trennung gefunden werden. Nach Bestimmung der optimalen Lage der Prototypen  $v_c$

$$v_c = \frac{\sum_{j=1}^N (u_{j,c})^m x_j}{\sum_{j=1}^N (u_{j,c})^m} \text{ und } u_{j,c} = \frac{[d_c^2(x_j, v_c)]^{\frac{1}{1-m}}}{\sum_{i=1}^C [d_i^2(x_j, v_i)]^{\frac{1}{1-m}}} \quad (5)$$

wird zudem für jedes Datum eine Zugehörigkeit  $u_{j,c}$  zum jeweiligen Zentrum berechnet. Das Datum wird dem Cluster zugeordnet, bei dem die Zugehörigkeit maximal ist. Ausreißer können anhand eines Schwellwertes für die minimale Zugehörigkeit zu den jeweiligen Prototypen erkannt werden. Abb. 7 zeigt das (bereits bezeichnete) Clusterergebnis für eine FCM-Clusterbildung ( $c = 4$ ). Der Partitionskoeffizient beträgt 0.84, die Partitionsentropie 0.34 begründet mit der kreisförmigen Gruppierungseigenschaft des FCM und der dazu ellipsoiden Lage der eigentlichen Gruppen. Im Sinne einer strengeren Zuordnung der Daten zu den jeweiligen Clustern wird versucht, durch Einsatz modifizierter FCM-Algorithmen ein optimaleres Ergebnis zu erreichen.

### 3.5 Gustafson-Kessel (GK)

Der Gustafson-Kessel-Algorithmus erweitert den FCM durch Einbeziehung der Datendichte. Dazu erfolgt die Berechnung der Fuzzy-Kovarianzmatrix, die für jeden Cluster unterschiedlich ist und somit eine spezifische Lage und Form, nicht aber die Datendichte, betrachtet wird (Lit. [3]). Der Abstand zwischen einem Datum und den jeweiligen Prototypen wird berechnet nach Gl. 4, wobei  $M \equiv A$ , so dass  $A$  ermittelt wird durch Bestimmung der clustereigenen Norm.

$$A = \sqrt[3]{\det(S_c)} S_c^{-1} \text{ mit } S_c = \left[ \sum_{j=1}^N u_{j,i}^m (x_j - v_i)(x_j - v_i)^T \right] * \left[ \sum_{j=1}^N u_{j,i}^m \right]^{-1} \quad (6)$$

$S_c$  bezeichnet die Fuzzy-Kovarianzmatrix für jeden Cluster und  $S_c^{-1}$  ihre Inverse. Der GK-Algorithmus unterstützt teilweise die intuitive Annahme der Clusterstruktur aus Abb. 8. Im linken oberen Bereich wurde zusätzlich die Kontur der Trennlinien der Zugehörigkeitsgrade dargestellt.

### 3.6 Gath-Geva (GG)

Der Gath-Geva-Algorithmus (Lit. [4]) erweitert den GK durch Betrachtung der Clustergröße und -dichte. Die Variablen des Datensatzes werden als normalverteilt angenommen. Die Distanzen werden mittels des Kehrwertes der Wahrscheinlichkeit  $\frac{1}{P}$ , mit dem ein Datum zu einem Cluster gehört, berechnet.

$$d^2 : D \times C \rightarrow \mathcal{R}, (x, v, S_c, P) \rightarrow \frac{1}{P} \sqrt{\det(S_c)} e^{\frac{1}{2}(x-v)^T S_c^{-1}(x-v)} \quad (7)$$

$S_c$  bezeichnet die Fuzzy-Kovarianzmatrix (Gl. 6).  $P$  wird berechnet mit

$$P = \sum_{j=1}^N u_{j,c}^m * \left[ \sum_{j=1}^N \sum_{l=1}^C u_{j,l}^m \right]^{-1} \quad (8)$$

Abb. 9 zeigt die resultierende und bereits *bezeichnete* Kontur. Der GG separiert die Clusterdaten durch eine Zugehörigkeitswertberechnung von 0 und 1. Die strikte Trennung und damit die Nutzung des GG für die Anwendung der Klassifikation muss noch überprüft werden.

### 3.7 Vergleichende Übersicht FCM-GK-GG

Für den Beispieldatensatz liegen keine weiteren Messungen vor, so dass noch keine Aussage über die Ausbreitung der Cluster (Ellipse oder Korridor) getroffen werden kann. Da der GG die Daten fehlerfrei unterteilt, wird dieses Verfahren zunächst als das geeignete angesehen. Vergleichend seien in den Abb. 10 bis Abb. 12 die Konturen der Cluster dargestellt. Die Maßzahlen zur Clusteranalyse, wie Partitionskoeffizient und Partitionsentropie, unterstützen die Annahme zur Auswahl des Algorithmus. Der Partitionskoeffizient bezeichnet dabei den Grad der Trennung und die Partitionsentropie die Verteilung der Daten im Merkmalsraum zur Bestimmung der Eindeutigkeit der Zuordnungen. Je größer der Partitionskoeffizient und je kleiner die Entropie ist, desto eindeutiger ist die Zuordnung der Daten zu den jeweiligen Clustern. Beispiel: Bei vollständig gleichverteilten Daten wäre die Entropie 1 und damit jedes Datum jedem Cluster zu gleichen Anteilen zugeordnet. Anhand der Berechnungszeit und der Maßzahlen (Tab. 3.7) lassen sich die Ergebnisse vergleichen.

### 3.8 Selbstorganisierende Karten (SOM, Lit. [5])

Bei den oben verwendeten Verfahren musste die Clusteranzahl vordefiniert werden. Mittels SOM wird nun untersucht, ob die vordefinierte Anzahl der Cluster als richtig angenommen werden kann. (Der vereinfachte Datensatz erlaubt dies offensichtlich,

	t (ms)	it	PC	PE
FCM	520.93	12	0.84	0.34
GK	676.93	24	0.93	0.17
GG	569.30	22	0.99	0.00

Tabelle 1: Maßzahlen

jedoch erfolgen die Untersuchungen bereits in Voraussicht der Analyse komplexerer und dichter gruppierterer Daten.) SOM kategorisieren automatisch (Training) anhand der Varianzen der Eingangsdaten. Sie sind eine Abbildung des Eingangsdatenmusters. Eine Selbstorganisierende Karte (SOM) findet Cluster unbekannter Anzahl durch Veränderung ihrer Topologie aus einem Initialisierungszustand. Die topologischen Gruppen müssen nach der Datenrepräsentation einem Label-Verfahren aus a-priori Prozesswissen unterzogen werden. Vorteilhaft bei der SOM ist dabei die Abbildung von n-dimensionalen Merkmalsvektoren auf ihre 2-dimensionale anschauliche Struktur. In diesem Zusammenhang werden die Klassen durch eine Anzahl von Prototypen repräsentiert, die als Ansatz für eine Struktur im Datenraum verwendet werden können. Die gekennzeichnete SOM (Label) kann als Entscheidungsfläche für die Zielanwendung dienen. In diesem Zusammenhang wird untersucht, ob anhand topologischer Veränderungen der Karte, Entscheidungsräume modifiziert werden müssen und damit Rückschlüsse auf die Datenqualität zu ziehen sind.

### 3.8.1 Aufbau

Die SOM besteht aus einer 2-dimensionalen Schicht (x-y-Anordnung) von Neuronen. Die Gewichtsvektoren der Neuronen sollen möglichst so im n-dimensionalen Datenraum verteilt werden, dass sie diesen hinreichend gut repräsentieren. Zwischen den Neuronen existiert eine lokale Nachbarschaftsbeziehung. Durch das Training im Eingaberaum werden die Gewichtsvektoren der Neuronen so modifiziert, dass sie diesen Raum geordnet wiedergeben. Durch die lokale Nachbarschaftsfunktion werden jeweils auch die Nachbarschaftsneuronen des jeweiligen Gewinnerneurons geändert. Der SOM werden sequentiell alle Eingabemuster präsentiert. Der jeweilige Gewinner  $j^{win}$  bezeichnet das Neuron, dessen Gewichtsvektor den kleinsten Abstand zu einem Eingabedatum besitzt.

$$j^{win} = \arg \min_j \| \underline{x}(t) - \underline{w}_j(t) \|; j = 1, \dots, M \times N \quad (9)$$

Die Gewichtsvektoren  $\underline{w}_j$  werden in Richtung des Eingabevektors unter Verwendung einer (dynamisch angepassten) Lernrate  $\eta$  geändert.

$$\underline{w}_j(t+1) = \underline{w}_j(t) + \eta_t \text{nei}_{t,j^{win}}(\underline{x}(t) - \underline{w}_j(t)) \text{ mit } \eta_t = \eta_0 \left( \frac{1.0 - it}{it_{max}} \right) \quad (10)$$

Die Lernrate wird mit der Anzahl der Neuronen verringert. Die Nachbarschaftsfunktion  $\text{nei}$  kann über verschiedene Funktionsdefinitionen, z. B. der logistischen, berechnet werden.

$$\text{nei}_t = d_0 e^{\left( \frac{-\log(d_0)it}{it_{max}} \right)} \quad (11)$$

$d_0$  bezeichnet den maximal zulässigen Abstand. Für die Nachbarschaftsfunktion muss zudem ein Verfahren für die Kartenränder festgelegt werden. Im Modus *ranged*

endet die Funktion am Kartenrand, im Modus *wrapped* wird sie auf der gegenüberliegenden Kartenseite fortgeführt (Abb. 13). Die SOM wurde durch eine 64x64-Karte initialisiert, als Konvergenzkriterium wurden 500 Iteration in 5 Epochen, eine Startlernrate von 0.9 und  $d_0 = 30$  festgelegt. Die Gaußsche Nachbarschaftsfunktion passt die hexagonal um das Gewinnerneuron angeordneten Neuronen iterativ dem Dateneingaberaum an.

### 3.8.2 U-Matrix

Die Visualisierung des 2-dimensionalen Merkmalsraumes wurde in Lit. [6] vorgestellt. Die U-Matrix veranschaulicht Informationen zur lokalen Struktur der Abstände der Datensätze durch Analyse der modifizierten Gewichtsvektoren der Neuronen der SOM. Wenn  $v$  ein Neuron der Karte ist und  $nei_v$  die Anzahl der betrachteten Nachbarneuronen und  $w_v$  der Gewichtsvektor, dann werden die Höheninformationen für U berechnet mit

$$h_v = \sum_{m \in nei_v} d(w_v, w_m) \quad (12)$$

wobei  $d$  der Abstand für die Erstellung der Karte ist. Die Gruppen im Eingaberaum werden durch die verdichteten Konturen getrennt. Abb. 14 stellt die an einer SOM generierte U-Matrix anhand des Beispieldatensatzes dar. Die entsprechenden Cluster können in den Tälern der Matrix bezeichnet werden.

## 4 Zusammenfassung

Durch die Aufnahme von Spektren im Bereich der Resonanz des Sensors wurde zunächst nach einem geeigneten analytischen Modell für die Nachbildung des Funktionsverlaufs der Resonanzkurve gesucht. Die aus einer nichtlinearen Funktionsapproximation gewonnenen Parameterdaten bilden die Merkmale für die Clusteranalyse (Lernphase). Geeignete Clusteranalysemethoden erzeugen anhand der Merkmalsdaten Entscheidungsmuster für die Online-Datenanalyse auf Mikrocontrollern (Arbeitsphase). Die Auswahl der Merkmale definierenden Parameter kann aufgrund der Ergebnisse der Datenanalyse als hinreichend genau angenommen werden. Im Fokus eines neuen Sensorkonzeptes und Messgerätes unter Ausnutzung verschiedener Messmodi wurden mit einem Beispiel-Datensatz verschiedene Clustermethoden getestet. Die possibilistische Clusteranalyse ergab eine hinreichend genaue Clustering mittels Gath-Geva-Algorithmus.

## 5 Ausblick

Im 2-dimensionalen Datenraum (Frequenz-Frequenz-Plot) kann das Muster einer *gelernten* SOM als reduzierter Datenraum aufgefasst werden. Jeder Gewichtsvektor eines SOM-Neurons bildet einen Repräsentanten aus. Für jeden Repräsentanten werden die Zugehörigkeiten eines Datensatzes mittels eines possibilistischen Algorithmus berechnet. Dadurch wird eine Gruppe nicht mehr nur durch einen sondern

durch eine Anzahl von Prototypen beschrieben. Die Zusammenfassung der Prototypen ermöglicht eine komplexere Ausbildung der Cluster Grenzen und lokale Betrachtung der Lage- und Dichteigenschaften des Datensatzes im Merkmalsraum. Die Kombination aus SOM und possibilistischen Clusteranalysealgorithmus wird als *Fuzzy Kohonen Clustering Network* (FKCN, Lit. [10]) bezeichnet. Das FKCN gruppiert nach FCM-Methoden unter Zuhilfenahme der SOM-Repräsentanten. Die Untersuchungen werden auf die FCM-Ableitungen GK und GG erweitert. Erste Untersuchungen mittels FCM und GK auf den Beispieldatensatz in Abb. 15 ergaben einen Konturplot der die Dichteinformation noch unbeachtet lässt (zu erkennen an den durch Kreuze markierten den überbestimmten Repräsentanten), aufbauend auf einer  $6 \times 6$  SOM. Ziel der Ausweitung der Untersuchungen auf FKCN soll die Möglichkeit der Abbildung des 2-dimensionalen Datensatzes auf einen reduzierten Repräsentanten-Datensatz und damit eine effizientere, schnellere, jedoch hinreichend genaue Bestimmung der Gaszusammensetzungen in der Arbeitsphase sein.

## Literatur

- [1] Richter, D.; Fritze, H.; et. al.: *Selectivity improvement of high temperature resonant gas sensors by application of different electrode layouts*, Proceedings, Sensors, 2005.
- [2] Schneider, T.; et. al.: *Novel impedance interface for resonant high-temperature gas sensors*, Eurosensors XVIII., Rome, 2004, Proceedings, pp. 103-104.
- [3] Gustafson, D. E., Kessel, W. C.: *Fuzzy clustering with a fuzzy covariance matrix*, IEEE Press, New York, 1992.
- [4] Gath, I.; Geva, A. B.: *Unsupervised Optimal Fuzzy Clustering*, IEEE Trans. Pattern Analysis and Machine Intelligence 11 (1989), pp. 773-781.
- [5] Kohonen, T.: *Self-organizing maps*, Springer-Verlag, Berlin, 2001.
- [6] Ultsch, A.; Simon, H. P.: *Kohonen's Self Organizing Feature Maps for Exploratory Data Analysis*, Kluwer Academic Press, Dordrecht (Netherlands), 1990, pp. 305-308.
- [7] Höppner, F.; Klawonn, F.; Kruse, R.: *Fuzzy-Clusteranalyse. Verfahren für die Bilderkennung, Klassifikation und Datenanalyse*, Vieweg, Braunschweig, Wiesbaden, 1997.
- [8] Niebling, G.: *Strategien der Signalverarbeitung in der Chemosensorik. Konventionelle Methoden und neuronale Netze*, Herbert Utz Verlag Wissenschaft, München, 1996.
- [9] Press, W. H.; et. al.: *Numerical Recipes in Pascal*, Cambridge University Press, New York, 1989.
- [10] Tsao, E. C.-K.; Bezdek, J. C.; Pal, N. R.: *Fuzzy Kohonen Clustering Networks*, Pattern Recognition, Vol. 27, No. 5, 1994, pp. 757-764.

[11] Bauersfeld, N.; Kramer, K.-D.: *BMBF-Zwischenbericht 03*, Wernigerode, Hochschule Harz 2005.

[12] Fritze, H.; Richter, D.: *BMBF-Zwischenbericht 04*, Clausthal-Zellerfeld: TU Clausthal 2005.

## A Abbildungen

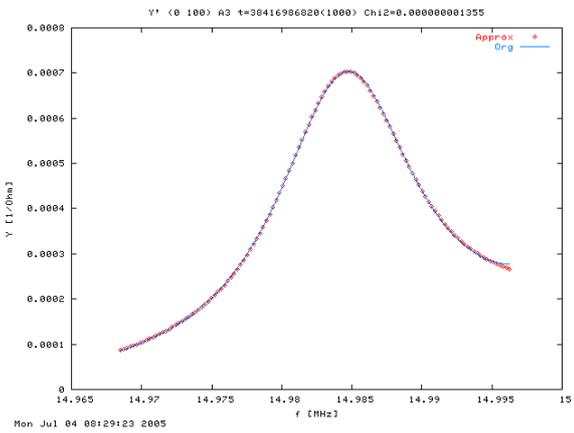
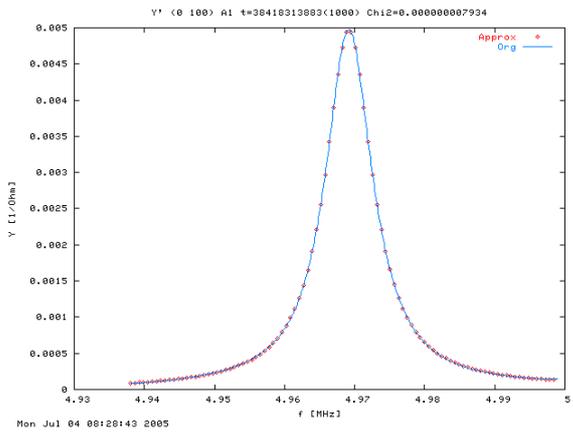


Abbildung 1: Lorentz-Fit, 1. Oberton

Abbildung 2: Lorentz-Fit, 3. Oberton

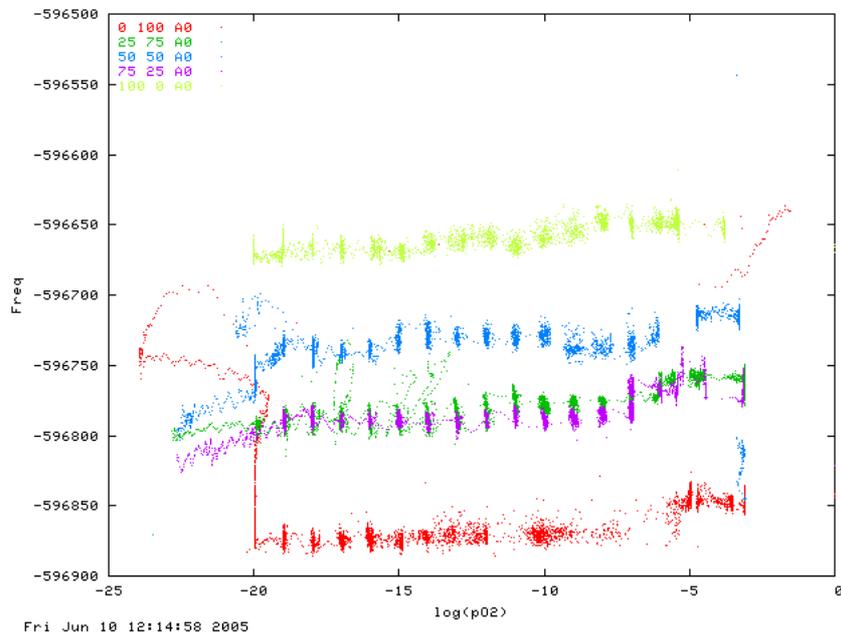


Abbildung 3: Resonanzfrequenz A

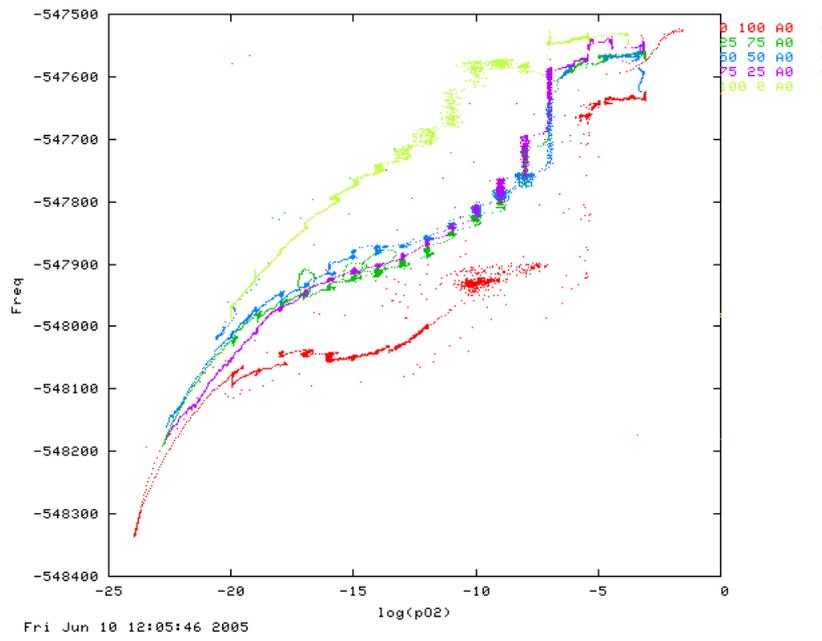


Abbildung 4: Resonanzfrequenz B

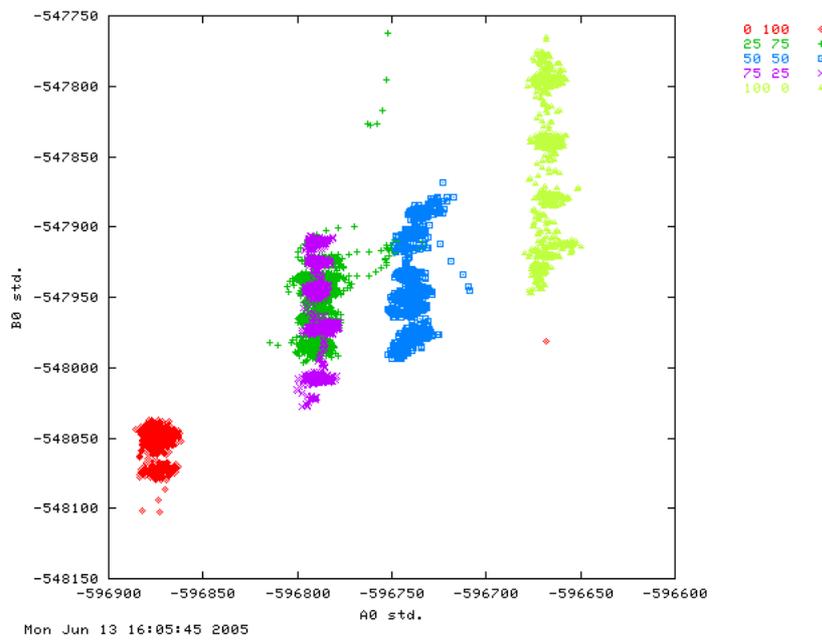


Abbildung 5: Frequenz-Frequenz-Diagramm

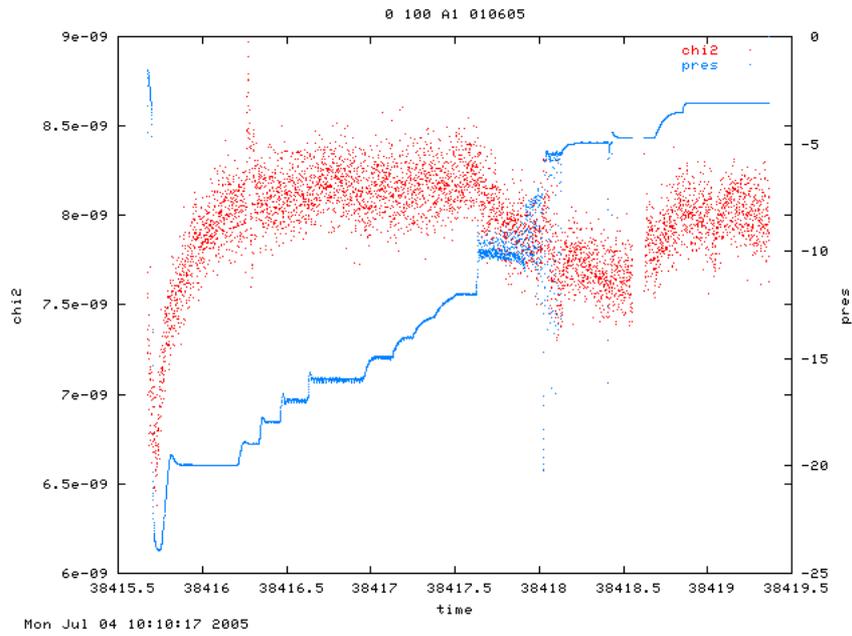


Abbildung 6: Approximationsfehler

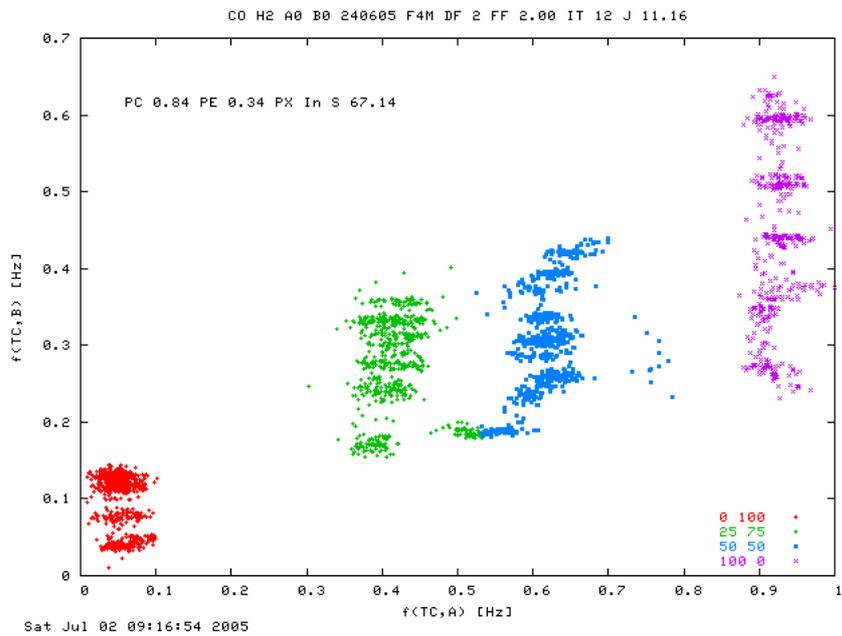


Abbildung 7: F4M-Frequenz-Frequenz-Daten

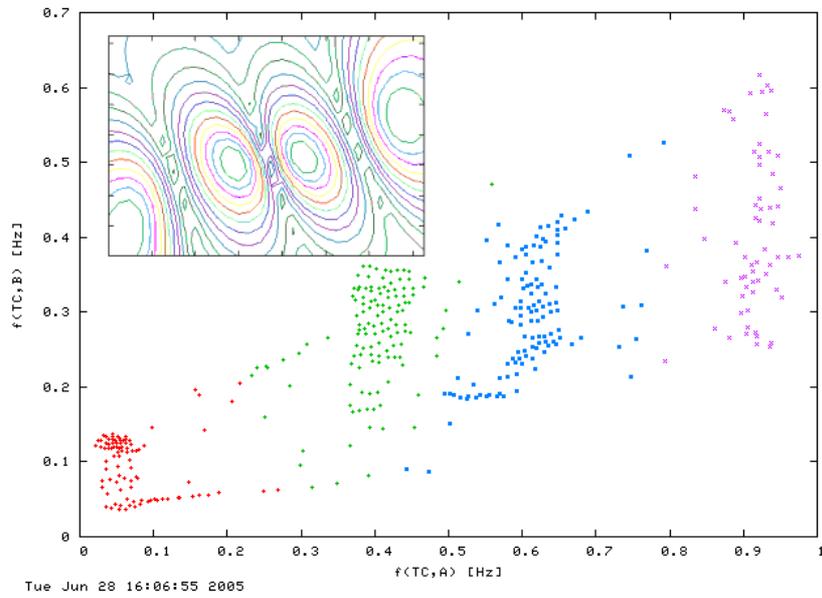


Abbildung 8: Frequenz-Frequenz-Plot mit GK-Kontur

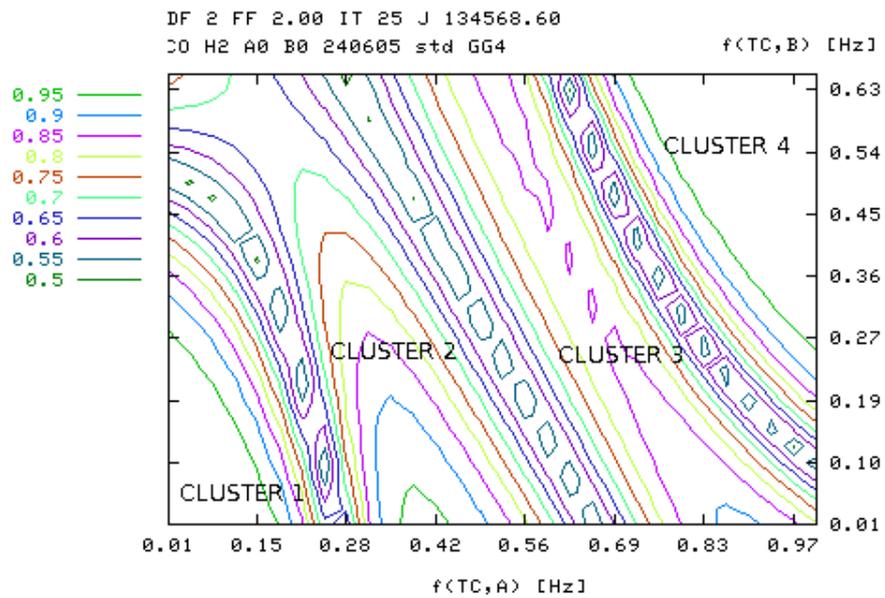


Abbildung 9: GG-Kontur und resultierende Cluster

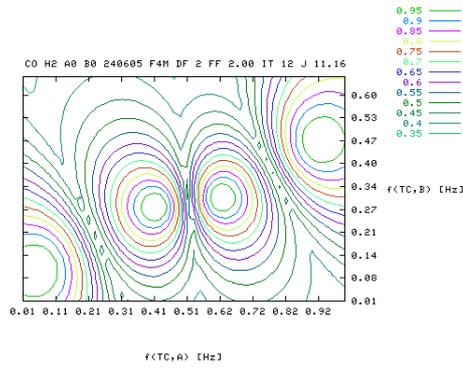


Abbildung 10: FCM-Kontur

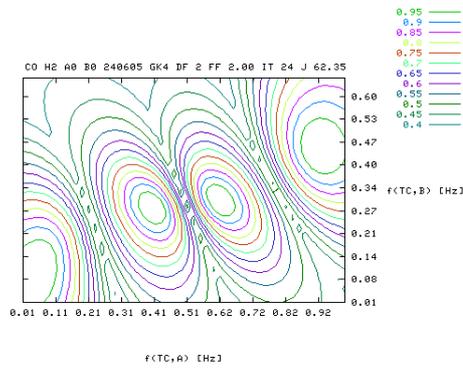


Abbildung 11: GK-Kontur

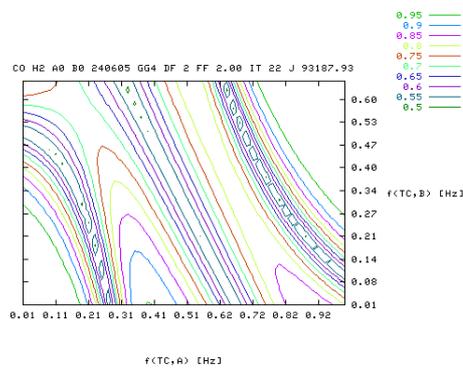


Abbildung 12: GG-Kontur

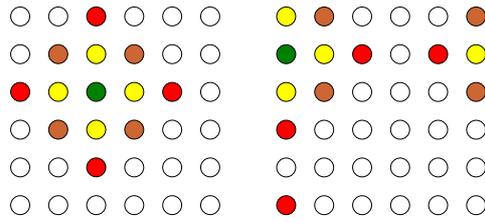


Abbildung 13: Nachbarschaftsfunktion

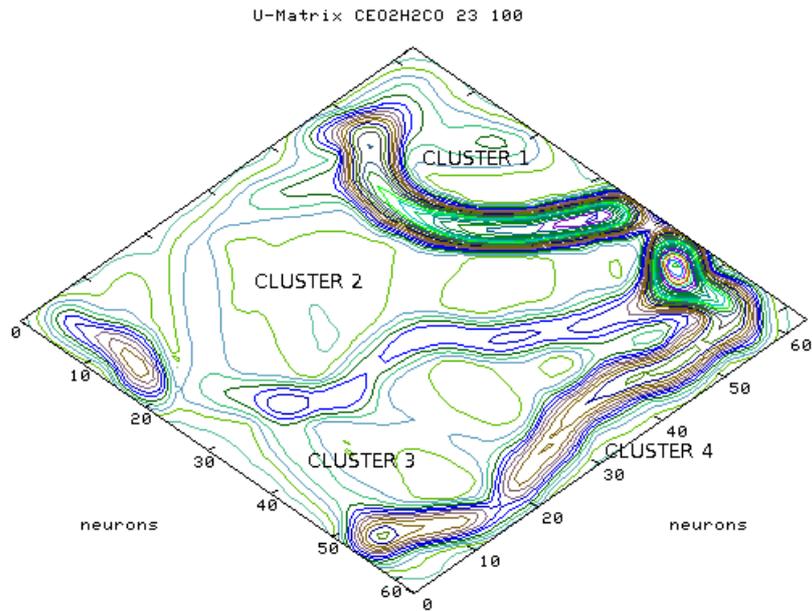


Abbildung 14: SOM-U-Matrix

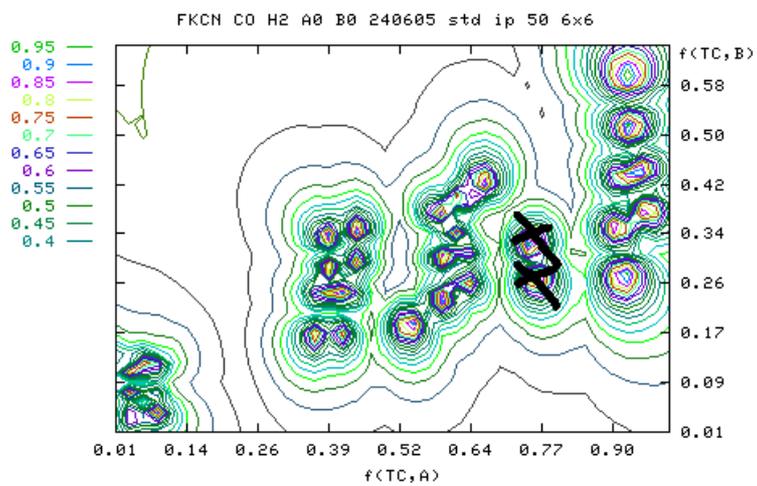


Abbildung 15: FKCN: SOM-GK-Frequenz-Frequenz-Plot

# Fuzzy Arithmetic – A New Approach to Modeling and Simulation of Uncertain Systems

**Michael Hanss**

Institut für Angewandte und Experimentelle Mechanik,  
Universität Stuttgart,  
Pfaffenwaldring 9, 70550 Stuttgart  
Tel.: (0711) 685-6273  
Fax: (0711) 685-6282  
E-Mail: hanss@mecha.uni-stuttgart.de

## **Abstract**

As a new approach to modeling, simulation and analysis of uncertain systems, a general method is presented in theory and practice. In this concept, the uncertain parameters of the system are modeled by fuzzy numbers in contrast to random numbers used in stochastic approaches. The fuzzy numbers form a special class of fuzzy sets and comply perfectly with the human perception of quantifying uncertainty. After an introduction to the general problem of fuzzy arithmetic, the transformation method is presented as a practical implementation which avoids the serious drawbacks of conventional fuzzy arithmetic, such as the overestimation effect, and which allows the inclusion of fuzzy arithmetic in any existing and commercially available software environment for system simulation without expensive rewriting of the program code. As a result of the simulation of the model, it is possible to quantify the overall uncertainty of the output values, providing a worst-case scenario. Additionally, as a result of the analysis of the model, the uncertain model parameters can be rated with respect to their individual influence on the overall output uncertainty. As an example from mechanical engineering, given by the finite element model of an automotive component, the simulation of the vibration behavior of an engine hood is presented and discussed in the presence of uncertain model parameters.

## **1 Introduction**

To achieve reliable results for the numerical simulation of real-world systems, exact values for the parameters of the model equations should be available. In practice, however, those exact values can often not be provided, and the parameters of the models usually exhibit a rather high degree of uncertainty. These uncertainties can be classified and assigned to different groups according to the type of their origin. In this context, we can distinguish between ‘unintentional uncertainties’, which arise due to partial lack or complete absence of information, and ‘intentional uncertainties’, which are usually the consequence of simplification. Some typical examples for the prevalent unintentional uncertainties are: scatter or variability, which arises due to irregularities in fabrication, measurement noise or other unmodeled disturbance signals that impair the identification of model parameters, vagueness, which is, for example, present if verbal characterizations, such as ‘nearly clamped’, are to be incorporated, and finally, idealization which is always inherent in modeling procedures when real-world systems are represented by mathematical models.

A very practical approach to the modeling and simulation of uncertain systems is the numerical implementation of the uncertain model parameters as fuzzy numbers, and then, the use of fuzzy arithmetic based on the transformation method [3, 4], as outlined in Section 2. By using this technique for the simulation of systems with uncertain model parameters, the complete information about the uncertainties can be included in the model and one can demonstrate how the uncertainties are propagated through the calculation procedure. The transformation method avoids the possibly serious drawbacks of conventional fuzzy arithmetic [3, 7], and, as an additional advantage, it can also be used to determine the degrees of influence of each fuzzy parameter, i.e., the proportion to which the uncertainty of each model parameter contributes to the overall uncertainty of the model output. Finally, in Sections 3, the transformation method is applied to the simulation and analysis of the vibration behavior of an engine hood in the presence of uncertain model parameters.

## 2 Fuzzy arithmetical concept

### 2.1 Fuzzy sets and fuzzy numbers

The theory of fuzzy sets was introduced by Zadeh [12] as an extension or generalization of classical set theory. According to the formulation of Cantor, presented in the late nineteenth century, a classical or so-called crisp set  $A$ ,  $A \subseteq U$ , can be defined as a collection of objects or elements  $x \in U$ , which are characterized by some well-defined property. Hence, if an element shows this property, it belongs to the set  $A$ , otherwise, it is excluded.

When we consider, as an example, the continuous and non-countable universal set  $U$  of possible outside temperatures  $x$  in degrees Celsius, we can use classical set theory and define the set  $A$  of ‘freezing temperatures’ by

$$A = \{x \in U \mid x < 0\} , \quad (1)$$

or alternatively by the *characteristic function*

$$\mu_A(x) = \begin{cases} 1 & \text{for } x \leq 0 \\ 0 & \text{for } x > 0 \end{cases} , \quad x \in U . \quad (2)$$

Thus, the property ‘ $x$  is the freezing temperature’ allows a non-ambiguous definition of the set  $A$ , that is, it allows a clear distinction between the elements that belong to  $A$ , and those which do not.

Classical set theory, however, reaches its limits when the property that determines the membership of an element to a set is formulated in such a way that a clear distinction between either membership or exclusion is no longer possible. As an extension of the example above, we can consider the following question: How does the set  $\tilde{A}$  of ‘low temperatures’ look like? Even though the classification of temperatures is, of course, very much dependent on the personal perception of ‘low temperature’, or ‘cold’, respectively, it is obvious that a clear division of the universal set into elements that definitely belong to the set, and those that are completely excluded, does

no longer make sense. The notion of a fuzzy property ‘ $x$  is a low temperature’ for the set  $\tilde{A}$  necessitates an extension of classical set theory towards a generalized set theory, where in addition to membership and exclusion there is also the possibility for the provision of gradations between the two groups. Against this background, so-called fuzzy sets can be introduced as a generalization of conventional sets by allowing elements of a universal set not only to entirely belong or not to belong to a specific set, but also to belong to the set to a certain degree [12]. For the description of fuzzy sets, the characteristic function  $\mu_A$  of a crisp set  $A$ , where  $\mu(x) \in \{0, 1\}$ , can be generalized to a *membership function*  $\mu_{\tilde{A}}$  for a fuzzy set  $\tilde{A}$ , where  $\mu(x) \in [0, 1]$ .

A very promising application of the theory of fuzzy sets, which is rather different from the well-established use of fuzzy set theory in fuzzy control, is the numerical implementation of uncertain model parameters as fuzzy numbers [2]. Basically, fuzzy numbers can be considered as a special class of fuzzy sets showing some specific properties [8]. On this basis, closed intervals and crisp numbers of the form

$$[a, b] = \{x \mid a \leq x \leq b\} \quad \text{and} \quad c = \{x \mid x = c\}, \quad x \in \mathbb{R}, \quad (3)$$

can be considered as classical subsets of the universal set  $U = \mathbb{R}$  which can also be expressed by

$$\mu_{[a,b]}(x) = \begin{cases} 1 & \text{for } a \leq x \leq b \\ 0 & \text{otherwise} \end{cases} \quad \text{and} \quad \mu_c(x) = \begin{cases} 1 & \text{for } x = c \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

when using the membership function  $\mu(x) \in \{0, 1\}$ ,  $x \in \mathbb{R}$  (Figure 1). Fuzzy numbers, instead, are defined as convex fuzzy sets over the universal set  $\mathbb{R}$  with their membership functions  $\mu(x) \in [0, 1]$ , where  $\mu(x) = 1$  is true only for one single value  $x = \bar{x} \in \mathbb{R}$ , the so-called center value  $\bar{x}$ . For example, a fuzzy number  $\tilde{p}_1$  of quasi-Gaussian shape and expressed by the abbreviated notation [5]

$$\tilde{p}_1 = \text{gfn}^*(\bar{x}_1, \sigma_1, \sigma_r), \quad (5)$$

can be defined by the membership function

$$\mu_{\tilde{p}_1}(x) = \begin{cases} 0 & \text{for } x \leq \bar{x}_1 - 3\sigma_1 \\ \exp \left[ -\frac{(x - \bar{x}_1)^2}{2\sigma_1^2} \right] & \text{for } \bar{x}_1 - 3\sigma_1 < x < \bar{x}_1 \\ \exp \left[ -\frac{(x - \bar{x}_1)^2}{2\sigma_r^2} \right] & \text{for } \bar{x}_1 \leq x < \bar{x}_1 + 3\sigma_r \\ 0 & \text{for } x \geq \bar{x}_1 + 3\sigma_r \end{cases} \quad \forall x \in \mathbb{R}. \quad (6)$$

A fuzzy number  $\tilde{p}_2$  of triangular (linear) shape, expressed by the abbreviated notation [5]

$$\tilde{p}_2 = \text{tfn}(\bar{x}_2, \alpha_1, \alpha_r), \quad (7)$$

can be defined by the membership function

$$\mu_{\tilde{p}_2}(x) = \begin{cases} 0 & \text{for } x \leq \bar{x}_2 - \alpha_1 \\ 1 + (x - \bar{x}_2)/\alpha_1 & \text{for } \bar{x}_2 - \alpha_1 < x < \bar{x}_2 \\ 1 - (x - \bar{x}_2)/\alpha_r & \text{for } \bar{x}_2 \leq x < \bar{x}_2 + \alpha_r \\ 0 & \text{for } x \geq \bar{x}_2 + \alpha_r \end{cases} \quad \text{or} \quad (8)$$

$$\mu_{\tilde{p}_2}(x) = \min \left\{ \max \left[ 0, 1 - \frac{(\bar{x}_2 - x)}{\alpha_1} \right], \max \left[ 0, 1 - \frac{(x - \bar{x}_2)}{\alpha_r} \right] \right\} \quad \forall x \in \mathbb{R}. \quad (9)$$

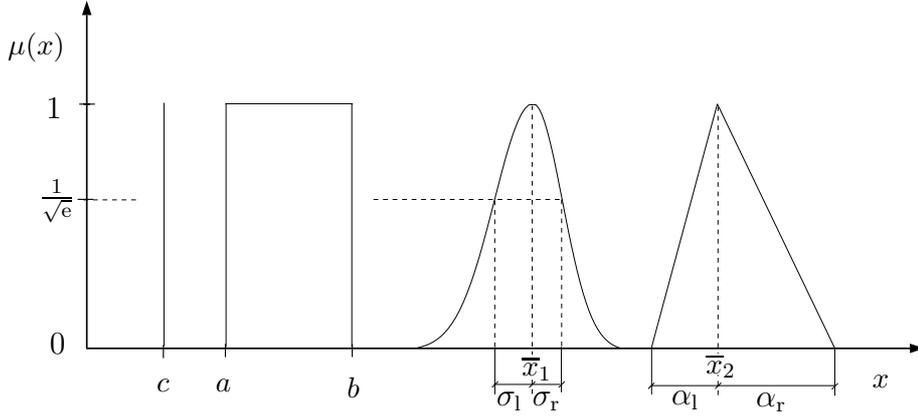


Figure 1: Closed interval  $[a, b]$  and crisp number  $c$  as well as quasi-Gaussian fuzzy number  $\tilde{p}_1$  and triangular fuzzy number  $\tilde{p}_2$  expressed by their membership functions.

## 2.2 The transformation method

To carry out arithmetical operations between fuzzy numbers, a so-called standard fuzzy arithmetic [5] has been defined [1, 8], where each fuzzy number is decomposed into a set of intervals for the different levels of membership, so that conventional interval arithmetic [11] can be applied separately for each level of membership. Upon closer examination, however, it is clear that the application of standard fuzzy arithmetic to the simulation of real-world systems with uncertain model parameters is undesirable. It has been shown that, as a serious drawback of this method, the results calculated by standard fuzzy arithmetic normally do not reflect the real result of the problem, but a result of much higher uncertainty [3, 7]. This drawback, however, can effectively be avoided by using the transformation method proposed by HANSS in [3, 4] as a practical implementation of fuzzy arithmetic. In the following, the transformation method is presented in its general and its reduced form, which can be used for the simulation as well as the analysis of systems with uncertain parameters, so-called fuzzy-parameterized systems.

### 2.2.1 Definition of fuzzy-parameterized systems

Fuzzy-parameterized systems, in general, consist of the following three key components:

- A set of  $n$  independent fuzzy-valued parameters  $\tilde{p}_i$  given by the membership functions  $\mu_{\tilde{p}_i}(x_i)$ ,  $x_i \in \mathbb{R}$ ,  $i = 1, 2, \dots, n$ .
- The system model itself, consisting of  $N$  functions  $F_r$ ,  $r = 1, 2, \dots, N$ , that perform some operations on the input fuzzy numbers  $\tilde{p}_i$ ,  $i = 1, 2, \dots, n$ .
- $N$  fuzzy-valued output variables  $\tilde{q}_r$  with the membership functions  $\mu_{\tilde{q}_r}(z_r)$ ,  $z_r \in \mathbb{R}$ ,  $r = 1, 2, \dots, N$ , that are obtained as the result of the functions  $F_r$ .

Without loss of generality, the number of fuzzy-valued output variables can be set to  $N = 1$  in the following, that is, the fuzzy-parameterized system shall be given by

$$\tilde{q} = F(\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_n) . \quad (10)$$

Note that this is an abbreviated form that focuses primarily on the dependency of the fuzzy-valued parameters  $\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_n$ . It does, of course, not exclude the existence of independent input variables such as the time  $t$  in case of dynamic systems. In this connection, (10) may also be considered as a short form of

$$\tilde{q}(t) = F(t; \tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_n) . \quad (11)$$

### 2.2.2 Simulation of fuzzy-parameterized systems

For the fuzzy arithmetical simulation of fuzzy-parameterized systems on the basis of the transformation method, the following procedure can be formulated:

#### 1. Decomposition of the input fuzzy numbers

In a first step, the interval  $[0, 1]$  of the  $\mu$ -axis is subdivided into  $m$  intervals of length  $\Delta\mu = 1/m$  according to Figure 2, and the discrete values  $\mu_j$  of the  $(m + 1)$  levels of membership are then given by

$$\mu_j = \frac{j}{m} , \quad j = 0, 1, \dots, m . \quad (12)$$

In a second step, the input fuzzy numbers  $\tilde{p}_i$ ,  $i = 1, 2, \dots, n$ , of the fuzzy-parameterized model are decomposed into so-called  $\alpha$ -cuts, leading to the decomposed representations

$$P_i = \left\{ X_i^{(0)}, X_i^{(1)}, \dots, X_i^{(m)} \right\} , \quad i = 1, 2, \dots, n , \quad (13)$$

of the fuzzy numbers  $\tilde{p}_i$ , where each set  $P_i$  consists of the  $(m + 1)$  intervals

$$X_i^{(j)} = [a_i^{(j)}, b_i^{(j)}] = \text{cut}_{\mu_j}(\tilde{p}_i) , \quad a_i^{(j)} \leq b_i^{(j)} , \quad j = 1, 2, \dots, m , \quad (14)$$

$$X_i^{(0)} = [a_i^{(0)}, b_i^{(0)}] = [w_{l_i}, w_{r_i}] \quad \text{with} \quad ]w_{l_i}, w_{r_i}[ = \text{supp}(\tilde{p}_i) . \quad (15)$$

The operators  $\text{cut}_\alpha(\tilde{p}_i)$  and  $\text{supp}(\tilde{p}_i)$  represent, respectively, the  $\alpha$ -cut and the support set of the fuzzy number  $\tilde{p}_i$  [5].

#### 2. Transformation of the input intervals

In case of the *reduced transformation method*, the intervals  $X_i^{(j)}$ ,  $i = 1, 2, \dots, n$ , of each level of membership  $\mu_j$ ,  $j = 0, 1, \dots, m$ , are transformed into arrays  $\widehat{X}_i^{(j)}$  of the form

$$\widehat{X}_i^{(j)} = \left( \overbrace{(\alpha_i^{(j)}, \beta_i^{(j)}), (\alpha_i^{(j)}, \beta_i^{(j)}), \dots, (\alpha_i^{(j)}, \beta_i^{(j)})}^{2^{i-1} \text{ pairs}} \right) \quad (16)$$

with

$$\alpha_i^{(j)} = \underbrace{(a_i^{(j)}, \dots, a_i^{(j)})}_{2^{n-i} \text{ elements}} , \quad \beta_i^{(j)} = \underbrace{(b_i^{(j)}, \dots, b_i^{(j)})}_{2^{n-i} \text{ elements}} . \quad (17)$$

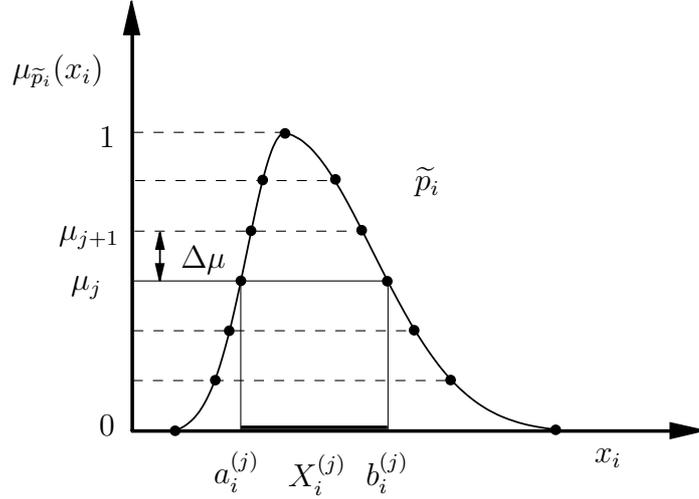


Figure 2: Decomposition of a fuzzy number  $\tilde{p}_i$  into intervals.

Obviously, only the boundary values  $a_i^{(j)}$  and  $b_i^{(j)}$  of the intervals  $X_i^{(j)}$ ,  $i = 1, 2, \dots, n$ ,  $j = 0, 1, \dots, m$ , are considered in this transformation scheme. This proves to be sufficient, however, if the problem  $F(\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_n)$  is monotonic, that is, if  $dF/dx_i \neq 0$  for  $x_i \in \text{supp}(\tilde{p}_i)$ ,  $i = 1, 2, \dots, n$ , or else if the problem is characterized by only one uncertain model parameter, that is, if  $n = 1$ .

If the fuzzy-parameterized model, instead, is expected to show non-monotonic behavior with respect to a number of  $\bar{n}$ ,  $1 \leq \bar{n} \leq n$ , out of the  $n > 1$  fuzzy-valued parameters  $\tilde{p}_i$ ,  $i = 1, 2, \dots, n$ , the *general transformation method* is recommended to be applied. In this case, additional points within the intervals  $X_i^{(j)}$ ,  $i = 1, 2, \dots, n$ ,  $j = 0, 1, \dots, m - 2$ , are considered for the transformation scheme, and the intervals are transformed into arrays  $\hat{X}_i^{(j)}$  of the form

$$\hat{X}_i^{(j)} = \left( \underbrace{(\gamma_{1,i}^{(j)}, \gamma_{2,i}^{(j)}, \dots, \gamma_{(m+1-j),i}^{(j)})}_{(m-j+1)^{i-1}}, \dots, \underbrace{(\gamma_{1,i}^{(j)}, \gamma_{2,i}^{(j)}, \dots, \gamma_{(m+1-j),i}^{(j)})}_{(m-j+1)\text{-tuples}} \right) \quad (18)$$

with

$$\gamma_{l,i}^{(j)} = \underbrace{(c_{l,i}^{(j)}, \dots, c_{l,i}^{(j)})}_{(m-j+1)^{n-i} \text{ elements}} \quad (19)$$

and

$$c_{l,i}^{(j)} = \begin{cases} a_i^{(j)} & \text{for } l = 1 \\ & \text{and } j = 0, 1, \dots, m, \\ \frac{1}{2} (c_{l-1,i}^{(j+1)} + c_{l,i}^{(j+1)}) & \text{for } l = 2, 3, \dots, m - j \\ & \text{and } j = 0, 1, \dots, m - 2, \\ b_i^{(j)} & \text{for } l = m - j + 1 \\ & \text{and } j = 0, 1, \dots, m. \end{cases} \quad (20)$$

### 3. Evaluation of the model

Assuming that the fuzzy-parameterized model is given by the functional expression  $F$  in (10), its estimation is carried out by evaluating the expression

separately at each of the columns of the arrays, using the conventional arithmetic for crisp numbers. That is, if the output  $\tilde{q}$  of the system can be expressed in its decomposed and transformed form by the arrays  $\widehat{Z}^{(j)}$ ,  $j = 0, 1, \dots, m$ , the  $k$ th element  ${}^k\hat{z}^{(j)}$  of the array  $\widehat{Z}^{(j)}$  is then determined by

$${}^k\hat{z}^{(j)} = F \left( {}^k\hat{x}_1^{(j)}, {}^k\hat{x}_2^{(j)}, \dots, {}^k\hat{x}_n^{(j)} \right), \quad (21)$$

where  ${}^k\hat{x}_i^{(j)}$  denotes the  $k$ th element of the array  $\widehat{X}_i^{(j)}$ .

#### 4. Retransformation of the output array

The decomposed representation of the fuzzy-valued model output  $\tilde{q}$ , expressed by the set

$$Q = \{Z^{(0)}, Z^{(1)}, \dots, Z^{(m)}\} \quad (22)$$

of the  $(m + 1)$  intervals

$$Z^{(j)} = [a^{(j)}, b^{(j)}] = \text{cut}_{\mu_j}(\tilde{q}), \quad a^{(j)} \leq b^{(j)}, \quad j = 1, 2, \dots, m, \quad (23)$$

$$Z^{(0)} = [a^{(0)}, b^{(0)}] = [w_l, w_r] \quad \text{with} \quad ]w_l, w_r[ = \text{supp}(\tilde{q}) \quad (24)$$

can be obtained by retransforming the arrays  $\widehat{Z}^{(j)}$  according to the recursive formulas

$$a^{(j)} = \min_k (a^{(j+1)}, {}^k\hat{z}^{(j)}) \quad \text{and} \quad b^{(j)} = \max_k (b^{(j+1)}, {}^k\hat{z}^{(j)}), \quad (25)$$

$$j = 0, 1, \dots, m - 1,$$

with

$$a^{(m)} = \min_k ({}^k\hat{z}^{(m)}) = \max_k ({}^k\hat{z}^{(m)}) = b^{(m)}. \quad (26)$$

#### 5. Recomposition of the output intervals

By recomposing the intervals  $Z^{(j)}$ ,  $j = 0, 1, \dots, m$ , of the set  $Q$  according to their levels of membership  $\mu_j$ , the output  $\tilde{q}$  of the fuzzy-parameterized system can be obtained.

### 2.2.3 Analysis of fuzzy-parameterized systems

Until now, the fuzzy-valued result  $\tilde{q}$  of a problem only reflects the overall influence of all the  $n$  uncertain model parameters  $\tilde{p}_i$ ,  $i = 1, 2, \dots, n$  together. A general methodology for quantitatively determining the proportions to which the  $n$  fuzzy-valued parameters  $\tilde{p}_i$  of a fuzzy-parameterized system separately contribute to the overall uncertainty of the output  $\tilde{q}$  is also provided by the transformation method as an additional feature. Instead of reducing the arrays  $\widehat{Z}^{(j)}$  immediately to the intervals  $Z^{(j)}$ ,  $j = 0, 1, \dots, m$ , as done in the retransformation step of the transformation method, the supplementary information encoded in the values and the arrangement of the elements in  $\widehat{Z}^{(j)}$  can be used. With the objective of uncovering this information, we can determine the coefficients  $\eta_i^{(j)}$ ,  $i = 1, 2, \dots, n$ ,  $j = 0, 1, \dots, (m - 1)$ , which for the *reduced transformation method* are defined by

$$\eta_i^{(j)} = \frac{1}{2^{n-1}(b_i^{(j)} - a_i^{(j)})} \sum_{k=1}^{2^{n-i}} \sum_{l=1}^{2^{i-1}} ({}^{s_2}\hat{z}^{(j)} - {}^{s_1}\hat{z}^{(j)}) \quad (27)$$

with

$$s_1(k, l) = k + (l - 1) 2^{n-i+1}, \quad (28)$$

$$s_2(k, l) = k + (2l - 1) 2^{n-i}, \quad (29)$$

and for the *general transformation method* by

$$\eta_i^{(j)} = \frac{1}{(m - j + 1)^{n-1} (b_i^{(j)} - a_i^{(j)})} \sum_{k=1}^{(m-j+1)^{n-i}} \sum_{l=1}^{(m-j+1)^{i-1}} (s_2 \hat{z}^{(j)} - s_1 \hat{z}^{(j)}) \quad (30)$$

with

$$s_1(k, l) = k + (m - j + 1)(l - 1)(m - j + 1)^{n-i} \quad (31)$$

$$= k + (l - 1)(m - j + 1)^{n-i+1}, \quad (32)$$

$$s_2(k, l) = k + ((m - j + 1)l - 1)(m - j + 1)^{n-i}. \quad (33)$$

As defined in (14) and (15), the values  $a_i^{(j)}$  and  $b_i^{(j)}$  are the lower and upper bounds of the interval  $X_i^{(j)}$ , and  $s_1 \hat{z}^{(j)}$  and  $s_2 \hat{z}^{(j)}$  denote, respectively, the  $s_1$ th and  $s_2$ th element of the array  $\hat{Z}^{(j)}$ . The coefficients  $\eta_i^{(j)}$  can be interpreted as gain factors that express the effect of the uncertainty of the  $i$ th parameter  $\tilde{p}_i$  on the uncertainty of the output  $\tilde{q}$  of the problem at the membership level  $\mu_j$ . More explicitly, within the range of uncertainty covered at the membership level  $\mu_j$ , deviations  $\Delta z^{(j)}$  from the modal value  $\bar{z}$  of the output fuzzy number  $\tilde{q}$  can be considered as being related to the corresponding deviations  $\Delta x_i^{(j)}$  from the modal values  $\bar{x}_i$  of the fuzzy parameters  $\tilde{p}_i$ ,  $i = 1, 2, \dots, n$ , by the approximation

$$\Delta z^{(j)} \approx \sum_{i=1}^n \eta_i^{(j)} \Delta x_i^{(j)}. \quad (34)$$

The validity of this relation has been proven by HANSS AND KLIMKE in [6]. To obtain a non-dimensional form of the influence measures with respect to the usually different physical dimensions of  $\tilde{p}_i$ , the standardized mean gain factors  $\kappa_i$  can be determined as an overall measure of influence according to

$$\kappa_i = \frac{\sum_{j=1}^{m-1} \mu_j \left| \eta_i^{(j)} (a_i^{(j)} + b_i^{(j)}) \right|}{2 \sum_{j=1}^{m-1} \mu_j} = \frac{1}{m-1} \sum_{j=1}^{m-1} \mu_j \left| \eta_i^{(j)} (a_i^{(j)} + b_i^{(j)}) \right|. \quad (35)$$

Finally, as a relative measure of influence, the normalized values  $\rho_i$  can be determined for  $i = 1, 2, \dots, n$  according to

$$\rho_i = \frac{\kappa_i}{\sum_{q=1}^n \kappa_q} = \frac{\sum_{j=1}^{m-1} \mu_j \left| \eta_i^{(j)} (a_i^{(j)} + b_i^{(j)}) \right|}{\sum_{q=1}^n \sum_{j=1}^{m-1} \mu_j \left| \eta_q^{(j)} (a_q^{(j)} + b_q^{(j)}) \right|}, \quad (36)$$

satisfying the consistency condition

$$\sum_{i=1}^n \rho_i = 1. \quad (37)$$

The gain factors  $\kappa_i$  quantify in a standardized way the effect of the  $i$ th fuzzy-valued model parameter  $\tilde{p}_i$  on the overall uncertainty of the output  $\tilde{q}$  of the problem by assuming every input parameter to exhibit the same amount of relative uncertainty with respect to its modal value.

In addition to many other advantages of the transformation method [5], its characteristic property of reducing fuzzy arithmetic to multiple crisp-number operations entails that the transformation method can be implemented without major problems into an existing software environment for system simulation. Expensive rewriting of the program code is not required. Instead, the steps of decomposition and transformation as well as of retransformation and recomposition can preferably be coupled to existing software by a separate pre- and postprocessing tool. For this purpose, the program FAMOUS (**F**uzzy **A**rithmetical **M**odeling **O**f **U**ncertain **S**ystems) has been developed, which carries out the transformation method and provides an interface to commercial finite element software.

### 3 An application to mechanical engineering: simulation and analysis of the vibrations of an engine hood

Addressing a finite element problem of industrial relevance, the transformation method is applied in the following to simulate and analyze the vibration behavior of an engine hood with uncertain model parameters. Specifically, we consider the engine hood of the roadster Mercedes SLK, which is shown in its top and bottom view in Figure 3. Basically, the engine hood consists of two parts: the upper sheet metal of thickness  $t_1$ , featuring a light gray color in Figure 3, and the bottom metal frame of thickness  $t_2$ , pictured in dark gray. Both components are made of the same material, characterized by the Young's modulus  $E$ , the Poisson's ratio  $\nu$ , and the mass density  $\rho$ . Whereas the Young's modulus  $E$  and the Poisson's ratio  $\nu$  are set to the crisp values

$$E = 2 \cdot 10^{11} \text{ N m}^{-2} \quad \text{and} \quad \nu = 0.314 , \quad (38)$$

the thickness parameters  $t_1$  and  $t_2$  as well as the mass density  $\rho$  are considered as uncertain, forming  $n = 3$  independent uncertain parameters of the model. They are quantified by symmetric quasi-Gaussian fuzzy numbers  $\tilde{p}_i$ ,  $i = 1, 2, 3$ , given by

$$\tilde{p}_1 = \tilde{t}_1 = \text{gfn}^*(\bar{t}_1, 3\% \bar{t}_1, 3\% \bar{t}_1) , \quad \bar{t}_1 = 0.8 \text{ mm} = 8 \cdot 10^{-4} \text{ m} , \quad (39)$$

$$\tilde{p}_2 = \tilde{t}_2 = \text{gfn}^*(\bar{t}_2, 3\% \bar{t}_2, 3\% \bar{t}_2) , \quad \bar{t}_2 = 0.7 \text{ mm} = 7 \cdot 10^{-4} \text{ m} , \quad (40)$$

$$\tilde{p}_3 = \tilde{\rho} = \text{gfn}^*(\bar{\rho}, 3\% \bar{\rho}, 3\% \bar{\rho}) , \quad \bar{\rho} = 8400 \text{ kg m}^{-3} , \quad (41)$$

where the definition of the standard deviations corresponds to a worst-case deviation of  $\pm 9\%$  from the modal values.

For the simulation of the uncertain system, the program FAMOUS uses the transformation method in its reduced form in conjunction with the commercial finite element software package MSC.Marc. The engine hood is discretized according to the mesh in Figure 3, consisting of 1 889 nodes and 2 235 thin-shell elements – most

of them four-node elements, some of three-node type. As the output values of the system, the first six uncertain eigenfrequencies

$$\tilde{q}_r = \tilde{f}_r, \quad r = 1, 2, \dots, 6, \quad (42)$$

of the engine hood are determined, which satisfy the condition

$$\det(\tilde{\mathbf{K}} - \tilde{\mathbf{M}} \tilde{\omega}_r^2) = 0, \quad \tilde{\omega}_r = 2\pi \tilde{f}_r, \quad (43)$$

where  $\tilde{\mathbf{K}}$  and  $\tilde{\mathbf{M}}$  are the fuzzy-valued global stiffness matrix and the fuzzy-valued mass matrix, respectively. Finally, to simulate the bearings of the engine hood on the car body, the displacements of some nodes at the locations A, B<sub>1</sub>, and B<sub>2</sub> in Figure 3 are set to zero.

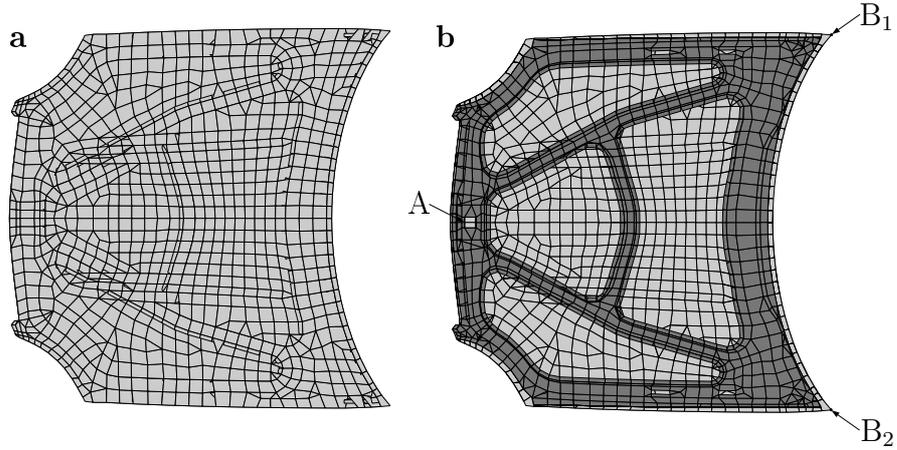


Figure 3: Mercedes SLK engine hood including the finite element mesh and the locations A, B<sub>1</sub>, and B<sub>2</sub> of the bearings: (a) top view; (b) bottom view (courtesy of DaimlerChrysler AG, Stuttgart).

As result of the finite element simulation using the reduced transformation method with the decomposition number  $m = 10$ , the uncertain eigenfrequencies  $\tilde{f}_r$  can be achieved with their membership functions  $\mu_{\tilde{f}_r}(f)$ ,  $r = 1, 2, \dots, 6$ , plotted in Figure 4.

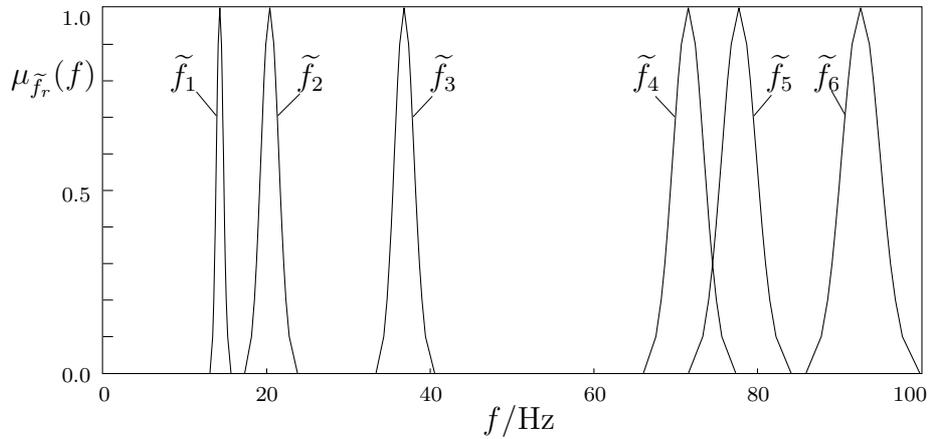


Figure 4: Membership functions  $\mu_{\tilde{f}_r}(f)$  of the first six uncertain eigenfrequencies  $\tilde{f}_r$ ,  $r = 1, 2, \dots, 6$ , of the engine hood.

We can see that the absolute imprecision of the uncertain eigenfrequencies increases with the magnitude of their modal values; however, the relative worst-case deviations from the modal values appear constant at about  $\pm 9\%$ , which coincides with the originally assumed worst-case deviations of the uncertain model parameters. Furthermore, to visualize the type of oscillations assigned to the different eigenfrequencies  $\tilde{f}_1, \tilde{f}_2, \dots, \tilde{f}_6$ , the corresponding eigenmodes of the engine hood are presented in Figures 5 to 10. In these diagrams, the amplitudes of the displacements are quantified by a gray scale, where darker gray tones symbolize higher amplitudes.

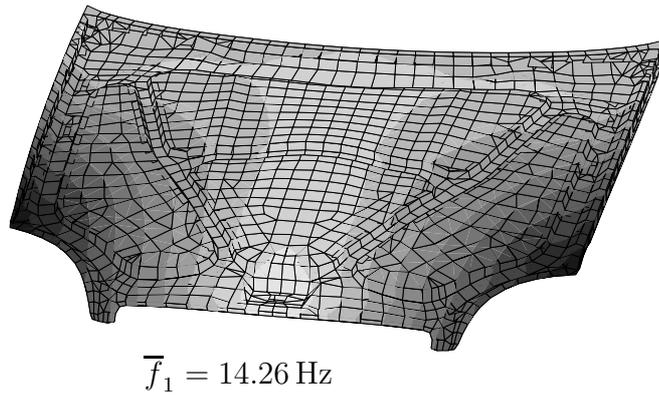


Figure 5: Eigenmode of the engine hood for the first eigenfrequency with its modal value  $\bar{f}_1 = 14.26 \text{ Hz}$ .

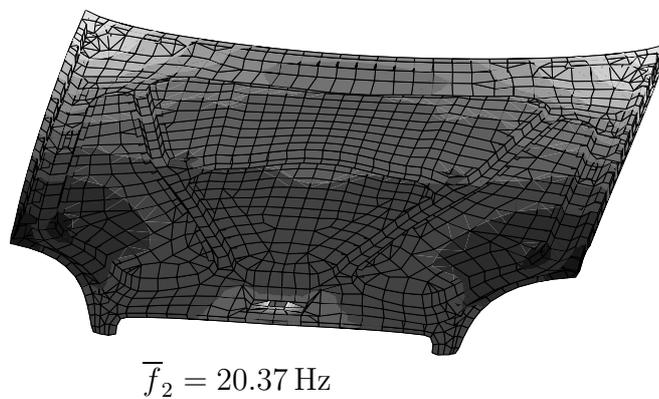
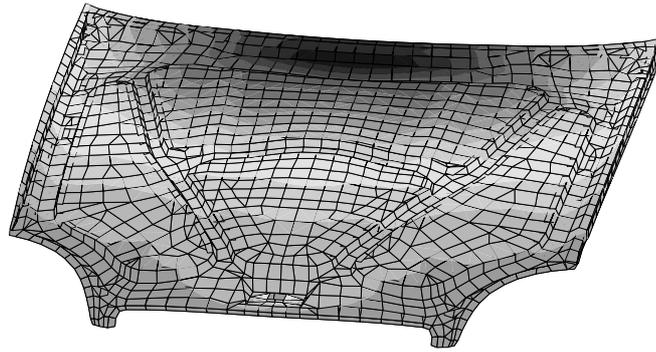
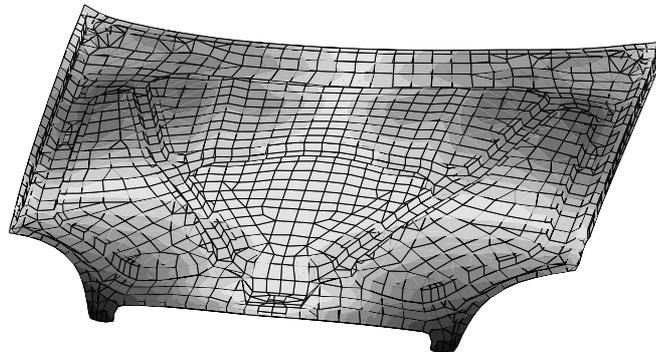


Figure 6: Eigenmode of the engine hood for the second eigenfrequency with its modal value  $\bar{f}_2 = 20.37 \text{ Hz}$ .



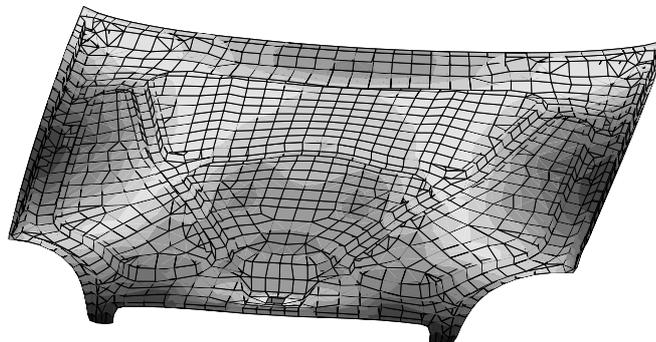
$$\bar{f}_3 = 36.79 \text{ Hz}$$

Figure 7: Eigenmode of the engine hood for the third eigenfrequency with its modal value  $\bar{f}_3 = 36.79 \text{ Hz}$ .



$$\bar{f}_4 = 71.54 \text{ Hz}$$

Figure 8: Eigenmode of the engine hood for the fourth eigenfrequency with its modal value  $\bar{f}_4 = 71.54 \text{ Hz}$ .



$$\bar{f}_5 = 77.73 \text{ Hz}$$

Figure 9: Eigenmode of the engine hood for the fifth eigenfrequency with its modal value  $\bar{f}_5 = 77.73 \text{ Hz}$ .

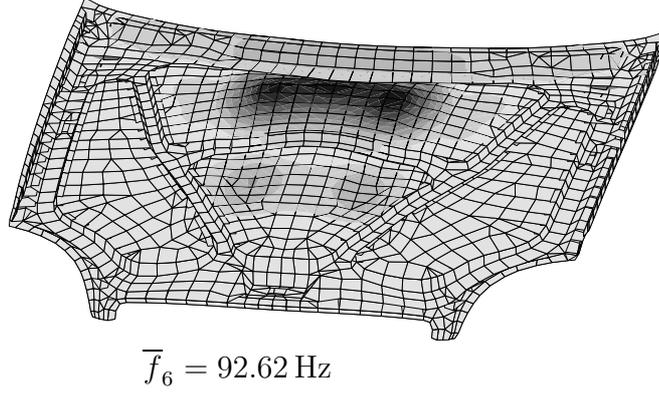


Figure 10: Eigenmode of the engine hood for the sixth eigenfrequency with its modal value  $\bar{f}_6 = 92.62 \text{ Hz}$ .

By evaluating the analysis part of the transformation method, the standardized mean gain factors  $\kappa_i(\tilde{f}_r)$  as well as the normalized values  $\rho_i(\tilde{f}_r)$  can be determined, providing a measure for the influence of the uncertainty of the  $i$ th model parameter  $\tilde{p}_i$  on the uncertainty of the  $r$ th eigenfrequency  $\tilde{f}_r$ . As an absolute measure of influence, the standardized mean gain factors  $\kappa_i(\tilde{f}_r)$ ,  $i = 1, 2, 3$ ,  $r = 1, 2, \dots, 6$ , are presented in Figure 11a, while Figure 11b shows the relative degrees of influence  $\rho_i(\tilde{f}_r)$ ,  $i = 1, 2, 3$ ,  $r = 1, 2, \dots, 6$ . Focusing on the influence of the uncertain parameters  $\tilde{p}_1 = \tilde{t}_1$  and  $\tilde{p}_2 = \tilde{t}_2$ , that is, on the thicknesses of the upper sheet metal and the bottom metal frame, we can see that the influence of  $\tilde{t}_2$  is significantly higher than the influence of  $\tilde{t}_1$  if the engine hood oscillates in its second eigenmode. However, if the sixth eigenmode is considered, we notice the opposite situation.

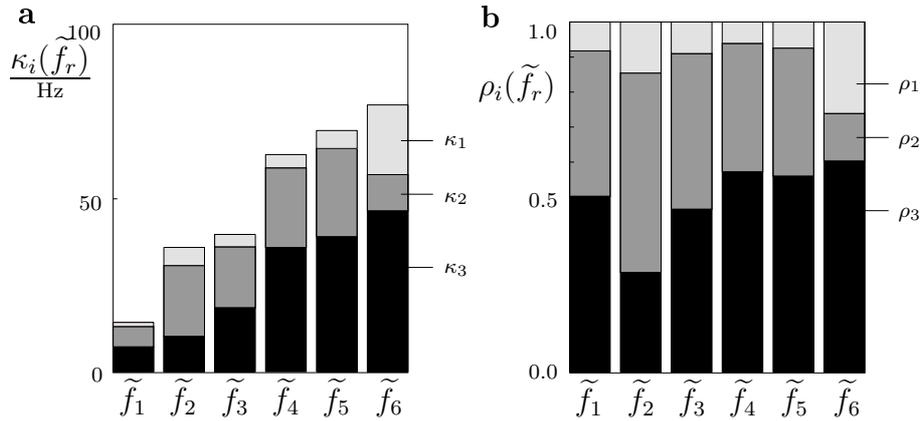


Figure 11: **(a)** Standardized mean gain factors  $\kappa_i(\tilde{f}_r)$  and **(b)** normalized degrees of influence  $\rho_i(\tilde{f}_r)$  with light gray for  $i = 1$ , dark gray for  $i = 2$ , and black for  $i = 3$ .

This result of the analysis of the transformation method can qualitatively be verified by comparing the pictures of the second and sixth eigenmode of the engine hood in Figures 6 and 10. The second eigenmode in Figure 6 represents the first bending mode of the hood about an in-plane axis that is orthogonal to the axis of symmetry

of the hood. In this eigenmode, the stiffness-inducing bottom frame of the hood, which is characterized by the uncertain thickness parameter  $\tilde{t}_2$ , will definitely be the determining factor of the oscillation, exhibiting a predominant influence on the uncertainty of the eigenfrequency  $\tilde{f}_2$ . Alternatively, the sixth eigenmode in Figure 10 represents a localized form of vibration, which primarily affects the upper sheet of the hood in the area located inside the bottom frame. For this reason, the uncertainty of the eigenfrequency  $\tilde{f}_6$  is considerably more influenced by the uncertain thickness parameter  $\tilde{t}_1$  of the upper sheet, than by the respective property  $\tilde{t}_2$  of the bottom frame.

## 4 Conclusions

A fuzzy arithmetical approach for the simulation and analysis of systems with uncertain parameters has been presented. In this concept, the uncertain parameters are modeled by fuzzy numbers, and the fuzzy-parameterized models are evaluated by the use of the transformation method in conjunction with existing simulation environments, such as commercial finite element software. A drawback of the transformation method may be seen in the possibly sizable number of system evaluations that is required, in particular if the general form of the transformation method is considered together with a significant number of fuzzy-valued model parameters. This may lead to high computational costs, especially if large-scale models are intended to be simulated. This drawback, however, can often be narrowed substantially by simply being aware of the fact that the number of effective model evaluations is not necessarily as high as the sum of entries required for the overall number of output arrays in the transformation method. That is, the set of combinations of parameter values, for which the model is to be evaluated, often exhibits a multiple occurrence of certain combinations, which is especially true if the general transformation method is applied for parameters with membership functions of symmetric triangular shape [5]. In those cases, the respective entries of the output arrays can simply be replicated from existing entries and do not call for new model evaluations. For appropriate applications, further reductions of the computational costs can be achieved by the use of the so-called extended transformation method [4] or a special approach based on piecewise multilinear sparse-grid interpolation [9, 10].

## References

- [1] D. Dubois and H. Prade. *Fuzzy Sets and Systems: Theory and Applications*. Mathematics in Science and Engineering, Vol. 144. Academic Press, New York - London, 1980.
- [2] M. Hanss. On the implementation of fuzzy arithmetical operations for engineering problems. In *Proc. of the 18th International Conference of the North American Fuzzy Information Processing Society – NAFIPS ‘99*, pages 462–466, New York, NY, USA, 1999.
- [3] M. Hanss. The transformation method for the simulation and analysis of systems with uncertain parameters. *Fuzzy Sets and Systems*, 130(3):277–289, 2002.

- [4] M. Hanss. The extended transformation method for the simulation and analysis of fuzzy-parameterized models. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 11(6):711–727, 2003.
- [5] M. Hanss. *Applied Fuzzy Arithmetic – An Introduction with Engineering Applications*. Springer, Berlin, 2005.
- [6] M. Hanss and A. Klimke. On the reliability of the influence measure in the transformation method of fuzzy arithmetic. *Fuzzy Sets and Systems*, 143(3):371–390, 2004.
- [7] M. Hanss and K. Willner. A fuzzy arithmetical approach to the solution of finite element problems with uncertain parameters. *Mechanics Research Communications*, 27:257–272, 2000.
- [8] A. Kaufmann and M. M. Gupta. *Introduction to Fuzzy Arithmetic*. Van Nostrand Reinhold, New York, 1991.
- [9] A. Klimke. Uncertainty modeling using fuzzy arithmetic based on dimension-adaptive sparse grids. In *Proc. of the 20th Canadian Congress of Applied Mechanics – CANCAM 2005*, pages 596–597, Montréal, QC, Canada, 2005.
- [10] A. Klimke, K. Willner, and B. Wohlmuth. Uncertainty modeling using fuzzy arithmetic based on sparse grids: applications to dynamic systems. *Int. Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 12(6):754–759, 2004.
- [11] R. E. Moore. *Interval Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1966.
- [12] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

# Nutzung negativer Regeln in TSK-Fuzzy-Systemen

Martin Sternke, Peter Krause<sup>1</sup>, Harro Kiendl<sup>2</sup>

Lehrstuhl für Systemanalyse

Universität Dortmund, 44221 Dortmund

Tel.: (0231) 9700-951 Fax: (0231) 9700-959

E-Mail: martin.sternke@cs.uni-dortmund.de

<sup>1</sup>NuTech Solutions GmbH

Martin-Schmeisser-Weg 15, 44227 Dortmund

Tel.: (0231) 725463-10 Fax: (0231) 725463-29

E-Mail: krause@nutechsolutions.de

<sup>2</sup>E-Mail: kiendl@hyperfuzzy.de

## Zusammenfassung

Eine Methode zur datenbasierten Modellierung von Prozessen durch Mamdani-Fuzzy-Systeme ist das Fuzzy-ROSA-Verfahren<sup>1</sup>. Dieses ermöglicht u. a. die Generierung und das manuelle Einfügen von negativen Mamdani-Regeln. Im Gegensatz zu den üblichen positiven Regeln sprechen diese keine Empfehlungen aus, sondern Warnungen bzw. Verbote. Negative Regeln erlauben daher eine transparente und kompakte Repräsentation von Expertenwissen, das häufig auch in dieser Form vorliegt. Mamdani-Fuzzy-Systeme zeichnen sich durch eine gute Interpretierbarkeit, Takagi-Sugeno-Kang-Fuzzy-Systeme (im Folgenden verkürzt als TSK-Fuzzy-Systeme bezeichnet) durch eine meist bessere Modellgenauigkeit aus. Um beide für Anwender wichtige, im Allgemeinen aber gegenläufige, Gesichtspunkte berücksichtigen zu können, ist in dem Fuzzy-ROSA-Verfahren die Möglichkeit geschaffen worden, die generierten Mamdani-Regeln unter Erhaltung ihrer Interpretierbarkeit in TSK-Regeln zu transformieren. Diese Transformation ist bisher auf positive Mamdani-Regeln beschränkt, sodass das Potenzial negativer Mamdani-Regeln hierfür nicht genutzt werden kann. In diesem Beitrag wird ein Ansatz vorgestellt, der diese Lücke schließt.

## 1 Einleitung

Mit zunehmender Komplexität moderner technischer wie auch nicht technischer Prozesse besteht ein steigender Bedarf an Verfahren zur Modellierung solcher Prozesse. Dabei werden in den Bereichen des Data-Minings und der datenbasierten Modellierung verstärkt Verfahren aus der Computational Intelligence eingesetzt. Fuzzy-Systeme haben sich dabei für viele Aufgabenstellungen als geeignet erwiesen, und sie werden von Anwendern wegen ihrer Transparenz häufig bevorzugt akzeptiert.

Es gibt verschiedene Typen von Fuzzy-Systemen, die sich im Hinblick auf ihre Interpretierbarkeit und ihre Genauigkeit unterscheiden. Grundsätzlich sind beide Eigenschaften wünschenswert, bei vielen bekannten Verfahren muss jedoch einer der Vorteile zugunsten des anderen aufgegeben werden.

---

<sup>1</sup>Regelorientierte statistische Analyse

Zur automatischen datenbasierten Generierung von Mamdani-Systemen kann z. B. das Fuzzy-ROSA-Verfahren [5] eingesetzt werden. Dieses Verfahren basiert auf der statistischen Relevanzanalyse von Hypothesen zur Generierung von Mamdani-Regeln, die für sich betrachtet jeweils einen relevanten Teilaspekt des zu modellierenden Prozesses beschreiben. Auf diese Weise wird eine sehr kompakte Regelbasis erzeugt. Durch eine nachgeschaltete Transformation des Mamdani-Systems in ein TSK-System kann die Modellgenauigkeit oftmals noch deutlich verbessert werden.

Der grundlegende Ansatz des Fuzzy-ROSA-Verfahrens wird in Abschnitt 2 zusammengefasst. Eine Besonderheit des Fuzzy-ROSA-Verfahrens ist die Verwendung negativer Regeln, mit denen sich Warnungen bzw. Verbote auf kompakte Weise ausdrücken lassen. Für die Verwendung negativer Regeln in Mamdani-Systemen ist ein zweisträngiges Konzept entwickelt worden, das in Abschnitt 2.2 erläutert wird. Darauf aufbauend wird ein Ansatz vorgestellt, der die Anwendung negativer Regeln auch in TSK-Systemen ermöglicht. In Abschnitt 4 werden exemplarisch Ergebnisse aufgeführt, die mit dem hier vorgestellten Ansatz erzielt worden sind.

Die vorgestellten Arbeiten werden in enger Kooperation zwischen der Universität Dortmund und der NuTech Solutions GmbH im Rahmen eines von der Deutschen Forschungsgemeinschaft geförderten Transferbereiches durchgeführt.

## 2 Das Fuzzy-ROSA-Verfahren

Die zwei gängigsten Typen von Fuzzy-Systemen mit den Eingangsgrößen  $x_1, \dots, x_n$  und der Ausgangsgröße  $u$  sind das Mamdani-System [11] und das TSK-System [15]. Ein Mamdani-System verwendet Regeln folgender Art:

$$\text{WENN } \langle P \rangle \text{ DANN } \langle C \rangle \quad . \quad (1)$$

Dabei ist  $P$  eine vom Eingangsvektor  $\mathbf{x} = (x_1, \dots, x_n)$  abhängige Prämisse und  $C$  eine von der Ausgangsgröße  $u$  abhängige Konklusion der Form  $u = \mathcal{S}_{u,i}$ , wobei  $\mathcal{S}_{u,i}$  ein linguistischer Wert ist. Im Gegensatz dazu haben die Regeln in einem TSK-System die Form:

$$\text{WENN } \langle P \rangle \text{ DANN } \langle f(\mathbf{x}) \rangle \quad , \quad (2)$$

d. h. 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 Eingangsvektor  $\mathbf{x}$  (im Folgenden auch als Eingangssituation bezeichnet) wird bei einem Mamdani-System zunächst die aussgangsseitige 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  $C_i$  spezifizierten linguistischen Wert der Ausgangsgröße. Die ausgangsseitige Zugehörigkeitsfunktion wird dann mit einem wählbaren Verfahren – z. B. mit der Schwerpunkt-Methode – defuzzifiziert. Bei einem TSK-System erfolgt die Berechnung des scharfen Ausgangsgrößenwertes  $u_D$  meist durch eine gewichtete Mittelwertbildung über die Empfehlungen aller durch die Eingangssituation  $\mathbf{x}$  aktivierten Regeln:

$$u_D(\mathbf{x}) = \frac{\sum_{i=1}^R \mu_{P_i}(\mathbf{x}) \cdot f_i(\mathbf{x})}{\sum_{i=1}^R \mu_{P_i}(\mathbf{x})} \quad . \quad (4)$$

## 2.1 Datenbasierte Generierung von relevanten 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 Regel 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. Details zu den unterschiedlichen Test- und Bewertungsstrategien finden sich in [4, 7, 10, 13, 14].

Jeder so generierten Mamdani-Regel wird im Fuzzy-ROSA-Verfahren durch die Bewertung ein Glaubensgrad  $\rho \in [0, 1]$  zugeordnet, der angibt, wie gut diese Regel durch die zugrunde liegenden Lerndaten statistisch abgesichert ist. Dieser Glaubensgrad kann in einem Mamdani-System bei der Inferenz entsprechend berücksichtigt werden:

$$\mu(\mathbf{x}, u) = \bigvee_{i=1}^R [\rho_i \wedge \mu_{P_i}(\mathbf{x}) \wedge \mu_{U_{C_i}}(u)] \quad . \quad (5)$$

Bei einem TSK-System wird dieser Glaubensgrad wie folgt beachtet:

$$u_{D_{TSK}}(\mathbf{x}) = \frac{\sum_{i=1}^R \rho_i \cdot \mu_{P_i}(\mathbf{x}) \cdot f_i(\mathbf{x})}{\sum_{i=1}^R \rho_i \cdot \mu_{P_i}(\mathbf{x})} \quad . \quad (6)$$

## 2.2 Negative Regeln: Hyperinferenz und Hyperdefuzzifizierung

Eine übliche Mamdani-Regel gemäß (1) spricht zu einer gegebenen Eingangssituation  $\mathbf{x}$  eine bestimmte linguistische *Empfehlung* für die Ausgangsgröße  $u$  aus. Daher werden Mamdani-Regeln dieses Typs auch als *positive Regeln* bezeichnet. Menschliches Wissen liegt aber oftmals nicht nur in Form von Empfehlungen, sondern auch in Form von *Warnungen* bzw. *Verboten* vor. Solches Wissen lässt sich ggf. zwar auch durch positive Regeln formulieren, die Repräsentation ist dann aber vergleichsweise umfangreich und nur bedingt nachvollziehbar. Um Wissen dieser Art in kompakter und transparenter Weise darstellen zu können, wurden von KIENDL die *negativen Regeln* gemäß (7) eingeführt:

$$\text{WENN } \langle P \rangle \text{ DANN NICHT } \langle C \rangle \quad . \quad (7)$$

Dabei wurde für die Verarbeitung von positiven und negativen Regeln im Fuzzy-ROSA-Verfahren ein zweisträngiges Konzept [6] entwickelt, das im Folgenden kurz skizziert wird. Die Regelbasis, die sowohl positive als auch negative Regeln enthält, wird zunächst in zwei Teil-Regelbasen geteilt, die dann jeweils nur positive bzw. nur negative Regeln enthalten. Für beide Teil-Regelbasen (Stränge) wird dann jeweils die Inferenz gemäß Gleichung (5) durchgeführt. Der Vorteil dieser Vorgehensweise liegt u. a. darin, dass für die beiden Stränge ggf. unterschiedliche Operatoren für die UND- bzw. ODER-Verknüpfungen gewählt werden können. Es ergeben sich so zwei ausgangsseitige Zugehörigkeitsfunktionen:  $\mu^+(u)$  für die Empfehlungen der positiven Regeln, und  $\mu^-(u)$  für die Warnungen der negativen Regeln<sup>2</sup>. Diese beiden ausgangsseitigen Zugehörigkeitsfunktionen werden durch die Hyperinferenz zu dem resultierenden Empfehlungsgebirge  $\mu(u)$  vereinigt. Im Fuzzy-ROSA-Verfahren sind drei unterschiedliche Strategien für diese Vereinigung realisiert worden. Bei Verwendung der *Strong-Veto-Hyperinferenz* wird die ausgangsseitige Zugehörigkeitsfunktion  $\mu^-(u)$  der negativen Regeln unmittelbar als Verbot interpretiert:

$$\mu(u) = \begin{cases} 0, & \text{falls } \mu^-(u) > 0, \\ \mu^+(u) & \text{sonst} \end{cases} . \quad (8)$$

Im Gegensatz dazu wird bei der *Weak-Veto-Hyperinferenz* die ausgangsseitige Zugehörigkeitsfunktion  $\mu^-(u)$  der negativen Regeln erst dann als Verbot interpretiert, wenn der Grad der Warnung gleich oder größer dem Grad der Empfehlung ist:

$$\mu(u) = \begin{cases} 0, & \text{falls } \mu^-(u) \geq \mu^+(u), \\ \mu^+(u) & \text{sonst} \end{cases} . \quad (9)$$

Die *Fuzzy-Veto-Hyperinferenz* bildet einen Kompromiss zwischen den beiden zuvor genannten Strategien:

$$\mu(u) = \mu^+(u) \wedge \neg\mu^-(u) . \quad (10)$$

Abbildung 1 zeigt beispielhaft die unterschiedlichen Resultate bei dem Einsatz verschiedener Hyperinferenzstrategien. Von links nach rechts sind dargestellt:  $\mu^+(u)$ ,  $\mu^-(u)$ ,  $\mu(u)$  für die Strong-Veto-Hyperinferenz sowie  $\mu(u)$  für die Weak-Veto-Hyperinferenz.

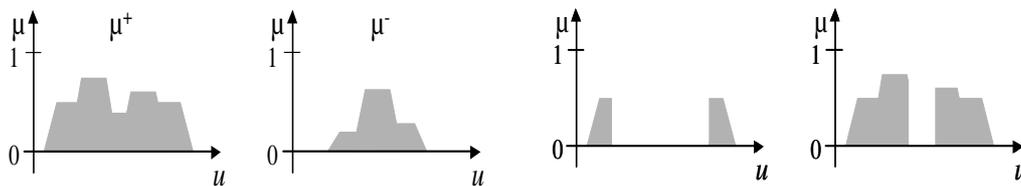


Abbildung 1: Beispiel für Hyperinferenzstrategien

Die Aufgabe der Hyperdefuzzifizierung besteht darin, aus dem resultierenden Empfehlungsgebirge  $\mu(u)$  einen sinnvollen Ausgangsgrößenwert  $u_D$  zu erzeugen, für den

<sup>2</sup>Es wird hier statt  $\mu(\mathbf{x}, u)$  verkürzt  $\mu(u)$  geschrieben.

in keinem Falle  $\mu(u_D) = 0$  gelten sollte. Dabei ist zu bedenken, dass die Hyperinferenz für einen Ausgangsgrößenwert  $u_i$  aus qualitativ unterschiedlichen Gründen zu  $\mu(u_i) = 0$  führen kann. Erstens kann  $\mu^+(u_i) = 0$  und  $\mu^-(u_i) = 0$  gelten. Der Ausgangsgrößenwert  $u_i$  wird dann weder empfohlen, noch verboten. Zweitens kann  $\mu^+(u_i) > 0$  und  $\mu^-(u_i) > 0$  gelten. Der Ausgangsgrößenwert  $u_i$  wird dann sowohl empfohlen, als auch verboten. Drittens kann  $\mu^+(u_i) = 0$  und  $\mu^-(u_i) > 0$  gelten. Der Ausgangsgrößenwert  $u_i$  wird dann nicht empfohlen, sondern ausschließlich verboten.

Für die Bestimmung des scharfen Ausgangsgrößenwertes  $u_D$  des Mamdani-Systems sind verschiedene Strategien entwickelt worden. Für die nachfolgenden Abschnitte wird jedoch nur eine dieser Strategien genutzt, die hier kurz skizziert wird: Das resultierende Empfehlungsgebirge  $\mu(u)$  wird dabei zunächst in Teil-Empfehlungsgebirge  $\mu_i(u)$  zerlegt. Diese sind jeweils in den Intervallen erklärt, in denen  $\mu(u) > 0$  ist. Jedem Teil-Empfehlungsgebirge  $\mu_i(u)$  wird ein Gewicht  $g_i$  zugeordnet. Das Gewicht  $g_i$  kann dabei beispielsweise die eingeschlossene Fläche unter  $\mu_i(u)$  berücksichtigen oder ein anderes charakteristisches Merkmal. Das Teil-Empfehlungsgebirge  $\mu_j(u)$  mit dem größten Gewicht  $g_j$  wird dann ausgewählt und auf herkömmliche Weise defuzzifiziert. Abbildung 2 greift das Beispiel aus Abbildung 1 auf. Aus dem resultierenden Empfehlungsgebirge  $\mu(u)$ , das sich durch die Weak-Veto-Hyperinferenz ergibt (siehe Abbildung 1, rechts), ist das Teil-Empfehlungsgebirge  $\mu_1(u)$  isoliert worden, das die größte Fläche einschliesst. Dieses Teil-Empfehlungsgebirge kann nun z. B. nach der *Mean of Maximum*-Methode (Abbildung 2, links) oder nach der Schwerpunkt-Methode (Abbildung 2, rechts) defuzzifiziert werden. Da das Gewicht  $g_i$  hier die eingeschlossene Fläche unter  $\mu_i(u)$  beschreibt, entspricht letztere Strategie im Prinzip der Wirkungsweise der auch für herkömmliche Mamdani-Systeme bekannten *Center of Largest Area*-Defuzzifizierung.

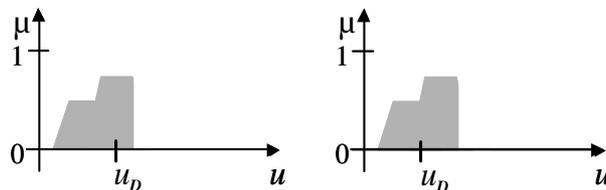


Abbildung 2: Beispiel für Hyperdefuzzifizierungsstrategien

### 2.3 Transformation eines Mamdani-Systems in ein TSK-System

Ein mit dem Fuzzy-ROSA-Verfahren generiertes Mamdani-System kann nachgelagert in ein korrespondierendes TSK-System transformiert werden [8, 9, 14]. Ziel dieser Transformation ist es, die Genauigkeit des letztlich verwendeten Fuzzy-Modells zu erhöhen.

### 2.3.1 Gängiger Ansatz für die Konklusion einer TSK-Regel

Für die Konklusion  $f(\mathbf{x})$  einer TSK-Regel können prinzipiell beliebig komplexe Funktionsansätze verwendet werden. Dabei ist aber zu bedenken, dass mit zunehmender Komplexität zum einen die Gefahr eines Overfitting anwächst, und zum anderen der Berechnungsaufwand steigt. Im Fuzzy-ROSA-Verfahren wird daher folgender allgemeiner Funktionsansatz:

$$f(\mathbf{x}) = \sum_{j=0}^m a_j \cdot g_j(\mathbf{x}) \quad (11)$$

verwendet. Dass heißt,  $f(\mathbf{x})$  wird als Linearkombination verschiedener Elementarfunktionen  $g_j(\mathbf{x})$  angesetzt, die sowohl lineare als auch nichtlineare Funktionen der Eingangsgrößen sein können. Die hier gewählte Funktion  $f(\mathbf{x})$  ist also linear in den Parametern  $a_j$ . Dies erlaubt es, optimale Werte – z. B. im Sinne der Minimierung der Summe der quadratischen Fehlerwerte – für diese Parameter mit vergleichsweise geringem Aufwand zu bestimmen. Bei einer entsprechenden Wahl der  $g_j(\mathbf{x})$  als lineare Funktionen lässt sich auch der im Fuzzy-ROSA-Verfahren verwendete Standard-Funktionsansatz realisieren:

$$f(\mathbf{x}) = a_0 + \sum_{j=1}^n a_j \cdot x_j \quad . \quad (12)$$

### 2.3.2 Transformation einer positiven Mamdani-Regel in eine TSK-Regel

Nach Wahl von  $f(\mathbf{x})$  z. B. gemäß Gleichung (12) müssen für jede Mamdani-Regel die freien Parameter in  $f(\mathbf{x})$  bestimmt werden. Dies erfolgt dabei auf Grundlage derjenigen Datenpunkte des zugrunde liegenden Lerndatensatzes, die die zu transformierende Mamdani-Regel stützen.

Der zugrunde liegende Lerndatensatz  $\mathcal{D}_L = \{\mathbf{d}_1, \dots, \mathbf{d}_D\}$  bestehe aus  $D$  Datenpunkten  $\mathbf{d}_i$ , wobei ein Datenpunkt durch  $\mathbf{d}_i = (x_{i,1}, \dots, x_{i,n}, u_{i,1})$  beschrieben wird. Von den  $D$  Datenpunkten erfüllen  $D_P$  Datenpunkte die Prämisse  $P$  der zu transformierenden Regel jeweils im Grade  $\mu_P(\mathbf{x}) > 0$ . In vielen Fällen werden jedoch nicht alle  $D_P$  Datenpunkte auch die Konklusion  $C$  der Regel im Grade  $\mu_{U_C}(u) > 0$  aktivieren. Für die Anzahl der Datenpunkte, die sowohl die Prämisse der Regel erfüllen als auch die spezifizierte Konklusion aktivieren, gilt somit  $D_{P \wedge C} \leq D_P$ . Bei der Transformation einer Mamdani-Regel in eine TSK-Regel sollten i. d. R. nur die  $D_{P \wedge C}$  Datenpunkte für die Bestimmung der freien Parameter in  $f(\mathbf{x})$  verwendet werden, damit die Gefahr einer ungewollten Verschiebung der resultierenden Funktion aus dem Wertebereich der Konklusion  $C$  möglichst gering ist [8]. Man geht davon aus, dass gilt:

$$\mu_P(\mathbf{x}) > 0 \Rightarrow \mu_{U_C}(f(\mathbf{x})) > 0 \quad . \quad (13)$$

Mit  $\mathbf{g}^T = (1, x_1, \dots, x_n)$  und  $\mathbf{a}^T = (a_0, \dots, a_n)$  kann in vektorieller Form  $f(\mathbf{x}) = \mathbf{g}^T \mathbf{a}$  geschrieben werden. Für einen gegebenen Datenpunkt  $\mathbf{d}_j$  wird der für diese Eingangssituation gemessene Ausgangsgrößenwert  $u_j$  durch  $f(\mathbf{x}_j) = \hat{u}_j$  approximiert, wobei in möglichst guter Näherung  $\hat{u}_j = u_j$  gelten soll. Aus den zuvor ausgewählten

Datenpunkten, die die zu transformierende Regel stützen, kann so ein lineares Gleichungssystem aufgebaut werden, das das zu lösende Problem für die Transformation einer Regel beschreibt:

$$\mathbf{G} \cdot \mathbf{a} = \mathbf{u} \quad . \quad (14)$$

Das lineare Gleichungssystem (14) kann dabei unterbestimmt oder eindeutig lösbar oder überbestimmt sein. Im Falle eines unterbestimmten Gleichungssystems kann z. B. auf eine einfach zu berechnende Mittelwert-Prognose zurückgegriffen werden. Dabei werden  $a_0 = \bar{u}_P$  und  $a_1 = \dots = a_n = 0$  gesetzt, wobei  $\bar{u}_P$  der mittlere Ausgangsgrößenwert der regelstützenden Datenpunkte ist. Ein aufwendigerer Ansatz besteht darin, eine Lösung  $\mathbf{a}$  zu berechnen, die  $k = \text{rang}(\mathbf{G})$  von Null verschiedene Komponenten aufweist [2, 14]. Im Falle eines überbestimmten Gleichungssystems kann z. B. ein Least-Squares-Algorithmus zur Bestimmung des Parametersatzes  $\mathbf{a}$  eingesetzt werden. Dieser Ansatz ist jedoch vergleichsweise anfällig gegenüber Ausreißern. In dem hier betrachteten Anwendungsfall der Regeltransformation ist der Begriff Ausreißer in erster Linie situationsbezogen zu verstehen. Bei der Transformation einer Regel sind immer nur die zuvor ausgewählten  $D_P$  bzw.  $D_{P \wedge C}$  Datenpunkte des zugrunde liegenden Gesamt-Lerndatensatzes  $\mathcal{D}_L$  relevant. Auch wenn der Gesamt-Lerndatensatz keine Ausreißer bzgl. der Ausgangsgrößenwerte im üblichen Sinne aufweist, so kann dies auf den für eine Regel extrahierten Teil-Lerndatensatz  $\mathcal{D}_P$  bzw.  $\mathcal{D}_{P \wedge C}$  durchaus zutreffen, nämlich dann, wenn die extrahierten Ausgangsgrößenwerte stark ungleichmäßig verteilt sind. Es ist daher oftmals von Vorteil, aufwendigere Verfahren für die Bestimmung der Parameter einzusetzen, die robuster gegenüber solchen Ausreißern sind. In [14] wurde gezeigt, dass beispielsweise ein Ansatz basierend auf einem *Iteratively Reweighted Least-Squares-Algorithmus* [1, 3] hierfür geeignet ist.

### 3 Negative Regeln in TSK-Systemen

#### 3.1 Problem der Transformation einer negativen Mamdani-Regel

Wir suchen nun nach Möglichkeiten zur Übertragung der oben beschriebenen Transformation von positiven Mamdani-Regeln in TSK-Regeln auch auf negative Mamdani-Regeln. Die Aufgabe negativer Regeln – unabhängig welchen Typs – besteht darin, bei einer Eingangssituation  $\mathbf{x}$  vor bestimmten Ausgangsgrößenwerten zu warnen bzw. diese zu verbieten. Im Folgenden wird vereinfachend angenommen, dass eine negative Regel stets als Verbot zu verstehen ist.

Es werde angenommen, dass eine negative Mamdani-Regel, die den Ausgangsgrößenwert  $u_\lambda$  verbietet, in eine ebenfalls den Ausgangsgrößenwert  $u_\lambda$  verbietende und deshalb als negativ bezeichnete TSK-Regel der Form (2) transformiert werden kann, wobei die Berechnung des Ausgangsgrößenwertes  $u_{D_{TSK}}(\mathbf{x})$  gemäß Gleichung (6) erfolgt. Bei einer TSK-Regelbasis mit mehreren positiven Regeln und einer einzigen negativen Regel würde dies bedeuten, dass diese eine negative Regel einen solchen Beitrag zu  $u_{D_{TSK}}$  liefern müsste, dass sich  $u_{D_{TSK}} \neq u_\lambda$  einstellt. Um dies zu erreichen, müsste bekannt sein, welche positiven Regeln durch die Eingangssituation  $\mathbf{x}$  aktiviert werden. Erst dann könnte die Konklusion der negativen Regel entsprechend

entworfen werden. Weitergehend würde sich dann die Frage stellen, ob die Konklusion der negativen Regel so zu wählen ist, dass  $u_{D_{TSK}} < u_\lambda$  oder  $u_{D_{TSK}} > u_\lambda$  gilt.

Nimmt man den Fall an, dass zu einer Eingangssituation  $\mathbf{x}$  neben mehreren positiven TSK-Regeln auch zwei negative TSK-Regeln aktiviert werden, so müsste der oben beschriebene Beitrag zu  $u_{D_{TSK}}$  im Prinzip auf die beiden negativen TSK-Regeln aufgeteilt werden. Es würde sich so eine Vielzahl von Entwurfsfreiheiten ergeben, was die Transparenz des Fuzzy-Modells stark beeinträchtigen würde.

Die in Abschnitt 2.3.2 beschriebene Transformation für positive Mamdani-Regeln lässt sich daher nicht unmittelbar auf negative Mamdani-Regeln übertragen. Im Folgenden wird jedoch eine Vorgehensweise aufgezeigt, mit der negative Mamdani-Regeln auch in TSK-Systemen gewinnbringend genutzt werden können. Diese Vorgehensweise zielt nicht darauf ab, eine Transformation der negativen Mamdani-Regeln in negative TSK-Regeln durchzuführen, sondern verfolgt vielmehr den Ansatz, die negativen Mamdani-Regeln dazu zu verwenden, eine situationsbezogene Selektion geeigneter positiver TSK-Regeln vorzunehmen.

### 3.2 Nutzung negativer Mamdani-Regeln in TSK-Systemen

Zur Nutzung negativer Mamdani-Regeln in TSK-Systemen wird folgende Strategie vorgeschlagen. Es wird ein hybrides Fuzzy-System mit zwei Regelbasen verwendet. Die erste Regelbasis enthält die positiven und negativen Regeln des Mamdani-Systems, das transformiert werden soll. Die zweite Regelbasis beinhaltet die aus den positiven Mamdani-Regeln transformierten TSK-Regeln. Für eine gegebene Eingangssituation  $\mathbf{x}$  wird zunächst die Mamdani-Regelbasis ausgewertet. Nach Durchführung der Hyperinferenz ergibt sich so das resultierende Empfehlungsgebirge  $\mu(u)$ . Aus diesem wird ein Intervall  $[u_{min}, u_{max}]$  bestimmt, das den prädestinierten (d. h. zulässigen im Sinne der Verbote, und am stärksten empfohlenen im Sinne der Empfehlungen) Wertebereich für den scharfen Ausgangsgrößenwert  $u_D$  des Fuzzy-Systems bezogen auf die gegebene Eingangssituation  $\mathbf{x}$  festlegt. Für die Berechnung des Ausgangsgrößenwertes  $u_D$  des Fuzzy-Systems werden jedoch nicht die Mamdani-Regeln, sondern die TSK-Regeln verwendet. Aus denjenigen TSK-Regeln, die durch die Eingangssituation  $\mathbf{x}$  aktiviert werden, wird anhand des oben genannten Intervalls jedoch ggf. noch eine situationsbezogene Regelselektion vorgenommen. Nur die so selektierten TSK-Regeln werden letztlich zur Bestimmung des Ausgangsgrößenwertes  $u_D$  verwendet.

Im Einzelnen wird in diesem hybriden Fuzzy-System für eine gegebene Eingangssituation  $\mathbf{x}$  zunächst die Mamdani-Regelbasis ausgewertet. Es ergibt sich nach der in Abschnitt 2.2 beschriebenen Hyperinferenz das resultierende Empfehlungsgebirge  $\mu(u)$ . Dieses kann aus mehreren Teil-Empfehlungsgebirgen  $\mu_i(u)$  bestehen, die jeweils in den Intervallen  $[u_{i,min}, u_{i,max}]$  erklärt sind, in denen  $\mu(u) > 0$  ist. Es wird das Teil-Empfehlungsgebirge  $\mu_j(u)$  mit dem größten Gewicht  $g_j$  bestimmt, wobei das Gewicht (wie in Abschnitt 2.3.2 beschrieben) verschiedene Merkmale eines Teil-Empfehlungsgebirges berücksichtigen kann. Würde das so bestimmte Teil-Empfehlungsgebirge  $\mu_j(u)$  defuzzifiziert werden (z. B. nach der Schwerpunkt- oder der Maximum-Methode), so läge der scharfe Ausgangsgrößenwert  $u_{D_{MAM}}$  des Mam-

dani-Systems dann in jedem Fall innerhalb des zugehörigen Intervalls  $[u_{j,min}, u_{j,max}]$ .

Die beiden linken Grafiken in Abbildung 3 zeigen diese Teilschritte nochmals anhand des Beispiels, das bereits in Abschnitt 2.2 behandelt worden ist. Die Auswertung der TSK-Regelbasis, die aus positiven Mamdani-Regeln hervorgegangen ist, liefert einen Ausgangsgrößenwert  $u_{D_{TSK}}$  gemäß Gleichung (6). In den beiden rechten Grafiken in Abbildung 3 sind die verschiedenen Empfehlungen der TSK-Regeln für dieses Beispiel in Form von Singletons dargestellt. Werden alle aktivierten TSK-Regeln für die Berechnung herangezogen, so ergibt sich ein Wert für  $u_{D_{TSK}}$ , der in der zweiten Grafik von rechts in Abbildung 3 skizziert ist. Dieser liegt deutlich außerhalb des Intervalls  $[u_{1,min}, u_{1,max}]$ , das sich durch die Hyperinferenz in dem Mamdani-System ergibt. Folglich weicht dieser Ausgangsgrößenwert  $u_{D_{TSK}}$  auch sehr stark von dem Ausgangsgrößenwert  $u_{D_{MAM}}$  des Mamdani-Systems ab. In diesem Beispiel tritt sogar der Fall ein, dass  $\mu(u_{D_{TSK}}) = 0$  ist, also der so bestimmte Ausgangsgrößenwert des TSK-Systems von dem zugrunde liegenden Mamdani-System als verboten eingestuft wird.

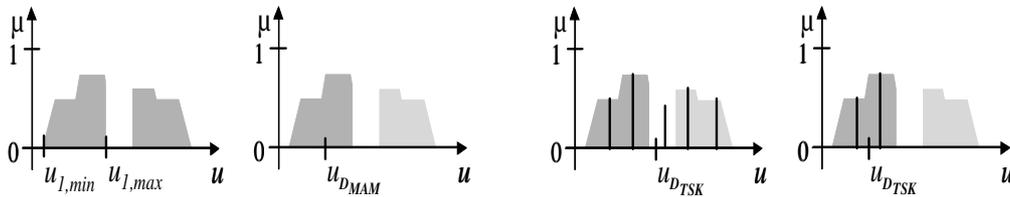


Abbildung 3: Beispiel für die Anwendung negativer Regeln in einem TSK-System

Um dies zu verhindern, werden nicht alle TSK-Regeln bei der Auswertung berücksichtigt, sondern nur diejenigen, für die  $u_{j,min} \leq f_i(\mathbf{x}) \leq u_{j,max}$  gilt. So wird gewährleistet, dass der Ausgangsgrößenwert  $u_{D_{TSK}}$  des TSK-Systems innerhalb des gewünschten Intervalls liegt, wie die rechte Grafik in Abbildung 3 für das obige Beispiel zeigt. Da dieser Ausgangsgrößenwert gleichzeitig auch der des hybriden Fuzzy-Systems ist, also  $u_D = u_{D_{TSK}}$  ist, erfüllt somit auch das hybride Fuzzy-System diese Anforderung an den Ausgangsgrößenwert.

### 3.3 Anwendung im Falle nur positiver Regeln

Der in Abschnitt 3.2 vorgestellte Ansatz bietet auch dann Vorteile, wenn ausschließlich positive Mamdani-Regeln bzw. deren korrespondierende TSK-Regeln vorhanden sind. Dabei sind zwei Szenarien voneinander zu unterscheiden.

Das erste Szenarium betrifft die Generierung eines u. U. ungeeigneten Ausgangsgrößenwertes  $u_D$  durch Überlagerung einzelner plausibler Empfehlungen für den Ausgangsgrößenwert. Wenn nur positive Mamdani-Regeln vorhanden sind, liefert die Hyperinferenz das resultierende Empfehlungsgebirge  $\mu(u) = \mu^+(u)$ . Dieses kann durchaus aus mehreren Teil-Empfehlungsgebirgen bestehen. Folglich gibt es in einem solchen Fall dann auch Ausgangsgrößenwerte, für die  $\mu(u_i) = 0$  gilt. Im Gegensatz

zu dem Fall des Vorliegens positiver und negativer Regeln kann dies in dem Fall nur positiver Regeln aber nicht bedeuten, dass Verbote verletzt werden, sondern nur, dass diese Ausgangsgrößenwerte von keiner einzigen Mamdani-Regel empfohlen werden. Um zu vermeiden, dass  $\mu(u_{D_{MAM}}) = 0$  ist, kann z. B. die *Center Of Largest Area*-Defuzzifizierung eingesetzt werden. Für die korrespondierenden TSK-Regeln gelten ebenfalls die Überlegungen aus Abschnitt 3.2. Durch die gewichtete Mittelwertbildung über einzelne plausible Empfehlungen mit  $\mu(f_i(\mathbf{x})) > 0$  kann sich ein Ausgangsgrößenwert  $u_{D_{TSK}}$  mit  $\mu(u_{D_{TSK}}) = 0$  ergeben. Es kann daher auch bei nur positiven Regeln sinnvoll sein, eine geeignete Auswahl der TSK-Regeln in dem oben beschriebenen Sinne vorzunehmen.

Das zweite Szenarium betrifft die Generierung eines u. U. ungeeigneten Ausgangsgrößenwertes  $u_D$  durch den Einfluss einzelner unplausibler Empfehlungen für den Ausgangsgrößenwert. In Abschnitt 2.3.2 wird beschrieben, wie die Transformation einer Mamdani-Regel in eine TSK-Regel vollzogen wird. Ziel dieser Transformation ist es, die Modellgenauigkeit zu erhöhen und dabei die Interpretierbarkeit der Regel zu erhalten. Damit die Konklusion der zugrunde liegenden Mamdani-Regel und die Konklusion der so generierten TSK-Regel ineinander überführbar bleiben, sollte die Konklusion der TSK-Regel möglichst nur solche Empfehlungen für den Ausgangsgrößenwert liefern, die im Wertebereich des linguistischen Wertes  $\mathcal{S}_{u,i}$  liegen, auf den sich die Elementaraussage in der Konklusion der Mamdani-Regel bezieht. Diese Überlegung motiviert die Auswahl der Datenpunkte, die für die Bestimmung der freien Parameter in der Konklusion der TSK-Regel herangezogen werden. Wie in Abschnitt 2.3.2 beschrieben, sollten hierfür i. d. R. nur die  $D_{P \wedge C}$  Datenpunkte des zugrunde liegenden Lerndatensatzes  $\mathcal{D}_L$  verwendet werden, die sowohl die Prämisse der Mamdani-Regel erfüllen, als auch deren Konklusion aktivieren. Im Fuzzy-ROSA-Verfahren kann die Zugehörigkeitsfunktion für einen linguistischen Wert rechteckförmig, trapezförmig oder dreieckförmig sein. Eine Zugehörigkeitsfunktion lässt sich in diesen Fällen durch vier Stützstellen  $s_i$  beschreiben, für die  $s_1 \leq s_2 \leq s_3 \leq s_4$  gilt. Der Wertebereich des linguistischen Wertes  $\mathcal{S}_{u,i}$  ist somit durch  $[s_{u,i,1}, s_{u,i,4}]$  gegeben.

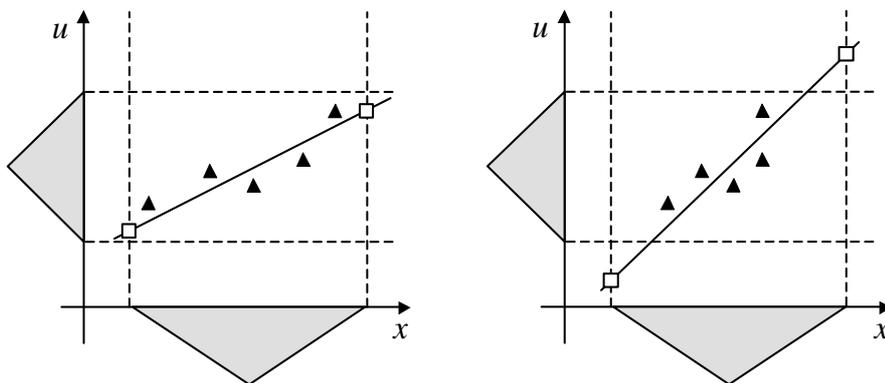


Abbildung 4: Beispiele für ein unkritische Extrapolation (links) und eine eventuell ungünstige Extrapolation (rechts)

Abbildung 4 zeigt zwei unterschiedliche Szenarien für die Transformation einer Regel. Dabei sind die  $D_{P\wedge C}$  regelstützenden Datenpunkte des Lerndatensatzes  $\mathcal{D}_L$  jeweils durch die schwarzen Dreiecke dargestellt. Die dreiecksförmigen Zugehörigkeitsfunktionen für die in diesem Beispiel relevanten linguistischen Werte (sowohl der Eingangs- als auch der Ausgangsgröße) sind in Abbildung 4 grau dargestellt. Die freien Parameter in der Konklusion  $f_i(\mathbf{x})$  der TSK-Regel werden mit einem Least-Squares-Algorithmus bestimmt. In dem Beispiel aus Abbildung 4 mit nur einer Eingangsgröße  $x_1$  ist  $f_i(\mathbf{x}) = a_0 + a_1 \cdot x_1$ . Die resultierende Ausgleichsgerade ist jeweils als durchgezogene schwarze Linie dargestellt. Die regelstützenden Datenpunkte haben in beiden Fällen identische Ausgangsgrößenwerte, ihre Eingangsgrößenwerte unterscheiden sich aber. Folglich unterscheiden sich auch die Ausgleichsgeraden, die sich in den beiden Fällen ergeben. Für die  $D_{P\wedge C}$  Datenpunkte des Lerndatensatzes trifft in diesen Beispielen die Annahme nach (13) zu, es ist also  $\mu_{U_{C_i}}(f_i(\mathbf{x})) > 0$  bzw.  $s_{u,i,1} \leq f_i(\mathbf{x}) \leq s_{u,i,4}$ . Grundsätzlich ist jedoch nicht auszuschließen, dass diese Annahme für einen der  $D_{P\wedge C}$  bzw. der  $D_P$  Datenpunkte im Einzelfall nicht zutrifft. Diese Möglichkeit ist aber vor allem im Hinblick auf die Prognose bei Eingangssituationen zu bedenken, die nicht in den Lerndaten enthalten sind. Dieser Aspekt ist also für den späteren praktischen Einsatz des Fuzzy-Modells von großer Bedeutung. Solche Eingangssituationen – und die zugehörigen prognostizierten Ausgangsgrößenwerte – sind in den Grafiken in Abbildung 4 beispielhaft durch die weißen Quadrate dargestellt. Im ersten Fall liegen die durch  $f_i(\mathbf{x})$  prognostizierten Ausgangsgrößenwerte noch innerhalb des angestrebten Wertebereichs, im zweiten Fall liegen sie außerhalb. Die Empfehlung der TSK-Regel deckt sich also in der rechten Grafik im Prinzip nicht mehr mit der Mamdani-Regel.

Es stellt sich die Frage, wie eine solche Extrapolation zu bewerten, und in der Berechnung des scharfen Ausgangsgrößenwertes  $u_{D_{TSK}}$  zu berücksichtigen ist. Grundsätzlich kann der so empfohlene Ausgangsgrößenwert  $f_i(\mathbf{x})$  unverändert bei der Berechnung von  $u_{D_{TSK}}$  verwendet werden. Es besteht aber dann die Gefahr, dass ein solcher „Ausreißer“ zu einem unplausiblen Wert für  $u_{D_{TSK}}$  führt. Eine andere Möglichkeit kann darin bestehen, dass man jeder generierten TSK-Regel zusätzlich noch entsprechend eine untere und eine obere Grenze zuweist und den von dieser Regel empfohlenen Ausgangsgrößenwert dann ggf. hart auf  $f_i(\mathbf{x}) = s_{u,i,1}$  bzw.  $f_i(\mathbf{x}) = s_{u,i,4}$  begrenzt, falls dieser sonst unter- bzw. oberhalb der entsprechenden Grenze liegen würde. Damit wird die Extrapolation im Prinzip unterdrückt, wodurch aber auch evtl. Vorteile verloren gehen. Außerdem werden so auch solche Empfehlungen erzeugt, für die eigentlich  $f_i(\mathbf{x}) \ll s_{u,i,1}$  bzw.  $f_i(\mathbf{x}) \gg s_{u,i,4}$  gilt, bei denen also die Empfehlung der TSK-Regel und die der zugrunde liegenden Mamdani-Regel „überhaupt nicht mehr zueinander passen“. Eine Alternative besteht darin, für das TSK-System eine Regelselektion anhand des resultierenden Empfehlungsgebirges  $\mu(u)$  des Mamdani-Systems in der oben beschriebenen Form durchzuführen. Bei dieser Vorgehensweise müssen keine Grenzen zu den TSK-Regeln gespeichert werden. Die oben beschriebene Extrapolation wird so zwar zugelassen, aber nur in dem Maße, wie es durch die Gesamtempfehlung aller aktivierten Mamdani-Regeln angemessen erscheint.

## 4 Anwendungen

Eine systematische Untersuchung der Leistungsfähigkeit des hier vorgestellten Ansatzes an verschiedenen Benchmark-Problemen ist Gegenstand aktueller Arbeiten. Nachfolgend werden erste Ergebnisse dieser Untersuchung für ein aus der Literatur bekanntes Modellierungsproblem [12] präsentiert.

### 4.1 Mackey-Glass Zeitreihe

Die Aufgabe besteht darin, auf der Basis von vier Messwerten  $x(t - 18)$ ,  $x(t - 12)$ ,  $x(t - 6)$  und  $x(t)$  eine Prognose für den zukünftigen Messwert  $x(t + 6)$  zu treffen. Der zugrunde liegende Datensatz besteht aus 1000 Datenpunkten, die zufällig in zwei gleich große Teildatensätze getrennt worden sind. Mit Hilfe dieser zwei Teildatensätze ist eine Kreuzvalidierung durchgeführt worden. Die Ergebnisse in den nachfolgenden Tabellen geben die gemittelten Werte für die Anzahl der Regeln und den mittleren absoluten Fehler (MAE) auf den Lerndaten (LD) bzw. den Validierungsdaten (VD) an. Die zulässige Kombinationstiefe [13] in der Prämisse einer Regel ist zu  $c_{max} = 2$  gewählt worden. Für jede Variable sind die Zugehörigkeitsfunktionen jeweils automatisch durch ein unidimensionales Clusterverfahren ermittelt worden. In diesem Fall sind dabei für jede Variable vier trapezförmige Zugehörigkeitsfunktionen erstellt worden.

Modelltyp	Regeltyp	Regeltest	$\rho_{min}$	#Regeln	MAE <sub>LD</sub>	MAE <sub>VD</sub>
Mamdani	Pos	$J_{CHR}$	0.0	13	0.194	0.193
			0.1	10	0.197	0.196
			0.2	8	0.185	0.189
			0.3	5	0.191	0.192
TSK	Pos	$J_{CHR}$	0.0	13	0.083	0.084
			0.1	10	0.087	0.087
			0.2	8	0.105	0.108
			0.3	5	0.132	0.133

Tabelle 1: Ergebnisse für Mackey-Glass ( $J_{CHR}$ , nur positive Regeln)

Als Test- und Bewertungsverfahren wird die *Konfidente Trefferquote* ( $J_{CHR}$ ) [13] verwendet. In Tabelle 1 sind die Modellierungsergebnisse für den Fall angegeben, dass ausschließlich positive Mamdani-Regeln generiert werden dürfen. Zunächst werden Mamdani-Modelle generiert, wobei die Relevanzschwelle  $\rho_{min}$ , ab der eine Regel tatsächlich in die Regelbasis übernommen wird, schrittweise erhöht wird. Es werden relativ wenig Regeln generiert und die Modellierung mit den Mamdani-Modellen gelingt vergleichsweise schlecht. Die so generierten Mamdani-Modelle werden dann in korrespondierende TSK-Modelle transformiert, wodurch die Qualität der Modellierung verbessert wird. In Tabelle 2 sind die Modellierungsergebnisse für den Fall angegeben, dass sowohl positive als auch negative Mamdani-Regeln generiert werden dürfen. Es fällt auf, dass deutlich mehr Regeln generiert werden, der Anteil negativer Regeln ist also entsprechend groß. Gegenüber den Mamdani-Modellen mit nur positiven Regeln wird eine deutliche Verbesserung erzielt. Ausgehend von diesen

Mamdani-Modellen werden mit dem in Abschnitt 3.2 vorgestellten Ansatz hybride Fuzzy-Modelle erstellt, die hier mit  $\text{TSK}_{\text{Hybrid}}$  bezeichnet werden. Auch hier tritt eine deutliche Verbesserung gegenüber den Mamdani-Modellen auf. Verglichen mit den Ergebnissen der TSK-Modelle aus Tabelle 1 liefern die hybriden Fuzzy-Modelle in diesem Fall eine wesentlich bessere Modellierung. Es fällt insbesondere auf, dass es hier von Vorteil ist, für die Relevanzschwelle  $0.1 \leq \rho_{\min} \leq 0.2$  zu wählen. Auf diese Weise werden Regeln (positive wie auch negative) ausselektiert, deren Relevanz eher fragwürdig ist. Solche Regeln können einen negativen Einfluss auf die Qualität der Modellierung haben und es ist oftmals von Vorteil, diese nicht zu beachten. Werden jedoch zu viele Regeln ausselektiert, so kann auch dies nachteilig sein.

Modelltyp	Regeltyp	Regeltest	$\rho_{\min}$	#Regeln	$\text{MAE}_{\text{LD}}$	$\text{MAE}_{\text{VD}}$
Mamdani	Pos&Neg	$J_{\text{CHR}}$	0.0	135	0.136	0.141
			0.1	125	0.134	0.139
			0.2	102	0.128	0.137
			0.3	77	0.169	0.176
$\text{TSK}_{\text{Hybrid}}$	Pos&Neg	$J_{\text{CHR}}$	0.0	135	0.063	0.067
			0.1	125	0.028	0.031
			0.2	102	0.028	0.030
			0.3	77	0.062	0.066

Tabelle 2: Ergebnisse für Mackey-Glass ( $J_{\text{CHR}}$ , positive und negative Regeln)

Wie in [7] ausgeführt, haben die für das Fuzzy-ROSA-Verfahren entwickelten Test- und Bewertungsverfahren im Hinblick auf die Generierung negativer Regeln gewisse Schwachstellen. Diese liegen im Wesentlichen darin, dass die Anforderungen für die Generierung negativer Regeln weniger anspruchsvoll sind als die für die Generierung positiver Regeln. Wie aus den Tabellen 1 und 2 zu ersehen ist, werden ungleich mehr negative als positive Regeln generiert. Um diese Schwachstellen zu überwinden, und eine gleichberechtigte Behandlung positiver und negativer Regeln zu erreichen, ist von KIENDL der *Inkrementelle Relevanzindex* ( $J_{\text{IRI}}$ ) vorgeschlagen worden [7]. Tabelle 3 zeigt die Ergebnisse, die mit diesem Ansatz für das hier untersuchte Beispiel erzielt werden.

Modelltyp	Regeltyp	Regeltest	$\rho_{\min}$	#Regeln	$\text{MAE}_{\text{LD}}$	$\text{MAE}_{\text{VD}}$
Mamdani	Pos&Neg	$J_{\text{IRI}}$	0.0	352	0.116	0.118
			0.1	118	0.110	0.111
			0.2	41	0.067	0.070
			0.3	12	0.110	0.109
$\text{TSK}_{\text{Hybrid}}$	Pos&Neg	$J_{\text{IRI}}$	0.0	352	0.051	0.052
			0.1	118	0.047	0.049
			0.2	41	0.055	0.057
			0.3	12	0.084	0.086

Tabelle 3: Ergebnisse für Mackey-Glass ( $J_{\text{IRI}}$ , positive und negative Regeln)

Bei diesem Test- und Bewertungsansatz werden stets positive und negative Regeln

erzeugt. Bedingt durch die Arbeitsweise dieses Ansatzes werden typischerweise zunächst sehr viele Regeln generiert. Die Anzahl der Regeln kann oftmals jedoch bereits durch eine etwas höhere Relevanzschwelle (z. B.  $\rho_{min} = 0.1$ ) deutlich reduziert werden, ohne dass die Modellierung dadurch signifikant schlechter wird. Dies ist für dieses Beispiel auch der Fall, und die so generierten Mamdani-Modelle schneiden im Vergleich mit denen aus den Tabellen 1 und 2 bzgl. des Modellierungsfehlers am besten ab. Durch den Übergang zu hybriden Fuzzy-Modellen kann der Modellierungsfehler verbessert werden, allerdings sind die Ergebnisse fast alle weniger gut als die der hybriden Fuzzy-Modelle aus Tabelle 2. Im Rahmen einer systematischen Untersuchung ist zu prüfen, ob dieses Beispiel in dieser Hinsicht einen Ausnahmefall darstellt oder ob dies wiederholt auftritt.

## 5 Zusammenfassung und Ausblick

In diesem Beitrag wird ein Ansatz für ein hybrides Fuzzy-System, bestehend aus einem zweisträngigen Mamdani-System und einem TSK-System, vorgestellt. Das hybride Fuzzy-System erlaubt es, das Potenzial der negativen Regeln des Mamdani-Teilsystems für die Modellierung mit dem TSK-Teilsystem zu nutzen. Da die Verarbeitung negativer Regeln auf Mamdani-Systeme beschränkt ist, also keine direkte Transformation in TSK-Regeln möglich ist, wird so eine bisher bestehende Lücke zu den TSK-Systemen in gewisser Weise geschlossen. Das hybride Fuzzy-System kann von industriellen Anwendern mit dem Fuzzy-ROSA-Verfahren in gleicher Weise erzeugt werden wie die den Anwendern bereits bekannten Mamdani- oder TSK-Systeme und erfordert so keine weitere Einarbeitung oder Umstellung. So sollen industrielle Anwender für den potenziellen Nutzen negativer Regeln sensibilisiert werden. Eine systematische Untersuchung des Potenzials negativer Regeln sowohl für Mamdani-Systeme als auch für die hier vorgestellten hybriden Fuzzy-Systeme wird gegenwärtig durchgeführt.

## 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] DuMouchel, W. H. ; O'Brien, F. L. Integrating a robust option into a multiple regression computing environment. In: *Computer Science and Statistics: Proceedings of the 21st Symposium on the Interface, American Statistical Association*, Alexandria, VA (1989), S. 297–301.
- [2] Goodall, C. R.: Computation using the QR decomposition. *Handbook in Statistics, Volume 9. Statistical Computing*, Amsterdam, NL, 1993.

- [3] Holland, P. W. ; Welsch, R. E. Robust regression using iteratively reweighted least-squares. In: *Communications in Statistics - Theory and Methods*, v. A6 (1977), S. 813–827.
- [4] Jessen, H.: *Test- und Bewertungsverfahren zur regelbasierten Modellierung und Anwendung in der Lastprognose*. Düsseldorf : VDI, 2000 (Fortschritt-Berichte VDI, Reihe 8, Nr. 836)
- [5] Kiendl, H. ; Krabs, M.: Ein Verfahren zur Generierung regelbasierter Modelle für dynamische Systeme. In: *at - Automatisierungstechnik* 37 (1989), Nr. 11, S. 423–430.
- [6] Kiendl, H.: *Fuzzy Control methodenorientiert*. München : Oldenbourg 1997.
- [7] 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* (Mikut, R.; Jäkel, J., Hrsg.), S. 1–14. Karlsruhe : Wissenschaftliche Berichte Forschungszentrum Karlsruhe, FZKA 6509.
- [8] Krause, P.: Generierung von Takagi-Sugeno-Fuzzy-Systemen aus relevanten Fuzzy-Regeln. In: *Proceedings 10. Workshop Fuzzy Control des GMA-FA 5.22* (Mikut, R.; Jäkel, J., Hrsg.), S. 84–97. Karlsruhe : Wissenschaftliche Berichte Forschungszentrum Karlsruhe, FZKA 6509.
- [9] Krause, P.: *Datenbasierte Generierung von transparenten und genauen Fuzzy-Modellen für mehrdeutige Daten und komplexe Systeme*. Düsseldorf : VDI, 2001 (Fortschritt-Berichte VDI, Reihe 10, Nr. 691)
- [10] Krone, A.: *Datenbasierte Generierung von relevanten Fuzzy-Regeln zur Modellierung von Prozesszusammenhängen und Bedienstrategien*. Düsseldorf : VDI, 1999 (Fortschritt-Berichte VDI, Reihe 10, Nr. 615)
- [11] Mamdani, E. H. ; Gaines, B. R.: *Fuzzy Reasoning and its Applications*. London : Academic Press. 1981.
- [12] Mackey, M. ; Glass, L. Oscillation and Chaos in Physiological Control Systems. In: *Science* 197 (1977), S. 287–289.
- [13] Slawinski, T.: *Analyse und Generierung von relevanten Fuzzy-Regeln in hochdimensionalen Suchräumen*. Düsseldorf : VDI, 2001 (Fortschritt-Berichte VDI, Reihe 10, Nr. 686)
- [14] Sterneke, 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* (Mikut, R.; Reischl, M., Hrsg.), S. 55–68. Karlsruhe: Universitätsverlag Karlsruhe.
- [15] Takagi, T ; Sugeno, M.: Fuzzy Identification of Systems and its Application to Modeling and Control. In: *IEEE Transactions on Systems, Man, and Cybernetics* 15 (1985), Nr. 1, S. 116–132.

# Mikrocontrollerbasiertes Regelungssystem für Ventilaktuatoren mit Fuzzy-Control

**Thomas Stolze, Steffen Braune, Klaus-Dietrich Kramer**

Institut für Automatisierung und Informatik GmbH

Schlachthofstraße 4, D-38855 Wernigerode

An-Institut der Hochschule Harz

Tel.: +49(0)3943 / 659-325

Fax: +49(0)3943 / 659-399

E-Mail: u16571@hs-harz.de, sbraune@hs-harz.de, kkramer@hs-harz.de

## 1 Einführung – Variabler Ventiltrieb

Die Automobilindustrie ist ständig bestrebt, neue und innovative Produkte herzustellen. Diese Entwicklung lässt sich auch im Motorenbau beobachten. Besonders bezüglich umweltpolitischen Gesichtspunkten schreitet die Entwicklung immer weiter voran, konkurriert jedoch mit den Forderungen nach immer leistungsstärkeren Fahrzeugen.

Eine mögliche Lösung dieses Problems ist eine variable Ventilsteuerung. Dabei erhalten die Gaswechselventile eines Verbrennungsmotors verschiedene Freiheitsgrade, die es ermöglichen, sowohl die Öffnungszeiten des jeweiligen Ventils als auch den Ventilhub selbst zu beeinflussen. Zusätzlich erlaubt diese Technologie den durch Temperaturänderungen und Verschleiß notwendigen Ventilspielausgleich elektronisch zu realisieren.

Diese Variabilität ermöglicht: [1]

- eine Erhöhung des Drehmoments vor allem bei geringen Drehzahlen
- eine Reduzierung der Schadstoffemissionen um bis zu 50%
- eine Reduzierung des Kraftstoffverbrauchs um bis zu 20%

Konventionelle Motoren öffnen und schießen die Gaswechselventile über Nocken, die sich auf einer rotierenden Nockenwelle befinden. Durch die Nockenkontur können Ein- und Auslasszeitpunkte sowie der Ventilhub bestimmt werden. Allerdings ist dieses System stark eingeschränkt bezüglich der Freiheitsgrade, denn die Nockenkontur ist unveränderlich.

Seit einigen Jahren gibt es Untersuchungen zum Einsatz vollvariabler (elektromagnetischer) Ventilsteuerungen, bei denen jedes Ventil einen eigenen Antrieb (Aktuator) besitzt. Als ein aussichtsreich angesehenes System sei hier das 2-Magnetprinzip genannt, bei dem die Ventilbewegung durch Druckfedern bestimmt wird, wobei in den jeweiligen Endlagen durch einen Elektromagneten die notwendige Haltekraft aufgebracht wird. Zu diesem System gibt es vielfältige Untersuchungen, die allerdings auch die Schwachstellen dieser Anordnung aufzeigen. So ist es hierbei nicht oder nur sehr bedingt möglich, Teilhübe zu realisieren. Die Bewegungstrajektorien sind durch das Feder-Masse-System vorgegeben und nur geringfügig zu beeinflussen. Der

Ventilspielausgleich wird weiterhin hydraulisch vorgenommen. Der Leistungsbedarf ist vergleichsweise hoch. Als Alternative zu diesem Konzept werden seit einigen Jahren Linearmotoren als Ventilaktuatoren untersucht und sind bislang auf Grund ihrer Komplexität, ihres Bauvolumens und ihres Leistungsbedarfes nicht in Serienmotoren eingesetzt worden. Am Institut für Automatisierung und Informatik Wernigerode (IAI) wurden, gefördert im Rahmen eines MAHREG-Projektes, diese Entwicklungen aufgegriffen und verschiedene Linearmotorkonzepte und deren zugehörige Regelungsstrategien untersucht. Diese Linearaktuatoren, welche die Realisierung von Teilhuben, genauso wie einen elektronischen Ventilspielausgleich erlauben, zeichnen sich durch hohe Dynamik, geringe bewegte Masse und kleines Bauvolumen aus. Dieser Beitrag thematisiert die Regelung eines solchen Ventilaktuatoren.

## 2 Elektromagnetischer Ventilaktuator

### 2.1 Funktion

Aus den diversen Entwicklungen hat sich ein Konzept herauskristallisiert, das als „Moving-Magnet-Prinzip“ zu bezeichnen ist. Kennzeichen dieses Aktuators sind stationäre Spulen, in denen ein Paket von Permanentmagneten bewegt wird, das eine Kraftwirkung auf das Gaswechselventil ausübt. (vgl. Abb. 2-1).

Fließt nun ein Strom durch die Spulen, so wird ein magnetischer Fluss erzeugt. Der Fluss wird durch einen Eisenrückschluss zu Polschenkeln geleitet. Er wirkt bei einem Polschenkel in Magnetisierungsrichtung des Permanentmagneten, beim anderen Polschenkel entgegen der Magnetisierungsrichtung. Die dadurch entstehenden variierenden Flussverhältnisse bewirken eine Kraft in senkrechter Richtung auf den Läufer, die sich bei umgekehrtem Vorzeichen des Spulenstromes in umgekehrter Richtung mit gleichem Betrag ausbildet. Somit ist es möglich, durch geeignete Bestromung der Spulen das mit dem Läufer kontaktierte Ventil zu bewegen. Die Ventilbewegung wird zusätzlich von einem Federsystem unterstützt. Die Bewegungsrichtung wird durch den Richtungsvektor der Kraft bestimmt, die sich als Resultierende aus Federkraft und Magnetkraft ergibt. Die Bewegung des Läufers bestimmt so die Hubkurve des Ventils und kann durch Bestromung der Spulen variiert werden.

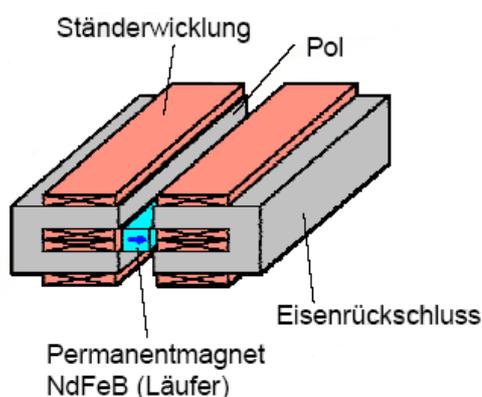


Abb. 2-1: Moving-Magnet-Aktuator [1]

## 2.2 Besonderheiten

Der Moving-Magnet-Ventilaktuator ermöglicht einen Hub von bis zu 12 mm. Aufgrund des Flusses der Permanentmagnete ist er in der Lage, bei bestimmten Auslenkungen das Ventil stromlos gegen die Federkraft in Ruhelage zu halten. Dies kann genutzt werden, um die benötigte elektrische Leistung gering zu halten.

Die wichtigste Eigenschaft des Aktuators ist allerdings sein nichtlinearer Kraftverlauf. Der Aktuator ist in der Lage, Kräfte größer 600 N zu erzeugen. Er kann so auch für Auslassventile verwendet werden, die einen hohen inneren Gasdruck beim Öffnen nach dem Verbrennungsvorgang überwinden müssen. Die verfügbare Kraft ist für ausgewählte Ströme in Abbildung 2-2 dargestellt. Bei 4 mm ist das Ventil vollständig geschlossen, bei entsprechend kleiner werdenden Positionen öffnet es sich. Die verfügbare Kraft ist gerade bei ca. 3,5 mm und  $-3,5$  mm am stärksten. Dies ist vorteilhaft, da in diesem Bereich die Öffnungs- und Schließvorgänge gestartet bzw. beendet werden. Die Richtung der erzeugten Kraft ändert beim Durchfahren der Mittellage ihr Vorzeichen. Die Hubkurve des Ventils entsteht, indem der Läufer des Aktuators eine vorgegebene Trajektorie nachfährt (vgl. Abb 5-1).

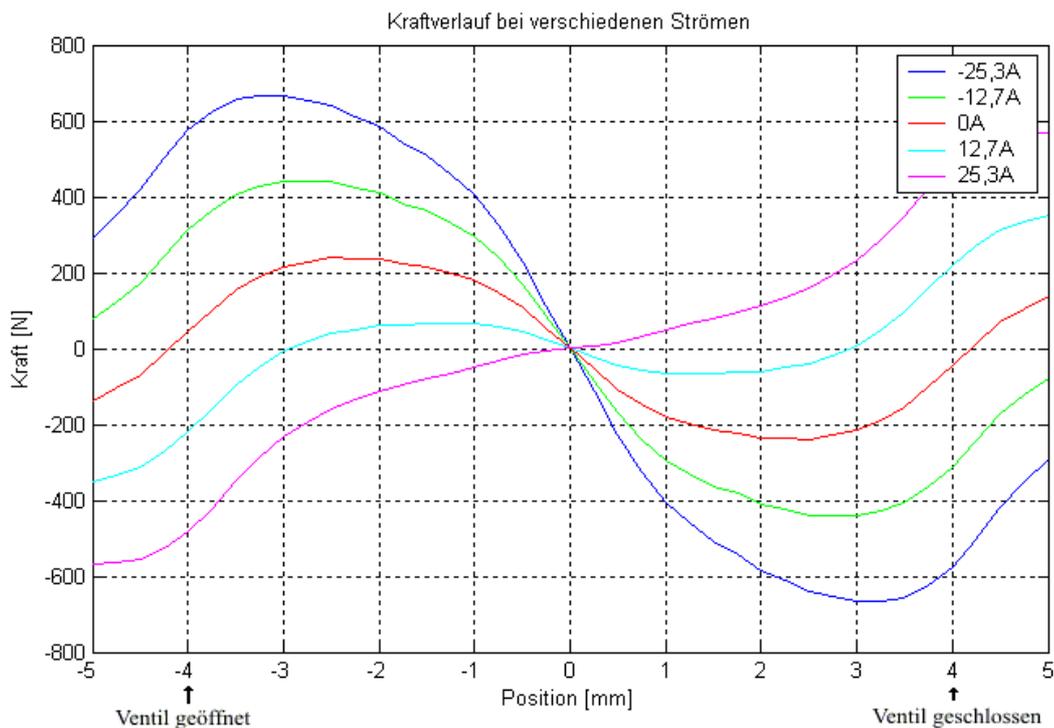


Abb. 2-2: Kraftverlauf bei verschiedenen Strömen

### **3 Konzepte mit Fuzzy-Control**

Das vorgestellte Aktuatorsystem kann mit Hilfe von Matlab / Simulink in einem Modell simuliert werden. Auf diese Weise ist es möglich, das Verhalten mit verschiedenen Regelungskonzepten zu analysieren. Allerdings sind nicht alle Einflussgrößen bekannt und können so auch nicht modelliert werden. Teilweise ist jedoch eine intuitive Beschreibung des Systemverhaltens möglich. Im realen System treten zudem kontinuierliche Veränderungen der Einflussgrößen auf. Es ist nur bedingt möglich, dieses Verhalten nachzubilden. Trotzdem muss ein Regler für dieses System robust genug sein, um diese Veränderungen zu kompensieren.

Aufgrund des beschriebenen nichtlinearen Verhaltens und der Tatsache, dass die Gesamteigenschaften nicht vollständig mathematisch beschrieben werden können, ist der Einsatz von Fuzzy-Control für die zu erstellende Regelung erwogen worden. Die mit Fuzzy-Control erstellten Regler werden im nächsten Kapitel vorgestellt.

#### **3.1 Fuzzy-Controller 1**

Der Controller arbeitet nach dem Min-Max-Prinzip, dem Max-Operator bei Aggregation, dem Min-Operator bei Implikation sowie dem Center-of-Area-Verfahren bei der Defuzzifizierung.

Es wurden für den Regler vier Eingangsgrößen vorgesehen: Aktuelle Position, Positionsabweichung (Regelfehler), Geschwindigkeit der Soll- und Istpositionskurve zueinander sowie die Richtung der Ventilbewegung.

Dabei sind die dritte und vierte Eingangsgröße von besonderer Bedeutung. Die Geschwindigkeit der Kurven zueinander, also die Ableitung der Regelabweichung, wird genutzt, um eine zusätzliche Größe zur Verfügung zu stellen, um die Dynamik des Systems zu beschreiben. Bewegen sich die beiden Kurven mit relativ hoher Geschwindigkeit voneinander weg, so kann davon ausgegangen werden, dass eine nicht unerhebliche Abweichung die Folge sein wird, auch wenn die aktuelle Position noch gut mit der Sollposition übereinstimmt. Solche Ereignisse lassen sich vor allem an Punkten der Solltrajektorie beobachten, an denen stark beschleunigt oder verzögert wird, also besonders zu Beginn und am Ende des Öffnungs- und Schließvorgangs. Somit kann hier bereits dem Entstehen solcher Abweichungen entgegengewirkt werden, da die Systemdynamik frühzeitig ausgewertet werden kann.

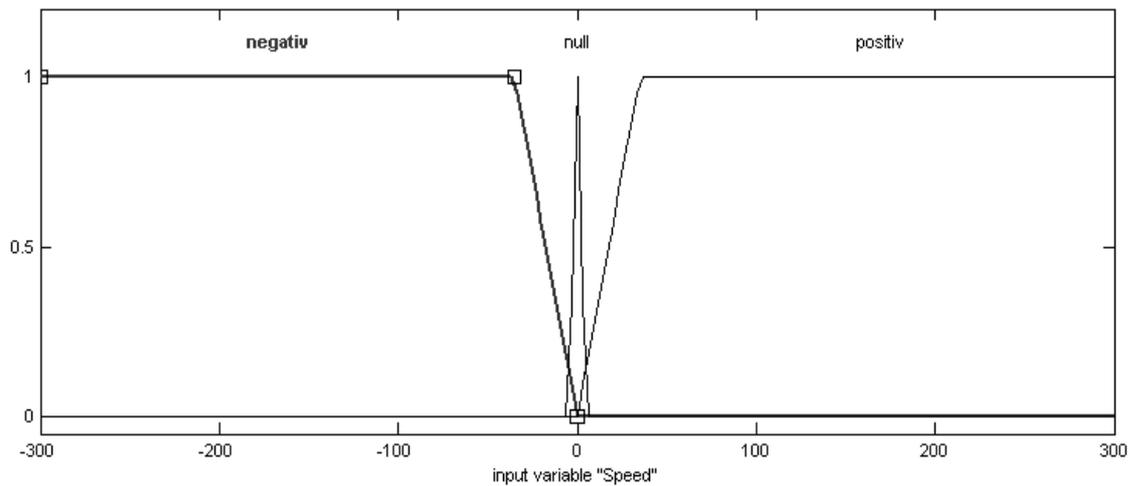


Abb.3-1: Zugehörigkeitsfunktionen der Bewegungen von Soll- und Istposition zueinander

Solange sich Ist- und Sollwert nicht mit großer Geschwindigkeit zueinander bewegen, wird die Eingangsvariable Geschwindigkeit nahezu null sein. Sobald sich jedoch größere Bewegungen andeuten, kann aufgrund der stark steigenden Zugehörigkeitsfunktionen dieses erkannt und der Bewegung entgegengewirkt werden. Damit werden große Abweichungen, wie sie durch hohe Geschwindigkeiten in diesem Zusammenhang auftreten könnten, bereits im Ansatz verhindert.

Die Angabe der Bewegungsrichtung hingegen verschafft dem Fuzzy-Controller einen weiteren Vorteil. Da die Solltrajektorie nicht symmetrisch ist, können mit Hilfe der Angabe der Bewegungsrichtung die Regeln für Abwärts- und Aufwärtsbewegung asymmetrisch aufgestellt werden. Es ist somit nicht notwendig, asymmetrische Kurventeile mit gleichen Regeln zu bearbeiten.

Des weiteren verfügt der Regler über die Eingangsgrößen Position und Positionsänderung. Damit kann, bezogen auf die Positionsänderung, eine angemessene Stärke der Reaktion mit Hilfe der Zugehörigkeiten beim Reglerausgang festgelegt werden. Darüber hinaus ermöglicht die Kenntnis der Position Rückschlüsse auf bestimmte Ereignisse. So wird z.B. im positiven Bereich bei etwa 4 mm stark beschleunigt, um das Ventil zu öffnen. Tritt hier eine hohe Positionsänderung auf, so bedeutet dies, dass der Läufer offenbar Probleme hat, dem vorgegebenen Lauf der Solltrajektorie zu folgen. Dies ist insbesondere in der Hinsicht kritisch, da dieser Punkt entscheidend für Verlustleistung und das entsprechende Bremsmanöver bei geöffnetem Ventil ist. Somit muss hier unter allen Umständen versucht werden, die Regelabweichung so klein wie möglich zu halten.

Um möglichst vielfältige und damit gut anzupassende Sollstromwerte für den Aktuator am Ausgang des Reglers zu erhalten, wurden die Ausgangszugehörigkeiten entsprechend so gewählt, dass jeweils sehr hohe, mittlere und geringe Ströme durch geeignete Regelauswahl den Eingangsgrößen zugeordnet werden können. Dabei muss das Vorzeichen des Stromes mit berücksichtigt werden. Zusätzlich gibt es auch die Möglichkeit, den Strom gleich null zu setzen. Dies ist insbesondere im mittleren Bereich der Flugbahn bei etwa 0 mm vorteilhaft, da hier trotz hoher Ströme kaum Änderungen an der Flugbahn des Ventils vorgenommen werden können. Selbst durch

hohe Ströme können hier nur sehr geringe Kräfte aufgebracht werden, was ungünstig hinsichtlich der Verlustleistung ist.

Insgesamt kommen bei diesem Reglerkonzept 54 Regeln zu Einsatz, die das Verhalten des Controllers beschreiben. Dadurch besitzt dieser Controller einen nicht unerheblichen Rechenaufwand.

### 3.2 Fuzzy-Controller 2

Der zweite entwickelte Fuzzy-Controller basiert auf den Erfahrungen des ersten Fuzzy-Controllers. Auch dieser Regler verwendet das Min-Max-Verfahren, den Max-Operator für Aggregation, den Min-Operator für Implikation und die Center-of-Area-Methode bei der Defuzzifizierung. Es wurde versucht, zusätzliche Verbesserungen in punkto Stabilität und Anpassungsfähigkeit vorzunehmen. Somit verwendet dieser Fuzzy-Controller fünf Eingangsgrößen: Position, Positionsdifferenz, Beschleunigung, Beschleunigungsdifferenz sowie die aktuelle Bewegungsrichtung. Dabei werden Position und Positionsdifferenz sowie die Bewegungsrichtung wie beim ersten Fuzzy-Controller verwendet. Neu hingegen ist die Verwendung von Beschleunigung und Beschleunigungsdifferenz.

Die Sollbeschleunigung wird aus der vorliegenden Solltrajektorie ermittelt, indem diese zweimal abgeleitet wird. Somit kennt der Regler in jedem Punkt die erforderliche Beschleunigung und kann, entsprechende Regeln vorausgesetzt, in geeigneter Weise auf die Vorgaben reagieren.

Die Beschleunigungsdifferenz wird dann aus den Werten der vorliegenden Ist- und Sollbeschleunigung errechnet. Die Zugehörigkeitsfunktionen beider Eingänge ähneln sich.

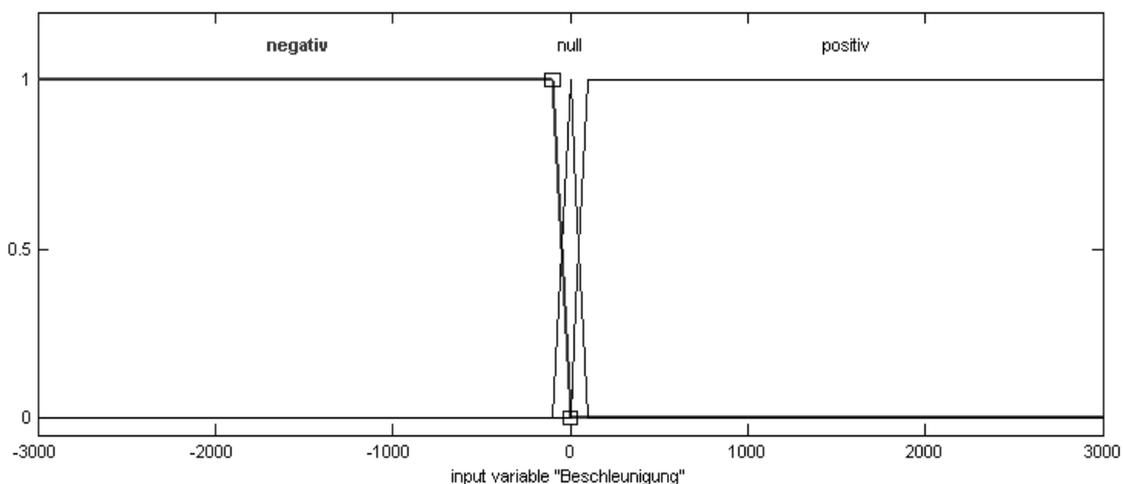


Abb. 3-2: Zugehörigkeitsfunktionen der Eingangsvariable Beschleunigung

Es gibt, wie in Abb. 3-2 zu sehen ist, einen sehr schmalen Bereich, in dem die Beschleunigung bzw. Beschleunigungsdifferenz sehr klein bzw. null ist. Jeweils zur positiven und negativen Seite erreichen die Werte dann durch steile Anstiege den Wert

1. Das bewirkt, dass der Controller nur kleine Abweichungen „toleriert“ und bei darüber hinaus gehenden Abweichungen bereits andere Regeln zum Einsatz kommen, die hohe Korrekturströme zur Folge haben. Es ist jedoch gerade bei den Regeln der Beschleunigungsdifferenz darauf zu achten, wie sich Istwert und Sollwert zueinander verhalten. Durch unterschiedliche Positionierungen können Differenzen der Beschleunigung entstehen, die das Vorzeichen wechseln, wie Abb. 3-3 zeigt.

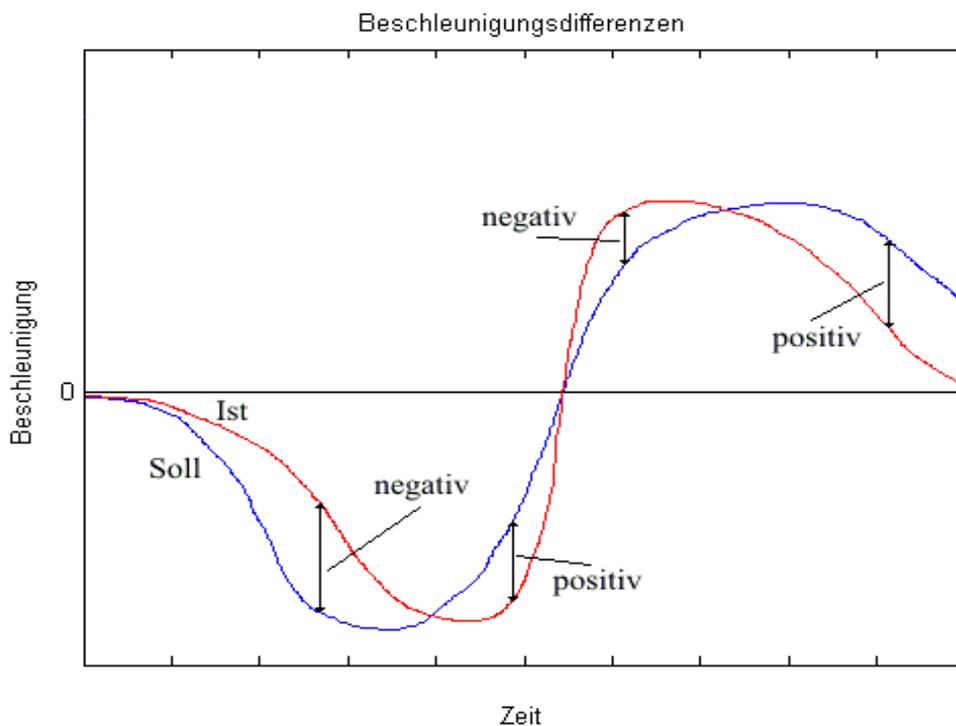


Abb. 3-3: Vorzeichenwechsel der Beschleunigungsdifferenz bei variierender Beschleunigung

Somit ist bei der Regelerstellung dieser Effekt besonders berücksichtigt worden, um nicht Gefahr zu laufen, dass sich die Regeln widersprechen. Die Beschleunigung und die Beschleunigungsdifferenz können in Simulink durch entsprechende Differenzierungen der Positionswerte ermittelt werden. Treten nun beim Ventilhub an bestimmten Kurvenpunkten der Position Defizite in Form von Abweichungen auf, so erfolgt die Auswertung der Beschleunigung und deren Differenz zu diesem Zeitpunkt. Es kann so ermittelt werden, welcher Fall dazu geführt hat, dass beispielsweise der Läufer nicht schnell genug auf eine bestimmte Position fahren konnte. Liegt das Problem nicht an einer Begrenzung des Stromes, so kann eine Regelanpassung mit Hilfe von Beschleunigung und Beschleunigungsdifferenz dazu genutzt werden, dem Controller in genau diesem Arbeitspunkt ein verändertes Verhalten zu geben. Natürlich müssen die Auswirkungen der Regeländerungen erneut getestet werden. Der Fuzzy-Controller besitzt damit zwei hochdynamische Eingangsgrößen, die in solchen Entscheidungsfällen zugunsten einer positiven Beeinflussung des Regelergebnisses genutzt werden können.

Es entsteht durch den Einsatz der fünf Eingänge ein relativ komplexes Regelwerk, das sehr vielfältige Möglichkeiten bietet, den Regler an besondere Kurvenverläufe anzupassen. Zudem können durch die Bewegungsrichtung auch Asymmetrien beachtet werden.

Der Fuzzy-Controller besitzt einen Ausgang. Die Zugehörigkeitsfunktionen des Ausgangs, der wiederum den Sollstrom angibt, ähneln denen des ersten Fuzzy-Controllers. Auch hier wird zwischen sehr großem, mittleren und kleinem Strom sowie null unterschieden. Der Ausgang wird durch sieben Zugehörigkeitsfunktionen bestimmt.

Insgesamt fällt jedoch der Rechenaufwand des Reglers mit 108 Regeln hoch aus, denn bedingt durch die fünf Eingangsgrößen steigt bei Beachtung aller wichtigen Kombinationen die Anzahl verwendeter Regeln deutlich an.

### **3.3 Fuzzy-Controller 3**

Wie auch die anderen beiden Controller verwendet dieser das Min-Max-Verfahren, den Max-Operator für Aggregation, den Min-Operator für Implikation sowie das Flächenschwerpunktverfahren bei der Defuzzifizierung. Es sei angemerkt, dass zusätzlich auch Tests mit anderen Verfahren durchgeführt wurden, so beispielsweise das Max-Prod-Verfahren bei Aggregation und Implikation oder das Sum-of-Maxima-Verfahren bei der Defuzzifizierung. Jedoch konnte keine Verbesserung des Reglerverhaltens gegenüber den ursprünglichen Einstellungen festgestellt werden.

Dennoch unterscheidet sich Fuzzy-Controller 3 grundlegend von den beiden vorangegangenen Konzepten. Er verfügt über drei Eingänge, wobei jedem Eingang genau ein Ausgang zugewiesen wurde.

Der erste Eingang erhält als Eingangsgröße die aktuelle Positionsdifferenz  $\Delta s$  zwischen Sollposition und Istposition. Mit Hilfe von fünf Regeln wird der zugehörige Ausgabewert ermittelt. Die Zugehörigkeitsfunktionen und die Regeln beeinflussen das Verhalten des Ausgangs derart, dass der Ausgangswert immer stärker nichtlinear ansteigt, je größer die Abweichung von der Mittellage wird. Dieses Verhalten stellt sicher, dass bei kleinen Abweichungen keine zu heftige Reaktion des Ausgangs stattfindet, und so etwa ein Schwingen entstehen könnte. Große Abweichungen werden hingegen energisch verhindert, da der Ausgabewert hier umso größer ist. Das in Abbildung 3-4 dargestellte Verhalten ermöglicht eine genauere Regelung, als dies mit einem linearen Verhalten möglich ist..

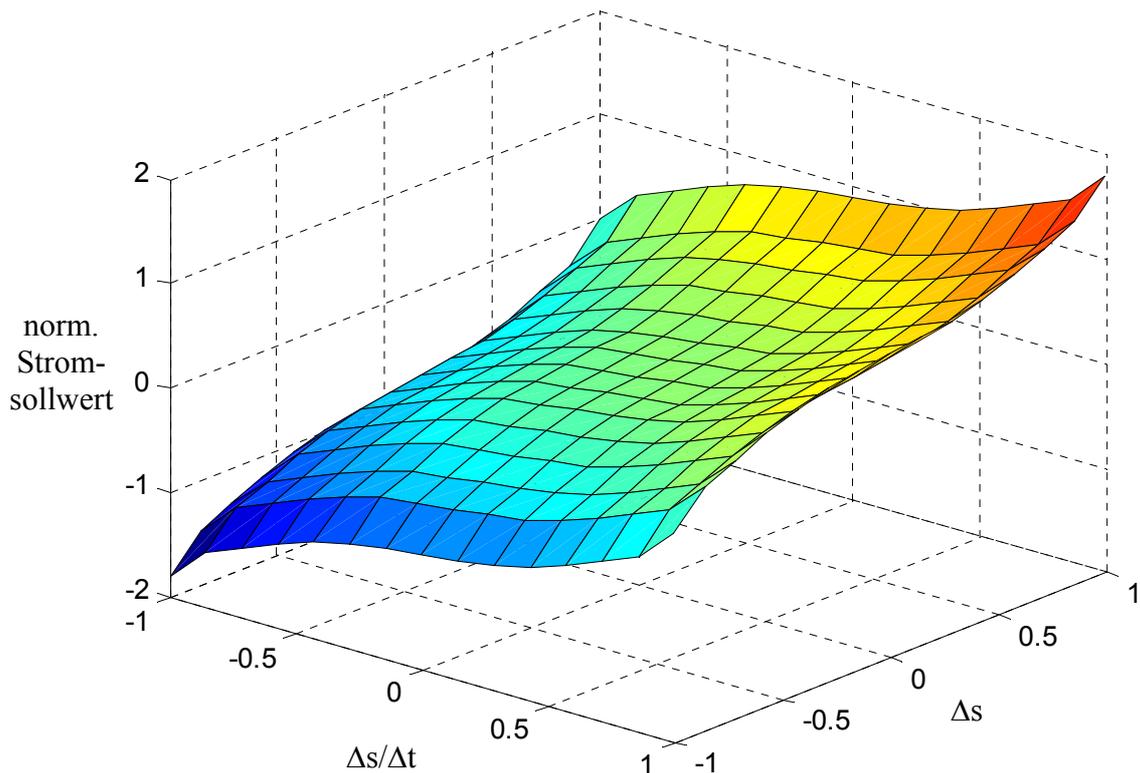


Abb. 3-4: 3-D-Surface-Ansicht des normierten Stromsollwertes

Der zweite Eingang wird mit der ersten Ableitung der Positionsänderung  $\Delta s/\Delta t$  beschaltet. Somit kann der Controller die Änderung der Positionsänderung erkennen. Diese Änderung wird ebenfalls mit fünf Regeln auf den zugehörigen Ausgang umgesetzt. Dabei wird ein ähnliches Verhalten wie das des ersten Eingangs verwendet. Somit wird auch hier sichergestellt, dass bei geringen Änderungen der Differenz nur geringe Korrekturen, bei großen Abweichungen aber umso größere Korrekturen vorgenommen werden.

Die Ausgabewerte von erstem und zweitem Controllerausgang werden zusätzlich mit Faktoren multipliziert, da die Zugehörigkeitsfunktionen der Ausgänge normiert worden sind. Auf diese Weise können leicht Veränderungen der Verstärkungsfaktoren vorgenommen werden, ohne dass die Charakteristik des Controllers verändert werden muss. Anschließend erfolgt eine Summierung beider Ausgabewerte.

Der dritte Eingangswert ist die aktuelle Ventilposition. Anhand dieses Wertes kann der Regler wiederum durch fünf Regeln eine dynamische Verstärkung errechnen, die es ermöglicht, an bestimmten kritischen Positionen die Gesamtverstärkung zusätzlich zu erhöhen oder zu verringern. So wird die Verstärkung entsprechend Abbildung 3-5 bei etwa 4 mm angehoben, um den Öffnungsvorgang zu unterstützen und beim Schließvorgang ein sanftes Aufsetzen des Ventils auf dem Ventilsitz zu ermöglichen. Gerade in diesen Bereichen kommt es sonst zu Abweichungen infolge zu geringer Verstärkung. Beim Durchschreiten der Mittellage wird die Gesamtverstärkung auf etwa 70 % abgesenkt, so dass hier eine Verringerung der Verlustleistung erzielt werden kann, denn es können hier ohnehin nur sehr geringe Kräfte erzeugt werden.

Der Ausgabewert des dritten Ausgangs enthält somit die globale Verstärkung und wird multiplikativ mit der Summe der beiden anderen Ausgänge (normierter Stromsollwert) verknüpft.

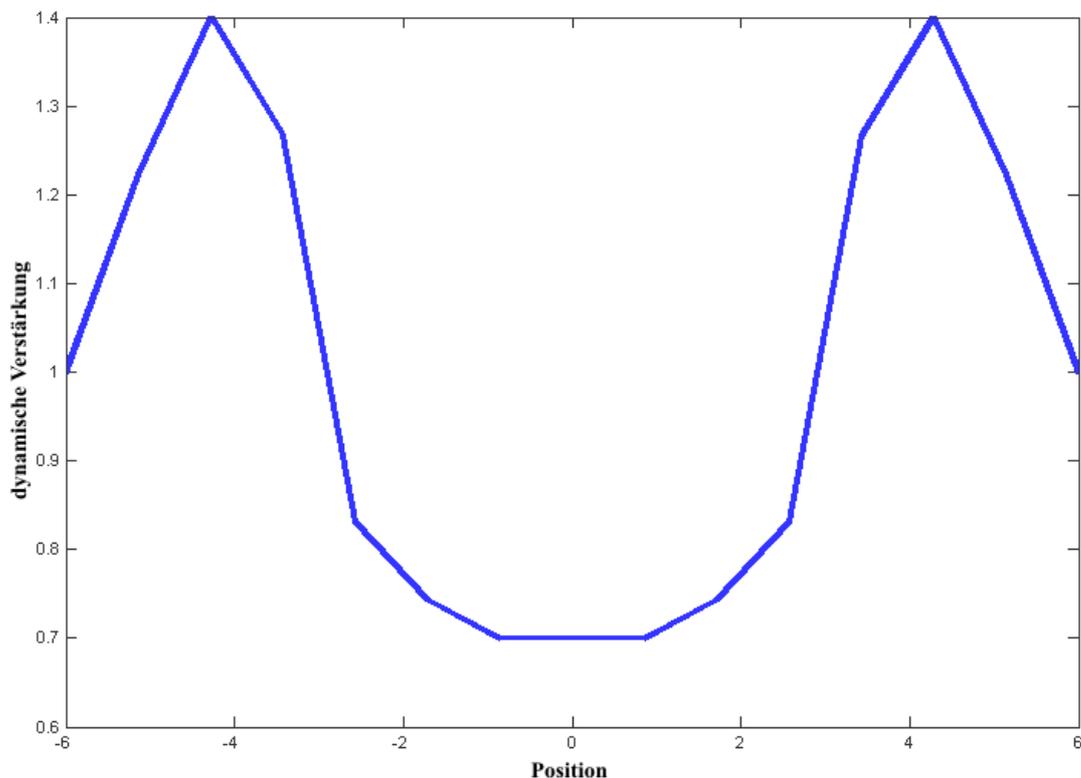


Abb. 3-5: Dynamische Verstärkung des Fuzzy-Controllers

Insgesamt besitzt dieser Controller nur 15 Regeln, was auf einen geringeren Rechenaufwand schließen lässt. Dieser Controller eignet sich daher besser zur Implementierung auf einem Mikrocontroller.

## 4 Test und Bewertung der Konzepte

Die verschiedenen Fuzzy-Controller wurden in Matlab / Simulink in einem Modell des Aktuatorsystems getestet. Dabei wurden unter anderem die mittlere Verlustleistung, die mittlere quadratische Abweichung von Soll- und Istposition sowie die Landegeschwindigkeit des Ventils auf dem Ventilsitz analysiert. Um auch die variierenden Bedingungen des realen Systems simulieren zu können, wurden verschiedene Tests durchgeführt, bei denen die Parameter des Modells verändert wurden. Bei den Tests stellte sich heraus, dass die ersten beiden Fuzzy-Controller teilweise sehr gute Ergebnisse erzielen konnten. Allerdings blieben diese Konzepte nicht bei allen Tests stabil. Somit musste davon ausgegangen werden, dass gerade beim realen Einsatz Probleme entstehen würden, wenn keine Idealbedingungen vorliegen. Der dritte Fuzzy-Controller war jedoch bei allen Tests stabil. Er erzielte zudem die

insgesamt besseren Ergebnisse und besitzt dennoch nur 15 Regeln. Die Implementierung auf dem Zielsystem basiert daher auf dem dritten Fuzzy-Controller.

## **5 Realisierung mit Mikrocontroller Infineon XC167**

Die Regelung wurde auf dem Mikrocontroller XC167 mit 20 MHz Taktfrequenz implementiert. Dieser Mikrocontroller bietet das beste Schnittstellenkonzept für die Ansteuerung, verfügt über ausreichend RAM- und Flash-Speicher und besitzt ein effizientes Interruptsystem. Außerdem existieren von diesem Controller viele Derivate, so dass eine Auswahl entsprechend den Anforderungen getroffen werden konnte.

Dennoch ist der XC167 kein Controller, der speziell für Fuzzy-Algorithmen entwickelt wurde. Dies wurde deutlich, als die Laufzeit des von Matlab / Simulink exportierten C++-Quellcodes des Fuzzy-Controllers auf dem XC167 ermittelt wurde. Da diese selbst nach Optimierungen um ein Vielfaches höher war, als dies die Echtzeitanforderungen der Regelung zulassen, musste die Implementierung des Fuzzy-Controllers mit Hilfe von Look-Up-Tables manuell erfolgen. Nur so konnte der komplette Berechnungszyklus für einen aktuellen Ausgabewert in weniger als 50  $\mu$ s erfolgen. Die analogen Prozessgrößen werden dem Regler durch den internen ADU des XC167 mit einer Auflösung von 10 Bit bereit gestellt. Der vom Controller berechnete Stromsollwert wird mit ebenfalls 10 Bit Genauigkeit als Spannungswert ausgegeben und über eine nachgeschaltete Leistungselektronik dem Aktuator aufgeprägt.

Der Regelalgorithmus wurde in einen komplexen Steuerungsablauf integriert. Mit Hilfe einer seriellen bidirektionalen Kommunikationsverbindung erfolgt die Parametrierung und der Datenaustausch der Regelungsergebnisse mit einem PC. Der PC wird zur Mensch-Maschine-Kommunikation eingesetzt. Die eigenentwickelte Software „SCI“ übernimmt die Benutzerinteraktion und die Visualisierung und Speicherung der Ergebnisse.

Als Beispiel eines ausgeführten Ventilhubes im Vergleich zur verwendeten Solltrajektorie soll Abbildung 5-1 betrachtet werden. Es ist zu erkennen, dass noch Abweichungen bei der Aktuatorbewegung oberhalb 4 mm und im Bereich des Öffnungsvorgangs bestehen. Es muss davon ausgegangen werden, dass dies eines der ersten Ergebnisse der Regelung mit diesem System ist. Erwartungsgemäß kann daher das Ergebnis nicht bereits optimal sein. Der Fuzzy-Controller sollte daher zukünftig so abgestimmt werden, dass gerade in diesen Bereichen die Regelung noch verbessert wird. Durch eine zusätzliche Verstärkungsanpassung kann die Präzision des Reglers erhöht werden.

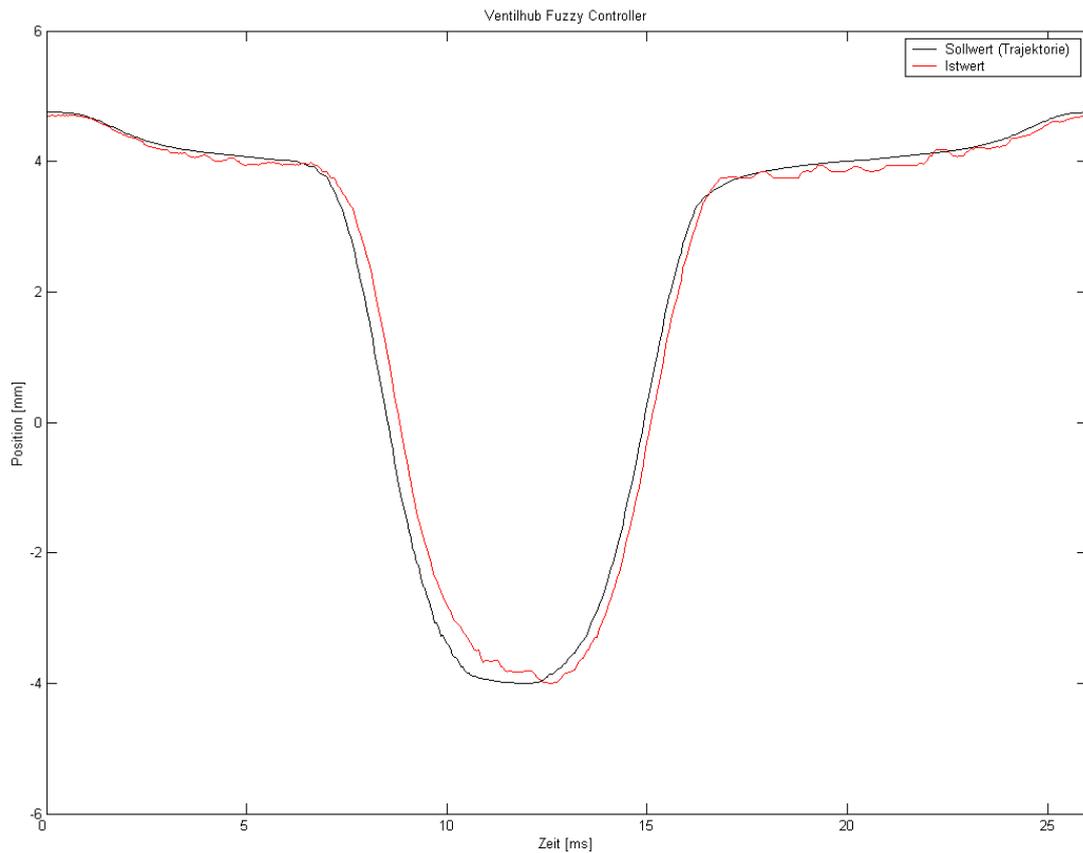


Abb. 5-1: Realer Hub des Regelungssystems

## 6 Weiterführende Arbeiten

Da nun die Funktion der Regelung grundlegend nachgewiesen worden ist, sollte in weiteren Versuchen eine Optimierung der Regelung erfolgen. Um die gewünschten Freiheitsgrade der variablen Ventilsteuerung zu erreichen, müssen alternative Trajektorien getestet werden. Durch Variation der Öffnungs- und Schließzeitpunkte sowie der Haltezeiten, kann eine Anpassung an verschiedene Drehzahlen und Lastbedingungen des Verbrennungsmotors durchgeführt werden.

## 7 Literatur

- [1] Braune, S.: Untersuchungen zum Einsatz von Linearmotoren als Ventilaktuator zur Betätigung von Gaswechselventilen in Verbrennungsmotoren. Institut für Automatisierung und Informatik GmbH Wernigerode; 2005
- [2] Schultes, R.; Pohle, I.: 80C166 Mikrocontroller. Franzis-Verlag, Poing: 1994
- [3] Braess, H.;Seiffert, U.: Handbuch Kraftfahrzeugtechnik (2.Auflage). Vieweg-Verlag, Braunschweig / Wiesbaden: 2001
- [4] Kahlert, F.; Frank, H.: Fuzzy-Logik und Fuzzy-Control (2.Auflage). Vieweg-Verlag, Braunschweig / Wiesbaden: 1994
- [5] Zimmermann, H.: Neuro+Fuzzy Technologien – Anwendungen. VDI-Verlag, Düsseldorf: 1995

# Datenanalyse-Plattform InformationMiner

**Frank Rügheimer, Rudolf Kruse**

Institut für Wissens- und Sprachverarbeitung, Universität Magdeburg

Universitätsplatz 2, 39106 Magdeburg

Tel.: (0391) 67 18182

Fax: (0391) 67 12018

E-Mail: {ruegheim, kruse}@iws.cs.uni-magdeburg.de

## Zusammenfassung

Data-Mining-Verfahren haben in der Vergangenheit ihre Nützlichkeit bei der Bearbeitung verschiedenster Fragestellungen gezeigt. Umfangreiche Softwarepakete stellen heute Methoden und Datenaufbereitungsverfahren bereit und begleiten ihre Anwender bei vielen Schritten des Wissensentdeckungsprozesses [6]. Die für Spezialanwendungen erforderlichen maßgeschneiderten Analyselösungen können jedoch mit den bisher verfügbaren Werkzeugen oft nur unzureichend unterstützt werden und durch die Beschränkung auf ein vorher festgelegtes Methodenrepertoire stehen vielversprechende oder optimierte Verfahrenvarianten mitunter nicht zur Verfügung. Mit einer von unserer Arbeitsgruppe entwickelten Software versuchen wir diese Aspekte stärker zu berücksichtigen und eine erweiterbare Plattform zur effizienten Lösung von anwendungsspezifischen Data-Mining-Problemen bereitzustellen.

## 1 Motivation

Gefördert durch Weiterentwicklung von Techniken zur Erhebung, Speicherung und Verarbeitung von Daten sowie die zunehmende Verfügbarkeit geeigneter Analysemethoden, konnte Data-Mining in den letzten Jahren zur Bearbeitung einer Vielzahl unterschiedlichster Problemstellungen aus Wissenschaft und Wirtschaft eingesetzt werden. Aktuelle Werkzeuge unterstützen gewöhnlich mehrere Phasen des Data-Mining-Prozesses und stellen zahlreiche etablierte Verfahren bereit. Diese Bandbreite ermöglicht die Bearbeitung durchaus unterschiedlicher Problemklassen.

Als nachteilig erweist sich jedoch mitunter die beschränkte Erweiterbarkeit dieser Pakete. Gerade der Einsatz neuerer Analysemethoden kann hierdurch erschwert werden. So kann ein potentiell Erfolg versprechendes Verfahren in der Praxis unberücksichtigt bleiben, da innerhalb des vorhandenen Programmpakets keine entsprechende Implementierung bereitgestellt wird. Eventuell verfügbare externe Implementierungen dagegen, sind gegebenenfalls nur mit erheblichen Aufwand in den Data-Mining-Prozess einzubinden; dem Anwender ist es dann oftmals nicht möglich von den im Paket implementierten unterstützenden Tools, z.B. zur Vorverarbeitung oder Evaluierung, zu profitieren.

Einige der in Forschung und Wirtschaft auftretenden komplexen Fragestellungen lassen sich bei Beschränkung auf Standardverfahren allein nicht hinreichend detailliert behandeln. In vielen Fällen gelingt es jedoch unter Ausnutzung anwendungsspezifischen Hintergrundwissens, etwa mittels einer Komplexitätsreduktion anhand

bekannter Nebenbedingungen, Erfolge zu erzielen. Dagegen können geschickte Kombinationen unterschiedlicher Verfahren den Zugang zu Problemen erschließen, sich mit einer Methode allein nicht bearbeiten lassen. Gemeinsam mit der großen Variabilität in Anforderungen und Zielen bei Data-Mining-Problemen inspiriert dies ein System für die Entwicklung hochgradig flexibler und vom Nutzer spezifisch konfigurierbarer Lösungen. Ein geeigneter Ansatz findet sich im Konzept eines Software-Baukastens. Hierbei können vorgefertigte Module vom Nutzer zu einer Systemlösung zusammengestellt werden. Da enthaltenen Module zudem einzeln konfiguriert werden können, lässt sich das gewünschte Maß an Flexibilität realisieren. Zugleich ist jedoch zu berücksichtigen, dass die effiziente Nutzung eines derartigen Systems nach außen hin eine Reduktion des Konfigurationsbedarfes erfordert, so dass anwendbare Lösungen mit geringem Zeitaufwand erstellt werden können. Der scheinbare Gegensatz lässt sich durch Bereitstellung von Mechanismen zur weitestgehend automatischen Vorkonfiguration aufheben. Auch die Bereitstellung spezieller sachbereichsbezogener Verfahrensbibliotheken kann die Anwender entscheidend bei der zügigen Lösung von Data-Mining-Problemen unterstützen.

Schließlich sollten neue mit aktuellen Anwendungsfeldern einhergehende Anforderungen Berücksichtigung finden. Aufgaben aus dem Bereich der Biologie, z.B. die Analyse von Genexpressionsmustern zeichnen sich durch besonders hochdimensionale Datenräume aus. Kennzeichnend für die Analyse multimedialer Daten ist die Notwendigkeit der Verarbeitung stark heterogener Datenbestände. Für die Bearbeitung derartiger Probleme geeignete Algorithmen sollen daher in Zukunft ebenfalls in die Methodenbibliotheken eingehen.

## 2 Systemstruktur

Die in der Arbeitsgruppe entwickelte Software InformationMiner orientiert sich an einer graphischen Beschreibung der Verarbeitungsstroms beim Data-Mining. In dieser prozessorientierten Sicht werden Teilfunktionalitäten, wie das Einlesen oder Vorverarbeiten von Daten, die Visualisierung, die Verwendung von Analyseverfahren oder Anwendung der gewonnenen Modelle jeweils als Knoten in einem gerichteten azyklischen Graphen repräsentiert. Die Kanten in diesem Graphen weisen auf eine Übertragung von Information zwischen den beteiligten Knoten hin (Pipes&Filter-Konzept [8]). Die graphische Darstellung des Verarbeitungsstroms ist als wesentliches Element in die mit Java implementierte Nutzeroberfläche eingebunden, wo die Anwender durch Anpassung dieses Graphen die von ihnen gewünschten Abläufe spezifizieren. Abbildung 1 zeigt einen einfachen Verarbeitungsstrom zur Erzeugung eines Fuzzy-Clustermodells [2, 10].

### 2.1 Verfahrensbibliothek

Als Grundlage für die Erstellung von Verarbeitungsströmen dient ein umfangreiches Repertoire vorimplementierter Methoden, die entsprechenden Knoten zugeordnet sind. Diese unterschiedlichen Knotentypen werden über Verfahrensbibliotheken bereitgestellt. Entsprechend ihrer Funktionalität können Knoten Kategorien zugeordnet werden. Neben modellerzeugenden Verfahren, wie der Induktion von

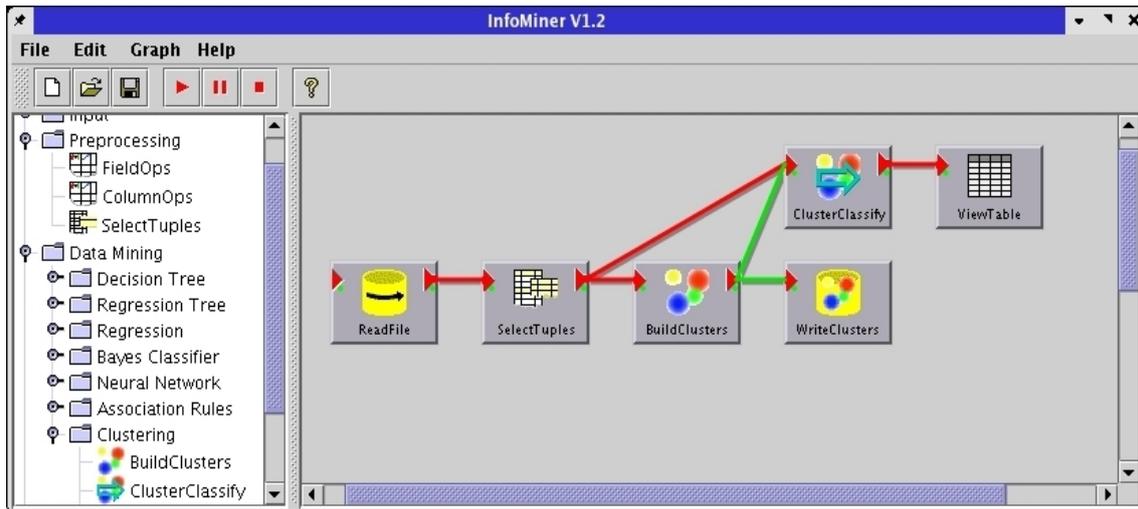


Abbildung 1: InformationMiner-Oberfläche einfachem Verarbeitungsstrom

Entscheidungs- und Regressionsbäumen [14, 4], Regellernern [1], Clusteringalgorithmen [2, 9, 10] und Neuronalen Netzen, werden Tools zur Unterstützung weiterer Abschnitte des Data-Mining-Prozesses bereitgestellt. Die Standardbibliothek unterscheidet hier Datenimport, Vorverarbeitung, Data-Mining, Visualisierung, Evaluierung oder Ausgabe. Durch weitere Unterteilung dieser Kategorien ergibt sich eine baumartige Struktur. Einigen Verfahren sind mehrere Knotenmodelle zugeordnet, da beispielsweise die Generierung eines Entscheidungsbaummodells von dessen Einsatz zur Datenklassifikation entkoppelt werden soll. Zur Erstellung eines Verarbeitungstromes werden die benötigten Knotenmodelle vom Verfahrensbaum auf die Arbeitsfläche gezogen, wo dann eine entsprechende Instanz erzeugt wird. Der neue Knoten lässt sich anschließend mit weiteren Knotenelementen verbinden und über nach außen geführte Parameter konfigurieren.

Auf technischer Seite werden die verfügbaren Knotenmodelle durch spezielle Java-Klassen definiert, die eine standardisierte Parameter- und Schnittstellenspezifikation und den Aufruf der dem Knoten zugeordneten Operation bereitstellen. Jeder Eintrag im Register der Verfahrensbibliothek verweist auf eine derartige Knotendefinitionsklasse und erklärt deren Positionierung(en) im Verfahrensbaum. Je nach geplantem Einsatzbereich kann die Verfahrensbibliothek von Entwicklerseite aus spezifisch angepasst und gegebenenfalls um dort benötigte Methoden erweitert werden. Das Softwarepaket selbst kann also in individuellen Konfigurationen ausgeliefert werden und so ein dem jeweiligen Sachbereich angemessenes Verfahrenrepertoire bereitstellen (vertikales System). Dieser Ansatz gestattet später eine zügigere Lösungsentwicklung, da sich die Anwender lediglich mit den in ihrem Bereich sinnvoll einsetzbaren Methoden vertraut machen müssen.

## 2.2 Verbindungskomponente

Die Verbindungskomponente von InformationMiner ermöglicht es Knoten zu komplexen Verarbeitungsströmen zu verknüpfen. Die Kopplung von Knoten erfolgt über Konnektoren [8], mit denen jeweils die Menge der Ein- bzw. Ausgaben der von den Knoten dargestellten Prozesse repräsentiert wird. Durch Klicken und Ziehen

lassen sich die Ausgaben eines Knotens einem anderen Knoten als Eingabe zuordnen. Um die Zulässigkeit derart spezifizierter Verarbeitungsströme zu sichern, wird vor Ergänzung einer Kante ermittelt, ob hierdurch Zyklen im Graphen geschlossen würden und überprüft, dass mindestens eine der Eingaben des nachgeschalteten Knotens mit der Ausgabe des neuen Vorgängers kompatibel ist. Eine automatische Zuordnung von Knotenausgaben zu den vom Nachfolger benötigten Eingaben erleichtert gleichzeitig den Anwendern die Konfiguration der von ihnen erstellten Data-Mining-Lösungen (siehe auch 2.3).

Für die eigentliche Verarbeitungen werden die im Graphen beschriebenen Abhängigkeiten aufgelöst, um eine Ausführungssequenz zu generieren. Hier erklärt sich auch das oben erwähnte Verbot gerichteter Zyklen, da sich in so einem Fall keine adequate Sequenz finden ließe. Nach Durchführung der Berechnungen können Teilergebnisse in den Knoten zwischengespeichert werden. Hierdurch lassen sich bei späterer Modifikation am Graphen oder nach Rekonfiguration einzelner Teilprozesse, noch gültige Resultate weiterverwenden, was die Abarbeitungszeit z.T. merklich reduziert. Eine Ausführung einzelner Teilpfade im Graphen ist ebenfalls möglich. Die genannten Eigenschaften erlauben eine zügige Entwicklung von Streams und gestatten es Nutzern die Auswirkungen von Modifikationen direkt zu beobachten.

Eine Besonderheit der Software besteht darin, dass sich durch Verbinden von Verarbeitungsknoten neben reinen Daten (rote Kanten) auch komplette Modelle (grün) übertragen lassen (vgl. Abbildung 1). Diese Eigenschaft gestattet es auch Anwendern ohne besondere Vorkenntnisse im Bereich Datenanalyse mit Hilfe einmal erstellter Verarbeitungsströme wiederholt auf jeweils aktuellen Daten basierende Modelle zu erstellen. Hiermit wird der Erfahrung Rechnung getragen, dass datengenerierende Prozesse oftmals in produzierenden Abteilungen von Unternehmen angesiedelt sind und die Ergebnisse der Auswertungen direkt vor Ort benötigt werden.

### **2.3 Konfiguration von Teilprozessen**

Um die erwünschte angemessene Berücksichtigung anwendungsspezifischer Vorgaben zu erreichen, ist ein hohes Maß an Konfigurierbarkeit der zur erstellenden Analyselösungen unerlässlich. Wie bereits im Abschnitt 1 erwähnt sollte der Nutzer aber bei der Wahrnehmung von Konfigurationsaufgaben entlastet werden. Bei der Plattform InformationMiner werden Streams bereits während ihrer Erstellung mit einer Basiskonfiguration versehen. Diese basiert einerseits auf der automatischen Zuordnung der Knoten Ein- und Ausgaben (vgl. 2.2), andererseits auf Ergänzung noch fehlender notwendiger Parameter durch sinnvolle Standardeinstellungen. Der Nutzer gelangt so nach Ergänzung nur weniger Angaben (z.B. dialoggestützte Auswahl der Datenquellen) zu einsatzbereiten Verarbeitungsströmen, mit denen sich gewöhnlich bereits akzeptable Analyseergebnisse erzielen lassen.

Für eine detailliertere Konfiguration stehen für alle Knoten generische Dialoge zur Verfügung (Abbildung 2). Diese gestatten den Zugriff auf die mit dem jeweiligen Prozess assoziierten Parameter und werden direkt aus den in der Knotenspezifikation abgelegten Angaben erzeugt.

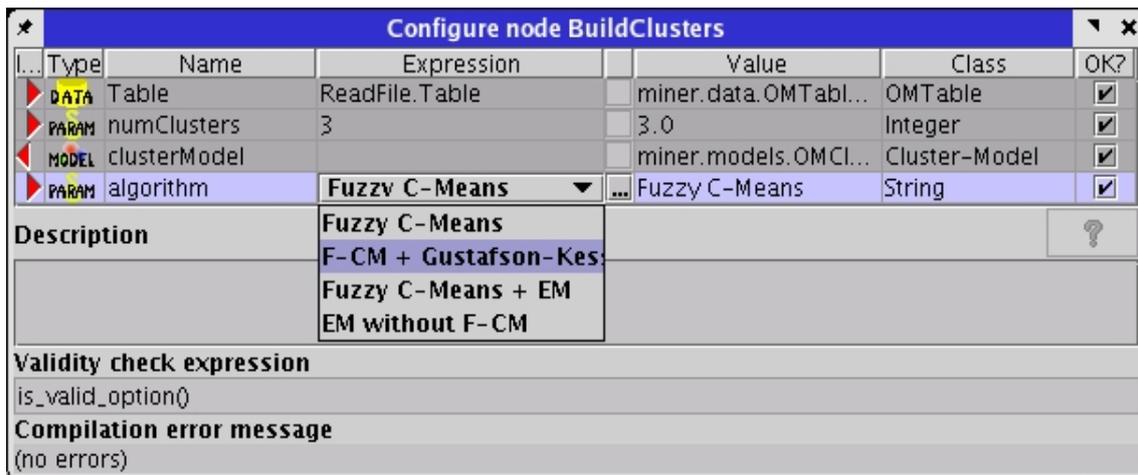


Abbildung 2: Beispiel eines Dialoges zur Knotenkonfiguration

## 2.4 Einbindung externer Programme

InformationMiner weist eine offene Architektur auf, die eine Integration externer Software gezielt fördern soll. Somit können bei Anwendungen, für die bereits Verfahren z.B. zur spezifischen Datenaufbereitung vorhanden sind, jene in die Gesamtlösung übernommen werden. Auch die Berücksichtigung von Algorithmenvarianten, die für das real vorliegende Problem optimiert wurden, ist so möglich. Weiterhin lassen sich auf diese Weise einfache kommandozeilenbasierte Implementierungen neuer Algorithmen zügig in ein komplexeres Data-Mining-System integrieren.

Die Implementierung kann dadurch extern und mittels einer vom Entwickler selbst wählbaren Programmiersprache erfolgen. Da die externen Programme nur auf dem Zielsystem ausgeführt werden müssen, können hier nämlich, im Gegensatz zu den plattformunabhängigen Oberflächenkomponenten, Binärprogramme eingesetzt werden. Durch Ausnutzung sprachspezifischer Bibliotheken und Konstrukte kann beispielsweise der Entwicklungsaufwand reduziert werden. Auch Optimierungen, die mit den in Java verfügbaren Datenstrukturen oder mit einem rein objektorientierten Paradigma nicht realisierbar wären, lassen sich so gezielter durchführen. Obwohl in einigen Fällen zusätzlicher Rechenaufwand, z.B. für die Konvertierung von Datenformaten zu berücksichtigen ist, erweisen sich derartige Lösungen insgesamt oft als vorteilhaft. Tatsächlich wurde die Mehrheit der im Basissystem enthaltenen Lernverfahren auf diese Weise bereitgestellt. Beim Aufruf wird dabei auf eine entsprechende plattformspezifische Variante des Binärprogramms zurückgegriffen. Die eigentliche Anbindung an die von InformationMiner bereitgestellte Oberflächenkomponente erfolgt über die bereits erwähnten Knotendefinitionsklassen, die als Wrapper [5] fungieren. Diese Klassen enthalten Knotenbezeichnung, Namen und Typ der Ein- und Ausgabeparameter sowie, im Falle externer Programme, die Generierung des notwendigen Kommandozeilenaufrufe, weiterhin optional einen Verweis auf ein zu verwendendes Icon. Die Erzeugung und Verwaltung der Knotenobjekte selbst sowie der Konfigurationsdialoge kann mit Hilfe der dort abgelegten Information automatisiert werden. Auf Grundlage dieses Frameworks ist von Entwicklerseite eine zügige Auslieferung sachbereichsspezifischer Paketvarianten möglich.

### 3 Anwendung

InformationMiner wird gegenwärtig im Rahmen einer Kooperation mit dem Deutschen Sparkassen- und Giroverband (DSGV) eingesetzt. Die Software dient dabei als Basisplattform zur Bearbeitung der im Finanzwesen auftretenden Fragestellungen. Zusätzlich zu den bereitgestellten Standardkomponenten werden in Zusammenarbeit mit dem Kooperationspartner fortwährend weitere, für den Anwendungsbereich relevante Verfahren integriert. Gleichzeitig werden so Erfahrungen aus dem Einsatz unter realen Bedingungen gewonnen.

Insgesamt betont das System eine stark anwendungsorientierte Perspektive, in der die Berücksichtigung problemspezifischer Restriktionen und gegebenenfalls die Verwendung angepasster Verfahren als wichtiges Mittel zur Verbesserung der Modellierungs- und Vorhersagequalität gesehen wird. Diese Ausrichtung spiegelt sich auch in der von InformationMiner unterstützten Trennung des Designs von Datenanalyselösungen von deren produktiver Anwendung durch die Endnutzer dieser Lösungen wider. Das nachfolgende einfache Beispielszenario soll stellvertretend einige Abläufe bei der Arbeit mit InformationMiner verdeutlichen. Ziel soll hier die Erzeugung eines Klassifikators für die Irisdaten sein.

#### 3.1 Datenimport und Datenexploration

Der verwendete Datensatz stammt aus dem UCI Machine Learning Repository (<http://www.ics.uci.edu/~mllearn/MLRepository.html>) und enthält in Form einer ASCII-Tabelle abgelegte Beispiele zur Blütengeometrie dreier Orchideenarten. Einer Analyse geht zunächst eine einfache Exploration des Datensatzes voraus:

Um die Daten zu importieren, wird aus der Methodenbibliothek der Knoten für die Dateieingabe gewählt und für das Einlesen der Quelldatei eingerichtet (Abbildung 3). Eine einfache Darstellung der darin enthaltenen Beispiele kann durch

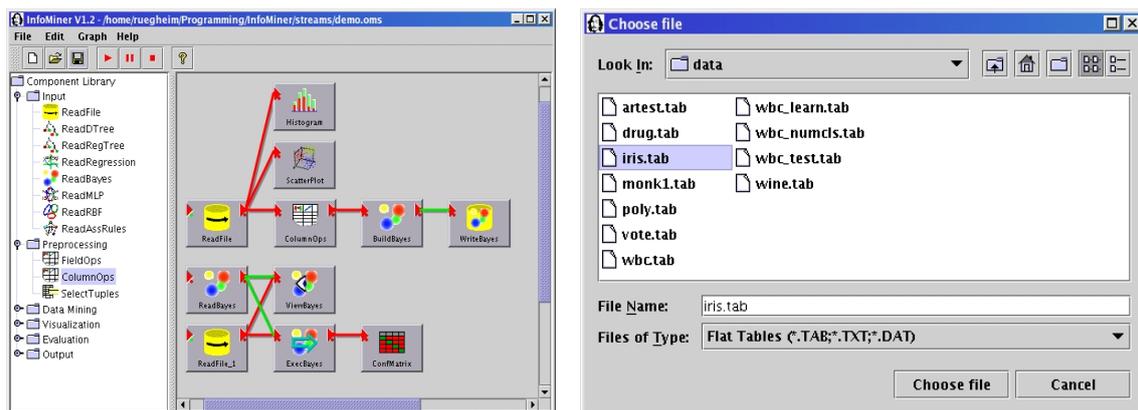


Abbildung 3: Verarbeitungsströme (l.) und Dateiauswahldialog (r.)

einen Scatterplot geleistet werden. Anhand einfacher Histogramme ist eine erste Beurteilung der Verteilung der Attributwerte für die jeweiligen Irisspezies möglich. Die entsprechenden Knoten werden daher mit dem Eingabeknoten verbunden. Bei Ausführung des Verarbeitungströmes werden beide Tools gestartet und auf die eingelesenen Daten angewendet. Wie eine sich zeigt, lässt sich eine der drei Klassen

deutlich abgrenzen, während zwischen den verbleibenden eine geringe Überlappung besteht (Abbildung 4).



Abbildung 4: Werkzeuge zur Datenexploration

Die Konvertierung oder Filterung der Daten ist mittels der bereitgestellten Vorverarbeitungsknoten möglich. Die im Beispiel eingesetzten Daten müssen nicht aufbereitet werden. Daher wird lediglich der `ColumnOps`-Knoten ergänzt, der Berechnungen unter Rückgriff auf die Attributwerte eines Eintrages gestattet und u.A. etwa Normalisierungen ermöglicht.

### 3.2 Modellbildung und Evaluierung

Zur Bearbeitung der Klassifikationsaufgabe soll hier ein Bayes-Klassifikator genutzt werden. Der Knoten für die Erstellung des Modells kann direkt mit der Preprocessingausgabe verbunden werden. Das von diesem Knoten erzeugte Modell wird zur späteren Wiederverwendung gespeichert. Eine Beurteilung ist durch Anwendung des wieder importierten Modells auf Testdaten möglich. Mit Hilfe des Knotens zur Berechnung der Verwechslungsmatrix kann das Klassifikationsergebnis zusammengefasst werden. Um auch das Modell selbst darzustellen, wird schließlich der `ViewBayes`-Knoten ergänzt und mit dem Modellimport sowie den Testdaten verbunden. Prinzipiell wäre es auch möglich gewesen, die Evaluierung direkt, also ohne Speicherung des erzeugten Klassifikators durchzuführen. Die hier gewählte Variante mit separater Erzeugung eines Modells entspricht jedoch eher den in der Praxis zu erwartenden Bedingungen. Der zweite, in der unteren Bildschirmhälfte dargestellte Verarbeitungsstrom repräsentiert eine Anwendung des fertigen Modells, auch wenn dies hier lediglich zum Zwecke der Evaluierung erfolgt. Auf ganz ähnliche Weise können andere Klassifikatoren induziert werden, so dass sich Methoden auch direkt vergleichen lassen.

### 3.3 Feineinstellung der Modellparameter

Bei Verwendung des Bayes-Klassifikators ohne weitere Konfiguration kommt die naive Variante des Verfahrens zum Einsatz. Dabei werden jedoch implizit starke Unabhängigkeitsannahmen getroffen. Ein flexibleres Modell erhält man durch Einstellen eines vollen Bayes-Klassifikators. Der Konfigurationsdialog für `BuildBayes`

bietet diese Option. Nach Modifikation der Verfahrensparameter werden von InformationMiner sämtliche noch in von BuildBayes abhängigen Knoten gespeicherten Zwischenergebnisse als ungültig markiert. Die anschließende Wiederholung der Berechnungen liefert ein verbessertes Ergebnis mit einer verringerten Anzahl fehlerhafter Zuordnungen (Visualisierung siehe Abbildung 5).

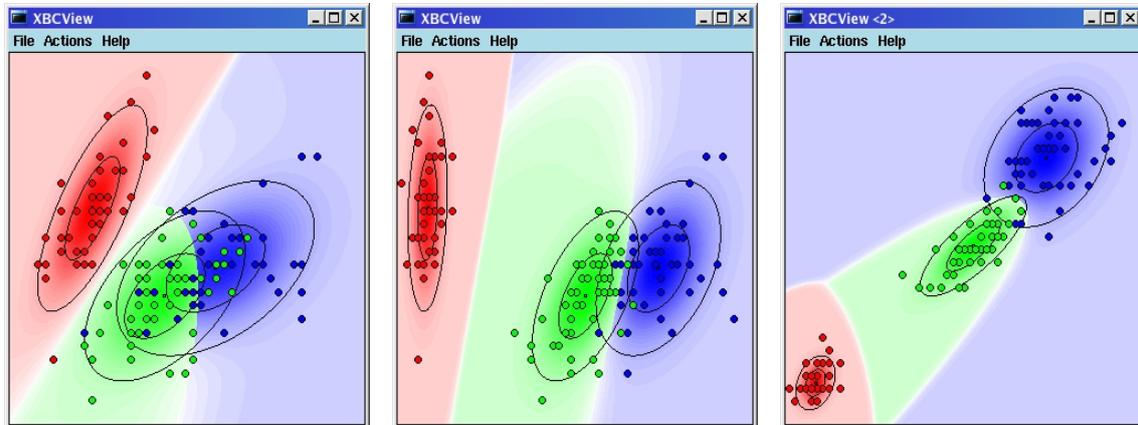


Abbildung 5: Visualisierung eines vollen Bayes-Klassifikators

Für den praktischen Einsatz bietet sich eine Trennung von Modellerzeugung und -einsatz an. Wie oben demonstriert ist das durch die Speicherung erzeugter Modelle in externen Dateien realisierbar. Das Bereitstellen fertiger Streams zur Datenanalyse und Modellbildung ermöglicht auch nicht in der Verwendung von Data-Mining-Methoden geschultem Personal die Durchführung von Untersuchungen oder Modellaktualisierungen. Insbesondere dann, wenn die Ergebnisse nah am datenerzeugenden Prozess benötigt werden, ist diese Vorgehensweise von Vorteil, da Analysetätigkeiten so abteilungsintern durchgeführt werden können.

Ein Vorteil der modularen Prozesskonzeption von InformationMiner tritt hingegen zu Tage, wenn Änderungen in Abläufen eine Anpassung des Analyseprozesses verlangen. Bei einer Umstellung der Datenhaltung kann etwa statt der Nutzung lokaler Dateien der Zugriff auf einen Datenbankserver erwünscht sein. Im Verarbeitungsstrom wären in diesem Fall lediglich die Datenimport- und -exportknoten zu ersetzen.

## 4 Geplante Erweiterungen

Obwohl die bisher beschriebenen Funktionen bereits zum gegenwärtigen Zeitpunkt ein weites Anwendungsspektrum eröffnen, konnten einige der in der Einleitung benannten Aspekte bisher noch nicht vollständig umgesetzt werden. Geplante Erweiterungen der Kernfunktionalität und der Methodenbibliothek sollen diese offenen Zielstellungen realisieren sowie die Arbeit mit der Plattform weiter vereinfachen.

### 4.1 Wizard für Knoteneinbettung

Die gegenwärtig genutzte einheitliche Knotenschnittstelle von InformationMiner erlaubt zwar bereits eine zügige Anbindung externer Software an die Data-Mining-

Umgebung, jedoch erfordert dies eine Einarbeitung in die von InformationMiner bereitgestellten Schnittstellen- und Parameterklassen sowie grundlegende Java-Programmierkenntnisse. Die vereinheitlichte Definition der Schnittstellen würde jedoch auch eine Integration mittels eines dialoggesteuerten Tools, welches die relevanten Daten vom Nutzer abfragt und in die entsprechenden Quelltexte umsetzt, gestatten. Hierdurch sollte eine weitere Vereinfachung der Verfahrensintegration erreicht werden können.

## 4.2 Umgang mit hochdimensionalen Datenräumen

Das in der Einleitung angesprochene Problem hochdimensionaler Datenräume ist im Data-Mining seit längerem bekannt. Derartige Problemstellungen wurden in der Vergangenheit häufig nicht direkt gelöst, sondern durch eine vereinfachte Beschreibung approximiert. Hierdurch wurde eine Bearbeitung zwar oft erst ermöglicht, doch gingen diese Ansätze mit einer verringerten Qualität der Ergebnisse einher. Nicht immer ist das für die Anwendung unproblematisch und viele aktuelle Fragestellungen, gerade aus dem Bereich der Bioinformatik, erfordern die Berücksichtigung einer sehr großen Zahl miteinander in Beziehung zu setzender Variablen.

Erfreulicherweise wurden in den vergangenen Jahren einige Verfahren veröffentlicht, die effiziente Operationen auf hochdimensionalen Datenräumen ermöglichen. Hier sind zum Einen verschiedene Ausprägungen der Graphischen Modelle [12, 13] zu nennen, die unter Ausnutzung vorhandener (bedingter) Unabhängigkeiten eine erhebliche Reduktion des Speicherbedarfs und der für Schlußfolgerungen erforderlichen Berechnungen gestatten, ohne die Ergebnisqualität signifikant zu beeinträchtigen. Implementierungen dieser Verfahren werden in absehbarer Zeit für InformationMiner verfügbar sein. Auf der anderen Seite finden sich Verfahren, die auf hierarchischen Fuzzy-Regeln basieren [7]. Diese erlauben den Anwendern u.a. eine Datenexploration auf unterschiedlichen Detailstufen, sind in der Lage Impräzision wiederzugeben und weisen eine hohe Verständlichkeit auf.

## 4.3 Verarbeitung heterogener Daten

Während Implementierungen vieler Data-Mining Verfahren von einheitlich strukturierten Daten ausgehen, ist diese Eigenschaft in der Praxis nicht immer gegeben. Bei der Verarbeitung von multimedialen Daten ist hiermit sogar definitionsgemäß zu rechnen. Je nach Anwendung und Ausprägung der Heterogenität eignen sich verschiedenste Lösungsansätze. Im Falle unvollständig klassifizierter Daten können Verfahren zum teilüberwachten Lernen [11], die derart inhomogene Daten direkt verarbeiten, genutzt werden. An anderen Fällen ist etwa eine Verwendung unterschiedlicher Importmodule bei zunächst getrennter Verarbeitung der Datenströme möglich. Jene ließen sich dann, z.B. nach einer Featureextraktion, zusammenführen und gemeinsam weiterverarbeiten. Als Alternative böte sich eine völlig separate Bearbeitung der Daten aus unterschiedlichen Quellen mit der Erstellung gegebenenfalls konkurrierender Modelle an. Die damit erstellten Prognosen ließen sich im Anschluß wiederum zusammenfassen.

## 4.4 Superknoten

Ein bisher für InformationMiner noch nicht realisiertes Konzept besteht in der Nutzung sogenannter Superknoten. Hierunter sind kombinierte Strukturen zu verstehen, die sich zwar der Schnittstelle nach wie einzelne Verarbeitungsknoten verhalten, deren Funktionalität in Wirklichkeit selbst aber mittels einer Kombination anderer Knoten realisiert wurde. Auf diese Weise lassen sich zusätzliche Abstraktionsebenen einführen oder wiederkehrende Teillösungen als funktionale Einheiten zusammenfassen. Zur Konfiguration der internen Knoten könnte Parameter vom Superknoteninterface weitergereicht werden. Dies ist vor allem dann sinnvoll, wenn intern mehrere Knoten mit identischer Parametrisierung verwendet werden sollen (mögliche Implementierung des Baggingverfahrens [3]). Die hierfür erforderliche, der Übertragung von Daten und Modellen analoge Weiterreichung von Parameterwerten ist in InformationMiner vorgesehen, zum gegenwärtigen Zeitpunkt jedoch ebenfalls noch nicht vollständig implementiert.

## 5 Fazit

Das Data-Mining-System InformationMiner behandelt viele der bisher nur wenig beachteten Probleme bei der Erstellung praxisnaher Data-Mining Lösungen. Durch ein hohes Maß an Flexibilität und die mögliche Integration neuer Algorithmen oder im jeweiligen Problemumfeld erprobter Verfahrensvarianten sollen in kurzer Zeit anwendungsspezifische Lösungen unter möglichst guter Ausnutzung des Kontextwissens zusammengestellt werden können. Eine Fokussierung auf spezialisierte Sachbeiche sowie Maßnahmen zur Unterstützung der Nutzer gestatten es, diese Tätigkeiten mit verhältnismäßig geringem Aufwand durchzuführen. Durch das Konzept adaptierbarer Verarbeitungströme kann der Einsatz von Data-Mining nah an die datengenerierenden Prozesse geführt werden. Somit können Analysen unmittelbar dort erfolgen, wo sie benötigt werden.

Auch den bereits erkennbaren Anforderungen an die Lösung künftiger Datenanalyseprobleme wird das Softwarepaket dank geplanter Erweiterungen in absehbarer Zeit genügen können. Kombiniert mit der Einführung neuer Konzepte werden sich somit relevante aktuelle Fragestellungen unter Ausnutzung der spezifischen Informations- und Datensituation effizient beantworten lassen.

### Danksagung

Die Entwicklung der in diesem Aufsatz beschriebenen Software wurde durch die DFG gefördert (Förderkennzeichen KR 521/4-1). Die Autoren danken Aljoscha Klose, Christian Borgelt und den zahlreichen weiteren Entwicklern für ihre Arbeit am Grundsystem und ihre Beiträge zum Methodenpool.

## Literatur

- [1] AGRAWAL, Rakesh ; IMIELINSKI, Tomasz ; SWAMI, Arun N.: Mining Association Rules between Sets of Items in Large Databases. In: BUNEMAN, Peter

- (Hrsg.) ; JAJODIA, Sushil (Hrsg.): *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*. Washington, D.C., 26–28 1993, 207–216
- [2] BEZDEK, James C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York, NY, USA : Plenum Press, 1981
- [3] BREIMAN, Leo: Bagging Predictors. In: *Machine Learning* 24 (1996), Nr. 2, 123-140. [citeseer.ist.psu.edu/breiman96bagging.html](http://citeseer.ist.psu.edu/breiman96bagging.html)
- [4] BREIMAN, Leo ; FRIEDMAN, Jerome H. ; OLSHEN, Richard A. ; STONE, Charles J.: *Classification and regression tree*. Belmont, CA : Wadsworth International Group, 1984
- [5] DELINE, Robert: Avoiding packaging mismatch with flexible packaging. In: *ICSE '99: Proceedings of the 21st international conference on Software engineering*. Los Alamitos, CA, USA : IEEE Computer Society Press, 1999, S. 97–106
- [6] FAYYAD, Usama ; PIATETSKY-SHAPIRO, Gregory ; SMYTH, Padhraic: From Data Mining to Knowledge Discovery in Databases. In: *AI Magazine* (1996), S. 37–54
- [7] GABRIEL, Thomas R. ; BERTHOLD, Michael R.: Constructing Hierarchical Rule Systems. In: BERTHOLD, Michael R. (Hrsg.) ; LENZ, Hans-Joachim (Hrsg.) ; BRADLEY, Elizabeth (Hrsg.) ; KRUSE, Rudolf (Hrsg.) ; BORGELT, Christian (Hrsg.): *Proc. 5th International Symposium on Intelligent Data Analysis (IDA 2003)*, Springer Verlag, 2003 (Lecture Notes in Computer Science (LNCS)), S. 76–87
- [8] GARLAN, David ; SHAW, Mary: An introduction to software architecture. In: AMBRIOLA, Vincenzo (Hrsg.) ; TORTORA, Genoveffa (Hrsg.): *Advances in Software Engineering and Knowledge Engineering*. Singapore : World Scientific Publishing Company, 1993, S. 1–39
- [9] GATH, Isak ; GEVA, Amir B.: Unsupervised optimal fuzzy clustering. In: *IEEE Transactions on pattern Analysis and Machine Intelligence* 11 (1989), S. 773–781
- [10] GUSTAFSON, Donald E. ; KESSEL, William C.: Fuzzy Clustering with a Fuzzy Covariance Matrix. In: *Proc. of the IEEE Conference on Decision and Control*, 1979, S. 761–766
- [11] KLOSE, Aljoscha ; KRUSE, Rudolf: Information Mining with Semi-Supervised Learning. In: *Advances in Soft Computing: Soft Methodology and Random Information Systems*. Berlin, Heidelberg : Springer-Verlag, 2004
- [12] LAURITZEN, S. L. ; SPIEGELHALTER, D. J.: Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. In: *Journal of the Royal Statistical Society, Series B* 2(50) (1988), S. 157–224
- [13] PEARL, J.: *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, USA : Morgan Kaufman, 1988. – (2nd edition 1992)
- [14] QUINLAN, J. R.: Induction of Decision Trees. In: SHAVLIK, Jude W. (Hrsg.) ; DIETTERICH, Thomas G. (Hrsg.): *Readings in Machine Learning*. Morgan Kaufmann, 1990. – Originally published in *Machine Learning* 1:81–106, 1986

- [15] RÜGHEIMER, Frank ; KRUSE, Rudolf: Information Miner - a Data Analysis Platform. In: MONTSENY, Eduard (Hrsg.) ; SOBREVILLA, Pilar (Hrsg.): *Proceedings of the Joint EUSFLAT-LFA Conference 2005, September 7-9, Barcelona*. Barcelona, Spain, 2005, S. 1182–1187

# **Zeitvariante Klassifikatoren für Brain Machine Interfaces und Neuroprothesen**

**Ole Burmeister, Markus Reischl, Ralf Mikut**

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**

Ein „Brain Machine Interface“ (BMI) ist gemäß Wolpaw eine Schnittstelle zwischen dem Gehirn und einem Gerät, die nicht von Aktivitäten in peripheren Nerven oder Muskeln abhängt [1]. BMIs ermöglichen eine Interaktion mit der Umgebung über neue Kanäle. Sie bieten sich insbesondere bei Patienten an, deren Nervenleitungen unterbrochen oder beeinträchtigt sind.

Hirnsignale für BMIs können auf verschiedene Arten aufgenommen werden. Eine nicht-invasive Technik ist das Elektroenzephalogramm (EEG), bei der Elektroden auf der Schädeloberfläche angebracht werden. EEG-basierte Systeme beruhen darauf, dass Hirnsignale willkürlich beeinflussbar sind (Übersicht in [1]). EEG-Sensoren messen Summenpotenziale einer relativ großen Anzahl oberflächennaher Nervenzellen. Die Signale sind durch das Hirnwasser und die Schädeldecke örtlich und zeitlich stark tiefpassgefiltert und anfällig für Artefakte [2, 3]. Invasive Techniken beinhalten z.B. Aufnahmen mit implantierten Elektroden (subkortikal) [4] oder epikortikal (auf der Hirnrinde) platzierte Elektrodenarrays (Elektrokortikogramm, engl: electrocorticogram, ECoG) [5]. Sie ermöglichen eine bessere Signalqualität, stellen aber hohe technische Anforderungen und sind einem höheren klinischen Risiko ausgesetzt [3, 6]. Bisher werden in erster Linie Versuche an Affen durchgeführt. Diese Systeme sind zum Teil in der Lage, Bewegungsabsichten zu erkennen und einen Roboterarm oder Cursor durch die aufgezeichneten Signale in zwei oder drei Dimensionen zu steuern (z.B. [4, 7]).

Bereits vorhandene Einsatzgebiete für BMIs sind Kommunikationssysteme für so genannte „Locked-In“ Patienten, die wegen der Lähmung nahezu aller Muskeln nicht mit ihrer Umwelt kommunizieren können. Diese Systeme sind in der Regel EEG-basiert (z.B. [8]). Eine Anwendung sind virtuelle Computer-Keyboards: Das Alphabet wird in eine Baumstruktur gebracht und der Patient wählt durch mehrere hierarchische Entscheidungen einen Buchstaben aus. Die erreichbare Anzahl an Buchstaben pro Minute hängt stark von den Fähigkeiten des Patienten ab, rekonstruierbare Signale zu erzeugen. Im Mittel sind zwischen 0.5 und 3 Zeichen pro Minute möglich.

Komplexere Aufgaben wie die Ansteuerung von Prothesen und Neuroprothesen beruhen derzeit meist auf mechanischen Schaltern (z.B. Freehand-System) oder der Auswertung von Muskelsignalen (Elektromyogramm, EMG) (z.B. Prothesenansteuerungen in [9–11]). Einsatzfähige BMIs lassen sich voraussichtlich nur mit invasiven Techniken erreichen [4]. Allerdings gibt es auch erste Versuche mit EEG-basierten BMIs für Neuroprothesen [12] und Rollstühle [13].

Die Ansteuerung von BMIs führt in der Regel auf Klassifikationsprobleme mit wenigen Klassen und geeignet vorverarbeiteten Hirnsignalen als Merkmalen (Übersicht in [14]). In den verschiedenen Anwendungen sind dem Klassifikator unterschiedliche Informationen zugänglich. Bei einem virtuellen Keyboard ist bekannt, in welchem zeitlichen Bereich eine für die Klassifikation relevante Aktivität vorhanden ist, da das BMI ein Startsignal (Trigger) generiert. Das System kann seinen Klassifikator selbstständig in einen aktiven bzw. inaktiven Zustand versetzen. Für eine Prothesensteuerung ist dieses Vorgehen nicht sinnvoll, da die Kontrolle der Prothese ausschließlich vom Patienten durchgeführt werden soll. Weiterhin ist es bei den virtuellen Keyboards akzeptabel, wenn Zeitreihen oder Merkmale über einen bestimmten Zeitraum betrachtet werden und nicht zu jedem Abtastpunkt eine Entscheidung getroffen wird. Im Gegensatz dazu darf die Auswahl des Kommandos für die Prothesensteuerung nicht zu lange dauern.

Die Verwendung mehrerer Klassifikatoren oder das Anlernen des Klassifikators zu verschiedenen Zeitpunkten ist sinnvoll, wenn in den Signalen Zeitvarianzen enthalten sind. Im Bereich der Mustererkennung werden bereits Multi-Klassifikatoren-Systeme verwendet. Solche Systeme kombinieren die Ergebnisse von Klassifikatoren oder wählen anhand gegebener Entscheidungsfunktionen den besten Klassifikator aus [15]. Im Gegensatz dazu erfolgt in dieser Arbeit die Auswahl der Klassifikatoren ausschließlich in Abhängigkeit von der Zeit. Es gibt einige Systeme, die zeitabhängige Klassifikatoren für ein BMI verwenden (z.B. [16]). In jüngerer Zeit häufen sich auch Arbeiten zur automatischen Detektion klassifikationsrelevanter Zeitabschnitte (z.B. [17]). Die selbstständige Aktivierung des Klassifikators und die Verwendung von zeitabhängigen Informationen ist ein großer Vorteil und führt zu deutlich besseren und robusteren Klassifikationsergebnissen. Eine systematische Kategorisierung und ein Vergleich der Klassifikatoren fehlen aber bisher.

Dieser Beitrag nimmt eine Kategorisierung von Klassifikatoren vor (Abschnitt 2) und beschreibt einige Realisierungen der Strategien (Abschnitt 3). Anhand eines künstlichen Datensatzes und eines experimentellen Datensatzes aus der BCI Competition III werden die Klassifikatoren verglichen (Abschnitt 4).

## 2 Systematische Kategorisierung der Klassifikation

Gegeben sei ein Klassifikationsproblem mit Klassen  $c = 1 \dots C$ . Ein Klassifikationssystem bezeichne ein System zur Lösung des Klassifikationsproblems. Es enthält mindestens einen Klassifikator und enthält als Eingabe Biosignale (z.B. Elektromyogramm: EMG, Elektroenzephalogramm: EEG) eines Nutzers. Weitere Klassifikatoren und Eingabekanäle sind erlaubt. Für die Steuerung des Klassifikationssystems wird ein übergeordneter Zustandsautomat mit folgenden Zuständen eingeführt (siehe Bild 1): *OFF* (Aus), *ON* (Ein), *IDLE* (Bereitschaft, Leerlauf), *CLASS* (Klassifizieren).

Im Zustand *OFF* ist das Klassifikationssystem ausgeschaltet, das Einschalten (Zustand *ON*) geschieht durch ein Triggersignal (Tr2). Das Klassifikationssystem befindet sich zunächst im Leerlauf (Zustand *IDLE*), Signale des Nutzers werden nicht für die Generierung von Aktionen verwendet. Ein weiteres Triggersignal (Tr3) schaltet das Klassifikationssystem in den Zustand *CLASS*. In diesem Zustand generiert der Klassifikator des Klassifikationssystems anhand der Eingabe des Nutzers eine Lösung für das Klassifikationsproblem und eine entsprechende Aktion. Ein erneutes Triggersignal (Tr4) führt das Klassifikationssystem in den Leerlauf. Bild 2 skizziert den zeitlichen Ablauf der Übergänge.

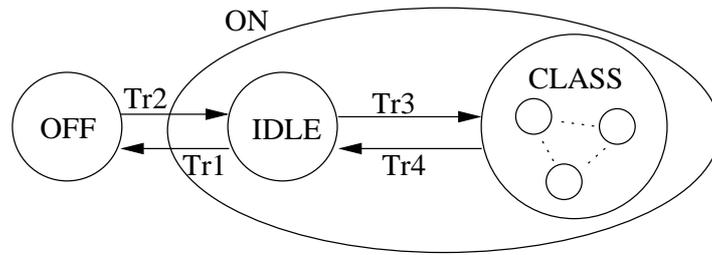


Bild 1: Zustandsübergänge in einem Klassifikationssystem. Innerhalb des Zustands *CLASS* befindet sich ein weiterer Automat, der das Klassifikationsproblem beschreibt.

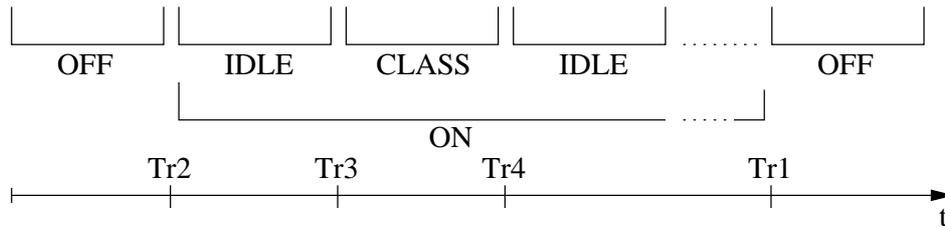


Bild 2: Zeitlicher Ablauf der Zustandsübergänge in einem Klassifikationssystem

Beim virtuellen Keyboard generiert das Klassifikationssystem den Trigger *Tr3*. In der Regel wird die Unterscheidung von zwei Buchstabenblöcken als Zwei-Klassen-Problem gelöst: Zunächst ist das Klassifikationssystem ausgeschaltet (*OFF*). Nach dem Einschalten (*ON*) sind keine Buchstabenblöcke auf dem Bildschirm vorhanden, der Nutzer kann keine Entscheidung treffen (*IDLE*). Sobald ein Buchstabenblock erscheint (*Tr3*), schaltet das Klassifikationssystem in den Zustand *CLASS*. In diesem Zustand wird die Eingabe des Nutzers für die Klassifikation verwendet und einer der Buchstabenblöcke gewählt. Nach einer bestimmten Zeit schaltet das Klassifikationssystem in den Leerlauf (*Tr4*).

Die Literatur unterscheidet bezüglich der Entstehung des Triggersignals für das Umschalten zwischen *IDLE* und *CLASS* zwei Systemklassen [17]. Ein *synchrones* Klassifikationssystem gibt die Zeitabschnitte vor, in denen der Nutzer Kontrolle ausüben kann (*Tr3*, *Tr4* vom System). Im Gegensatz dazu generiert bei *asynchronen* Klassifikationssystemen der Nutzer die Triggersignale *Tr3* und *Tr4* selbst (Bild 3).

Formal bedeutet diese Unterscheidung, dass Klassenänderungen zu Zeitpunkten  $k_{\text{trig}}$  mit  $y[k_{\text{trig}} - 1] \neq y[k_{\text{trig}}]$  auftreten. Im synchronen Fall sind die Zeitpunkte  $k_{\text{trig}}$  bekannt, asynchrone Klassifikationssysteme schätzen die Zeitpunkte:  $\hat{k}_{\text{trig}} \approx k_{\text{trig}}$ . Probleme ohne zeitabhängige Klassenänderung werden hier nicht betrachtet.

Die Triggersignale werden durch Bedienen eines Schalters oder durch Generierung bestimmter Muster im Biosignal erzeugt. Im ersten Fall ist die Erkennung des Triggers sehr einfach. Für den zweiten Fall muss das Muster im Biosignal robuste Klassifikationen ermöglichen, insbesondere für das Ein- und Ausschalten ( $ON \leftrightarrow OFF$ ) des Klassifikationssystems. Weiterhin muss das Signal willkürlich erzeugbar sein und die Wahrscheinlichkeit einer unwillkürlichen Erzeugung nahe Null liegen. Für das Umschalten von *IDLE* in *CLASS* durch Biosignale gibt es bereits einige Arbeiten (z.B. [17]), durch Biosignale gesteuerte Schalter für  $ON \leftrightarrow OFF$  sind nicht vorhanden.

Bei asynchroner Klassifikation kann das Erkennen des Triggers bereits der Klassifika-

tionsentscheidung entsprechen (Bild 3a). Ein Beispiel ist [12], das nur das Auftreten eines Weberschaltsignals für eine neue Griffart erkennt. Das Klassifikationsproblem enthält somit nur eine Klasse. Im Klassifikationssystem aus Bild 3b wird der Trigger asynchron vom Nutzer ausgelöst (A2/B2). Ein Klassifikator erkennt den Trigger und ein weiterer Klassifikator behandelt das eigentliche Klassifikationsproblem. Das in Bild 3c dargestellte Klassifikationssystem für das virtuelle Keyboard arbeitet synchron. Demgegenüber steht das Klassifikationssystem in Bild 3d (A4/B4). Das Klassifikationssystem löst entweder zu jedem Zeitpunkt das Problem  $c = 1, \dots, C$ . Eine alternative Realisierung betrachtet den Zustand *IDLE* als zusätzliche Klasse und unterscheidet diese Klasse von den problem-spezifischen Klassen:  $c = 1, \dots, C + 1$ . Am Beispiel der Steuerung einer Handprothese wirkt sich der Unterschied zwischen Bild 3b und 3d wie folgt aus: Im ersten Fall wird zunächst überprüft, ob ein Wunsch besteht, die Prothese zu nutzen. Erst wenn ein solcher Wunsch festgestellt wird, wird das Kommando klassifiziert. Im zweiten Fall wird zu jeder Zeit das Kommando klassifiziert. Die Steuersignale dürfen nicht unwillkürlich auftreten, um eine unbeabsichtigte Bedienung der Prothese zu verhindern.

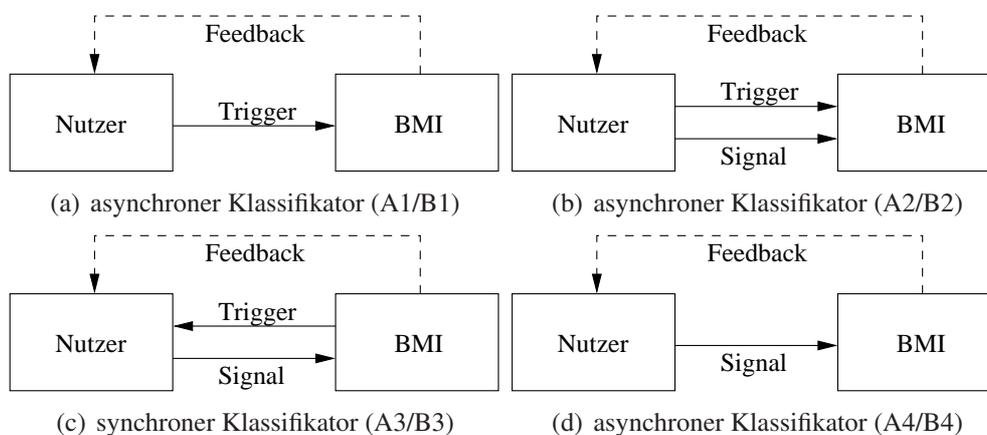


Bild 3: Synchroner und asynchrone Klassifikatoren mit verschiedenen Klassifikationsaufgaben gemäß Tabelle 1

Bei Einbeziehung mehrerer Abtastpunkte für eine Klassifikationsentscheidung  $\hat{y}(t)$  findet eine *zeitaggregierte* Entscheidung statt. Wird für jeden Abtastpunkt eine Entscheidung getroffen, handelt es sich um eine *zeitkontinuierliche* Klassifikation. Die zeitkontinuierliche Klassifikation ist ein Sonderfall der zeitaggregierten Klassifikation mit Fensterlänge 1. Die Unterschiede werden in zwei Kategorien zusammengefasst, „Klassifikationsaufgabe“ (synchron/asynchron und Erkennung von Trigger und/oder Klasse) sowie „zeitlicher Ablauf“ (zeitkontinuierlich/zeitaggregiert). Die Verwendung eines *zeitaggregierten* oder *zeitkontinuierlichen* Klassifikators hat insbesondere Konsequenzen für die Merkmalsgenerierung. Kontinuierliche Klassifikatoren sind auf kontinuierliche Merkmale angewiesen, z.B. Frequenzbänder oder gleitende Mittelwertschätzungen über Infinite-Impulse-Response (IIR) Filter, während zeitaggregierte Klassifikatoren Merkmale über einen gewissen Zeitraum generieren können, z.B. Kurzzeit-Spektrogramme oder summierte quadratische Abweichungen. Tabelle 1 gibt eine Übersicht über die möglichen Varianten inklusive einiger Beispiele.

A1 ist zwar schon in der Anwendung [18], es wird aber nur ein Trigger erkannt ( $C = 1$ ). Für die Fälle A2, A3, A4 und B3 gibt Tabelle 2 einen Überblick über mögliche Klassifi-

Klassifikations- aufgabe	zeitlicher Ablauf für die geschätzte Klasse $\hat{y}(t)$	
	zeitkontinuierlich (A)	zeitaggregiert (B)
asynchron (1) nur Trigger	EEG-Freehand [18]	
asynchron (2) Trigger & Klassen	BMI virtuelles Keyboard (3 Kl.) [19, 20] EMG-Prothese Herberths [10]	EMG-Prothese Reischl [21]
synchron (3) nur Klassen	BMI virtuelles Keyboard (2 Kl.) [11]	BMI-System mit Erkennung am Ende eines Signals [16]
asynchron (4) nur Klassen	ungetriggertes BMI für virtuelles Keyboard [11, 22, 23]	

Tabelle 1: Kategorisierung von Mensch-Maschine-Schnittstellen mit Biosignalen

katoren, Bild 4 eine grafische Übersicht über den zeitlichen Ablauf. B2 hat starke Ähnlichkeit zu B3 und wird daher nicht explizit betrachtet.

Der Klassifikator K1 verwendet einen (einheitlichen) Merkmalsatz für alle Abtastpunkte. Die Merkmale werden zwar zu bestimmten Zeitpunkten ausgewählt (in Bild 4 durch  $\boxtimes$  gekennzeichnet), angelernt wird aber über zusammengefasste Datentupel der bestimmten Zeitpunkte. Im Gegensatz dazu wird K3 separat zu jedem Abtastpunkt angelernt. Der Merkmalsatz wird, wie bei K1, zu einem oder mehreren bestimmten Abtastpunkten ermittelt. K2 verwendet einen zusammengefassten Merkmalsatz aus  $k$  Abtastpunkten und berechnet die Parameter des Klassifikators zu einem einzigen Abtastpunkt. Die Sicherheit dieses Klassifikators steigt tendenziell mit zunehmender Zahl an Abtastpunkten  $k_c$ . K4 bestimmt für jeden Abtastpunkt einen eigenen Merkmalsatz und lernt jeweils einen eigenen Klassifikator an. Der zeitinvariante Klassifikator K5 verwendet die Information der Zeit seit dem Triggersignal als  $(s_z + 1)$ . Merkmal.

Mit Ausnahme vom Klassifikator K1 klassifizieren alle Klassifikatoren nur innerhalb des Zustands *CLASS*. K1, K2 und K5 verwenden nur einen einzigen Klassifikator, K3 und K4 für jeden Abtastpunkt einen. Bis auf K2 erzeugt jeder Klassifikator zu jedem Abtastpunkt ein Ergebnis. K2 erzeugt sein Ergebnis zum Zeitpunkt  $k_c + \hat{k}_{\text{trig}}$ . Je nach Struktur des Klassifikators ist die Ausgabe von Zwischenergebnissen möglich.

Die Verfahren der Klassifikatoren K1-K5 für die in [24] vorgestellten Teilschritte Merkmalsauswahl ( $S_2$ ), Merkmalsaggregation ( $S_3$ ) und Klassifikation ( $S_4$ ) sind dem jeweiligen Problem anzupassen. Für die Klassifikation sind u.a. Bayes-Klassifikatoren, Support-Vektor-Maschinen, Neuronale Netze und Nearest Neighbour Klassifikatoren möglich (Übersicht in [25]). Die Merkmalsauswahl kann z.B. über eine multivariate Varianzanalyse (multivariate analysis of variance, MANOVA [26]) oder informationstheoretische Maße [27] erfolgen, die Merkmalsaggregation über Hauptkomponenten- oder Diskriminanzanalyse [26]:

$$\mathbf{x}_{\text{Roh}} \xrightarrow{S_2 \text{ mit } \mathcal{I}} \mathbf{x} \xrightarrow{S_3} \mathbf{z} \xrightarrow{S_4} \hat{y}. \quad (1)$$

In (1) werden die einzelnen Schritte dargestellt. Ausgehend von einem Rohdatensatz  $\mathbf{x}_{\text{Roh}}$  der Dimension  $(N, s)$  werden  $s_m$  Merkmale ausgewählt. Die Indizes der ausgewählten Merkmale sind in der Indexmenge  $\mathcal{I}$  enthalten, der resultierende Merkmalsatz  $\mathbf{x}$  hat die Dimension  $(N, s_m)$ . Anschließend wird eine Aggregation der Merkmale auf einen

	Beschreibung	Anzahl Merkmale $s$	Anzahl Klassif.	Anzahl Datentupel ( $N$ )	Entwurf	
					$S_2$	$S_3, S_4$
K1	Zeitinvarianter Klassifikator (A4)	$s_z$	1	$N_z \cdot N_s$	BA	BA
K2	Klassifikator mit Ergebnis am Ende eines Zeitbereichs (B3)	$s_z \cdot k_c$	1	$N_z$	JS	JS
K3	Zeitvariante Klassifikationsparameter (A2/A3)	$s_z$	$K$	$N_z$	BA	JA
K4	Zeitvarianter Klassifikator (A2/A3)	$s_z(k)$	$K$	$N_z$	JA	JA
K5	Zeitinvarianter Klassifikator mit der Zeit als Zeitreihe	$s_z + 1$	1	$N_z \cdot K$	JA	JA

Tabelle 2: Anzahl Merkmale, Klassifikatoren und Entwurfsmethoden –  $S_2$ : Merkmalsauswahl,  $S_3$ : Merkmalsaggregation,  $S_4$ : Klassifikation [24], BA: ein oder mehrere bestimmte Abtastpunkte, JS: jedes Segment, JA: jeder Abtastpunkt,  $s_z$ : Anzahl Zeitreihen für die Klassifikation,  $K$ : Anzahl Abtastpunkte pro Zeitreihe,  $k$ : Abtastpunkt,  $k_c$ : Anzahl der Abtastzeitpunkte seit dem Trigger, nach der bei K2 die Entscheidung getroffen wird,  $N_z$ : Anzahl Realisierungen für Zeitreihen,  $N_s$ : Anzahl gewählter Abtastpunkte in BA

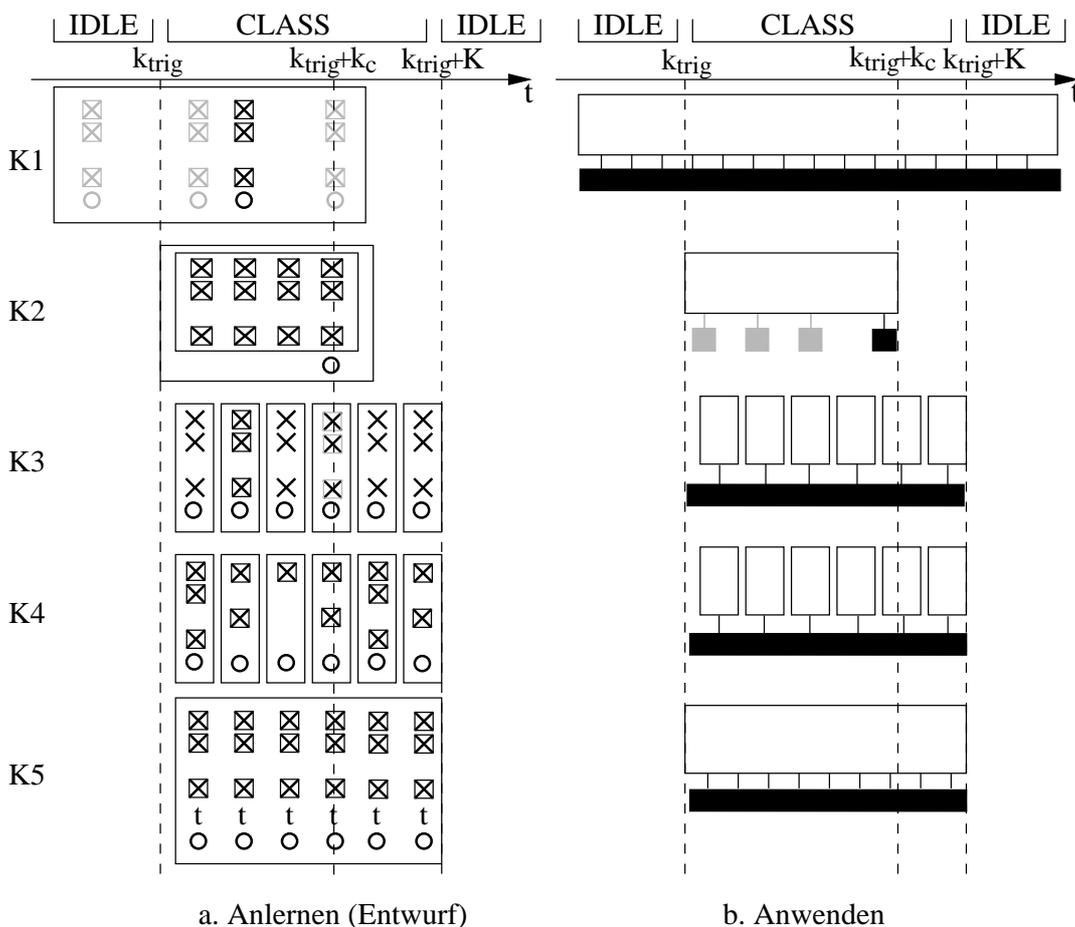


Bild 4: Zeitlicher Ablauf der Merkmalsauswahl, des Anlernens und der Klassifikation,  $\times$ : Merkmale nehmen am Entwurf der Merkmalsauswahl teil,  $\times$ : ausgewählte Merkmale,  $\circ$ : Klassenlabel für den Entwurf,  $t$ : Zeit seit Triggersignal als Merkmal,  $\square$ : Zusammengefasstes Merkmal (nur Anlernen bei K2) bzw. Klassifikatoren (Anlernen und Anwenden),  $\blacksquare$ : Klassifikation, Linien: Verbindung Klassifikator  $\rightarrow$  Klassifikation, grau: optionale Vorgänge

$(N, s_d)$ -dimensionalen Merkmalsatz  $\mathbf{z}$  durchgeführt, mit

$$\mathbf{z} = \mathbf{W}^T \cdot \mathbf{x} + \mathbf{b}. \quad (2)$$

$\mathbf{W}$  und  $\mathbf{b}$  können durch verschiedenen Verfahren bestimmt werden. Bei Verwendung der Diskriminanzanalyse (DA) gilt  $\mathbf{b} = 0$ . Klassifikatoren wie die Support-Vektor-Maschinen nehmen eine eigene Aggregation des Merkmalsatzes vor, so dass bei ihnen auf eine vorherige explizite Merkmalsaggregation verzichtet werden kann (siehe Abschnitt 3.3).

Alle Klassifikatoren außer K1 sind auf ein Triggersignal angewiesen. Für den Entwurf spielt es keine Rolle, ob die Klassifikatoren K2-K5 synchron oder asynchron arbeiten.

### 3 Material und Methoden

#### 3.1 Datensätze

Zur Veranschaulichung der verwendeten Verfahren und der angesprochenen Klassifikatoren wird ein Datensatz künstlich erzeugt, für den Nachweis der praktischen Bedeutung wird ein Datensatz (IIIb) aus der BCI Competition III [28] herangezogen (Daten unter [http://ida.first.fhg.de/projects/bci/competition\\_iii/](http://ida.first.fhg.de/projects/bci/competition_iii/)).

Der künstliche Datensatz beinhaltet  $C = 2$  Klassen,  $N_z = 200$  Datentupel und  $s_z = 4$  Zeitreihen ( $x_1, \dots, x_4$ ) mit jeweils  $K = 60$  Abtastpunkten (Bild 5). Der Trigger Tr3 tritt zum Zeitpunkt  $k_{Tr3} = 12$  auf, Tr4 zum Zeitpunkt  $k_{Tr4} = 60$ . Die Zeitreihen sind in kleine Bereiche geteilt, in denen verschiedene Zeitreihen für die Klassentrennung geeignet sind:  $k = 12 \dots 20$ ,  $k = 21 \dots 32$ ,  $k = 33 \dots 40$ ,  $k = 41 \dots 60$ . Alle Zeitreihen wurden mit einem normalverteilten Rauschen überlagert.

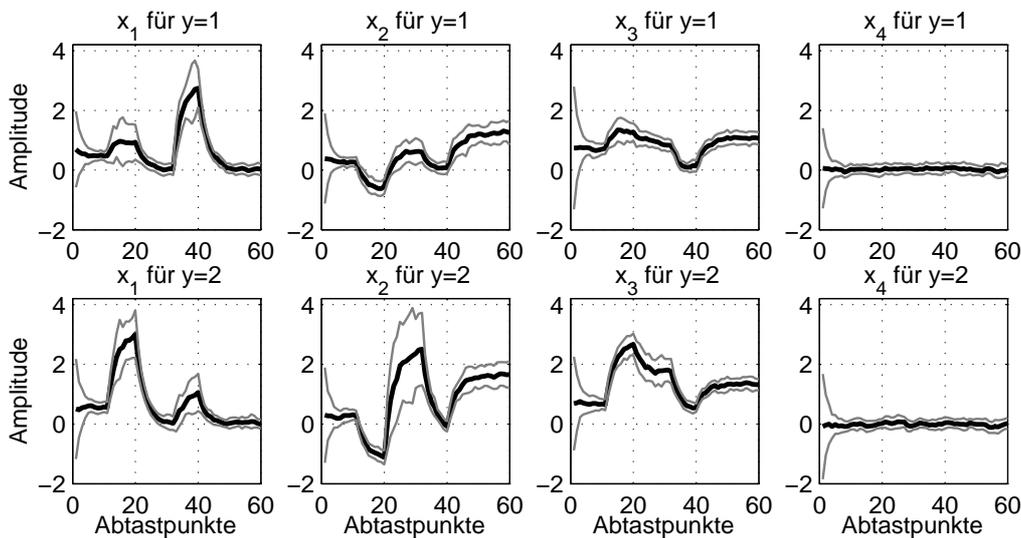


Bild 5: Zeitreihen des künstlichen Datensatzes, über Klassen gemittelt (schwarz), Standardabweichungen eingezeichnet (grau)

Im BCI Datensatz besteht das Klassifikationsproblem in der Unterscheidung einer Bewegungsvorstellung der linken bzw. rechten Hand ( $C = 2$ ). Für eine Beschreibung des Versuchsablaufs siehe [28]. Der Datensatz enthält Aufnahmen von drei Probanden, in

dieser Arbeit wird der Lerndatensatz der Aufnahme *S4b* verwendet. Er enthält  $N_z = 540$  Versuche mit jeweils  $K = 875$  Abtastpunkten ( $k_{Tr3} = 375$ ,  $k_{Tr4} = 875$ ). Neben den Originalzeitreihen (EEG-Aufnahme der Kanäle C3 und C4, Abtastrate: 125 Hz) wurden diverse Frequenzbänder und Verhältnisse von gleichen Frequenzbändern in verschiedenen Kanälen berechnet ( $s_z = 53$ ). Die Frequenzbänder wurden mit einem Butterworth-Filter 5. Ordnung bestimmt, gleichgerichtet und tiefpassgefiltert.

### 3.2 Generierung eines einheitlichen Merkmalsatzes

Die Klassifikatoren K1-K3 und K5 verwenden einen einheitlichen Merkmalsatz für alle Abtastpunkte und lernen über einen oder mehrere Abtastpunkte an. Eine Frage ist, wie der Merkmalsatz und die Abtastpunkte (BA) bestimmt werden. Bei Zeitreihen sind verschiedene Varianten denkbar:

1. Die maximale Merkmalsrelevanz über der Zeit liefert sowohl den Merkmalsatz als auch den Abtastpunkt zum Anlernen ( $k_{best}$ ), wobei  $M_{\mathcal{I}}[k]$  die Merkmalsrelevanz der Merkmale in  $\mathcal{I}$  zum Zeitpunkt  $k$  beschreibt:

$$k_{best}, \mathcal{I}_{best} = \operatorname{argmax}_{k, \mathcal{I}} M_{\mathcal{I}}[k]. \quad (3)$$

2. Für alle möglichen Merkmalskombinationen werden die Merkmalsrelevanzen abhängig von der Zeit bestimmt. Alle Merkmalskombinationen, die einen Wert von  $M_{krit}$  für den durchschnittlichen oder minimalen Wert unterschreiten, werden ausgeschlossen. Somit wird (3) durch eine der beiden folgenden Nebenbedingungen ergänzt:

$$\sum_k M_{\mathcal{I}}[k] > M_{krit} \quad (4)$$

$$\min_k M_{\mathcal{I}}[k] > M_{krit}. \quad (5)$$

3. Für alle möglichen Merkmalskombinationen werden die Merkmalsrelevanzen abhängig von der Zeit bestimmt und der Mittelwert gebildet. Die beste Kombination wird verwendet. Der Abtastpunkt ist der Zeitpunkt mit maximalem Wert für diese Kombination:

$$\mathcal{I}_{best} = \operatorname{argmax}_{\mathcal{I}} \sum_k M_{\mathcal{I}}[k], \quad k_{best} = \max_k M_{\mathcal{I}_{best}}[k]. \quad (6)$$

Der erste Punkt hat den Nachteil, dass die Merkmale u.U. nur in einem kleinen Bereich um den gewählten Abtastpunkt gute Klassifikationsgüten ermöglichen. Je mehr sich die Signale mit der Zeit ändern, desto schlechter wird das Klassifikationsergebnis. Das Anlernen über mehrere ( $k$ ) Abtastpunkte eines Merkmalsatzes kann den Durchschnitt erhöhen, führt aber eventuell zu einer Reduktion der maximalen Klassifikationsgüte.

Nachteil des zweiten Punktes mit der Nebenbedingung (4) ist, dass es Zeitabschnitte geben kann, in denen keine Zeitreihe eine Trennung von Klassen ermöglicht. Dann bleibt kein Merkmal für die Klassifikation übrig. Durch Einführung eines Schwellwertes für die Anzahl an Abtastpunkten, die diese minimale Merkmalsrelevanz unterschreiten dürfen, kann dieses Vorgehen abgeschwächt werden. Etwa gleichbedeutend damit ist der dritte Punkt, die Verwendung der mittleren Merkmalsrelevanz über der Zeit für verschiedene Merkmalskombinationen. Bei den letzten beiden Punkten ergibt sich außerdem das Problem der kombinatorischen Explosion. Dieser Beitrag beschränkt sich auf die Verwendung der maximalen Merkmalsrelevanz über der Zeit, der zumindest zu einem Zeitpunkt gute Klassifikationsgüten garantiert.

### 3.3 Verfahren zur Klassifikation

Für die Klassifikatoren werden verschiedene Verfahren eingesetzt, ein Bayes-Klassifikator, eine Support Vektor Maschine (SVM) und ein künstliches neuronales Netz vom Typ Multilayer-Perceptron (MLP). Im Allgemeinen wird bei der Verwendung des Bayes-Klassifikators die a-posteriori-Wahrscheinlichkeit für eine Klassifikation geschätzt. Mit Annahme einer Normalverteilung und identischen a-priori-Wahrscheinlichkeiten gilt:

$$\hat{y} = \underset{c}{\operatorname{argmax}} (\hat{p}(\mathbf{z}|y = c)) = \underset{c}{\operatorname{argmax}} \left( \frac{1}{(2\pi)^{\frac{s_d}{2}} \sqrt{|\boldsymbol{\Sigma}_c|}} e^{-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1}(\mathbf{z}-\boldsymbol{\mu}_c)} \right), \quad (7)$$

wobei  $c$  die Klassen bezeichnet,  $\mathbf{z}$  die beobachteten aggregierten Merkmale,  $y$  die aktuelle Klasse,  $\boldsymbol{\mu}_c$  den klassenspezifischen Vektor der Erwartungswerte der Merkmale und  $\boldsymbol{\Sigma}_c$  die klassenspezifische Kovarianzmatrix.

Eine Support Vektor Maschine erzeugt eine optimale Trennebene zur Separation zweier Klassen. Dies entspricht einer Lineartransformation des Merkmalsatzes auf einen skalaren Wert, die Klassen werden stets durch  $-1$  und  $1$  kodiert:

$$z = \mathbf{x}^T \cdot \mathbf{w} + b, \quad \hat{y} = \begin{cases} 1, & \text{falls } z \geq 0 \\ -1, & \text{falls } z < 0. \end{cases} \quad (8)$$

Mehrklassenprobleme mit  $C > 2$  werden durch die Zusammenfassung der Ergebnisse mehrerer Zwei-Klassen-Probleme gelöst. Mit einer Heuristik (wählbarer Faktor  $\alpha_p > 0$ ) kann eine geschätzte Wahrscheinlichkeit für die erste Klasse durch

$$\hat{p}_1 = \begin{cases} \frac{1}{2} \cdot \left( 1 + e^{-\alpha_p \cdot \frac{1}{z}} \right) & \text{für } z \geq 0 \\ \frac{1}{2} \cdot \left( 1 - e^{\alpha_p \cdot \frac{1}{z}} \right) & \text{für } z < 0 \end{cases} \quad (9)$$

angegeben werden. Da die SVM den Merkmalsatz selbst aggregiert, wird auf die Diskriminanzanalyse verzichtet.

Als Nebenbedingung gilt  $y_n(\mathbf{x}_n^T \cdot \mathbf{w} + b) - 1 \geq 0$ , für alle  $n = 1, \dots, N$ . Für Fälle, bei denen keine fehlerfreie Trennfläche existiert, werden für einzelne Klassifikationsfehler Korrekturwerte  $\xi_n > 0$  in die Nebenbedingung eingeführt:

$$y_n(\mathbf{x}_n^T \cdot \mathbf{w} + b) - 1 + \xi_n \geq 0, \text{ für alle } n = 1, \dots, N. \quad (10)$$

In Lagrange-Formulierung ergibt sich

$$Q_L = \frac{1}{2} \|\mathbf{w}\|^2 + D \sum_{n=1}^N \xi_n - \sum_{n=1}^N \lambda_n (y_n(\mathbf{x}_n^T \cdot \mathbf{w} + b) - 1 + \xi_n) - \sum_{n=1}^N \mu_n \xi_n, \quad (11)$$

mit  $D$ : Wichtungsfaktor für auftretende Klassifikationsfehler,  $\lambda_n$ : Lagrange-Multiplikatoren für die Nebenbedingungen (10),  $\mu_n$ : Lagrange-Multiplikatoren für die Ungleichungsbedingung  $\xi_n \geq 0$ . Offensichtlich beeinflussen nur Datentupel mit nicht verschwindenden Lagrange-Multiplikatoren das Ergebnis. Sie werden als *Support-Vektoren* bezeichnet:  $\lambda_{\text{sup}} = \{\lambda_n | \lambda_n > 0\}$ . Für  $z$  folgt damit:

$$z = \mathbf{x}^T \mathbf{w} + b, \quad \mathbf{w} = \sum_{n=1}^N \lambda_n y_n \mathbf{x}_n, \quad b = \frac{1}{\mathbf{x}_n^T \mathbf{w}}, \forall n | 0 < \lambda_n < C. \quad (12)$$

Der beschriebene Zusammenhang kann durch die Verwendung von nichtlinearen Kernoperationen verallgemeinert werden [29]. Für die Implementierung siehe [30].

Durch ein künstliches neuronales Netz wird das Verhalten realer Netze von Nervenzellen in stark vereinfachter Form nachgebildet. Beim Multilayer-Perceptron wird das Netz in  $H$  Schichten unterteilt, wobei jede Schicht  $j$  aus  $M_j$  künstlichen Neuronen besteht, die jeweils mehrere Eingänge und einen Ausgang besitzen. Die Eingabe ist für jedes Neuron identisch. Der Ausgang wird als Eingang für die nächste Schicht verwendet. Für die Ausgabe eines Neurons  $i$  einer Schicht  $j$  gilt:

$$z_{j,i} = f_j \left( \mathbf{a}_{j,i}^T \mathbf{z}_{j-1} + a_{j,i,0} \right), \quad (13)$$

mit  $f_j$ : Aktivierungsfunktion der Neuronen in Schicht  $j$ ,  $\mathbf{a}_{j,i}$ : Gewichtsvektor für Skalarprodukt der Eingangsgrößen in Schicht  $j$  und Neuron  $i$  und  $a_{j,i,0}$ : konstanter Summand für Neuron  $i$  in Schicht  $j$ . Dieser Summand wird häufig durch eine Erweiterung der Eingabe realisiert:  $\tilde{\mathbf{z}}_j^T = (1, \mathbf{z}_j)^T$ ,  $a_{j,i,0} = 0$ . Weiter gilt:  $\mathbf{z}_0 = \mathbf{x}$ . In dieser Arbeit wird pro Klasse ein Ausgangsneuron verwendet und als Entscheidung die Klasse mit maximalen Ausgangsneuron gewählt. Bei der Anwendung des MLPs wird ebenfalls auf eine Diskriminanzanalyse verzichtet. Ein Algorithmus zur Optimierung ist z.B. der Backpropagation-Algorithmus [31]. Verwendet wurde die Implementierung aus der MATLAB Neural Networks Toolbox [32].

### 3.4 Verwendete Parameter

Für die Auswertung des künstlichen Datensatzes werden für alle Klassifikatoren über MANOVA die besten  $s_m = 2$  Merkmale ausgewählt und bei den Bayes-Klassifikatoren durch eine Diskriminanzanalyse auf  $s_d = 1$  Merkmal reduziert. Eine Ausnahme bildet der Klassifikator K3. Die Verwendung der Diskriminanzanalyse führt in diesem speziellen Fall zu einem Verlust an Informationen, daher wurde auf sie verzichtet ( $\mathbf{z} = \mathbf{x}$ ).

Im BCI Datensatz werden über MANOVA die besten  $s_m = 4$  Merkmale ausgewählt und bei Verwendung der Bayes-Klassifikatoren durch eine Diskriminanzanalyse auf  $s_d = 1$  Merkmal reduziert. Auch hier gilt die Ausnahme für K3.

Die MLP-Klassifikatoren verwenden drei Schichten ( $H = 3$ ). Für die Anzahl der Neuronen gilt:  $M_1 = s_m$ ,  $M_2 = 5$ ,  $M_3 = C$ . Die Eingangsschicht leitet die Merkmale an die Zwischenschicht ohne Verarbeitung weiter. Als Aktivierungsfunktion kommt die *tansig*-Funktion ( $f_2$ ) [31] bzw. eine lineare Funktion ( $f_3$ ) zum Einsatz. Als Klassifikationsentscheidung wird die Klasse gewählt, deren Ausgangsneuron maximal ist. Tabelle 3 gibt einen Überblick über die verwendeten Klassifikatoren, ihre Parameter und den zeitlichen Verlauf. Eine Variante für K2 besteht darin, den Aggregationsvektor  $\mathbf{w}$  zu einem bestimmten Zeitpunkt zu ermitteln und für alle Zeitpunkte als aggregiertes Merkmal  $z$  anzuwenden (K2 DA, Bayes):

$$\hat{y}[k_c] = \underset{c}{\operatorname{argmin}} \left( \sum_{k_s=1}^{k_c} (z[k_s + k_{\text{trig}}] - \mu_c[k_s])^2 \right). \quad (14)$$

Hierbei ergibt sich ein einfacher Abstandsklassifikator, der eine Ausgabe von Zwischenergebnissen ermöglicht. Somit ist die Anwendung von SVM oder MLP nicht sinnvoll.

	Verfahren	Parameter	Zeitpunkt Merkmalsauswahl	Zeitpunkt Anlernen
K1	DA, Bayes	$\Sigma_j(k_{\text{best}}), \mu_j(k_{\text{best}})$	$k_{\text{best}}$	$k_{\text{best}}$
K1	SVM	$\mathbf{w}(k_{\text{best}}), b(k_{\text{best}})$	$k_{\text{best}}$	$k_{\text{best}}$
K1	MLP	$\mathbf{A}(k_{\text{best}})$	$k_{\text{best}}$	$k_{\text{best}}$
K2	DA, Bayes	$\Sigma_j = \sigma^2 I, \mu_j(k)$	$k_{\text{best}}$	$k$
K2	K4, SVM	$\mathbf{w}(k), b(k), \alpha_{K4}[k], \alpha_p$	$k$	$k$
K3	Bayes	$\Sigma_j(k), \mu_j(k)$	$k_{\text{best}}$	$k$
K3	SVM	$\mathbf{w}(k), b(k)$	$k_{\text{best}}$	$k$
K3	MLP	$\mathbf{A}(k)$	$k_{\text{best}}$	$k$
K4	DA, Bayes	$\Sigma_j(k), \mu_j(k)$	$k$	$k$
K4	SVM	$\mathbf{w}(k), b(k)$	$k$	$k$
K4	MLP	$\mathbf{A}(k)$	$k$	$k$

Tabelle 3: Überblick über die verwendeten Klassifikatoren, ihre Parameter und die Zeitpunkte der Merkmalsauswahl und des Anlernens (siehe auch Bild 4).  $k_{\text{best}}$ : ein oder mehrere bestimmte Abtastpunkte (BA),  $k$ : jeder Abtastpunkt (JA), DA: Diskriminanzanalyse

Eine weitere Alternative ist das Vorschalten eines K4-Klassifikators und die Weiterverarbeitung der Ergebnisse (geschätzte Wahrscheinlichkeiten der Klasse  $\hat{p}_{c,K4}$ ) als K2:

$$\hat{y}[k_c] = \underset{c}{\operatorname{argmax}} \frac{\sum_{k_s=1}^{k_c} \alpha_{K4}[k_s] \cdot \hat{p}_{c,K4}[k_s + k_{\text{trig}}]}{\sum_{k_s=1}^{k_c} \alpha_{K4}[k_s]}, \quad \alpha_{K4}[k] = \max \left( 0, 1 - \frac{F_{K4}[k]}{1 - \frac{1}{C}} \right). \quad (15)$$

Der Wichtungsfaktor  $\alpha_{K4}[k]$  bewertet die Güte des Klassifikators K4 über dem Lern Datensatz und wählt einen Wert von Eins bei einer fehlerlosen (Klassifikationsfehler  $F_{K4}[k] = 0$ ) und Null bei einer zufälligen Klassifikation ( $F_{K4}[k] = 1 - 1/C$ ) aus. In dieser Arbeit wird  $\hat{p}_{c,K4}$  mit der Heuristik aus (9) berechnet.

## 4 Ergebnisse

Die in diesem Abschnitt präsentierten Ergebnisse sind über eine 10-fache Crossvalidierung entstanden (Tabelle 4). In den Bildern 6 und 7 sind die Klassifikationsfehler der Klassifikatoren K1-K4 abgebildet, wobei für „K2 DA, Bayes“ und „K2 K4, SVM“ sämtliche Zwischenergebnisse eingezeichnet sind. Die Mittelwerte der Klassifikatoren wurden jeweils von  $k_{Tr3}$  bis  $k_{Tr4}$  gebildet ( $k = [12 \dots 60]$  im künstlichen Datensatz bzw.  $k = [375 \dots 875]$  im BCI Datensatz).

Beim künstlichen Datensatz ist gut zu erkennen, dass Klassifikator K1 nur in einem kleinen Bereich um seinen Anlernpunkt ( $k_{\text{best}} = 17$ ) gute Ergebnisse liefert. Im weiteren Verlauf wird sogar ein Fehler über 50% erreicht, da sich die klassenabhängigen Mittelwerte der Zeitreihen ändern (Bild 5). Durch die zeitabhängige Parameterbestimmung im Klassifikator K3 kann dieses Problem beseitigt werden. Im gesamten Verlauf erreicht der Klassifikator bessere Ergebnisse. Da K3 einen Merkmalsatz für alle Abtastpunkte verwendet, erreicht der Klassifikator K4 in bestimmten Zeitabschnitten bessere Ergebnisse.

Klassifikator	Fehler im Datensatz [%]					
	Minimum (Abtastpunkt)			Mittelwert ab Trigger		
	KDS (CV)	BCI (CV)	BCI Test	KDS (CV)	BCI (CV)	BCI Test
K1 DA, Bayes	4.4 (18)	20.0 (451)	22.6 (452)	41.9	52.1	52.3
K1 SVM	3.9 (18)	19.8 (452)	23.0 (452)	41.9	52.7	52.6
K1 MLP	5.0 (20)	19.6 (450)	22.6 (450)	41.7	51.1	50.8
K2 DA, Bayes	0.5 (30)	15.7 (778)	17.8 (753)	3.0	21.2	23.0
K2 K4, SVM	0.5 (26)	14.2 (819)	12.0 (846)	2.9	21.8	21.4
K3 Bayes	3.9 (18)	21.3 (455)	25.4 (545)	22.7	38.1	37.7
K3 SVM	4.4 (19)	19.1 (451)	21.5 (454)	22.7	36.5	36.3
K3 MLP	4.4 (18)	18.7 (441)	21.1 (452)	24.2	37.1	36.6
K4 DA, Bayes	4.5 (18)	19.8 (452)	22.4 (454)	21.3	31.1	30.6
K4 SVM	5.0 (18)	18.9 (455)	21.7 (455)	21.0	31.0	30.5
K4 MLP	4.5 (17)	20.3 (442)	20.7 (454)	21.9	31.5	31.3
Ergebnis BCI Competition	–	–	11.5-46.2	–	–	–

Tabelle 4: Minimaler und mittlerer Fehler der Klassifikatoren über eine 10-fache Crossvalidierung (CV). KDS: künstlicher Datensatz, BCI: BCI Datensatz, BCI Test: Testdaten des BCI Datensatzes

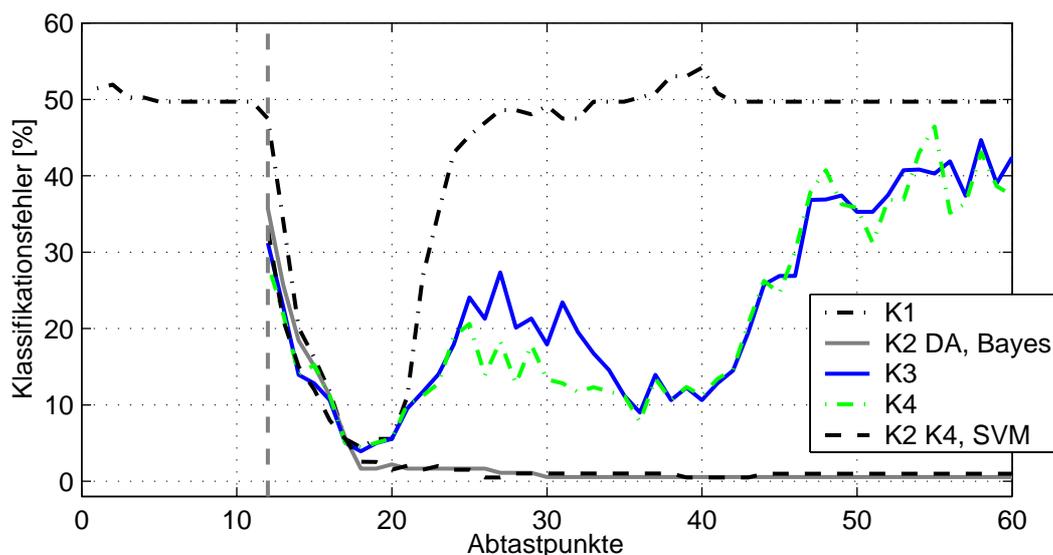


Bild 6: Mittlerer Klassifikationsfehler  $F[k]$  ausgewählter Klassifikatoren nach einer 10-fachen Crossvalidierung im künstlichen Datensatz

K4 bestimmt für jeden Abtastpunkt einen Merkmalsatz, ein Nachteil ist der entstehende immense Speicher- und Rechenaufwand.

Beim Klassifikator K2 erhöht sich die Sicherheit des Klassifikators tendenziell mit zunehmender Anzahl der Abtastpunkte  $k_c$ . Je nach Abtastrate und Parametern führt das zu erheblichen Speicheranforderungen. Die Rechenzeit stellt kein Problem dar, bereits berechnete Summen werden weiter verwendet. Innerhalb von Abschnitten, in denen keine trennenden Merkmale vorhanden sind, weist dieser Klassifikator nur schwach steigende Klassifikationsfehler auf. K2 erreicht das beste Ergebnis und bereits zu einem frühen

Zeitpunkt gute Klassifikationsgüten.

Die minimalen Klassifikationsfehler der Klassifikatoren K1, K3, K4 stimmen fast genau überein. Nur im zeitlichen Verlauf sind Unterschiede festzustellen. Die zeitaggregierten K2-Klassifikatoren erreichen gegenüber den anderen Klassifikatoren einen deutlich besseren minimalen und mittleren Fehler über der Zeit. Das Minimum wird zwar zu einem späten Zeitpunkt erreicht, zum Zeitpunkt der minimalen Fehler der Klassifikatoren K1, K3 und K4 ist der Klassifikator K2 aber nicht schlechter. Auf der anderen Seite ist K2 nur mit erheblichen Verzögerungen in der Lage, auf Klassenwechsel in einem Klassifikationssegment zu reagieren. Dieser Nachteil fällt bei beiden Datensätzen aber nicht auf, da dieser Fall nicht vorkommt. Das für den Klassifikator verwendete Verfahren hat in diesem Datensatz nur einen geringen Einfluss.

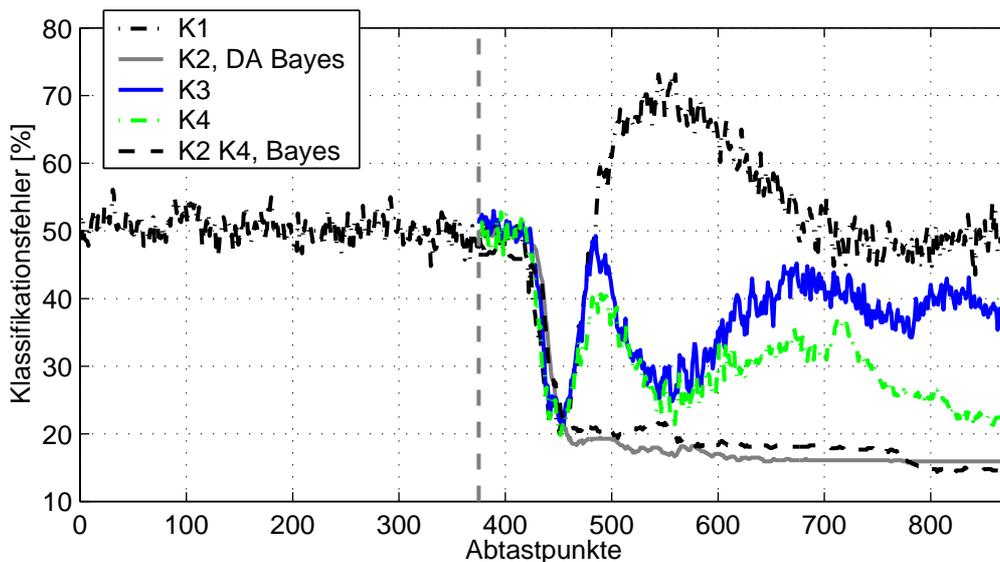


Bild 7: Mittlerer Klassifikationsfehler  $F[k]$  ausgewählter Klassifikatoren nach einer 10-fachen Crossvalidierung im BCI Datensatz, gestrichelte Linie: Triggerzeitpunkt

Auch beim BCI Datensatz ist der Anlernpunkt von K1 deutlich zu erkennen ( $k_{\text{best}} = 438$ ). Hier steigt der Fehler im weiteren Verlauf auf 70%, da keine Merkmale vorhanden sind, die den Fehler kompensieren können. Alle Klassifikatoren erreichen früh nach dem Triggerpunkt (ca. Abtastpunkt 375,  $t \approx 3s$ ) gute und ähnliche Ergebnisse. Dass der Klassifikator K1 zu diesem Zeitpunkt ein gutes Ergebnis erzielt, liegt am Anlernpunkt bei  $k_{\text{best}} = 438$  ( $t \approx 3.5s$ ). Die Zeitverzögerung wird im BCI Datensatz deutlicher (Zeitdifferenz  $> 2.5s$ ). Allerdings ist auch hier K2 zu keinem Zeitpunkt schlechter als ein anderer Klassifikator. Die verschiedenen K2-Klassifikatoren erreichen vergleichbare Ergebnisse, wobei das Minimum von „K2 K4, SVM“ kleiner ist. Im zeitlichen Verlauf ist dieser Klassifikator meist über dem einfacheren „K2 DA, Bayes“. Dies kann durch das unterschiedliche Design der Klassifikatoren erklärt werden. „K2 DA, Bayes“ verwendet Merkmale, deren Klassenmittelwerte zu späten Abtastpunkten nahezu identisch sind. Die Entscheidung (maximale quadratische Abweichung zu einem Klassenmittelwert) wird durch diese Merkmale fast nicht mehr beeinflusst. Demgegenüber aggregiert „K2 K4, SVM“ einzelne Klassifikationsentscheidungen eines K4-Klassifikators. Bereiche, die für Klassifikationsfehler anfällig sind, werden zwar schwach gewichtet, sie wirken sich aber dennoch auf die gesamte Entscheidung aus.

Die Klassifikatoren K1-K4 wurden außerdem auf den Testdaten der BCI Competition III des verwendeten Lerndatensatzes angewendet. Der in Abschnitt 3.1 beschriebenen Lern-datensatz wurde für das Anlernen des Klassifikators verwendet. Die minimalen Klassifikationsergebnisse lagen bei der BCI Competition III zwischen 11.5% und 46.2%. Die mittleren Fehler sind nicht veröffentlicht worden. Der Klassifikator „K2 K4, SVM“ erreicht mit 12.0% ein mit dem Gewinner vergleichbares Ergebnis.

## 5 Zusammenfassung und Ausblick

In diesem Beitrag wurde eine systematische Kategorisierung von Klassifikatoren für zeit-variante Probleme vorgenommen und die Bedingungen für einen sinnvollen Einsatz beschrieben. Anhand der Kategorisierung wurden einige Klassifikatoren eingeführt, deren Verfahren dem Problem angepasst werden können. Für ausgewählte Verfahren, der MANOVA zur Merkmalsauswahl, der Diskriminanzanalyse zur Merkmalsaggregation und dreier Klassifikationsverfahren, wurden die eingeführten Klassifikatoren implementiert. Die Klassifikatoren wurden auf zwei Datensätze angewendet und die erreichten Güten verglichen. Die zeitaggregierten Klassifikatoren konnten sich, was den minimalen Klassifikationsfehler angeht, deutlich von den restlichen Klassifikatoren absetzen. Ansonsten erreichten die Klassifikatoren vergleichbare Klassifikationsgüten, der zeitliche Verlauf variierte. Das führte zumindest im BCI Datensatz zu deutlichen Unterschieden in den mittleren Klassifikationsfehlern. Der zeitaggregierte Klassifikator „K2 K4, SVM“ erreicht Güten, die vergleichbar mit dem besten Klassifikator aus der BCI Competition III sind.

Die vorgestellte Kategorisierung legt nahe, durch einen Klassifikator (z.B. K1) Trigger-signale zu erkennen und anschließend mit Hilfe zeitabhängiger Klassifikatoren (z.B. K2) gute und robuste Ergebnisse zu produzieren. Auf diese Weise kann der Vorteil der syn-chronen, zeitabhängigen Klassifikatoren in asynchronen Systemen verwendet werden.

## Literatur

- [1] Wolpaw, J. R.; Birbaumer, N.; McFarland, D. J.; Pfurtscheller, G.; Vaughan, T. M.: Brain-Computer Interfaces for communication and control. *Clinical Neurophysiology* (2002) 113, S. 767–791.
- [2] Roberts, S. J.; Rezek, I. A.; Penny, W. D.; Everson, R. M.: The use of advanced information processing methods in EEG analysis. In: *Proc., IEE Colloquium on Intelligent Decision Support in Clinical Praxis*. London. 1998.
- [3] Leuthardt, E. C.; Schalk, G.; Wolpaw, J. R.; Ojemann, J. G.; Moran, D. W.: A brain-computer interface using electrocorticographic signals in humans. *Journal of Neural Engineering* 1 (2004), S. 63–71.
- [4] Carmena, J. M. e. a.: Learning to control a Brain-Machine Interface for reaching and grasping by primates. *PLoS Biology* 1(2) (2003), S. 193–208.
- [5] 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.
- [6] Wolpaw, J. R.; McFarland, D. J.: Control of a two-dimensional movement signal by a noninvasive Brain-Computer Interface in humans. *Proc., National Academy of Sciences* 101(51) (2004), S. 17849–17854.
- [7] Taylor, D. M.; Tillery, S. I. H.; Schwartz, A. B.: Information conveyed through Brain-Control: Cursor versus robot. *IEEE Transactions on Neural System and Rehabilitation Engineering* 11(2) (2003), S. 195–199.
- [8] Wolpaw, J. R.; McFarland, D. J.; Vaughan, T. M.; Schalk, G.: The Wadsworth Center Brain-Computer Interface (BCI) research and development program. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11(2) (2003), S. 204–207.

- [9] Rossdeutscher, W.: Steuerungsmöglichkeiten in der Armprothetik. *Orthopädie-Technik* 10(51) (2000), S. 865–868.
- [10] Herberts, P.; Almstroem, C.; Caine, K.: Clinical application study of multifunctional prosthetic hands. *Journal of Bone and Joint Surgery* 60-B (4) (1978), S. 552–560.
- [11] Reischl, M.; Mikut, R.; Schlögl, A.: Comparison of fuzzy and statistical classifiers for Brain-Computer Interfaces. *Biomedizinische Technik* 49(E1) (2004), S. 762–763.
- [12] Müller-Putz, G. R.; Scherer, R.; Pfurtscheller, G.; Rupp, R.: EEG-Based Neuroprosthesis Control: A Step Towards Clinical Practice. *Neuroscience Letters* 382(1-2) (2005), S. 169–174.
- [13] Tanaka, K.; Matsunaga, K.; Wang, H. O.: Electroencephalogram-based control of an electric wheelchair. *Accepted Paper, IEEE Transactions on Robotics* (2005).
- [14] Mason, S. G.; Birch, G. E.: A general framework for Brain-Computer Interface design. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11(1) (2003), S. 70–85.
- [15] Giacinto, G.; Roli, F.: A theoretical framework for dynamic classifier selection. In: *Proc., 15th International Conference on Pattern Recognition*, S. 8–11. Barcelona, Spain. 2000.
- [16] Kohlmorgen, J.; Blankertz, B.: Bayesian classification of single-trial event-related potentials in EEG. *International Journal of Bifurcations and Chaos* 14(2) (2004), S. 719–726.
- [17] Mason, S. G.; Birch, G. E.: A brain-controlled switch for asynchronous control applications. *IEEE Transactions on Biomedical Engineering* 47(10) (2000), S. 1297–1307.
- [18] Pfurtscheller, G.; Müller, G. R.; Pfurtscheller, J.; Gerner, H. J.; Rupp, R.: 'Thought' -control of functional electrical stimulation to restore hand grasp in a patient with tetraplegia. *Neuroscience Letters* 351(1) (2003), S. 33–36.
- [19] Reischl, M.; Mikut, R.: Auswertung EEG-Benchmark-Datensatz Brain-Machine-Interface. Techn. Ber., Forschungszentrum Karlsruhe, Institut für Angewandte Informatik. 2003.
- [20] Blankertz, B.; Curio, G.; Müller, K.-R.: Classifying single trial EEG: Towards Brain Computer Interfacing. In: *Proc., Advances in Neural Information Processing Systems*, Bd. 14, S. 157–164. 2001.
- [21] Reischl, M.; Mikut, R.; Pylatiuk, C.; Schulz, S.; Beck, S.; Bretthauer, G.: Steuerungs- und Signalverarbeitungskonzepte für eine multifunktionale Handprothese. *Automatisierungstechnik* 50(6) (2002), S. 279–286.
- [22] Scherer, R.; Müller, G. R.; Neuper, C.; Graimann, B.; Pfurtscheller, G.: An asynchronously controlled EEG-based virtual keyboard: Improvement of the spelling rate. *IEEE Transactions on Biomedical Engineering* 51(6) (2004), S. 979–984.
- [23] Townsend, G.; Graimann, B.; Pfurtscheller, G.: Continuous EEG classification during motor imagery – simulation of an asynchronous BCI. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 12(2) (2004), S. 258–265.
- [24] Reischl, M.; Mikut, R.: Validierung hierarchischer Klassifikatoren für Mehrklassenprobleme. In: *Proc., 14. Workshop Fuzzy-Systeme und Computational Intelligence*, S. 175–188. Karlsruhe: Universitätsverlag Karlsruhe. 2004.
- [25] Jain, A. K.; Duin, R. P. W.; Mao, J.: Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(1) (2000), S. 4–36.
- [26] Tatsuoka, M. M.: *Multivariate Analysis*. New York: Macmillan. 1988.
- [27] Shannon, C. E.: A mathematical theory of communication. *The Bell System Technical Journal* 27 (1948), S. 379–423.
- [28] Vidaurre, C.; Schlögl, A.; Cabeza, R.; Pfurtscheller, G.: A fully on-line adaptive Brain Computer Interface. *Biomedizinische Technik* 49 (2004), S. 760–761.
- [29] Burges, C.: A tutorial on support vector machines for pattern recognition. *Knowledge Discovery and Data Mining* 2(2) (1998), S. 121–167.
- [30] Canu, S.; Grandvalet, Y.; Rakotomamonjy, A.: SVM and kernel methods MATLAB toolbox. Perception Systèmes et Information, INSA de Rouen, Rouen, France. 2003.
- [31] Schürmann, J.: *Pattern classification*. New York: John Wiley & Sons. 1996.
- [32] The MathWorks: *Neural Network Toolbox for use with MATLAB*. 1998.

# SPAM-Abwehr auf Basis von Computational Intelligence (KNN und Bayes-Filer)

Volker Gördes, Katharina Henning, Martin Hühne\*, Ulrich Lehmann\*, Marco Pack  
(\*Fachhochschule Südwestfalen Iserlohn; [www-in.fh-swf.de](http://www-in.fh-swf.de))

## Abstract

Spam-Mails, frei übersetzt: unerwünschte Werbe-E-Mails, sind ein grundlegendes Problem für das Internet. Laut den Umfragen der Radicati Group geben bereits 94% der Anwender an, dass Spam für sie ein großes Problem ist.<sup>1</sup>

Aber nicht nur für die Anwender stellt Spam ein großes Problem dar.

Abbildung 1 zeigt die Entwicklung der zusätzlichen Kosten durch Spam auf. Gemessen an einem Unternehmen mit 10.000 Mitarbeitern. Die Kosten ergeben sich aus weiteren Servern, die für die Bearbeitung der Spam-Mails benötigt werden.

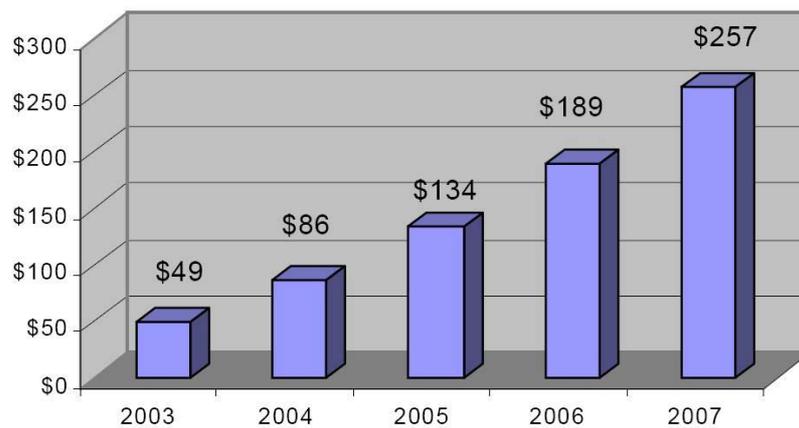


Abbildung 1 / Kosten von Spam/Jahr<sup>2</sup>

Im Vortrag werden die Architektur und die Arbeitsweise eines neuen Spam-Filters auf Basis von Methoden der Computational Intelligence (CI) vorgestellt. Zum Einsatz kommen Künstliche Neuronale Netze in Kombination mit Bayes-Filtern. Der Spam-Filter lernt mit der Zeit dazu. Der Filter kann sowohl einzeln (Stand-Alone-Server) als auch in Kombination mit bestehenden kommerziellen Spam-Filtern betrieben werden.

Für die Entwicklung des Filters wurden verschiedene CI-Methoden für ihren Einsatz bei der Spam-Abwehr bewertet, ausgewählt und optimiert. In der vorgestellten Lösung kommt ein Bayes-Filter für „Spam-Wörter“ in gewichteter Kombination mit zwei KNN für statistische Informationen und Spam-

---

<sup>1</sup> Antispam January 2004 – Radicati Group, Kalifornien Palo Alto

<sup>2</sup> Kosten Spam – Radicati Group, Kalifornien Palo Alto

Wörter sowie eine Whitelist zum Einsatz. Ein Feedback-Verfahren in Form einer geschlossenen Schleife regelt die Verbesserung der Filterwirkung im laufenden Betrieb.

Das Projekt wurde in Kooperation mit der mittelständischen Industrie, dem Arbeitskreis für IT-Sicherheit SISE und der Kompetenzplattform Computer Vision based on Computational Intelligence an der Fachhochschule Südwestfalen in Iserlohn bearbeitet. Siehe auch:

<http://www3.fh-swf.de/fbin/huehne/forschung.htm>

[http://www3.fh-swf.de/forschung/kopf\\_cv-ci/kopv\\_cv-ci.htm](http://www3.fh-swf.de/forschung/kopf_cv-ci/kopv_cv-ci.htm)

# On fuzzy descriptor systems and nonlinear model following control of two-link robot manipulators

Horst Schulte

Laboratory of Control Engineering and System Dynamics,  
Department of Mechanical Engineering, University Kassel

Tel:++49 561 8043248

Fax:++49 561 8047768

E-Mail: schulte@ieee.org

This paper presents the first steps towards a robust controller design for two-link manipulators using a fuzzy state-feedback gain-scheduling approach based on a Takagi Sugeno fuzzy descriptor form. Due to the circumstances that this form is more similar to a given original nonlinear equation as a Takagi Sugeno fuzzy state-space system, it allows to reduce the conservatism of the controller design by using common matrix structures. The control law here is equivalent to the well-known parallel distributed compensation scheme. The challenge of the investigated control problem in this case is the highly nonlinear dynamics of the dual-actuator drive powered by air-pressure that interacts with the dynamics of the robot manipulator.

## 1 Introduction

For several years Takagi Sugeno (TS) fuzzy systems [12] are widely used in a context of modeling for control design [1, 13, 15]. Also the linear descriptor system, which differs from a state-space representation, has generated a great deal of interest in control system design. The descriptor system describes a wider class of systems including physical models and nondynamic constraints [6]. It is much tighter than the state-space model for representing real independent parametric perturbations and uncertainties.

There exist a large number of papers on the stability analysis of TS fuzzy systems based on the state-space representation and also some applications in the control design process e.g. [3, 5, 9, 10]. But in contrast, the stability analysis of TS fuzzy systems based on descriptor representation as weighted combinations of linear descriptor systems (from now called *TS fuzzy descriptor systems*) have not been investigated until [14] and only once applied in the context of observer design with applications in biomechanics [2]. The purpose of this work is to propose a robust fuzzy controller design based on a first derived descriptor form of a two-link robot manipulator powered by pneumatic actuators as a reasonable example of a highly nonlinear complex system.

This paper is organized as follows: First, for a class of serial manipulators, the so-called SCARA-type robot manipulators [11], a nonlinear state-space model is formulated by the Newton-Euler equation of the manipulator arm coupled with a reduced model of the actuators with two bounded uncertainties. These uncertainties are caused by the influence of the model reduction on the differential equations of the pressure evolution and the time-variable leakage air flows between the actuator chambers.

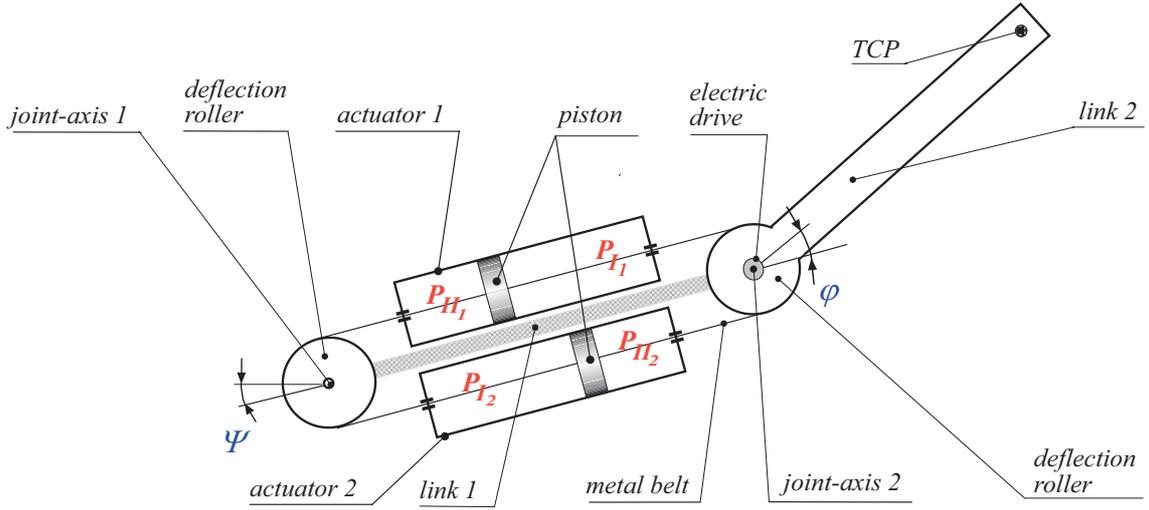


Figure 1: Two-link manipulator with a dual servo-pneumatic actuator as main drive ( $\psi$ ) and an electric direct-drive ( $\varphi$ )

Then the state-space model combined with dynamic constraints is transformed into a Takagi Sugeno fuzzy descriptor form [14]. After this the complete dynamic model based on a combination of the actuator models in descriptor form and the robot arm dynamics in descriptor form.

Finally, a global controller is designed as a robust fuzzy gain-scheduler of state-feedback gains. A systematic design procedure for a fuzzy controller is proposed through the parallel distributed compensation.

## 2 Physical System Modelling

In the following the dynamic model of a SCARA-type robot manipulator is presented. The robot arm is powered by an actuator mechanism based on tangential feed as the main drive and an electric direct-drive as the actuator of the second axis that is shown in Fig. 1. The main drive consists of two rodless cylinders with pistons, two deflection rollers and a metal belt as a friction-locked connection between the rollers and the pistons (also called Belt-motor [8]). The forces on the pistons caused by opposed pressure differences in Actuator 1 and 2 are transmitted by the belt to the deflection rollers, where the linear motion of the pistons is converted to angular motion of the robot link 1. We assume that the normal strain of the belt is negligibly small (infinitely stiff) and the tangent stiffness is pure elastic.

### 2.1 Mechanical model of the manipulator dynamics

Consider the two-link planar arm shown in Fig. 2 that represents a reduced mechanical system of the manipulator in Fig. 1. The robot arm has two vertical revolute joints, both joint axes are orthogonal to the xy-plane in Fig. 2. Here, the masses of the actuator cylinders and the mass of link 1 are combined to  $m_{sp}$ . Let  $m_e$  be the mass of the rotor of the electric drive, and  $m_t$  the mass of the tool and vertical drive in the tool center point (TCP, see Fig. 1), whereby the mass of link 2 is small

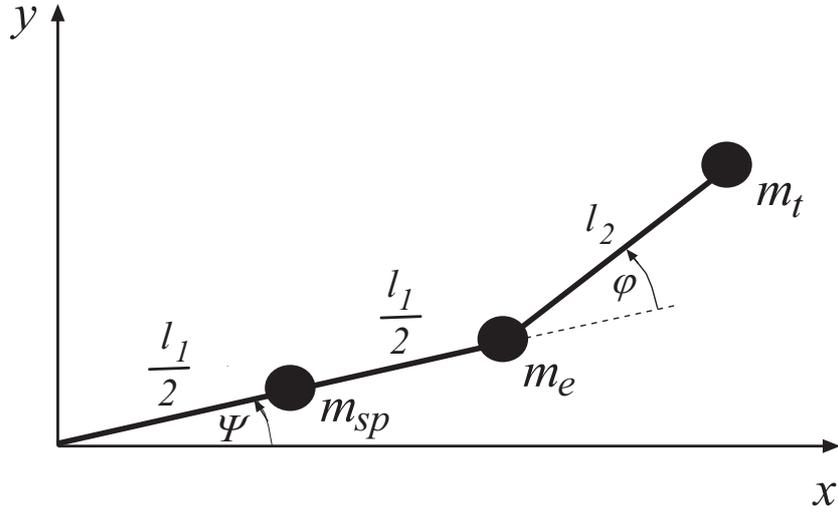


Figure 2: Reduced mechanical system of the robot manipulator arm in the xy-plane.

in comparison to  $m_t$ . The distance of the centers of mass of  $m_{sp}$  and  $m_e$  to the Joint-axis 1 is defined by  $l_1/2$ , and  $l_1$  as the length of link 1. The distance of the center of mass of  $m_t$  to the joint-axis 2 is defined by  $l_2$  as the length of link 2. The equation of motion for this mechanical system can be written in the known matrix form [11] which represents the joint space dynamics model:

$$\mathbf{H}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \boldsymbol{\tau} \quad (1)$$

where  $\mathbf{q} = [\psi \ \varphi]^T$  is the vector of generalized coordinates with  $\psi$  as the angle of link 1 from the x-axis, and with  $\varphi$  as the angle between link 1 and link 2. The inertia matrix

$$\mathbf{H}(\mathbf{q}) = \begin{bmatrix} H_{11}(\mathbf{q}) & H_{12}(\mathbf{q}) \\ H_{21}(\mathbf{q}) & H_{22}(\mathbf{q}) \end{bmatrix} \quad (2)$$

depends on the current arm configuration with

$$H_{11}(\mathbf{q}) = I_{L1} + m_{sp} \left( \frac{l_1}{2} \right)^2 + m_e l_1^2 + I_{L2} + m_t (l_1^2 + l_2^2 + 2 l_1 l_2 \cos \varphi) ,$$

$$H_{12}(\mathbf{q}) = H_{21}(\mathbf{q}) = I_{L2} + m_t l_2 (l_2 + l_1 \cos \varphi) ,$$

$$H_{22} = I_{L2} + m_t l_2^2 .$$

Note that for this application the assumption holds (the links are made of carbon-fiber-reinforced plastic) that the moments of inertias relative to the centers of mass of link 1 and link 2 as  $I_{L1}$  and  $I_{L2}$  can be neglected against the other terms, such as  $(m_e l_1^2)$ . So we set in the following considerations  $I_{L1} = I_{L2} = 0$ . The actuation torques in (2) are represented by  $\boldsymbol{\tau} = [\tau_1 \ \tau_2]^T$ . Finally, the vector

$$\mathbf{h}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} h_1(\mathbf{q}, \dot{\mathbf{q}}) \\ h_2(\mathbf{q}, \dot{\mathbf{q}}) \end{bmatrix} \quad (3)$$

in (1) represents the centrifugal effects and Coriolis effects with

$$h_1(\mathbf{q}, \dot{\mathbf{q}}) = m_t l_1 l_2 \sin \varphi (-\dot{\varphi}^2 - 2 \dot{\psi} \dot{\varphi}) ,$$

$$h_2(\mathbf{q}, \dot{\mathbf{q}}) = m_t l_1 l_2 \sin \varphi \dot{\psi}^2 .$$

## 2.2 Dual-Actuator model

The physical description of the main drive (see Fig.1) based on the continuity and enthalpy balance equation of the gas flow through the actuator and the constitutive relations of the air. Whereby the most important aspect of the actuator dynamics that interact with the robot arm is described by the pressure evolution of each chamber. Assuming isentropic behavior of the gas flow, the pressure evolution in the chambers of actuator  $j, j = 1, 2$  (see Fig.1) is modeled by the equations

$$\dot{p}_{I_j} (V_{0I} + A_K (x_K - x_{K0})) = \kappa [R_g T_I (\dot{m}_2 - \dot{m}_1) - p_I A_K \dot{x}_K] \quad (4a)$$

and

$$\dot{p}_{II_j} (V_{0II} - A_K (x_K - x_{K0})) = \kappa [-R_g T_{II} (\dot{m}_4 - \dot{m}_3) + p_{II} A_K \dot{x}_K] . \quad (4b)$$

The *variables* used in the above equations are:

$p_I, p_{II}$  as the pressure in actuator chambers  $I, II$  [N/m<sup>2</sup>],

$x_K$  as the actuator piston displacement [m],

$\dot{m}_n$  as the mass flow over the servovalve control edges  $n = 1, 2, 3, 4$  [kg/s]  
(see Fig. 3),

$T_I, T_{II}$  as the temperature in the chamber  $I, II$  [K], and

$T_S$  as the temperature associated with the system pressures  $p_S$  and  $p_R$  [K].

The *model parameters* are:

$x_{K0}$  as the start position of the actuator piston [m],

$V_{0I,0II}$  as the start volume of the actuator chamber  $I, II$  [m<sup>3</sup>],

$A_K$  as the actuator piston area [m<sup>2</sup>].

$R_g$  as the gas constant [ $\frac{J}{kg \cdot K}$ ], and

$\kappa$  as the adiabatic exponent of the gas [-].

The two variable volume chambers in each actuator cylinder are connected to a four-way servovalve. The configuration for one actuator cylinder in combination with the servovalve is shown in Fig. 3. In this case the servovalve controls the air-mass-flow  $\dot{m}_1, \dots, \dot{m}_4$  by adjusting four orifice areas between the constant pressure supply  $p_s$ , the exhaust pressure  $p_R$ , and the chamber pressures  $p_{I,II}$ . The four orifice areas are continuously controlled by the servovalve voltage  $u_v$ . The equations for the mass

flows are

$$\dot{m}_1(x_v, p_I) = -\alpha_{D1} A_1(x_v) \psi \left( \frac{p_R}{p_{II}} \right) p_I \sqrt{\frac{2}{R_g T_{II}}} , \quad (5a)$$

$$\dot{m}_2(x_v, p_I) = \alpha_{D2} A_2(x_v) \psi \left( \frac{p_{II}}{p_S} \right) p_S \sqrt{\frac{2}{R_g T_S}} , \quad (5b)$$

$$\dot{m}_3(x_v, p_{II}) = -\alpha_{D3} A_3(x_v) \psi \left( \frac{p_I}{p_S} \right) p_S \sqrt{\frac{2}{R_g T_S}} , \quad (5c)$$

$$\dot{m}_4(x_v, p_{II}) = \alpha_{D4} A_4(x_v) \psi \left( \frac{p_R}{p_I} \right) p_I \sqrt{\frac{2}{R_g T_I}} , \quad (5d)$$

with the servovalve displacement  $x_v$  and the orifice areas of the control edges

$$A_n(x_v) = \begin{cases} x_v \pi d_n & \text{for } x_v \geq x_{\ddot{u}_n} \\ 0 & \text{for } x_v < x_{\ddot{u}_n} \end{cases} \quad \text{for } n = 2, 4 , \quad (6a)$$

$$A_n(x_v) = \begin{cases} x_v \pi d_n & \text{for } x_v \leq -x_{\ddot{u}_n} \\ 0 & \text{for } x_v > -x_{\ddot{u}_n} \end{cases} \quad \text{for } n = 1, 3 , \quad (6b)$$

and the flow function

$$\psi \left( \frac{p_a}{p_b} \right) = \begin{cases} \psi_0 \sqrt{1 - \left( \frac{\frac{p_a}{p_b} - p_{krit}}{1 - p_{krit}} \right)^2} & \text{for } \frac{p_a}{p_b} \geq p_{krit} \\ \psi_0 & \text{for } \frac{p_a}{p_b} < p_{krit} \end{cases} . \quad (6c)$$

The not yet defined model parameters are

$\alpha_{D_n}$  as the flow coefficient of the control edge  $n = 1, \dots, 4$  ,

$d_n$  as the effective diameter of the control edge  $n$  [m],

$R_g$  as the gas constant [ $\frac{J}{kg \cdot K}$ ],

$\kappa$  as the adiabatic exponent of the gas [-],

$p_{krit}$  as the critical pressure ratio [-],

$\psi_0$  as the maximum value of the flow function  $\psi$ , and

$x_{\ddot{u}_n}$  as the valve overlap of the control edge  $n$  [m].

### 3 Reduced physical model in TS fuzzy descriptor form

#### 3.1 Reduced actuator model in descriptor form

The detailed physical description of the servo-pneumatic actuators is reduced by simplification of the mass flow relations (5) and by linearization of the switching functions for zero valve overlap ( $x_{\ddot{u}_n} = 0$  for  $n = 1, \dots, 4$ ). The influence of the model reductions on the differential equation of the chamber pressures  $p_I, p_{II}$  is condensed in two bounded uncertainties as  $\Delta \mathbf{B}$  and  $\Delta \mathbf{a}$ . The remaining nonlinear

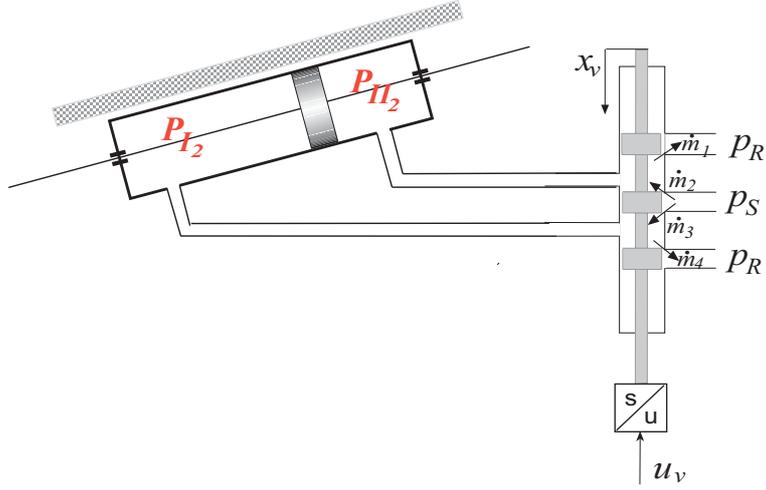


Figure 3: One cylinder of the dual-actuator with the accessory servovalve

terms are transferred into TS fuzzy relations using the lemma of Morère [7]. It leads to the following descriptor form

$$\sum_{j=1}^2 \nu_j(x_K) \mathbf{E}_j^P \dot{\mathbf{x}}^P = \sum_{i=1}^8 h_i(\dot{x}_K, p_I, p_{II}) [\mathbf{A}_i^P \mathbf{x}^P + \mathbf{B}_i^P x_v] + \Delta \mathbf{B} x_v + \Delta \mathbf{a} \quad (7)$$

with

$$\mathbf{x}^P = [p_I, p_{II}]^T, \quad (8a)$$

$$\mathbf{E}_j^P = \begin{bmatrix} \frac{V_{0I} + A_K(*_1 - x_{K0})}{\kappa} & 0 \\ 0 & \frac{V_{0II} - A_K(*_1 - x_{K0})}{\kappa} \end{bmatrix}, \quad *_1 \in \{\underline{x}_K, \bar{x}_K\} \quad (8b)$$

$$\mathbf{A}_i^P = \begin{bmatrix} A_K *_2 & 0 \\ 0 & A_K *_2 \end{bmatrix}, \quad *_2 \in \{\dot{\underline{x}}_K, \dot{\bar{x}}_K\}, \quad (8c)$$

$$\mathbf{B}_i^P = \begin{bmatrix} R_g T_I \alpha_D \pi d (\tilde{a}_{02} + \tilde{b}_{02} *_3) \\ -R_g T_{II} \alpha_D \pi d (\tilde{a}_{03} + \tilde{b}_{03} *_4) \end{bmatrix}, \quad *_3 \in \{\underline{p}_I, \bar{p}_I\}, \quad (8d)$$

$$*_4 \in \{\underline{p}_{II}, \bar{p}_{II}\}, \quad (8e)$$

and the uncertainty terms

$$\Delta \mathbf{a} = \begin{bmatrix} \Delta p_I \\ \Delta p_{II} \end{bmatrix}, \quad (8f)$$

$$\Delta \mathbf{B} = \begin{bmatrix} R_g T_I \alpha_D \pi d \Delta \tilde{a}_2 \\ -R_g T_{II} \alpha_D \pi d \Delta \tilde{a}_3 \end{bmatrix}. \quad (8g)$$

### 3.2 Model of the manipulator dynamics in TS fuzzy descriptor form

The equations (1),(2) can be directly written in the following "exact" TS fuzzy descriptor form

$$\sum_{j=1}^2 \nu_j(\varphi) \mathbf{E}_j^R \dot{\mathbf{x}}^R = \sum_{i=1}^4 h_i(\varphi, \dot{\varphi}, \dot{\psi}) \mathbf{A}_i^R \mathbf{x}^R + \mathbf{B}^R \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (9)$$

with

$$\mathbf{x}^R = \begin{bmatrix} \psi, \varphi, \dot{\psi}, \dot{\varphi} \end{bmatrix}^T, \quad (10a)$$

$$\mathbf{E}_j^R = \begin{bmatrix} \mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \begin{bmatrix} a + 2b *_{5} & c + b *_{5} \\ c + b *_{5} & c \end{bmatrix} \end{bmatrix}, \quad *_{5} \in \{\underline{f}_1, \bar{f}_1\}, \quad (10b)$$

$$(10c)$$

$$\mathbf{A}_i^R = \begin{bmatrix} \mathbf{0}_{2 \times 2} & \mathbf{I}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \begin{bmatrix} 2b *_{6} & b *_{6} \\ -b *_{7} & 0 \end{bmatrix} \end{bmatrix}, \quad *_{6} \in \{\underline{g}_1, \bar{g}_1\}, \quad (10d)$$

$$*_{7} \in \{\underline{g}_2, \bar{g}_2\}, \quad (10e)$$

and

$$\mathbf{B}^R = \begin{bmatrix} \mathbf{0}_{2 \times 2} \\ \mathbf{I}_{2 \times 2} \end{bmatrix}, \quad (10f)$$

whereby

$$\begin{aligned} f_1 &= \cos \varphi, \\ g_1 &= \dot{\varphi} \sin \varphi, \\ g_2 &= \dot{\psi} \sin \varphi, \\ a &:= m_{sp} \left( \frac{l_1}{2} \right)^2 + m_e l_1^2 + m_t (l_1^2 + l_2^2), \\ b &:= m_t l_1 l_2, \\ c &:= m_t l_2^2. \end{aligned}$$

Based on the assumptions that, first, the piston area of the actuators are equal  $A_{K_1} = A_{K_2} =: A_K$ , and second, the mass flow control of each actuator caused by opposed pressure differences  $p_{L_j} = p_{I_j} - p_{II_j}$ ,  $j = 1, 2$  (see Fig. 1) the magnitudes are

$$p_{L_1} = p_{L_2} =: p_L, \quad (11)$$

the driving torque of joint 1 can write as

$$\tau_1 = p_{L_1} A_{K_1} \frac{d_r}{2} + p_{L_2} A_{K_2} \frac{d_r}{2} = p_L A_K d_r, \quad (12)$$

with  $d_r$  as the roller diameter. In this consideration it is assumed that the electric actuator of joint 2, see Fig. 1, behaves as a ideal torque-controlled generator, which gives a proportional relation between the torque  $\tau_2$  and the control voltage  $u_m$  established by the motor constant  $k_m$ .

$$\tau_2 = k_m u_m \quad (13)$$

### 3.3 Complete dynamic model in descriptor form

The complete dynamic model based on an combination of the actuator model in descriptor form (7) and the robot arm dynamics in descriptor form (9). It was derived by using the following algebraic constraints between the translational motion of the actuator piston and the rotational motion of the deflection roller:

$$x_K = \psi \frac{d_r}{2} \quad , \quad \dot{x}_K = \dot{\psi} \frac{d_r}{2} . \quad (14)$$

With the assumption that the piston areas are  $A_K = A_{K_1} = A_{K_2}$  and the mass flow control of each actuator caused by opposed pressure difference in both actuators the chamber pressures are

$$p_I = p_{I_1} = p_{I_2} \quad , \quad p_{II} = p_{II_1} = p_{II_2} \quad ,$$

the driving torque in joint 1 is

$$\tau_1 = (p_{I_1} - p_{II_1}) A_{K_1} \frac{d_r}{2} + (p_{I_2} - p_{II_2}) A_{K_2} \frac{d_r}{2} = (p_I - p_{II}) A_K d_r \quad . \quad (15)$$

So the complete dynamic model can be presented as

$$\sum_{k=1}^4 \tilde{v}_k(\psi, \varphi) \mathbf{E}_k \dot{\mathbf{x}} = \sum_{l=1}^{32} h_l [\mathbf{A}_l \mathbf{x} + (\mathbf{B}_l + \Delta \mathbf{B}) \mathbf{u}] + \Delta \tilde{\mathbf{a}} \quad (16)$$

with  $\mathbf{E}_k \in \mathbb{R}^{6 \times 6}$ ,  $\mathbf{A}_l \in \mathbb{R}^{6 \times 6}$ ,  $\mathbf{B}_l \in \mathbb{R}^{6 \times 2}$ ,  $\Delta \tilde{\mathbf{a}} \in \mathbb{R}^{6 \times 1}$ , the right hand side weighting function

$$h_l = h_l(\varphi, \dot{\varphi}, \psi, \dot{\psi}, p_I, p_{II}) \quad , \quad (17a)$$

the input vector

$$\mathbf{u} = [x_v, \tau_1]^T \quad , \quad (17b)$$

and the state vector

$$\mathbf{x} = [\psi, \varphi, \dot{\psi}, \dot{\varphi}, p_I, p_{II}]^T \quad . \quad (17c)$$

In detail the matrices are

$$\mathbf{E}_1 = \begin{bmatrix} \mathbf{E}_1^R & \mathbf{0}_{4 \times 2} \\ \mathbf{0}_{2 \times 4} & \mathbf{E}_1^P \end{bmatrix} \quad , \quad \mathbf{E}_2 = \begin{bmatrix} \mathbf{E}_2^R & \mathbf{0}_{4 \times 2} \\ \mathbf{0}_{2 \times 4} & \mathbf{E}_2^P \end{bmatrix} \quad , \quad (18a)$$

$$\mathbf{E}_3 = \begin{bmatrix} \mathbf{E}_1^R & \mathbf{0}_{4 \times 2} \\ \mathbf{0}_{2 \times 4} & \mathbf{E}_2^P \end{bmatrix} \quad , \quad \mathbf{E}_4 = \begin{bmatrix} \mathbf{E}_2^R & \mathbf{0}_{4 \times 2} \\ \mathbf{0}_{2 \times 4} & \mathbf{E}_2^P \end{bmatrix} \quad , \quad (18b)$$

$$\mathbf{A}_l = \begin{bmatrix} \mathbf{0}_{2 \times 2} & \mathbf{I}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \begin{bmatrix} 2b * _6 & b * _6 \\ -b * _7 & 0 \end{bmatrix} & \begin{bmatrix} A_K d_r & -A_K d_r \\ 0 & 0 \end{bmatrix} \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \begin{bmatrix} A_K * _8 & 0 \\ 0 & A_K * _8 \end{bmatrix} \end{bmatrix} \quad , \quad (18c)$$

with  $*_8 \in \{\underline{\dot{x}}_K, \bar{x}_K\}$  where  $\underline{\dot{x}}_K = \underline{\psi} \frac{d_r}{2}$ ,  $\bar{x}_K = \bar{\psi} \frac{d_r}{2}$  and

$$\mathbf{B}_l = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ b_5(*_3) & 0 \\ b_6(*_4) & 0 \end{bmatrix}, \quad \Delta \mathbf{B} = \begin{bmatrix} \mathbf{0}_{4 \times 2} & \\ \Delta b_5 & 0 \\ \Delta b_6 & 0 \end{bmatrix} \quad (18d)$$

with

$$\begin{aligned} b_5(*_3) &= R_g T_I \alpha_D \pi d (\tilde{a}_{02} + \tilde{b}_{02} *_3) \quad , \\ b_6(*_4) &= -R_g T_{II} \alpha_D \pi d (\tilde{a}_{03} + \tilde{b}_{03} *_4) \quad , \\ \Delta b_5 &= R_g T_I \alpha_D \pi d \Delta \tilde{a}_2 \\ \Delta b_6 &= -R_g T_{II} \alpha_D \pi d \Delta \tilde{a}_3 \end{aligned}$$

and finally

$$\Delta \tilde{\mathbf{a}} = \begin{bmatrix} \mathbf{0}_{4 \times 1} \\ \Delta p_I \\ \Delta p_{II} \end{bmatrix} \quad . \quad (18e)$$

## 4 Controller design concept

The basic idea behind the investigated controller concept is that the model base of the *feedback loop controllers* will be increased so a less accurate knowledge of the system model is required in the *feedforward compensation*. In this paper we consider the special case of a structure without a common feedforward compensation as an appropriate starting point.

By defining  $\tilde{\mathbf{x}} = [\mathbf{x}^T \ \dot{\mathbf{x}}^T]$ , the fuzzy descriptor system can be rewritten as

$$\tilde{\mathbf{E}} \dot{\tilde{\mathbf{x}}} = \sum_{l=1}^{32} \sum_{k=1}^4 h_l \tilde{\nu}_k \left[ \tilde{\mathbf{A}}_{lk} \tilde{\mathbf{x}} + (\tilde{\mathbf{B}}_l + \Delta \tilde{\mathbf{B}}) \mathbf{u} \right] + \Delta \tilde{\mathbf{a}} \quad (19)$$

with  $h_l = h_l(\varphi, \dot{\varphi}, \psi, \dot{\psi}, p_I, p_{II})$  and  $\tilde{\nu}_k = \tilde{\nu}_k(\psi, \varphi)$  where

$$\begin{aligned} \tilde{\mathbf{E}} &= \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \tilde{\mathbf{A}}_{lk} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ \mathbf{A}_l & -\mathbf{E}_k \end{bmatrix}, \\ \tilde{\mathbf{B}}_l &= \begin{bmatrix} \mathbf{0} \\ \mathbf{B}_l \end{bmatrix}, \quad \Delta \tilde{\mathbf{B}} = \begin{bmatrix} \mathbf{0} \\ \Delta \mathbf{B} \end{bmatrix}, \quad \Delta \tilde{\mathbf{a}} = \begin{bmatrix} \mathbf{0} \\ \Delta \tilde{\mathbf{a}} \end{bmatrix}. \end{aligned}$$

We propose a modified parallel distributed compensation scheme (PDC)

$$\mathbf{u} = \sum_{l=1}^{32} \sum_{k=1}^4 h_l \tilde{\nu}_k \tilde{\mathbf{F}}_{lk} \dot{\tilde{\mathbf{x}}} \quad (20)$$

with  $\tilde{\mathbf{F}}_{lk} = [\mathbf{F}_{lk} \ \mathbf{0}]$  to stabilize the fuzzy descriptor system (19). The fuzzy controller design problem here is to determine the local feedback gains using the local models  $\{\tilde{\mathbf{A}}_{lk}, \tilde{\mathbf{B}}_l, \Delta \tilde{\mathbf{B}}, \Delta \tilde{\mathbf{a}}\}$  that globally stabilize the fuzzy descriptor system with uncertainties by (20). We currently solve this problem using a novel LMI (Linear Matrix Inequalities) condition derived in [4].

## 5 Conclusion

In this paper the possibilities of the use of a TS fuzzy descriptor form to describe the dynamics of a two-link manipulators with dual-actuators were analysed. A complete manipulator model using the combination of the actuator models in descriptor form and the robot arm dynamics in descriptor form was derived for a model-based control concept. We think this is an appropriate starting point for a robot control concept that systematically take into consideration the influence of model reduction by structured uncertainties.

## References

- [1] T. M. Guerra and L. Vermeiren. Control laws for Takagi-Sugeno fuzzy models. *Fuzzy Sets and Systems*, 120(1):95–108, 2001.
- [2] T. M. Guerra, K. Guelton and S. Delprat. A class of non linear observers in descriptor form: LMI based design with applications in biomechanics. In *2nd IFAC Workshop on Advanced Fuzzy-Neural Control*, pages 73–78, Oulu, Finland, Sept. 2004.
- [3] T. A. Johansen, K. Hunt, P. Gawthrop, and H. Fritz. Off-equilibrium linearisation and design of gain-scheduled control with application to vehicle speed control. *Control Engineering Practice*, 6(2):167–180, 1998.
- [4] J. Lauber, T. M. Guerra, and W. Perruquetti. LMI conditions for continuous uncertain TS models in closed-loop with an observer: application to engine speed control. In *IEEE VTS - Vehicle Power and Propulsion Proceedings*, Paris, France, Oct. 2004.
- [5] C. J. Lopez-Toribio, R. J. Patton, and S. Daley. Takagi-Sugeno Fuzzy Fault-Tolerant Control of an Induction Motor. *Neural Computing and Applications*, 9(1):19–28, 2000.
- [6] D. G. Luenberger. Dynamic Equations in Descriptor Form. *IEEE Trans. Automatic Control*, 22(3):312–321, 1977.
- [7] Y. Morère. *Mise en oeuvre de lois de commande pour les modèles flous de type Takagi Sugeno*. PhD thesis, University Valenciennes, LAMIH-SF, France, 2001.
- [8] K. Schillings. *Servopneumatische Antriebssysteme und Handhabungsgeräte*. PhD thesis, RWTH-Aachen, Institute of Fluid Power Drives and Controls, January 2000.
- [9] H. Schulte and H. Hahn. Fuzzy state feedback gain scheduling control of servopneumatic actuators. *Control Engineering Practice*, 12(5):639–650, 2004.
- [10] H. Schulte. Model-based Fuzzy Tracking Control of Two-Link Manipulators with Dual Actuators. In *2nd IFAC Workshop on Advanced Fuzzy-Neural Control*, pages 109–114, Oulu, Finland, Sept. 2004.

- [11] L. Sciavicco and B. Siciliano. *Modelling and Control of Robot Manipulators*. Springer-Verlag London, second edition, 1999.
- [12] T. Takagi and M. Sugeno. Fuzzy identification of systems and its application to modelling in control. *IEEE Transactions on System, Man and Cybernetics*, 15(1):116–132, 1985.
- [13] K. Tanaka, T. Ikeda, and H. O. Wang. Fuzzy regulators and fuzzy observers: Relaxed stability conditions and LMI-based designs. *IEEE Transactions on Fuzzy Systems*, 6(2):1–16, 1998.
- [14] T. Taniguchi, K. Tanaka and H.O Wang. Fuzzy Descriptor Systems and Non-linear Model Following Control. *IEEE Trans. on Fuzzy Systems*, 8(4):442–452, 2000.
- [15] H. O. Wang, K. Tanaka, and M. F. Griffin. An Approach to Fuzzy Control of Nonlinear Systems: Stability and Design Issues. *IEEE Transactions on Fuzzy Systems*, 4(1):14–23, 1996.

# Petri-Netze zur Aufgabenüberwachung in humanoiden Robotern

Arne Lehmann<sup>1</sup>, Ralf Mikut<sup>1</sup>, Tamim Asfour<sup>2</sup>

<sup>1</sup> Forschungszentrum Karlsruhe GmbH, Institut für Angewandte Informatik (IAI),  
D-76021 Karlsruhe, Postfach 3640, Telefon: (07247) 82-5749,  
Fax: (07247) 82-5786, E-Mail: arne.lehmann@iai.fzk.de

<sup>2</sup> Universität Karlsruhe (TH), Institut für Technische Informatik (ITEC),  
Technologiefabrik, Haid-und-Neu-Str. 7, D-76131 Karlsruhe,  
Telefon: (0721) 608-7379, Fax: (0721) 608-8270, E-Mail: asfour@ira.uka.de

## 1 Einleitung

Im Unterschied zu Industrierobotern agieren humanoide Roboter (wie z. B. ARMAR [1]) in komplexen und sich ständig verändernden Umgebungen. Sie benötigen deshalb eine leistungsfähige Steuerungsarchitektur zur Planung, Koordination und Überwachung der Ausführung komplexer Aufgaben, die auch in Ausnahmesituationen (Abweichungen vom Routineablauf) flexibel reagiert. Beispiele für Ausnahmesituationen sind Kollisionen, das Nichterreichen gewünschter Ziele oder neue Benutzeranweisungen während der Ausführung einer Aufgabe.

Viele Roboter-Steuerungsarchitekturen bestehen aus drei hierarchisch angeordneten Schichten mit Planungs-, Koordinierungs- und Ausführungsebene [2, 3, 4, 5]. Derzeit werden die meisten Routineabläufe und Behandlungsstrategien für Ausnahmesituationen beim Durchführen einer Aufgabe im Voraus zumindest strukturell geplant. Diese Pläne werden situationsabhängig aktiviert und aktualisiert. Die Koordination basiert auf erkannten externen und internen Ereignissen. Die dazu verwendeten Merkmale reichen von festgelegten Reaktionszeiten, einfachen Schwellwerten bei Sensorsignalen bis hin zu komplexen Erkennungsaufgaben mit Methoden der Computational Intelligence (z.B. Sprach- und Geräuscherkennung, fuzzy-basierte Erkennung von Stabilitätsproblemen und Hardwarefehlern). Die Koordinierungs- und die Ausführungsebene unterliegen dabei Echtzeitanforderungen.

Bei Routineabläufen mit sequentiellen und parallelen Aktionen hat sich eine Beschreibung in Form von Petri-Netzen [6] bewährt. Solche Beschreibungen finden sich sowohl für Industrieroboter [7, 8] als auch bei humanoiden Robotern [9, 10, 11] für die Planung, die Koordination und die Ressourcenverwaltung.

Ausnahmesituationen verlangen je nach Typ eine einheitliche (z.B. Notaus) oder situationsspezifische Behandlung. Letztere erfolgt derzeit meist durch zusätzliche Stellen in Petri-Netzen, für die bereits in der Aufgabenplanung von jeder Stelle aus entsprechende Transitionen einzuführen sind. Dadurch entstehen komplizierte Petri-Netze, die zudem die Koordination bei der späteren Fortsetzung von unterbrochenen Aufgaben erschweren. Deswegen ist die vollständige Integration der existierenden Arbeiten zur Erkennung von Ausnahmesituationen (z.B. mit Fuzzy-Regeln und Neuronalen Netzen) in die Steuerungsarchitektur eines humanoiden Roboters ein ungelöstes Problem.

Die vorliegende Arbeit skizziert ein Konzept, das für alle drei Ebenen einer hierarchischen Roboter-Steuerungsarchitektur separate Überwachungsmodule vorschlägt, um damit die

einfache und systematische Behandlung von Ausnahmesituationen zu ermöglichen (Abschnitt 2). Abschnitt 3 beschreibt detailliert die Umsetzung der Aufgabenkoordination der Steuerungsarchitektur mit den entsprechenden Petri-Netz-basierten Strategien zur Erkennung und Behandlung von Ausnahmesituationen. Die Grundidee besteht aus einer speziell strukturierten Dekomposition von Überwachungsmodulen und deren Kopplung mit geplanten Routineabläufen. Abschnitt 4 setzt das vorgestellte Konzept am Beispiel einer typischen Aufgabe im humanoiden Umfeld um und zeigt Ergebnisse einer Simulationsstudie, die die Funktionsfähigkeit des vorgeschlagenen Konzepts dokumentieren.

## 2 Steuerungsarchitektur

Zur Beherrschbarkeit der hohen Systemkomplexität bei humanoiden Robotern wird eine hierarchische Steuerungsarchitektur zugrunde gelegt, bei der eine gestellte Aufgabe auf unterschiedlichen Abstraktionsebenen gelöst wird, die in ihrer Gesamtheit eine sogenannte Abstraktionshierarchie bilden. Dabei wird eine vom Benutzer vorgegebene Aufgabe (engl.: task) von einer Aufgabenplanung auf Teilaufgaben (engl.: subtask) für ein oder mehrere Teilsysteme des Roboters abgebildet (z.B. eine Handhabungsaufgabe mit Teilaufgaben für eine Roboterplattform, einen Arm, eine Hand und für Perzeptionskomponenten beim Abstellen einer Tasse auf einem Tisch). Jede Teilaufgabe dient einem bestimmten Ziel (z.B. Tasse lokalisieren, Arm auf einer Trajektorie zu einer Zielposition bewegen). Das hierarchisch organisierte Handlungswissen des Roboters enthält Aktionsfolgen  $AF$  (engl. sequence of actions), um Aufgaben oder Teilaufgaben erfolgreich zu lösen. Eine Aktionsfolge besteht aus einer oder mehreren sequentiellen oder parallelen Aktionen (Bild 1). Eine Aktion  $A$  (engl. action) setzt sich wiederum aus einer oder mehreren sequentiellen oder (koordinierten) parallelen Elementaraktionen  $EA$  zusammen. Eine Elementaraktion umfasst eine oder mehrere parallele Elementaroperationen  $EO$  (engl. elementary operation), die je eine primitive sensomotorische Operation für einen einzelnen Freiheitsgrad (z.B. einen Regelkreis) beschreiben.

Aktionsfolgen, Aktionen, Elementaraktionen und Elementaroperationen werden hier als Stellen in hierarchisch organisierten Petri-Netzen umgesetzt. Ein Petri-Netz besteht aus Stellen (engl. places), Marken (engl. tokens), Transitionen (engl. transitions) und ihren

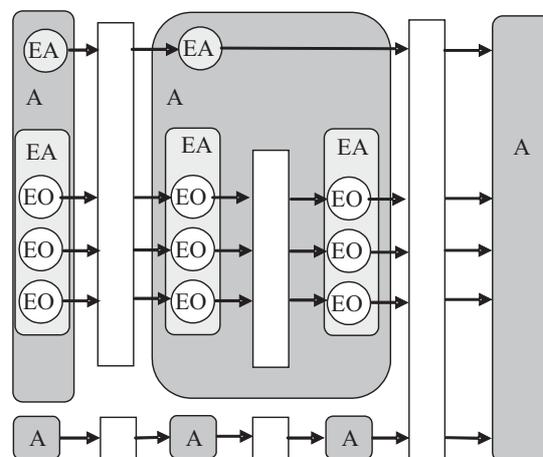


Bild 1: Beispiel für eine Aktionsfolge mit parallelen und sequentiellen Aktionen  $A$  und deren Bestandteilen (Elementaroperationen  $EO$  und Elementaraktionen  $EA$ )

Verbindungen (engl. arcs). Da eine Elementaraktion nur über parallele Elementaroperationen verfügt, haben alle zugehörigen Elementaroperationen eine identische Markierung. Die Transitionen werden über externe oder interne Ereignisse (z.B. Zeiten, Bedingungen an Sensormesswerte, Klassifikationsergebnisse über verfügbare Merkmale) geschaltet.

Die verwendete hierarchische Steuerungsarchitektur basiert sowohl auf Steuerungs- und Regelungsstrategien für Industrieroboter [12, 7] als auch auf eher forschungsorientierten Architekturen intelligenter Systeme und humanoider Roboter [2, 3, 4, 5, 9, 13, 14]. Die Besonderheit des Ansatzes ist die Gliederung in vertikale Module (Wissen, Durchführung und Überwachung) und horizontale Ebenen zur Aufgabenplanung, Aufgabenkoordination und Aufgabenausführung (Bild 2).

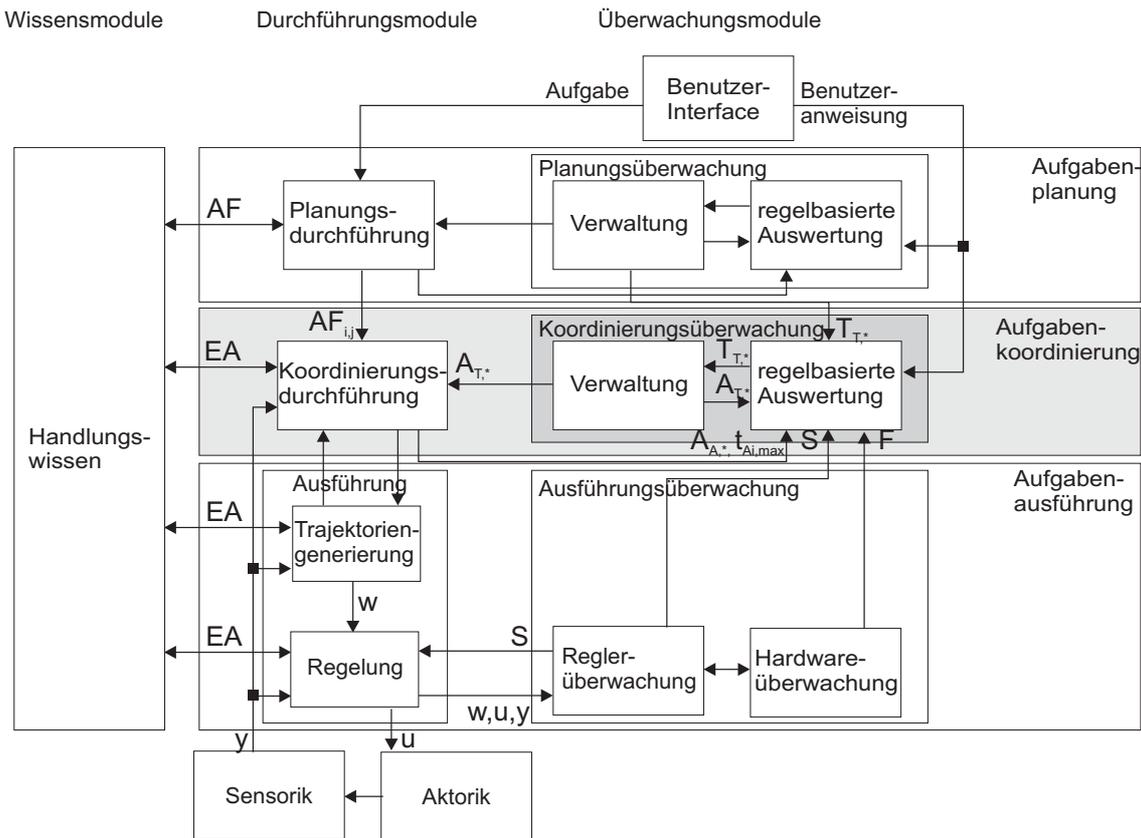


Bild 2: Hierarchische Struktur zur Planung, Ausführung und Überwachung von Aufgaben (grau: hier detailliert behandelte Komponenten)

Auf der oberen Hierarchieebene stellt die *Aufgabenplanung* (engl. task planning) aus den Anforderungen des Nutzers und dem Handlungswissen unter Berücksichtigung von Prioritäten und verfügbaren Ressourcen einen ausführbaren Plan in Form einer Aktionsfolge (einschließlich der zugehörigen Parametrierung) zusammen. Dieser Plan soll zumindest für den Routinebetrieb frei von Deadlocks sein. Zudem enthält dieser Plan vordefinierte Informationen für die Behandlung von Ausnahmesituationen auf der Ebene der Aufgabenkoordination. Derzeit wird lediglich auf vorhandene Aktionsfolgen zurückgegriffen, die im Handlungswissen für eine konkrete Aufgabe hinterlegt sind. Langfristig soll dieser Schritt von Lernalgorithmen übernommen werden [15, 16, 17].

Auf der mittleren Hierarchieebene führt die *Aufgabenkoordination* (engl. task coordination) die aktuelle Aktionsfolge als Petri-Netz aus und berechnet die zugehörigen Markie-

rungen. Ausgangsgrößen der Koordinierungsdurchführung sind Information über derzeit aktive Elementaraktionen.

Auf der unteren Hierarchieebene befindet sich die *Aufgabenausführung* (engl. task execution), die alle zugehörigen Funktionen zu denjenigen Elementaraktionen startet, die durch die Aufgabenkoordination aktiviert (markiert) werden. Zu den Elementaraktionen und -operationen gehören hierbei u.a. Funktionen mit stellenspezifischen Einstellungen für Reglerstrukturen und -parameter, Reflexe, Solltrajektorien und Überwachungsfunktionen. Elementaraktionen verallgemeinern somit das Konzept der Aktionsprimitive für Roboter in [7], die bereits zusammengehörige Bewegungen (mit Solltrajektorien, Reglern, Koordinatensystemen), Werkzeugkommandos und Abbruchbedingungen vorsehen.

Alle drei Hierarchieebenen enthalten zugehörige Überwachungsmodule (*Planungsüberwachung, Koordinierungsüberwachung, Ausführungsüberwachung*), die sich mit der Ausnahmebehandlung auf der jeweiligen Ebene beschäftigen. Das Ziel der vorgeschlagenen Grundstruktur besteht darin, die mittlere und die untere Hierarchieebene mit einem Höchstmaß an Autonomie zu versehen. Die entsprechenden Komponenten werden in die Lage versetzt, möglichst schnell ohne den direkten Eingriff der Aufgabenplanung auf auftretende Ereignisse (routinemäßig erwartete Situationen und berücksichtigte Ausnahmesituationen) zu reagieren. Dazu werden die Zustände aller Komponenten der Regelkreise (u.a. Regler, Aktoren, Sensoren) zur Laufzeit überwacht und entsprechend der geforderten Aufgabe bewertet (vgl. Ausführungsüberwachung Bild 2). Auf Basis dieser Informationen entscheiden die übergeordneten Ebenen über die Fortsetzung, die zeitweilige Unterbrechung oder den Abbruch von Aktionsfolgen.

Im Weiteren erfolgt die detaillierte Beschreibung der theoretischen Grundlagen der Aufgabenkoordination (Bild 2 grau hinterlegt).

### 3 Aufgabenkoordination

#### 3.1 Übersicht

Die Eingangsgröße des Durchführungsmoduls der Aufgabenkoordination (Bild 2, Koordinierungsdurchführung) ist die von der Aufgabenplanung vorgegebene Aktionsfolge (AF). Die Koordinierungsdurchführung lädt die zugehörigen Informationen für die einzelnen Elemente dieser Aktionsfolge (Aktionen, ..., Elementaroperationen) aus dem Handlungswissen. Die Aktionsfolge ist als sogenanntes Durchführungsnetz gegeben (Beispiele siehe Bild 3), das den Routineablauf als Petri-Netz ohne gesonderte Stellen für Ausnahmesituationen enthält.

Bild 3 zeigt zwei Beispiele für Ausschnitte aus Aktionsfolgen als Durchführungsnetz. Auf der linken Seite ist die Aktion "Objekt verfolgen" für die Halseinheit eines humanoiden Roboters dargestellt. Die rechte Seite zeigt einen Ausschnitt der Aktion "Greifen" [18]. Die Bedeutung der einzelnen Stellen und Transitionen wird im Abschnitt 4.1 detailliert beschrieben.

Bei Ausnahmesituationen stehen einem humanoiden Roboter prinzipiell verschiedene Bewältigungsstrategien zur Verfügung:

- Unterbrechen der Aufgabe und Start eines Klärungsdialogs mit dem Benutzer,

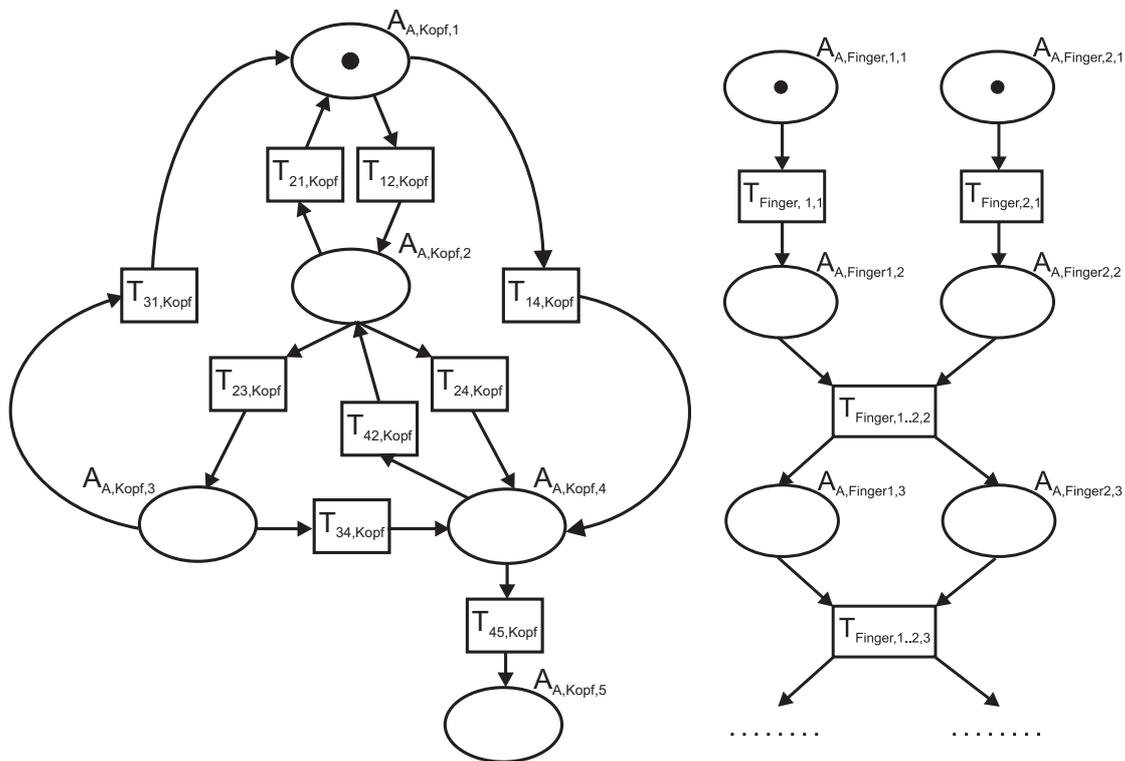


Bild 3: Ausschnitte aus Aktionsfolgen als Durchführungsnetz: "Objekt verfolgen" (links), Ausschnitt aus "Greifen" (rechts)

- Unterbrechen der Aufgabe und Start einer Problembehandlung durch externe Eingriffe (z.B. Telemanipulation),
- annähernd planmäßiges Beenden der Aufgabe und Ignorieren des Problems,
- Abbruch der Aufgabe und Übergang in einen sicheren Zustand und
- selbstständige Modifikation der geplanten Aufgabe.

Deswegen entscheidet der Roboter oder der Benutzer, je nach Schwere der Ausnahmesituation die entsprechenden Bewältigungsstrategien zu starten. Ein sicheres und sinnvolles Verhalten des Roboters beim Unterbrechen und späteren Fortsetzen bzw. beim Abbrechen hängt aber stark von der aktuellen Situation ab, die durch die jeweiligen markierten Elementaraktionen repräsentiert wird.

Die neue einheitliche Strategie besteht in der Idee,

- die Ausnahmebehandlung für ein Durchführungsnetz durch je ein Modul zur Verwaltung und zur regelbasierten Auswertung zu organisieren (Bild 2) sowie
- alle Stellen von Elementaroperationen mit zwei zusätzlichen Funktionen für eine stellenspezifische Ausnahmebehandlung zum Unterbrechen und Abbrechen von Aufgaben auszustatten.

Im Weiteren erfolgt nun die detaillierte Beschreibung der *Verwaltung* und der *regelbasierten Auswertung* sowie deren Zusammenspiel beim Auftreten von Ausnahmesituationen.

### 3.2 Verwaltung

Die *Verwaltung* der Aufgabe erfolgt in der Koordinierungsüberwachung (Bild 2) durch ein weiteres Petri-Netz, das als *Verwaltungsnetz* bezeichnet wird (Bild 4). Seine Stellen

stehen für eine inaktive (1. Sleeping  $A_{T,Sl}$ ), eine ausführungsbereite (2. Ready  $A_{T,Re}$ ), eine routinemäßig abzuarbeitende (3. Active  $A_{T,Ac}$ ), eine zeitweilige zu unterbrechende (4. Stopped  $A_{T,St}$ ), eine erfolgreich durchgeführte (5. Success  $A_{T,Su}$ ) oder eine irreversible abgebrochene (6. Cancelled  $A_{T,Ca}$ ) Aktionsfolge im zugehörigen Durchführungsnetz. Der Vorteil besteht darin, dass das Verwaltungsnetz keine Detailinformationen über die auszuführende Aktionsfolge benötigt. Verwaltungsnetze haben somit immer eine gleiche Netzstruktur, die nicht vom zugehörigen Durchführungsnetz abhängt.

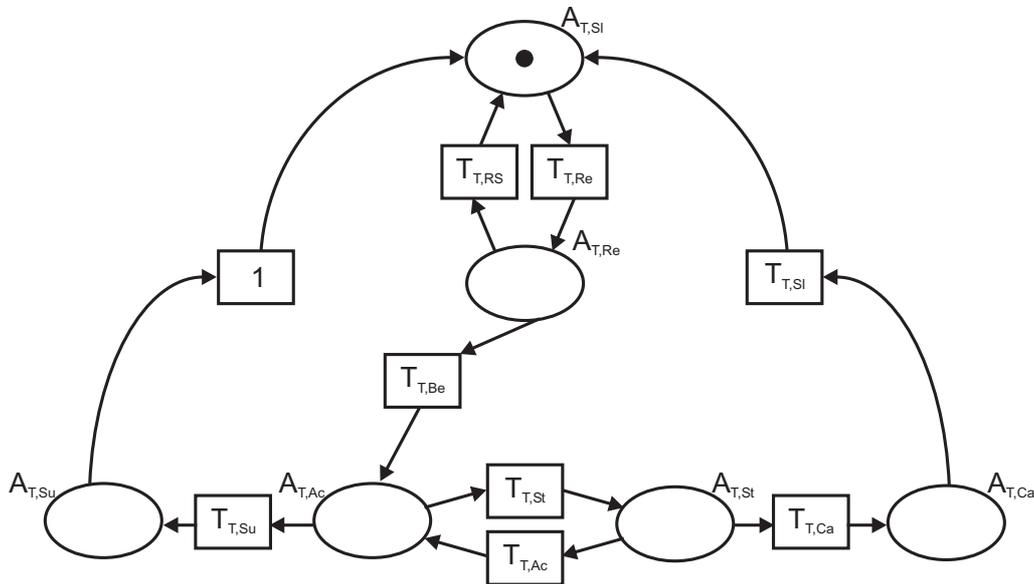


Bild 4: Verwaltungsnetz mit Stellen 1. Sleeping  $A_{T,Sl}$ , 2. Ready  $A_{T,Re}$ , 3. Active  $A_{T,Ac}$ , 4. Stopped  $A_{T,St}$ , 5. Success  $A_{T,Su}$  und 6. Cancelled  $A_{T,Ca}$

Die Transitionen im Verwaltungsnetz stammen aus der übergeordneten Planungsüberwachung oder aus den Konklusionen der Regeln im Modul regelbasierte Auswertung.

Das Starten einer Aktionsfolge wird von der Planungsüberwachung initiiert. Notwendige Voraussetzungen sind eine ausführungsbereite Aktionsfolge ( $A_{T,Re}$  im Verwaltungsnetz der Aktionsfolge) und die Verfügbarkeit der benötigten Hardwareressourcen [9]. Zusätzlich kann die Planungsüberwachung weitere initiale Bedingungen prüfen. Diese Bedingungen ergeben zusammen die Transition  $T_{T,Be}$  im Verwaltungsnetz. Das Verwaltungsnetz setzt bei Durchlaufen dieser Transition die im Handlungswissen hinterlegte Startmarkierung im zugehörigen Durchführungsnetz.

Erkannte Ausnahmesituationen aktivieren hierbei je nach Schweregrad die Transitionen  $T_{T,St}$  bzw.  $T_{T,St}$  und  $T_{T,Ca}$ . Sie bewirken somit für das Verwaltungsnetz einen Markenfluss auf die Stopped-Stelle  $A_{T,St}$  bzw. Cancelled-Stelle  $A_{T,Ca}$ . Die Prämissen der Regeln verarbeiten Informationen aus der Planungsüberwachung, der Koordinierungsdurchführung und der Ausführungsüberwachung (Bild 2).

Die Markenbelegung des Verwaltungsnetzes wird wiederum dem Durchführungsnetz mitgeteilt, das in seinen jeweils markierten Stellen die Funktionen für:

- eine routinemäßige Abarbeitung der Aktionsfolge (Markierung im Verwaltungsnetz auf  $A_{T,Ac}$ , Ziel: planmäßige Abarbeitung),
- eine zeitweilige Unterbrechung der Aktionsfolge (Markierung im Verwaltungsnetz auf  $A_{T,St}$ , Ziel: sichere Unterbrechung, um eine spätere Fortsetzung möglichst an der gleichen Markierung zu ermöglichen) oder

- einen Abbruch der Aktionsfolge (Markierung im Verwaltungsnetz auf  $A_{T,Ca}$ , Ziel: sicheres irreversibles Abbrechen)

ausführt. Die meisten Stellen im Durchführungsnetz enthalten bei Unterbrechung oder Abbruch nur einfache Wartefunktionen (mache nichts!). Allerdings ist bei bestimmten Elementaraktionen aus Sicherheitsgründen eine andere Funktion sinnvoll (z.B. halte gegriffenes Objekt gerade, ziehe den Arm nach einer Kollision zurück, starte einen Warn-dialog, setze Objekt definiert ab usw.). Zusätzlich können Stellen fest programmierte Informationen für eine aktionsspezifische regelbasierte Auswertung enthalten.

Eine Aktivierung der Transition  $T_{T,Ac}$  nach Beseitigung des Problems erfolgt derzeit hauptsächlich extern durch Befehl des Benutzers oder des Teleoperators. In einfachen Fällen (z.B. erkannte Tasse nach einem Dialog) kann diese Transition auch durch den Roboter selbst mittels der Planungsüberwachung aktiviert werden. Normalerweise wird das Durchführungsnetz an den gleichen Stellen  $A_{Ai}$  fortgesetzt, an denen es gestoppt wurde.

Allerdings können zwischenzeitliche Markenflüsse durch äußere Einflüsse (feuernde Transitionen im Durchführungsnetz) auftreten (z.B. Kollisionen, Verlust von gegriffenen Objekten usw.). Damit reagiert das Durchführungsnetz auch bei unterbrochenen oder abgebrochenen Aktionsfolgen auf veränderte äußere Situationen, solange sich das Verwaltungsnetz noch nicht in  $A_{T,sl}$  befindet.

Die finale Markenbelegung im Durchführungsnetz, die für die erfolgreiche Abarbeitung der Aktionsfolge steht, aktiviert die Transition  $T_{T,Su}$  im Verwaltungsnetz. Die Planungsüberwachung gibt nach der erfolgreichen Abarbeitung ( $T_{T,Su}$ ) oder nach einem abgeschlossenen Abbruch ( $T_{T,sl}$ ) die Ressourcen wieder frei und deaktiviert das Durchführungsnetz.

Grundsätzlich kann die übergeordnete Planungsüberwachung alle Transitionen im Verwaltungsnetz extern aktivieren. Sie übernimmt insbesondere bei  $T_{T,Su}, T_{T,sl}$  die volle Verantwortung für den weiteren Ablauf, während bei  $T_{T,St}, T_{T,Ac}, T_{T,Ca}$  das Durchführungsnetz noch aktiv eingreift.

Das Zusammenwirken von Durchführungs- und Verwaltungsnetz ergibt ein deutlich komplexeres Petri-Netz, das allerdings nicht mehr explizit implementiert werden muss. Jede Stelle im Durchführungsnetz enthält drei Unterstellen (Active, Stopped, Cancelled), die aus dem Verwaltungsnetz stammen, und zusätzliche Transitionen (Bild 5). Das Verlassen der zusammengefassten Stelle geschieht durch die ursprünglich definierte ausgangsseitige Transition<sup>1</sup>  $T_{i,Out}$ . Ausgangsseitige Transitionen können auch koordinierend sein. Zusätzlich kann jede Stelle aus der Unterstelle  $A_{Ai,Ca}$  verlassen werden, wenn im Verwaltungsnetz die Transition  $T_{T,sl}$  feuert.

Je nach Markenbelegung im Verwaltungsnetz führt das Durchführungsnetz nur die Funktion für die jeweilige Unterstelle aus. Ein Markenfluss im Verwaltungsnetz löst für *alle* markierten Stellen im Durchführungsnetz einen internen Markenfluss auf die jeweilige korrespondierende Unterstelle aus. Dennoch sind auch bei den jeweiligen Stopped- oder Cancelled-Unterstellen Markenflüsse zu nachfolgenden Stellen im Durchführungsnetz möglich.

Bei einem irreversiblen Fehler oder einem gewünschten Abbruch der Aktionsfolge (Transition  $T_{T,Ca}$ : Erkennung durch den Roboter selbst, durch den Benutzer oder durch Teleoperator) erfolgt ein ebenfalls irreversibler Übergang zur Stelle  $A_{T,Ca}$ . Hier stellt der

---

<sup>1</sup>Hier sind prinzipiell auch mehrere Transitionen möglich.

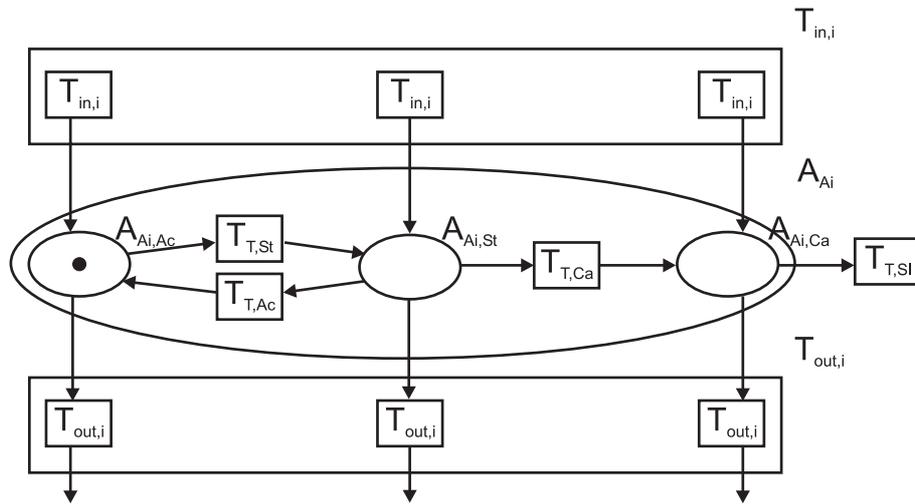


Bild 5: Ausschnitt für eine Stelle des Durchführungsnetzes mit den drei Unterstellen bei Verknüpfung mit dem Verwaltungsnetz

Roboter entweder selbstständig oder mit externer Hilfe einen sicheren Zustand her, der einem gezielten "Herunterfahren" der Aktionsfolge entspricht. Nachdem dieser erreicht wird, erfolgt der Übergang zur Stelle  $A_{T,SI}$  und die Aktionsfolge befindet sich wieder 'schlafend' im Handlungswissen.

### 3.3 Regelbasierte Auswertung

Bei der Abarbeitung des Durchführungsnetzes können verschiedene Situationen eintreten, welche den routinemäßigen Ablauf der Aktionsfolge stören:

- unerwünschte Ereignisse (z.B. Kollisionen in Annäherungsphasen an ein Objekt),
- Nichterreichen von gewünschten Ereignissen (z.B. ausbleibender Objektkontakt, Nichterreichen gewünschter Positionen),
- Stabilitätsprobleme von unterlagerten Regelkreisen,
- Ausfall von Hardwarekomponenten (z.B. Antriebe, Sensoren),
- nicht realisierbare Pläne,
- Änderung von Plänen durch den menschlichen Benutzer während der Ausführung einer Aufgabe.

Die regelbasierte Auswertung verarbeitet Informationen aus den einzelnen Stellen des Durchführungsnetzes. Die in der Regelbasis hinterlegten Regeln sind von der Form

WENN *Prämisse* DANN *Transition(en) aktivieren (=1) bzw. blockieren (-1)*.

Die Konklusion beeinflusst somit nur Transitionen im Verwaltungsnetz. Diese Form erzwingt eine strenge Dekomposition von Fehlererkennung und -bewertung (in der regelbasierten Auswertung) und Behandlung von Ausnahmesituationen (auszuführende Funktionen im Durchführungsnetz).

In der Prämisse befinden sich dabei Informationen wie z.B.

- die Zeitüberschreitung für den Aufenthalt einer Marke in einer Stelle  $t > t_{Ai,max}$ ,
- die regelbasierte Erkennung von Fehlern in einer Stelle durch spezielle Funktionen,
- erkannte Stabilitätsprobleme für mindestens eine aktive Regelung [19],

- regelbasiert erkannte Hardwarefehler [20],
- erkannte unzulässige Wettläufe in einem Petri-Netz mit Verzweigungen (gleichzeitiges Aktivieren mehrerer Transitionen an der Ausgangsseite einer markierten Stelle im Durchführungsnetz),
- Markenbelegungen bei (fest programmierten) Stellen, die Ausnahmesituationen kodieren,
- finale Markenbelegungen, die für sicher abbrechbare Aktionsfolgen stehen, sowie
- eine finale Markenbelegung, die für die erfolgreiche Abarbeitung der Aktionsfolge steht.

Damit ergeben sich Regeln wie

WENN *Stabilitätsgrad S klein* DANN  $T_{T,St} = 1$  und  $T_{T,Ac} = -1$   
 WENN *Hardwarefehler F* DANN  $T_{T,St} = 1$  und  $T_{T,Ca} = 1$  und  $T_{T,Ac} = -1$   
 WENN *Markenbelegung final* DANN  $T_{T,Su} = 1$   
 WENN  $\vdots$  DANN  $\vdots$

Bei Widersprüchen von Konklusionen haben von der Aufgabenplanung übermittelte Transitionen sowie Blockierungen Vorrang (Tabelle 1).

Transitionen aus der regelbasierten Auswertung der Planungsüberwachung	Transitionen aus der regelbasierten Auswertung der Koordinierungsüberwachung			
	0	1	-1	1 UND -1
0	0	1	0	0
1	1	1	0	0
-1	0	1	0	0
1 UND -1	0	1	0	0

Tabelle 1: Vorrangregeln für Transitionen (0: nicht aktiviert, 1: aktiviert) im Verwaltungsnetz bei regelbasierter Aktivierung (1), Blockierung (-1), Widersprüchen (1 und -1: mindestens eine aktivierende und mindestens eine blockierende Regel) oder fehlenden Aussagen (0) von Transitionen durch die Planungs- und Koordinierungsüberwachung

## 4 Beispiel: Bildüberwachtes Greifen

### 4.1 Umsetzung

Die Funktionsfähigkeit des vorgestellten Konzeptes wird anhand von Simulationsstudien von bildüberwachten Greifvorgängen eines humanoiden Roboters gezeigt. Die an dieser Aktionsfolge beteiligten Subsysteme eines humanoiden Roboters sind die bewegliche Plattform und der Torso, ein Arm, ein Greifer und der Kopf incl. Stereo-Kameras. Tabelle 2 zeigt die Hardware-Ressourcen der Aktionen  $A_{G,i}$  für das Beispiel "bildüberwachtes Greifen". Der zweite Arm und der zweite Greifer sind nicht eingebunden und müssen lediglich Minimalanforderungen (keine Kollisionen usw.) erfüllen. Die Aktionen bestehen aus weiteren subsystemspezifischen Aktionen, die wiederum Elementaraktionen und -operationen enthalten.

In diesem Beispiel wird nur die (einheitliche) Aktion für den Kopf und die subsystemspezifische Aktion in  $A_{G,2}$  für den ersten Greifer betrachtet (Tabelle 2 grau hinterlegt). Beide

$A_{G,i}$	$A_{G,1}$	$A_{G,2}$	$A_{G,3}$	$A_{G,4}$	$A_{G,5}$
Ressource	HINFAHREN +SUCHEN	TASSE GREIFEN	ZURÜCK FAHREN	ÜBERGEBEN	WARTEN
Plattform und Torso	Annähern	Warten	Transport	Warten	-
1. Arm	Annähern	Warten	Transport	Annähern	-
1. Greifer	Preshape	Zugreifen	Halten	Loslassen	-
2. Arm	-	-	-	-	-
2. Greifer	-	-	-	-	-
Kopf	Objekt verfolgen	Objekt verfolgen	Objekt verfolgen	Objekt verfolgen	-

Tabelle 2: Benötigte Hardware-Ressourcen der Aktionsfolge für das Beispiel "bildüberwachtes Greifen" (vereinfacht)

Aktionen setzen sich wiederum aus Folgen von Elementaraktionen zusammen. Diese Vorgehensweise setzt voraus, dass Plattform, Torso und der erste Arm die Aktion  $A_{G,1}$  bereits abgeschlossen haben, d. h. der Roboter so positioniert ist, dass ein Greifvorgang ohne weitere Bewegung von Plattform, Torso und Arm möglich ist.

Im Weiteren erfolgt nun zunächst die detaillierte Beschreibung der Aktion "Objekt verfolgen" (Subnetz der Aktionen  $A_{G,i}$  mit  $i = [1 \dots 4]$  für den Kopf in Tabelle 2). Bild 3 (links) zeigt die Aktion als Durchführungsnetz. In Tabelle 3 sind die Elementaraktionen als Stellen mit den zugehörigen Callbackfunktionen und die Transitionen für das Durchführungsnetz erläutert.

Stelle	Callbackfunktionen		
	Active ( $T_{T,Ac}$ )	Stopped ( $T_{T,st}$ )	Cancelled ( $T_{T,Ca}$ )
$A_{A,Kopf,1}$	Objekt verfolgen	Objekt verfolgen (+ Dialog)	-
$A_{A,Kopf,2}$	Objekt suchen	Objekt suchen (+ Dialog)	-
$A_{A,Kopf,3}$	Objekte verfolgen + Dialog	Objekte verfolgen + Dialog	-
$A_{A,Kopf,4}$	Benutzer verfolgen + Dialog	Benutzer verfolgen + Dialog	Dialog
$A_{A,Kopf,5}$	Beenden	Beenden	Beenden
Transition	Bedeutung		
$T_{12,Kopf}$	Objekt nicht lokalisiert		
$T_{14,Kopf}$	Dialoganforderung (Benutzer, Regel)		
$T_{21,Kopf}$	ein Objekt gefunden		
$T_{23,Kopf}$	mehrere Objekte gefunden		
$T_{24,Kopf}$	kein Objekt gefunden oder Dialoganforderung (Benutzer, Regel)		
$T_{31,Kopf}$	Objekt ausgewählt		
$T_{34,Kopf}$	kein Objekt ausgewählt oder Dialoganforderung (Benutzer, Regel)		
$T_{42,Kopf}$	Dialog beendet, Aktionsfolge fortsetzen		
$T_{45,Kopf}$	Dialog beendet, Aktionsfolge abbrechen		

Tabelle 3: Callbackfunktionen der Elementaraktionen (Stellen) und Transitionen der Aktion "Objekt verfolgen".

Ist das zu verfolgende Objekt (z. B. eine vom Roboter zu greifende Tasse) lokalisiert<sup>2</sup>, so ist die Stelle  $A_{A,Kopf,1}$  im Durchführungsnetz markiert. Falls das Objekt von der Kamera nicht mehr lokalisiert werden kann ( $T_{12,Kopf}$ ), startet der Kopf einen Suchvorgang (markierte Stelle  $A_{A,Kopf,2}$ ). Erfolgt vom Benutzer eine Dialoganforderung, dann feuern die Transitionen  $T_{14,Kopf}$  bzw.  $T_{24,Kopf}$  und der Roboter fragt beim Benutzer nach, wie es weitergehen soll. Ist der Roboter dabei, das zu verfolgende Objekt durch Kopfbewegungen zu suchen ( $A_{A,Kopf,2}$ ), so findet bei einem eindeutigen Erfolg ein Markenfluss mit  $T_{21,Kopf}$  zurück auf die Stelle  $A_{A,Kopf,1}$  und bei einem eindeutigen Misserfolg ein Markenfluss mit  $T_{24,Kopf}$  auf die Stelle  $A_{A,Kopf,4}$  statt. Die Stelle  $A_{A,Kopf,4}$  initiiert wieder einen Benutzerdialog. Identifiziert der Roboter mehrere Objekte, die die zu verfolgende Tasse sein könnten, dann findet ein Übergang  $T_{23,Kopf}$  zur Stelle  $A_{A,Kopf,3}$  statt. Dort erfolgt entweder die Identifikation des 'richtigen Objekts' mit Hilfe des Benutzers (Übergang zu der Stelle  $A_{A,Kopf,1}$ ), oder falls kein Objekt ausgewählt werden konnte, der Übergang zur Stelle  $A_{A,Kopf,4}$ . Ist der Benutzerdialog in der Stelle  $A_{A,Kopf,4}$  beendet, so wird die Aktion entweder durch Suchbewegungen des Kopfes fortgesetzt ( $T_{42,Kopf}$ ) oder ein Abbruch der ganzen Aktionsfolge veranlasst ( $T_{45,Kopf}$ ).

Bild 3 (rechts) zeigt einen Ausschnitt der Aktion "Greifen" als Subnetz zur Verfeinerung der Aktion  $A_{G,2}$  für den 1. Greifer (Tabelle 2) als Ausschnitt eines Durchführungsnetzes. In Tabelle 4 sind wiederum die Stellen mit zugehörigen Callbackfunktionen des Durchführungsnetzes sowie die Transitionen dargestellt.

Stelle	Callbackfunktionen		
	Active ( $T_{T,Ac}$ )	Stopped ( $T_{T,St}$ )	Cancelled ( $T_{T,Ca}$ )
$A_{A,Finger,n,1}$	Annähern	Warten (+Dialoganforderung)	-
$A_{A,Finger,n,2}$	Warten	Warten (+Dialoganforderung)	-
$A_{A,Finger,n,3}$	Zugreifen	Objekt gerade halten	Objekt gerade halten
Transition	Bedeutung		
$T_{Finger,j,1}$	Objektkontakt Finger $j$		
$T_{Finger,1..n,2}$	Objektkontakt alle $n$ Finger		
$T_{Finger,1..n,3}$	stabiler Griff für alle $n$ Finger		

Tabelle 4: Callbackfunktionen der Elementaraktionen (Stellen) und Transitionen für die Aktion "Greifen" als Bestandteile der Aktion  $A_{G2}$  für den 1. Greifer.

Ist der Robotergreifer in der für den geplanten Griff notwendigen Greifkonfiguration (Preshape) an der vorausberechneten Greifposition angekommen, so werden für alle am Griff beteiligten Finger  $n$  die Stellen  $A_{A,Finger,n,1}$  der Aktion "Greifen" markiert. Ergibt sich für einen oder mehrere Finger vorzeitig Kontakt mit dem zu greifenden Objekt, erfolgt für diese Finger  $j$  ein Übergang  $T_{Finger,j,1}$  zur Stelle  $A_{A,Finger,j,2}$ . Dort warten diese Finger, bis *alle* am Griff beteiligten Finger Kontakt haben  $T_{Finger,1..n,2}$ , damit dann ein koordinierter Markenfluss zu den Stellen  $A_{A,Finger,1..n,2}$  erfolgen kann. Nach Erreichen eines stabilen Griffs ( $T_{Finger,1..n,3}$ ) kann dann der Transport erfolgen (Übergang zu  $A_{G,3}$ ).

Analog dazu erfolgt nach Ende des Transports (erfolgreicher Abschluss  $A_{G,3}$ ) später ein koordiniertes Loslassen mit einem Lockern des Griffes, einem Warten auf den Kontaktverlust einzelner Finger und einer koordinierten Absetzbewegung.

<sup>2</sup>Lokalisiert: Im Sichtfeld der Kamera, interpolierte verdeckte Position oder bekannte Lage im Greifer.

Die regelbasierte Auswertung in der Aufgabenkoordinierung enthält für dieses Beispiel unter anderem die folgenden Regeln:

- $R_1$  : WENN  $t > t_{Ai,max}$  DANN  $T_{T,st} = 1$  und  $T_{T,Ac} = -1$   
 $R_2$  : WENN  $S < 0.5$  DANN  $T_{T,st} = 1$  und  $T_{T,Ac} = -1$   
 $R_3$  : WENN  $F[k] \neq 0$  DANN  $T_{T,st} = 1$  und  $T_{T,Ca} = 1$  und  $T_{T,Ac} = -1$   
 $R_4$  : WENN  $A_{A,Kopf,4}$  DANN  $T_{T,st} = 1$  und  $T_{T,Ac} = -1$   
 $R_5$  : WENN  $A_{A,Kopf,5}$  DANN  $T_{T,st} = 1$  und  $T_{T,Ca} = 1$  und  $T_{T,Ac} = -1$

Die erste Regel erkennt ein unerwünschtes Verharren in bestimmten Stellen durch eine Zeitüberschreitung. Wenn z.B. das Identifizieren eines Objekts ( $A_{A,Kopf,2}$ ) länger als  $t_{Ai,max} = 10s$  dauert, feuert die Transition  $T_{T,st}$  und ein Dialog wird initiiert (Tabelle 3). Weiterhin sollte z. B.  $A_{A,Finger1..n,2}$  nicht länger als  $t_{Ai,max} = 3 \dots 4s$  dauern (alle am Griff beteiligten Finger haben geplanten Kontaktzustand erreicht). Bleibt der erwartete Markenfluss aus, wird das Verwaltungsnetz gestoppt ( $T_{T,st}$  aktiv) und z.B. ein Dialog initiiert (Tabelle 4).

Falls die Basis-Regler der aktiven Elementaraktionen nicht in der Lage sind, die Vorgaben der Trajektoriengenerierung umzusetzen, gibt die jeweilige Reglerüberwachung Stabilitätsgrade kleiner 0.5 aus. Dies wertet die regelbasierte Auswertung als Ausnahmesituation und  $T_{T,st}$  feuert. Falls die jeweilige Hardwareüberwachung Fehlerzustandsvektoren  $F[k] \neq 0$  zurück liefert, bedeutet das einen diagnostizierten Hardwaredefekt (Roboterkopf: z. B. defekte Elektromotoren oder Winkelencoder, Robotergreifer: z. B. defekte Antriebseinheit (Pumpe) oder Leckageverluste in den Leitungen). Das Auftreten dieser u. U. schwerwiegenden Fehler verursacht sicherheitshalber ein gezieltes "Herunterfahren" der Aktionsfolge ( $T_{T,Ca}$  aktiv).

Die vierte Regel stoppt die Aktionsfolge z.B. nach einem Benutzereingriff. Die fünfte Regel entspricht einem finalen Abbruch wegen einem Misserfolg der Aktion "Objekt verfolgen". Regeln für ein erfolgreiches Abarbeiten der Aufgabe (Konklusion  $T_{T,Su}$ ) existieren erst für die finale Markenbelegung von Aktion  $A_{G,4}$  nach erfolgreichem Abschluss des kompletten Greifvorgangs.

Weitere Dialoge (z.B. der Roboter schafft es nicht, das Objekt anzuheben usw.) können nun regelbasiert bei jedem erkannten spezifischen Stopped-Zustand initiiert werden und müssen nicht extra geplant werden. Je nach Ergebnis des Dialogs wird die Aktionsfolge fortgesetzt oder definiert abgebrochen.

## 4.2 Simulationsergebnisse

In [21] wurden die mathematischen Modelle der Teilkomponenten des Roboterkopfes (Motor, Getriebe, Zahnriemen, Hals mit den vier Freiheitsgraden unteres Nicken, Halsneigen, Halsdrehen und oberes Nicken) sowie Regelalgorithmen zur Positionsregelung im Gelenkwinkelraum bzw. kartesischen Raum vorgestellt und als Gesamtsystem simulativ und experimentell evaluiert. Beim Robotergreifer des Forschungszentrums Karlsruhe [22] werden im Gegensatz zu herkömmlichen Roboterhänden [23, 24] die Freiheitsgrade nicht durch Elektromotoren, sondern fluidisch durch flexible Aktoren angetrieben. Die mathematischen Modelle der Teilkomponenten dieses flexiblen Robotergreifers (Pumpe, Ventile, Leitungen, Fingermechanik) und ein hybrider Momenten-Positionsregelalgorithmus wurden in [25] präsentiert. Auf diesen Modellen aufbauend erfolgt nun die simulative Umsetzung der in Abschnitt 4.1 vorgestellten Aktionsfolge "bildüberwachtes Greifen".

Bild 6 zeigt die Markierung von  $A_{A,Kopf,p}$  mit  $p = [1 \dots 5]$  Stellen der Aktion "Objekt verfolgen" (a), die Markierung von  $A_{A,Finger,n,q}$  mit  $n = 2$  am Griff beteiligten Fingern und die Stellen der Aktion "Greifen" (b) sowie die Markierung von  $A_{T,r}$  mit  $r = [1 \dots 6]$  Stellen im Verwaltungsnetz (c) über der Zeit.

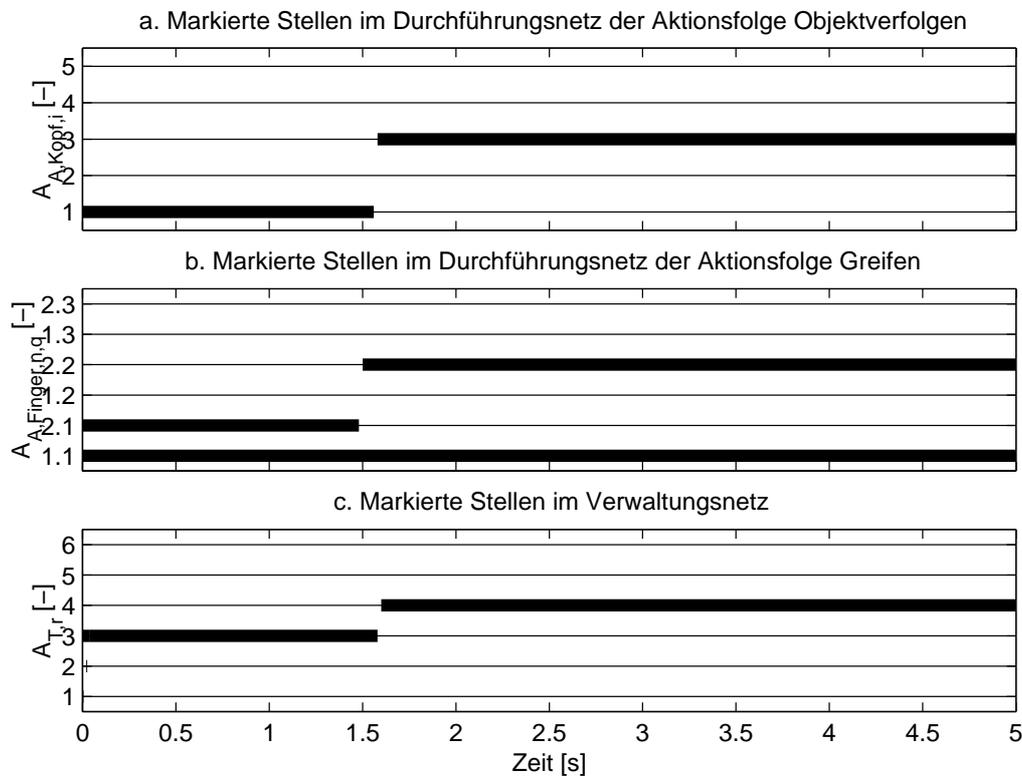


Bild 6: Simulationsergebnisse für die Aktionsfolge "bildüberwachtes Greifen".

Bis zum Zeitpunkt  $t = 1.68$  s befindet sich das zu greifende Objekt im Sichtfeld der Kameras und die Stelle  $A_{A,Kopf,1}$  der Aktion "Objekt verfolgen" ist markiert (Bild 6a). Das Ziel der Regelung der Roboter-Halseinheit ist es, den Bildpunkt des zu verfolgenden Objekts immer in der Nähe eines bestimmten Punkts im Bildkoordinatensystem der Kamera (z.B. Mittelpunkt) zu positionieren. Bei Objektbewegungen im Weltkoordinatensystem (WKS) muss der Regler die Halseinheit mitbewegen, um die Position im Bildkoordinatensystem zu halten. Der Greifer nähert sich dem zu greifenden Objekt an und für den ersten Finger ergibt sich zum Zeitpunkt  $t = 1.52$  s Objektkontakt (Bild 6b, Markierung von  $A_{A,Finger,2,2}$  im Durchführungsnetz). Der zweite am Griff beteiligte Finger hat noch keinen Objektkontakt (Bild 6b, Markierung von  $A_{A,Finger,1,1}$  im Durchführungsnetz).

Zum Zeitpunkt  $t = 1.68$  s erfolgt ein Eingriff in die geplante Aktionsfolge durch den Benutzer. Die Stelle  $A_{A,Kopf,4}$  der Aktion "Objekt verfolgen" wird markiert (Bild 6a). Da Benutzereingriffe 'oberste' Priorität haben, wird der Roboterkopf von der Aufmerksamkeitssteuerung in Richtung des lokalisierten Benutzers bewegt. Das zu greifende Objekt gerät außerhalb des Sichtfeldes der Kameras. Da ohne Blickkontakt keine erfolgreiche Durchführung der ursprünglich geplanten Greifvorgangs zu erwarten ist, stoppt die Koordinierungsüberwachung die Aktionsfolge ( $A_T = A_{T,3}$  in Bild 6c). Es werden nun die in der Stopped-Stelle hinterlegten Callbackfunktionen aller aktiven Elementaraktionen ausgeführt (Tabelle 3 und 4). Für die Aktion "Objekt verfolgen" bedeutet dies keine weitere Änderung der Strategie. Der Kopf wird in Richtung des lokalisierten Benutzers bewegt.

Ist das Ziel erreicht, wird die Bewegung gestoppt (Tabelle 3). Bei der Aktion "Greifen" hingegen erfolgt bei Ausführung der Stopped-Funktion ein Warten, d. h. Pumpe aus, Ventile zu ( $A_{A,Finger,n,1}$  und  $A_{A,Finger,n,2}$ ). Falls das Objekt schon gegriffen ist ( $A_{A,Finger,n,3}$ ), erfolgt stattdessen eine Strategie zur Stabilisierung des momentanen Griffs (Objekt gerade und halten).

Nach Abschluss des Dialogs und einem erneuten Lokalisieren des Objektes sowie einer weiteren expliziten Benutzerfreigabe und dem Prüfen aller Vorbedingungen kann der Griff fortgesetzt werden, was in der Simulation nicht mehr dargestellt ist.

Das Konzept gewährleistet eine weitgehende Entkopplung der Fehlerbehandlung und der routinemäßigen Durchführung einer Aktionsfolge. Eine dennoch erwünschte Kopplung bei der Situationseinschätzung kann bei Bedarf in die Callbackfunktionen der Stopped-Stellen integriert werden, was allerdings die genannten Vorteile wieder reduziert.

## 5 Zusammenfassung und Ausblick

In diesem Beitrag wurde eine hierarchische Steuerungsarchitektur für komplexe Robotersysteme vorgestellt, die neben der Planung und Koordination eine einfache und systematische Behandlung von Unterbrechungen und Ausnahmesituationen ermöglicht. Der Vorteil der vorgestellten Strategie liegt darin, dass der genaue Ablauf bei Ausnahmesituationen nicht mehr explizit im Voraus implementiert werden muss. Zudem können Rücksprünge in Aufgaben konsistent programmiert werden, indem innerhalb einer Aktionsfolge das Verwaltungsnetz gestoppt wird und die entsprechende Aktionsfolge abgearbeitet wird.

Eine experimentelle Umsetzung des vorliegenden Konzeptes steht derzeit noch aus und ist erst nach einer weitergehenden simulativen Erprobung geplant. Bisher wurden lediglich einige Teilkomponenten am humanoiden Roboter ARMAR [1] praktisch erprobt (z.B. handimplementierte Aktionsfolgen).

Die zukünftige Evaluierung und Weiterentwicklung des Konzepts wird sich hauptsächlich auf eine Untersuchung der umfangreichen Wechselwirkungen innerhalb der verschiedenen Komponenten, auf die Praktikabilität des Entwurfs der erforderlichen Bestandteile und die Einbindung von Lernalgorithmen konzentrieren. Nach dieser Evaluierung soll geprüft werden, inwieweit sich das Konzept für die standardisierte Nutzung in humanoiden Robotern eignet.

**Danksagung:** Diese Arbeit wird von der Deutschen Forschungsgemeinschaft (DFG) im Rahmen des Sonderforschungsbereiches (SFB) 588 "Humanoide Roboter- Lernende und kooperierende multimodale Roboter" gefördert.

## Literatur

- [1] Asfour, T.; Berns, K.; Dillmann, R.: The Humanoid Robot ARMAR: Design and Control. *Proceedings of the 1st IEEE-RAS International Conference on Humanoid Robots* (2000).
- [2] Bonasso, R.; Firby, R.; Gat, E.; Kortenkamp, D.; Miller, D. P.; Slack, M. G.: Experiences with an Architecture for Intelligent, Reactive Agents. *Journal of Experimental & Theoretical Artificial Intelligence* 9(2) (1997), S. 237–256.

- [3] Firby, R. J.: Building symbolic primitives with continuous control routines. In: *Proc. 1st Int. Conf. on Artificial Intelligence Planning Systems*, S. 62–69. ISBN 1-55860-250-X. 1992.
- [4] Connell, J.: SSS: A Hybrid Architecture Applied to Robot Navigation. In: *Proc., IEEE Conf. on Robotics and Automation (ICRA)*, S. 2719–2724. 1992.
- [5] Gat, E.: Three-Layer Architectures. In: *Artificial Intelligence and Mobile Robots: Case Studies of Successful Robot Systems* (Kortenkamp, D., Hg.), S. 195–210. AAAI Press. 1998.
- [6] Petri, C. A.: *Kommunikation mit Automaten*. Dissertation, Universität Bonn. 1962.
- [7] Finkemeyer, B.; Kröger, T.; Wahl, F.: Aktionsprimitive: Ein universelles Roboter-Programmierparadigma. *at-Automatisierungstechnik* (2005), S. 189–196.
- [8] Milutinovic, D.; Lima, P.: Petri Net Models of Robotic Tasks. In: *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, S. 4059 – 4064. 2002.
- [9] Asfour, T.; Ly, D.; Regenstein, K.; Dillmann, R.: Coordinated Task Execution for Humanoid Robots. *Proceedings of the 9th International Symposium on Experimental Robotics 2004 (ISER 04)* (2004).
- [10] Kobayashi, K.; Nakatani, A.; Takahashi, H.; Ushio, T.: Motion Planning for Humanoid Robots Using Timed Petri Net and Modular State Net. In: *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics*. 2002.
- [11] Kim, G.; Chung, W.; Kim, M.; Lee, C.: Tripodal Schematic Design of the Control Architecture for the Service Robot PSR. In: *Proc., IEEE Int. Conf. On Robotics and Automation (ICRA)*, S. 2792 – 2797. 2003.
- [12] Sciavicco, L.; Siciliano, B.: *Modelling and Control of Robot Manipulators*, Bd. 2. Springer Verlag. 2001.
- [13] Hörmann, A.: *Begleitende Montageablaufplanung für ein sensorgestütztes Zweiarm-Manipulatorsystem*. Dissertation, Universität Karlsruhe (TH). 1992.
- [14] Wang, F.-Y.; Saridis, G. N.: A coordination theory for intelligent machines. *Automatica* 26(5) (1991), S. 833–844.
- [15] Beetz, M.; McDermott, D.: Improving Robot Plans During Their Execution. *Proceedings of the AI Planning Systems Conference* (1994), S. 3–12.
- [16] McDermott, D.: Transformational Planning of Reactive Behavior. Techn. Ber. YALEU/DCS/RR-864, Yale University. 1991.
- [17] Fernandez, J. A.; Gonzalez, J.: *Multi-Hierarchical Representation of Large-Scale Space*. Kluwer Academic. 2001.
- [18] Lehmann, A.; Mikut, R.; Osswald, D.: Low-Level Finger Coordination for Compliant Anthropomorphic Robot Grippers. In: *Proc., CDC/ECC (Akzeptierter Beitrag)*. 2005.
- [19] Lehmann, A.; Mikut, R.; Bretthauer, G.: Online-Stabilitätsüberwachung strukturvariabler Roboterregelungen. In: *VDI-Bericht Nr. 1841*, S. 55–63. 2004.
- [20] Kuntze, H.-B.; Frey, C.; Giesen, K.; Milighetti, G.: On a smart structure variable supervisory control concept for humanoid robots. In: *Humanoids 2003 International Conference on Humanoid Robots, Karlsruhe and Munich (Conference Documentation and CD-ROM)*. 2003.
- [21] Angelbauer, K.: *Modellbildung und Reglerentwurf für die Halseinheit eines humanoiden Roboters*. Diplomarbeit, Universität Karlsruhe (TH). 2004.
- [22] Schulz, S.; Pylatiuk, C.; Reischl, M.; Martin, J.; Mikut, R.; Bretthauer, G.: A lightweight multifunctional prosthetic hand. *Robotica* 23(3) (2005), S. 293–299.
- [23] Butterfass, J.; Grebenstein, M.; Liu, H.; Hirzinger, G.: DLR Hand II: Next generation of a dextrous robot hand. In: *International Conference on Robotics and Automation, Seoul, Korea*. 2001.
- [24] Lovchik, C. S.; Diftler, M. A.: The Robonaut Hand: A Dexterous Robot Hand for Space. In: *Proceedings of the IEEE ICRA*, S. 907–913. Detroit. 1999.
- [25] Lehmann, A.; Martin, J.; Mikut, R.; Bretthauer, G.: Konzepte zur Regelung und Stabilitätsüberwachung beim Greifen mit flexiblen Robotergriffen. In: *Proceedings, 13. Workshop Fuzzy Systeme des GMA-Fachausschuss 5.22., Dortmund*, S. 79–98. 2003.

# Zur Modellbildung kontextabhängiger Systeme

**Hanns Sommer**

Universität Kassel

Mönchebergstr. 7

34109 Kassel

Tel: (0561) 804 3261

Fax: (0561) 404 7768

E-Mail: sommer@rts.maschinenbau.uni-kassel.de

## Zusammenfassung

Können mit der Fuzzy-Technologie Probleme angegangen werden, zu denen sonst kein anderer Zugang besteht? Im vorliegenden Beitrag soll begründet werden, dass diese Frage für kontextabhängige Systeme zu bejahen ist. Wir werden aufzeigen, dass Kontextabhängigkeit nicht nur auf die Quantenmechanik beschränkt ist, wo sie vom Theorem von Kochen und Specker gefordert wird, sondern auch in den Wirtschaftswissenschaften allgegenwärtig ist. Bezugnehmend auf eine Arbeit von D.M. Appleby, wird nachgewiesen, dass kontextabhängige Systeme nicht in konsistenter Weise mit klassischen mathematischen Methoden und insbesondere mit Verfahren der Wahrscheinlichkeitsrechnung modellierbar sind. Eine Modellbildung solcher Systeme, die auf den Methoden der Fuzzy-Logik und der Glaubenskalküle begründet ist, ist dagegen möglich. Es wird an eine Erweiterung des Dempster-Shaferschen Glaubenskalküls auf die Fuzzy-Logik erinnert und es wird gezeigt, wie mit diesem Ansatz, aus sehr einfachen und grundlegenden Prinzipien, Modelle für kontextabhängige Systeme gewonnen werden können. Als Folgerung werden dann einige Phänomene der Quantenmechanik einfach erklärbar.

## 1 Kontextabhängige Systeme

Kontextabhängige Systeme sind solche Systeme, bei denen jeder Teil seine Eigenschaften nur in seinem Bezug zum Gesamten erhält. So sollen zum Beispiel in einem Warenhaus die Waren so aufgestellt werden, dass sie:

1. von den Kunden gefunden werden können,
2. den Kunden "ins Auge springen", und
3. von diesen "als attraktiv" empfunden werden.

Diese Forderungen sind nicht widerspruchsfrei. Forderung 2 kann bedeuten, dass ein sachfremdes Produkt zwischen andern Produktgruppen aufgestellt wird, und Forderung 3, dass gleiche Produkte, die sich wegen Preisunterschieden gegenseitig "als unattraktiv" erscheinen ließen, auseinandergestellt werden müssen. Die üblicherweise in Warenhäusern durchgeführte Mischkalkulation von Billigprodukten (zur Kundenwerbung) und Standardprodukten (zur Gewinnerwirtschaftung) macht das Anordnungsproblem keineswegs einfach lösbar. Es geht darum, für jedes Produkt den ihm entsprechenden Kontext zu schaffen, in dem es möglichst viele Käufer findet. Am angeführten Beispiel können wir die allgemeine Definition kontextabhängiger Systeme ablesen:

In einem *kontextabhängigen System* passt jeder Teil bestmöglichst zum Gesamten, so dass dieses optimal wird.

Was unter "optimal" zu verstehen ist, ist dabei in der Regel nicht einfach zu definieren. Im Beispiel des Warenhauses soll der Verkaufsgewinn maximiert werden. Aber selbst diese Forderung ist nicht einfach zu definieren. Ein Kunde mag kurzfristig, weil er durch die Warenanordnung durch das Kaufhaus geführt wurde, mehr als geplant eingekauft haben, aber insgesamt enttäuscht sein und nicht wiederkommen, weil er "das was er wollte" nicht schnell finden konnte.

Insbesondere in den Wirtschaftswissenschaften und im Bereich des "economical engineering" wird in letzter Zeit in der Literatur verstärkt auf die Bedeutung kontextabhängiger Systeme hingewiesen [1], und es wird der Zusammenhang betont, den diese Systeme mit den bekanntesten kontextabhängigen Systemen der Physik, den quantenmechanischen Systemen, haben [2].

Die Kontextabhängigkeit quantenmechanischer Systeme wurde 1967 von Simon Kochen und Ernst Specker nachgewiesen und sie wurde seitdem als zentrale Eigenschaft quantenmechanischer Systeme erkannt. Ich möchte hier aufzeigen, dass kontextabhängige Systeme durchaus eine gemeinsame Systemklasse bilden, so dass Ergebnisse von kontextabhängigen ökonomischen Systemen zum Verständnis quantenmechanischer Systeme beitragen können und umgekehrt. Das Hauptproblem bei der Vereinheitlichung besteht in der Festlegung des Optimalitätsbegriffs. Wie wir am Beispiel des Warenhauses gesehen haben, ist uns dieser für ökonomische oder auch für biologische und soziale Systeme meist nicht direkt zugänglich. Optimalität hängt ab von Intentionen, die mit dem System verbunden sind, und diese sind uns oft verborgen.

"Sinn entsteht bezogen auf Intentionen"

Was die Physik im Vergleich zu sozialen Systemen einfacher macht ist, dass wir die Intentionen der Physiker eindeutig beschreiben können, durch das *Prinzip der bestmöglichen Vorhersage*:

"Ordne das physikalische Wissen so, dass Vorhersagen bestmöglichst ermöglicht werden."

Dieses Prinzip, das dem in der klassischen Physik bekannten Prinzip der kleinsten Aktion entspricht [8], erlaubt es uns quantenmechanische Systeme zu charakterisieren.

## **2 Bedeutung quantenmechanischer Systeme und einige Gründe, warum diese so schwer zu verstehen sind**

Insbesondere in der Nanotechnologie und beim Versuch Quantencomputer für schnelle Berechnungsverfahren einzusetzen, kommen auch Ingenieure immer stärker in die Notwendigkeit, quantenmechanische Systeme zu verstehen. Es gibt Prognosen, dass ab dem Jahre 2015 die Hälfte aller Innovationen von dieser Technologie abhängen werden [3]. Es interessiert uns daher zu wissen:

- Ab welcher Größenordnung müssen wir mit quantenmechanischen Effekten rechnen?

- Wie machen sich diese Effekte bemerkbar?
- Welche Denkweisen müssen wir zur Manipulation quantenmechanischer Systeme einsetzen?

Besonders die letzte Frage ist nicht einfach beantwortbar. Einer der Gründungsväter der Quantencomputation, Peter W. Shor, schreibt in einem Übersichtsartikel anlässlich des 20-jährigen Bestehens der ersten schnellen Quantenberechnungsverfahren, dass das Ausbleiben weiterer, größerer Fortschritte auf diesem Gebiet mit der "Andersartigkeit des Denkens" zu tun hat, das hier erforderlich wäre [4].

Der wesentliche Unterschied, der kontextabhängige von klassischen Systemen unterscheidet, besteht darin, dass wir uns bei klassischen Systemen auf "Wo-Fragen" beschränken können. Wir fragen:

"Wo ist ein Teilchen?" oder "Wo steht der Zeiger des Messinstruments?"

Bei kontextabhängigen Systemen müssen wir dagegen notwendigerweise auch "Was-Fragen" hinzunehmen. Schon beim Warenhausbeispiel haben wir gesehen, dass "Wo-" und "Was-Fragen" nicht getrennt werden können. Es geht nicht nur darum: "Wo platziere ich meine Ware?" sondern auch darum: "Was bedeutet Attraktivität?"

Wie entsteht Attraktivität in dem Kontext, in dem ich meine Ware präsentiere?

Wenn wir aber "Wo-" und "Was-Fragen" nicht mehr auseinanderhalten können, so können wir mittels Häufigkeitsbetrachtungen nicht mehr direkt zu einer Wahrscheinlichkeitstheoretischen Beschreibung kommen. Darauf hat D.M. Appleby für quantenmechanische Systeme in überzeugender Weise hingewiesen. Er schreibt sinngemäß [5]:

Wenn wir nicht wissen, "was für ein Ereignis" auftritt, so können wir dessen Häufigkeit nicht bestimmen.

Wenn Was- und Wo-Fragen verknüpft sind, so kommen wir ganz zwangsläufig zu einer der Fragestellungen, die Lotfi Zadeh als eigentliches Ziel der Fuzzy-Logik in seinen Arbeiten seit Jahren immer wieder fordert, dem Verständnis der Bedeutung der Granulierung [6]:

Wie entstehen Einheiten (Granuli) im Kontext einer Gesamtheit?

Wir kommen zu Fragestellungen, in denen die Fuzzy-Logik geeigneter erscheint als die Wahrscheinlichkeitsrechnung.

### **3 Zwei prinzipielle Sichtweisen kontextabhängiger quantenmechanischer Systeme**

Es gibt zwei prinzipielle Möglichkeiten, quantenmechanische Systeme zu beschreiben. Die erste Möglichkeit, diejenige der *Theorien verborgener Variablen (hidden variable theories)*, geht davon aus, dass uns ohne unser klassisches rationales Verstehen kein Verständnis möglich ist. Sie setzt also voraus, dass es eine klassisch beschreibbare Welt gibt, deren Beschreibungsvariablen uns aber weitgehend verborgen sind.

Diese Sichtweise entspricht der Kant'schen Philosophie, in der "das Ding an sich" unerkennbar und von unserem Wissen über es verschieden ist. Alle "seltsamen Phänomene der Quantenmechanik", wie die Kontextabhängigkeit und andere "No-Go-Theoreme", die Nichtkopierbarkeit von Zuständen, das No-Signaling-Theorem, die Unschärferelation, etc. entstehen nicht in der Welt selbst, sondern erst durch das Beobachtungsfiler, das uns den Zugang zur Welt gewährt.

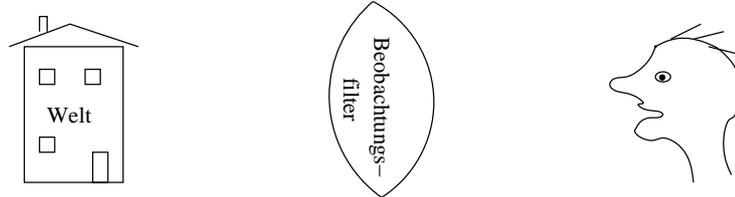


Abbildung 1: Beobachtung der Welt in der Kant'schen Sichtweise

Die zweite Sichtweise, die *epistemologische Sichtweise* versucht die Entstehung der Phänomene in der Welt zu erklären als Ereignisse einer äußeren Welt. Drei wesentliche Voraussetzungen sind notwendig zur Konstituierung einer objektiven äußeren Welt:

1. Die Vermittlung der äußeren Welt durch Messwerte. Da es uns hier um die Darstellung der Prinzipien geht, nehmen wir diese an, in der einfachsten Form als Messtupel:  $\{m_{i,j}\}_{i=1,\dots,N,j=1,\dots,K}$  oder  $\{m_i = (m_{i,j})_{j=1,\dots,K}\}_{i=1,\dots,N}$ .
2. Unsere Beziehung zur Welt, die festgelegt ist durch die Bedeutung, die die Begriffe: "groß, mittel, klein, sehr klein, etc." für uns haben. Diese Begriffe werden mittels der Fuzzy-Logik festgelegt.
3. Die Intentionen, mittels denen wir unser Bild der Welt formen. Für Physiker sind diese Intentionen eindeutig darstellbar mit dem *Prinzip der bestmöglichen Prognose*:

"Ordne das Wissen so, dass Prognosen bestmöglichst ermöglicht werden."

Mit diesen Voraussetzungen können wir dann Sätze bilden, z. B.:

$s$ : FALLS  $m_{i,2}$  klein und  $m_{i,3}$  groß und  $m_{i,4}$  mittel ist, DANN ist  $m_{i,5}$  groß.

Die Zugehörigkeit  $s(m_i)$  eines Messwerts  $m_i$  zum Satz  $s$  kann als Bestätigung dieses Satzes durch den Messwert aufgefasst werden. Mittels dem *Dempster-Shafer'schen Glaubenskalkül* ist dann die Glaubwürdigkeit, die Sätze ( $s$ ) durch die gesamten Messdaten  $\{m_i\}_{i=1,\dots,N}$  erhalten, festgelegt durch die Gleichung

$$Bel(s) := \bigvee_{i=1}^N s(m_i) = s(m_1) \vee s(m_2) \vee \dots \vee s(m_N)$$

mit  $a \vee b = 1 - (1 - a) \cdot (1 - b)$  für  $a, b \in [0, 1]$ .

Die Idee zur *Bildung des Wissens* besteht nun darin, die Sätze auszuwählen, die *relevante Aussagen* über die Messdaten machen. Dabei ist eine Aussage  $s$  dann relevant, wenn sie uns z. B. den Ort, an dem wir uns gerade befinden angibt, d. h. wenn sie bezüglich der Messwerte  $m_i$ , die an diesem Ort entstehen wahr ist ( $s(m_i) \approx 1$ ) und für Messwerte  $m_j$  anderer Orte nicht gilt ( $s(m_j) \approx 0$ ). Da wir außer den Messdaten zunächst über keinerlei weitere Informationen verfügen, so bedeutet dies:

Ein Satz  $s$  ist *relevant*, wenn er auf einem Teil der Messdaten möglichst stark bestätigt und auf den anderen Messdaten stark verneint ist.

Mittels dem Dempster-Shafer'schen Glaubenskalkül haben wir die Möglichkeit die relevanten Sätze auszuwählen. Dabei ist zusätzlich zu berücksichtigen, dass ein sehr langer, komplizierter Satz relevanter sein muss um ausgewählt zu werden, als ein kurzer Satz. Formal gehen wir so vor, dass wir Sätze mit einer universellen Turingmaschine konstruieren und die Relevanz dieser Sätze in Bezug setzen zu ihrer kürzesten Beschreibungslänge (Kolmogoroff-Komplexität). Wir betrachten also nicht die Bestätigung, die der Satz als Ganzer erhält, sondern diejenige für einen Buchstaben seiner kürzesten Beschreibung. Es kann gezeigt werden, dass diese Definition eindeutig ist, und nach einem Satz von Kolmogoroff ist sie auch unabhängig von der Auswahl der universellen Turingmaschine [7].

Ein Ort in Raum und Zeit ist durch Wissen, d. h. durch eine Anzahl von relevanten Sätzen, beschrieben, deren Zusammengehörigkeit von den Messdaten bestätigt ist.

Zur Definition der Topologie der Raumzeit verwenden wir das Prinzip der bestmöglichen Prognose und definieren diese so, dass Prognosen bestmöglichst ermöglicht werden. Dies bedeutet dass zwei Teile der Raumzeit um so näher liegen, je besser das Wissen aus einem der Teile Prognosen bezüglich des anderen gestattet.

Diese Ideen werden mittels folgender Definition formalisiert [8].

**Definition der Raumzeit:**

- (A)  $S_r$  beschreibe die Menge der ausgewählten relevanten Sätze.
- (B) Sätze, die durch die selben Messdaten bestätigt (bzw. verneint) werden, werden in Teilmengen  $S_p \subset S_r$  zusammengefasst.  
 $S_p$  ist die Beschreibung eines Punktes  $p$  der Raumzeit.  
 $W_{\text{opt}} = \{p \mid p \text{ ist durch } S_p \text{ beschrieben}\}$  beschreibt den Teil der Raumzeit, der durch die Messdaten  $M := \{m_i \mid i = 1, \dots, N\}$  erfasst wird.
- (C) Die Topologie auf  $W_{\text{opt}}$  wird bestimmt durch die Ähnlichkeit (Fuzzy-Übereinstimmung) der bestätigten (bzw. verneinten) Sätze in  $S_{p_1}$  und  $S_{p_2}$  für zwei Punkte  $p_1, p_2 \in W_{\text{opt}}$ .
- (D) Eine Uhr zur Bestimmung der Zeit ist definiert als eine Abbildung

$$t : [n_b, n_{b+1}, \dots, n_e] \times U \rightarrow W_{\text{opt}}$$

mit  $[n_b, n_{b+1}, \dots, n_e] \subset \mathbb{Z}$ ,  $U$  endliche Menge,  
so dass für ein festes  $T \in \mathbb{N}$  und alle  $n \in [n_b, n_{b+1}, \dots, n_{e-T}]$ ,  $u_1, u_2 \in U$  gilt:  
die Nachbarschaftsbeziehungen zwischen  $t(n, u_1)$  und  $t(n, u_2)$  sind die selben wie diejenigen zwischen  $t(n + T, u_1)$  und  $t(n + T, u_2)$ .  
(Gültigkeit bedeutet starke Bestätigung mittels dem Dempster-Shafer'schen Glaubenskalkül.)

Die *Zeit* wird bezüglich der Raumzeit  $W_{\text{opt}}$  durch die Uhr definiert, für die obige Eigenschaften am besten bestätigt sind.

*Physikalische Gesetze* sind Sätze, die auf  $W_{\text{opt}}$  stark bestätigt sind. Für solche Sätze erhalten wir auch eine Bestätigung außerhalb des Zeitintervalls  $[n_b, n_{b+1}, \dots, n_e]$  und können somit Prognosen für zukünftige Messungen mittels physikalischen Gesetzen gewinnen [7].

Ein *physikalischer Zustand* ist beschrieben durch unser Wissen in einem gewissen Gebiet der Raumzeit. Im Formalismus der Quantenmechanik wird dieses Wissen durch eine *Wellenfunktion* dargestellt.

Die Gesetze, die die Zeitentwicklung dieses Wissens beschreiben, bezeichnen wir als *Schrödingergleichung*.

Unsere Interpretation des physikalischen Zustandes ist rein epistemologisch. Die Quantenmechanik ist in dieser Sichtweise eine Theorie über Information [9],[10]. Es handelt sich dabei um die Information, die uns durch die Messdaten zugänglich ist.

Die Topologie der Raumzeit wird durch unser Wissen bestimmt. Die Gültigkeit identischer Aussagen lässt Raumgebiete zusammenrücken, wohingegen diese durch unterscheidende Aussagen auseinandergerückt werden. Alle Aussagen erhalten ihre Glaubhaftigkeit im Bezug zum gesamten Wissen. Ihre Gültigkeit ändert sich bei Ortswechseln in diskreten Gültigkeitsgraden, die im Bezug zur Gültigkeit der anderen Sätze festgelegt sind. Zwischen den Gültigkeitsgebieten verschiedener Aussagen-gruppen existieren Trennungsgebiete.

## 4 Eigenschaften quantenmechanischer Systeme

In der beschriebenen Raumzeitstruktur  $W_{\text{opt}}$  können viele Eigenschaften, die für quantenmechanische Systeme charakteristisch sind, wiederentdeckt werden. Dies ist im hier untersuchten Zusammenhang insofern interessant, da wir dadurch nachweisen, dass Kontextabhängigkeit nicht eine Eigenschaft quantenmechanischer Systeme ist, die diese neben vielen anderen Eigenschaften haben, sondern dass Kontextabhängigkeit die wesentliche, konstituierende Eigenschaft quantenmechanischer Systeme darstellt. In einer Reihe von Arbeiten wurden die folgenden Eigenschaften in der Struktur  $W_{\text{opt}}$  bisher nachgewiesen:

- Heisenberg's Unschärferelation [8]
- Quanten Zeno-Effekt [11]
- Verschränkung von Messwerten (EPR-Gedankenexperiment) [11]
- Existenz einer maximalen Geschwindigkeit [7]
- Feynman-El Naschie-Vermutung [8]
- Bedeutung von Einheiten und Elementarteilchen [8]
- Fraktalität der Raumzeit [8]
- Doppelspaltexperiment [8]

Hier soll nur auf das Young'sche Doppelspaltexperiment eingegangen werden, da dieses, wie Feynman festgestellt hat, das wesentliche Experiment zum Verständnis der Quantenmechanik ist.

Das *Young'sche Doppelspaltexperiment* zeigt, dass das Wissen (d. h. die Wellenfunktion) nicht nur von den reinen Experimentdaten, sondern auch von den Bedingungen abhängt, unter denen dieses konstituiert wird.

Wir betrachten das Doppelspaltexperiment in einer Anordnung, in der wir jeden Störeinfluss durch den Messprozess ausschließen.

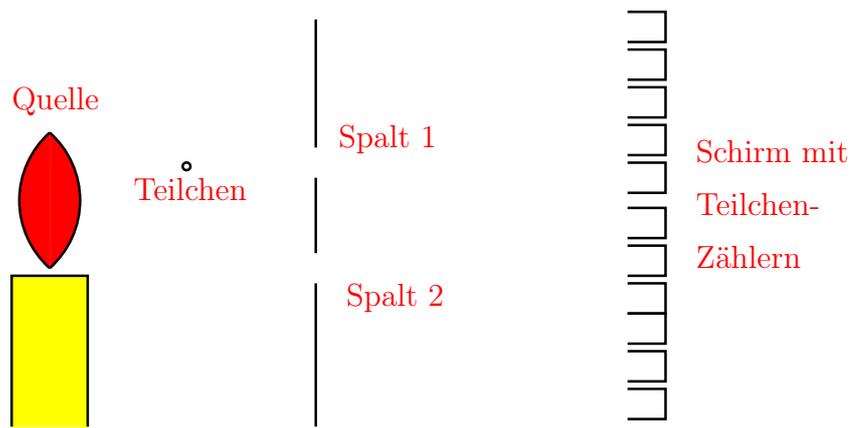


Abbildung 2: Young'sches Doppelspaltexperiment

Die Quelle emittiert Teilchen in so geringer Anzahl, dass jede Interaktion zwischen diesen ausgeschlossen werden kann. Jedes Teilchen fliegt alleine durch einen der Spalte.

Wir führen das Experiment auf zwei Arten durch:

- (A) Beide Spalte sind 1 Sekunde lang geöffnet.  
Am Schirm erhalten wir ein Interferenzmuster.
- (B) Eine Sekunde ist Spalt 1 geöffnet und Spalt 2 geschlossen. Anschließend wird umgekehrt 1 Sekunde Spalt 2 geöffnet und Spalt 1 bleibt geschlossen.  
In diesem Fall erhalten wir keine Interferenz sondern die Überlagerung der Ereignisse beider Zeitabschnitte.

Die unterschiedlichen Ergebnisse dieser beiden Ausführungsarten des Experiments stellen die klassische Denkweise vor große Erklärungsprobleme. Da jedes Teilchen einzeln durch einen der Spalte fliegt, so müssen wir in der klassischen Denkweise (die den Theorien verborgener Variablen und insbesondere der Bohmschen Interpretation der Quantenmechanik zugrundeliegt [12],[13]) annehmen, dass das Teilchen durch seine Wellenfunktion über die Situation (offen,geschlossen) auch an dem Spalt beeinflusst wird, den es selbst "niemals sieht". Die Wellenfunktion stellt in dieser Sichtweise eine "reale Größe" dar, durch deren Beeinflussung die Welt geändert werden kann (Dieses Ergebnis wird durch Untersuchungen des EPR-Experiments [14] und des Quanten-Zeno-Effekts [8] bestätigt).

Es liegt am Beobachtungsfiler, das unseren Zugang zur Welt beschränkt, dass wir diese Realität der Wellenfunktion nicht zur instantanen Nachrichtenübermittlung nutzen können (No-Signaling-Theorem).

Es erscheint nun zunächst unklar, wie eine epistemologische Interpretation der Wellenfunktion möglich sein könnte, in der diese unser Wissen darstellt, dessen Veränderung auf die Realität natürlich keinen direkten Einfluss haben kann.

Erklären wir das Doppelspaltexperiment in der epistemologischen Sichtweise, so haben wir zunächst im Fall (B), nur einen geöffneten Spalt. Wegen der, durch diesen bewirkten Ortsfixierung geht (auf Grunde der Unschärferelation) jede Information über den Teilchenimpuls verloren. Wir erhalten daher eine Gleichverteilung über die Spalt-Austrittswinkel der Teilchen. Die Überlagerung der Muster, die aus dieser Winkelgleichverteilung für beide Spalte entstehen, ergibt die Verteilung am Schirm.

Für jedes Teilchen besteht ein im voraus eindeutig festgelegter Kontext, in dem es auf den Schirm zufliegt.

Der Fall (A) unterscheidet sich wesentlich vom Fall (B) dadurch, dass nun zwei Ursachen für das Auftreffen eines jeden Teilchens am Schirm existieren. Diese Ursachen werden durch die Sätze  $S_1$  und  $S_2$  beschrieben:

$S_1$ : Das Teilchen hat Spalt 1 passiert.

$S_2$ : Das Teilchen hat Spalt 2 passiert.

Ankunftsorte, in denen  $S_1$  (bzw.  $S_2$ ) gleichgradige Glaubhaftigkeit (bezüglich der anderen Sätze, die unser Wissen repräsentieren) haben, werden wegen der übereinstimmenden Gültigkeit des gleichen Satzes  $S_1$  (bzw.  $S_2$ ) "einander nähergebracht", wohingegen Orte in denen gegenteilige Sätze gelten in der Topologie von  $W_{\text{opt}}$  voneinander entfernt werden. Durch die unterschiedlichen Rahmenbedingungen, die durch die Sätze  $S_1$  und  $S_2$  beschrieben werden, verläuft das Experiment daher für ein einzelnes Teilchen, je nachdem wie glaubhaft für dessen Ankunftsort am Schirm jede der Ursachen  $S_1$  (bzw.  $S_2$ ) ist, in unterschiedlichen Kontexten ab. Diese unterschiedlichen Kontexte wirken ein auf das Ergebnis am Schirm und sind der Grund dafür, dass Fall (A) und Fall (B) unterschiedliche Ergebnisse liefern. Im Fall (B) haben die Sätze  $S_1$  und  $S_2$  keine Relevanz und sind daher einflusslos.

Wir können nun weiter fragen, was ändert sich denn, wenn an Stelle der Teilchen kleine Flugzeuge durch die Spalte fliegen würden?

In der epistemologischen Interpretation ändert sich so lange gar nichts, so lange die Piloten der Flugzeuge nicht eingreifen und uns nicht mitteilen, über welchen Weg sie geflogen sind. Denn unser Wissen bildet sich ausschließlich in dem Kontext, in dem "wir" das Experiment sehen.

Viel komplizierter liegt der Fall in der klassischen Interpretation. Da sich die Bahnen der durch Spalt 1 fliegenden Flugzeuge durch Öffnen beziehungsweise Schließen von Spalt 2 ändern, so müssen wir annehmen, dass die Information (Spalt offen, Spalt geschlossen) ohne Zeitverlust vom Spalt 2 zum Spalt 1 übertragen wird. Diese Information kann aber (gemäß dem No-Signaling-Theorem) nicht zur Nachrichtenübermittlung an die Piloten der Flugzeuge verwendet werden. Es bleiben also die Fragen:

- Warum ist die Verwendung dieser Information durch die Piloten ausgeschlossen?
- Ist die Trajektorie der Flugzeuge von der Anwesenheit intelligenter Piloten abhängig, auch dann, wenn diese nicht eingreifen, sondern nur die Flugbahn beobachten?
- Kann der externe Beobachter die Anwesenheit intelligenter Piloten im Flugzeug erkennen?

Die letzte Frage ist bezüglich der epistemologischen Auffassung und auch auf Grund von Experimenten mit Quantenradierern, bei denen Information im Nachhinein gelöscht wird, mit "nein" zu beantworten. Aber wir sehen aus diesen Überlegungen, dass die realistische Interpretation der Quantenmechanik weniger scharf ist als die epistemologische und daher mehr Fragen offen lässt. Für den Ingenieur, dem es ja um Eingriffsmöglichkeiten in die Systeme geht, ist die Frage nach der Realität der Wellenfunktion von wesentlicher Bedeutung.

## 5 Folgerungen und Schlussbemerkungen

Aus den vorangegangenen Kapiteln können im wesentlichen die folgenden Ergebnisse festgehalten werden:

- I. Die Klasse der kontextabhängigen Systeme bildet eine große Systemklasse, die sowohl zur Beschreibung von Effekten der Sozialwissenschaften und der Wirtschaftswissenschaft, als auch für quantenmechanische Effekte wichtig ist. Anschauliches Verstehen aus den ersten beiden Wissenschaften könnte in der Quantenmechanik hilfreich sein. Deren mathematische Methoden könnten dagegen in ersteren Anwendung finden [2].
- II. Kontextabhängigkeit ist eine wesentliche Eigenschaft quantenmechanischer Systeme, aus der andere Effekte derselben gefolgert werden können.
- III. Die Fuzzy Logik ist 'die' geeignete Formulierungsmethode zur Beschreibung kontextabhängiger Systeme. Die mathematische Auswertung dieser Beschreibung ist jedoch sehr kompliziert (und nicht effektiv, d. h. nicht polynomial schnell berechenbar). Daher sind halb-klassische Darstellungen notwendig [15], zu deren Verständnis und Herleitung allerdings das Verständnis der Kontextabhängigkeit von Vorteil ist, z. B. bei der Frage: "Ab welcher Größenordnung müssen den quantenmechanische Effekte berücksichtigt werden?"

Die Antwort lautet: "Wenn der Kontext anfängt die Experimentergebnisse zu beeinflussen. Nicht die Größenordnung ist ausschlaggebend, sondern die Systemkonfiguration in ihrer Gesamtheit [16]."

Ein Beispiel für eine solche Einflussnahme des Kontexts ist der Casimir-Effekt:

"Zwischen zwei eng aneinanderstehenden Metallplatten besteht eine Anziehungskraft, da zwischen ihnen weniger quantenmechanische Zustände möglich sind als im Außenraum und somit diese überzähligen Zustände oder Teilchen einseitig auf die Platten wirken."

Die quantenmechanische Betrachtungsweise wird daher dann notwendig, wenn der Kontext, in dem Teilchen bestehen können, in die Betrachtung mit einbezogen werden muss.

- IV. Die letzte wesentliche Frage, die wir angesprochen haben, ist diejenige nach der Realität der Wellenfunktion. Diese Frage ist in den Ingenieurwissenschaften von großer Bedeutung, wenn es um die Einwirkung auf die Systeme geht. Wir haben gesehen, dass in der epistemologischen Sichtweise, die Wellenfunktion ausschließlich unser Wissen repräsentiert. Aber diese Frage ist offen.

Jede Wissenschaft muss ihren eigenen Zugang zu ihren Forschungsgegenständen finden. Vor dieser Aufgabe steht die Ingenieurwissenschaft heute bezüglich den quantenmechanischen, bzw. kontextabhängigen Systemen. Ziel dieses Vortrags ist es, ein Grundverständnis für diese Systeme zu vermitteln.

## Literatur

- [1] Markose, S.-M.: Computability and evolutionary complexity: Markets as complex adaptive systems. *The Economic Journal* 115 (2005), S. 159–192.

- [2] Haven, E.: Emergence of fuzzy preferences for the risk in a Birkhoff-von Neumann logics environment. *Fuzzy Sets and Systems* 153 (2005), S. 29–43.
- [3] Roco, M.-C.; Bainbridge, W.-S.: Societal implications of nanoscience and nanotechnology: Maximizing human benefit. *Journal of Nanoparticle Research* 7 (2005), S. 1–13.
- [4] Shor, P.-W.: Progress in Quantum Algorithms. *Quantum Information Processing* 3,1-5 (2004), S. 5–13.
- [5] Appleby, D.-M.: Facts, values and quanta. *Foundations of Physics* 35,4 (2005), S. 627–668.
- [6] Zadeh, L.-A.: From imprecise to granular probabilities. *Fuzzy Sets and Systems* 154,3 (2005), S. 370–374.
- [7] Sommer, H.-J.: Anticipation, an optimisation principle for cognition. *Int. J. of Computing Anticipatory Systems* 14 (2005), S. 710–719.
- [8] Sommer, H.-J.: A toy-world, that satisfies some Principles of El Naschies E-infinity Theory. Erscheint in: *Int. J. of Computing Anticipatory Systems* (2006).
- [9] Peres, A.: Einstein,Podolsky,Rosen and Shannon.  
*Foundations of Physics* 35,4 (2005), S. 511–514.
- [10] Bub, J.: Quantum Mechanics is about Quantum Information. *Foundations of Physics* 35,4 (2005), S. 541–560.
- [11] Sommer, H.-J.: Physical phenomena as sense determinate occurrences. *Chaos, Solitons and Fractals* 24 (2005), S. 7–12.
- [12] Genovese, M.: Research on hidden variable theories: A review of recent progresses. *Physics Reports* 413 (2005), S. 319–396.
- [13] Bokulich, P.; Bokulich, A.: Niels Bohrs generalisation of classical mechanics. *Foundations of Physics* 35,3 (2005), S. 347–371.
- [14] Ferrero, M.; Salgado, D.; Sanchez-Gomez, J.-L.: Is the epistemic view of quantum mechanics incomplete?. *Foundations of Physics* 34,12 (2004), S. 1993–2003.
- [15] Dauger, D.-E.; Decyk, V.-K.; Dawson, J.-M.: Using semiclassical trajectories for the time-evolution of interacting quantum-mechanical systems. *J. Computational Physics* 209 (2005), S. 559–581.
- [16] Davies, E.-B.: Some Remarks on the Foundations of Quantum Theory. *Brit. J. Phil. Sci.* 56 (2005), S. 521–539.

# Applied stochastic integer programming: hybrid evolution strategies vs. exact algorithms

Jochen Till, Guido Sand, Sebastian Engell

Process Control Lab (BCI-AST)  
Department of Biochemical and Chemical Engineering  
Universität Dortmund  
Emil-Figge-Str. 70, D-44221 Dortmund, Germany  
Tel.: +49 (231) 755-5172  
Fax: +49 (231) 755-5129  
E-Mail: s.engell@bci.uni-dortmund.de

## Abstract

This contribution addresses the solution of two-stage stochastic integer programs with mixed-integer recourse arising in dynamical batch scheduling problems. Due to a large probability space and integer variables in the second-stage, these programs are usually of a large scale and cannot be solved easily without incorporating problem specific knowledge.

A new algorithm for general stochastic integer programs based on stage decomposition and a hybrid evolution strategy has been proposed by the authors. The first-stage decisions are handled by an evolution strategy, while the second-stage subproblems are solved by exact mathematical programming techniques. This approach is benchmarked to an exact approach based on scenario decomposition by means of chemical batch scheduling problems under uncertainty. The performance of the algorithms depends strongly on properties of the problem, the evolution strategy based approach performed better for stochastic integer programs without relative complete recourse.

## 1 Introduction

Uncertainties are present in the most engineering and economic optimization problems. An important application of optimization under uncertainty is the scheduling of chemical batch plant operation [20]. Two-stage stochastic integer programming provides systematic methods to deal with these uncertainties, but the huge probability space leads to large-scale optimization problems [19]. However, the two-stage stochastic integer programs show a specific structure which is amenable to decomposition approaches. Recent surveys of theory and application of stochastic programming can be found for example in [4, 18, 19, 24].

The authors have recently proposed a new decomposition based hybrid evolution strategy for two-stage stochastic integer programs [23]. In this paper the new approach is conceptually and numerically compared to a state-of-the-art exact decomposition based approach by means of three test examples from chemical batch scheduling under uncertainty.

The paper is organized as follows: In Sec. 2 we introduce the chemical batch scheduling problem and motivate the formulation of two-stage stochastic integer programs.

Sec. 3 is dedicated to the algorithmic approaches; the proposed algorithm is presented and conceptually compared to alternative exact approaches. The test problem and the experimental setup are described in Sec. 4 and Sec. 5, respectively. The numerical comparison is presented in Sec. 6, before we conclude this paper in Sec. 7.

## 2 Motivation

### 2.1 Chemical batch scheduling under uncertainty

The globalization of markets and the increasing demand for diversified chemical products force the processing industry to utilize flexible production facilities and to operate them in an agile and flexible manner. An important class of flexible production facilities are multi-product batch plants, where different products follow the same sequence of processing steps.

Chemical batch scheduling problems are concerned with the optimal allocation of processing steps (e. g. reaction, separation) to resources (e. g. processing equipment, raw materials) over time with respect to an (economic) objective. This scheduling problems are typically subject to uncertainties e. g. in the equipment availability or in the product demand. The combinatorial nature together with the uncertainties make the scheduling problem hard to solve.

A widely used approach to deterministic chemical batch scheduling problems is mixed-integer linear programming (MILP) [1, 11, 21]. In recent works [10, 20], the deterministic MILP model for the scheduling of a multi-product batch plant for polymers was extended by the aspects of uncertainty, leading to a two-stage stochastic integer program. The general two-stage stochastic integer programming framework is described in the next subsection; the example problem is described in Sec. 4.

### 2.2 Two-stage stochastic integer programming

The information and decision structure of the problems under consideration is characterized by an alternation of decisions and realizations of the uncertain parameters. In two-stage stochastic integer programming, the case with two decision stages is considered. Some decisions (first-stage variables) have to be made under uncertainty in some of the parameters. It is assumed that the precise values of the uncertain parameters become available at a certain point and that the remaining decisions (second-stage, or recourse variables) can be taken on the basis of this knowledge, i. e. in a fully deterministic situation. There is a complex interaction between the stages because the first-stage decisions constrain the space of the second-stage decisions. Here, the uncertainty is modelled by random variables with discrete distribution and each combination of the realizations is called a scenario. A formal representation of a stochastic integer program (*SIP*) in its *extensive* form is given as follows:

$$z^* = \min_{\mathbf{x}, \mathbf{y}_1, \dots, \mathbf{y}_\Omega} \mathbf{c}^T \mathbf{x} + \sum_{\omega=1}^{\Omega} \pi_\omega \mathbf{q}_\omega^T \mathbf{y}_\omega \quad (2.1)$$

$$s.t. \quad \mathbf{A} \mathbf{x} \leq \mathbf{b}, \quad (2.2)$$

$$\mathbf{T}_\omega \mathbf{x} + \mathbf{W}_\omega \mathbf{y}_\omega \leq \mathbf{h}_\omega, \quad (2.3)$$

$$\mathbf{x} \in X, \mathbf{y}_\omega \in Y, \omega = 1, \dots, \Omega. \quad (2.4)$$

The uncertainties are represented by a finite number of scenarios  $\omega = 1, \dots, \Omega$  with corresponding probabilities  $\pi_\omega$ . The first-stage decisions  $\mathbf{x} \in X$  in (2.1) are subject to the deterministic first-stage constraints (2.2) and represent *here-and-now*-decisions which are applied before the realization of the uncertain parameters is known. The latter implies that the values of  $\mathbf{x}$  have to be identical for all scenarios. This property is called *nonanticipativity*. In contrast, the sets of second-stage decisions  $\mathbf{y}_\omega \in Y$  denote scenario-dependent recourses which are associated to particular realizations of the second-stage constraints (2.3). The symbols denote the first-stage and the second-stage coefficients  $\mathbf{c} \in \mathbb{R}^{n_1}$ , and  $\mathbf{q}_\omega \in \mathbb{R}^{n_2}$ , the matrix and the right-hand side of the first-stage constraints,  $\mathbf{A} \in \mathbb{R}^{m_1 \times n_1}$  and  $\mathbf{b} \in \mathbb{R}^{m_1}$ , the technology matrix  $\mathbf{T}_\omega \in \mathbb{R}^{m_2 \times n_1}$ , the recourse matrix  $\mathbf{W}_\omega \in \mathbb{R}^{m_2 \times n_2}$ , and the right-hand side  $\mathbf{h}_\omega \in \mathbb{R}^{m_2}$ . The objective is to minimize the first-stage costs plus the expected second-stage costs.

The decision variables of the optimization problem (2.1)-(2.4) are elements of rectangular mixed-integer sets  $X = \mathbb{Z}^{n'_1} \times \mathbb{R}^{n''_1}$ ,  $Y = \mathbb{Z}^{n'_2} \times \mathbb{R}^{n''_2}$ , where  $n_1 = n'_1 + n''_1$  and  $n_2 = n'_2 + n''_2$ . When integer requirements are present for the second stage decisions, the problem is called *two-stage stochastic integer program (SIP)*.

For a *SIP* with *relative complete recourse*, every solution  $\mathbf{x}$  that satisfies the first-stage constraints (2.2) has also a feasible completion in the second-stage. For *SIP* with *complete recourse*, every second-stage solution  $\mathbf{y}_\omega$  has also a feasible completion in the first-stage.

### 3 Algorithms for two-stage stochastic integer programs

The straightforward method to solve the *SIP* is to address the large-scale MILP (2.1)-(2.4) by a standard MILP solver. The standard solver for MILPs is based on a branch-and-bound algorithm which implicitly enumerates the integer search space in form of a decisions tree [14]. At each node, the integer relaxation of the MILP is solved by linear programming methods to obtain a lower bound (for minimization problems). All branches with a lower bound greater than the best known integer solution are discarded from the search tree. This algorithm is called *exact* because it guarantees to find an optimal solution and prove its optimality by using conservative gaps, the lower bounds [16].

The main complexity in solving the *SIP* in form of the large-scale MILP lies in its sheer size due to a large number of scenarios and in the resulting large number of integer variables in the second stage. However, the *SIP* has a so called *block-angular* constraint matrix, which is amenable to decomposition approaches [4],[18].

Although commercial MILP solvers exploit the MILP problem structure, they are not yet able to automatically detect and exploit the block-diagonal structure of the *SIP* [15]. Addressing the *SIP* in form of a large scale MILP by a standard algorithm is computational prohibitive for real world problems and thus also critical for online-applications [22].

The idea of *decomposition* is to separate the stochastic problem into two or more sets of variables and associated constraints. The (primal) stage-decomposition separates the variables in first-stage and second-stage decisions. The (dual) scenario decomposition separates the *SIP* by the constraints associated to a scenario. The advantage is, that the large and difficult stochastic program is replaced by a collection of smaller and easier problems. The subproblems are usually MILPs which can be solved by standard optimization software.

In practice, only moderately sized problems can be solved by exact methods because the run-time increases drastically with the problem size. For large-scale and combinatorial optimization problems, *metaheuristics* like evolutionary algorithms (*EAs*) are widely used [2]. In contrast to exact algorithms, metaheuristics have the potential to find good solutions in limited time but cannot guarantee or prove optimality [16]. A recent promising trend is the use of *hybrid* approaches combining metaheuristics with exact algorithms for solving large scale combinatorial optimization problems [16].

The authors have proposed a new integrative hybrid algorithm for *SIPs*, which incorporates exact algorithms in an evolution strategy (called *ES/SIP*) [23]. Based on stage decomposition, an *ES* performs the search on the first-stage master problem, the subproblems are solved by a standard MILP solver. Evolution strategies (*ES*) are a class of evolutionary algorithms which adapt the mutation strength during the course of evolution [3]. The variation operators work on the same representation (called object parameters) as the fitness evaluator. The algorithmic parameters of an *ES* are called strategy parameters. The exogenous strategy parameters are defined by the user, whereas the endogenous strategy parameters are adapted by the algorithm during the course of evolution (called self-adaption).

### 3.1 The hybrid evolution strategy based on stage decomposition

For the *ES* based approach, the *SIP* (2.1)-(2.4) is rewritten and addressed in its *intensive form*

$$z^* = \min_{\mathbf{x}} \quad \mathbf{c}^T \mathbf{x} + \sum_{\omega=1}^{\Omega} \pi_{\omega} Q(\mathbf{x}, \omega) \quad (3.1)$$

$$s.t. \quad \mathbf{A} \mathbf{x} \leq \mathbf{b}, \mathbf{x} \in X, \quad (3.2)$$

where the *second-stage value function*  $Q(\mathbf{x}, \omega)$  is defined by

$$Q(\mathbf{x}, \omega) = \min_{\mathbf{y}_{\omega}} \quad \mathbf{q}_{\omega}^T \mathbf{y}_{\omega} \quad (3.3)$$

$$s.t. \quad \mathbf{W}_{\omega} \mathbf{y}_{\omega} \leq \mathbf{h}_{\omega} - \mathbf{T}_{\omega} \mathbf{x}, \mathbf{y}_{\omega} \in Y. \quad (3.4)$$

When the first-stage decisions  $\mathbf{x}$  are fixed in (3.1), the second-stage decomposes and (3.3)-(3.4) can be separately solved for each scenario. This segmentation into the first-stage *master problem* (3.1)-(3.2) and the second-stage *subproblems* (3.3)-(3.4) is called stage (primal) decomposition, because it separates the variables by stages.

The key idea of *ES/SIP* is to tackle the nonlinear master problem (3.1)-(3.2) by an *ES* and to solve the decoupled mixed-integer linear subproblems (3.3)-(3.4) by a standard MILP solver. The main challenge in the development of *ES/SIP* lies in the design of an appropriate *ES* for the mixed-integer search space  $X$  which is subject to feasibility constraints.

The *population* of an *ES* consists of  $\mu$  parent individuals which are used to generate  $\lambda$  offspring per generation. The *individuals* of the *ES* interpret a first-stage candidate solution  $\mathbf{x}$  as their *object parameters*. The *fitness* of a feasible individual is evaluated by the objective function of (3.1):  $f(\mathbf{x}) = \mathbf{c}^T \mathbf{x} + \sum_{\omega=1}^{\Omega} \pi_{\omega} Q(\mathbf{x}, \omega)$ .

The object parameters  $\mathbf{x} \in X$  are subject to the explicit feasibility constraints (3.2) and for *SIPs* without relative complete recourse also subject to the implicit feasibility constraints of the second-stage (3.4). In the *EA*-literature [7] many *constraint handling techniques* are described, e. g. penalty functions, special representations and operators, repair algorithms, separation of objectives and constraints. Here the use of a *modified objective function* is chosen, because it is rather general and allows to apply standard *ESs*. The fitness function  $f(\mathbf{x})$  is extended to  $F(\mathbf{x})$  which always prefers feasible to infeasible solutions [7]:

$$F(\mathbf{x}) = \left\{ \begin{array}{ll} f(\mathbf{x}) & \text{if } \mathbf{x} \text{ is feasible w.r.t. (3.2)-(3.4)} \\ f_{max} + p(\mathbf{x}) & \text{otherwise} \end{array} \right\} \quad (3.5)$$

The parameter  $f_{max}$  denotes the objective of the worst feasible solution and is here easily calculated by maximizing the integer relaxation of the *SIP* (2.1)-(2.4). The positive penalty term  $p(\mathbf{x})$  defines a measure of the constraint violation and should be a measure of the distance to the feasible region [7]. In contrast to classical penalty functions which add a penalty to the fitness of an infeasible individual, the time consuming evaluation of the fitness function is not required here.

If a first-stage solution candidate  $\mathbf{x}$  is explicitly infeasible in (3.2), the distance to the feasible region is calculated by a weighted sum of the constraint violations  $p(\mathbf{x}) = \sum_{j=1}^J r_j^{-1} \max[0, (\mathbf{A}_j \mathbf{x} - b_j)]$  using a positive scaling factor  $r_j$  for each constraint. The scaling factor  $r_j$  is chosen as the maximum range of the elements of the vector  $\mathbf{x}$  with nonzero elements in the corresponding row of  $\mathbf{A}_j$ .

For *SIPs* without relative complete recourse, the infeasibility of first-stage feasible solutions is caused by infeasible subproblems. The second-stage constraints are not explicitly available for the master-problem, thus the distance measure for explicit constraints as described above can not be applied. Instead, it is checked if the second-stage is infeasible for at least one scenario. Then a death penalty  $p(\mathbf{x}) = +\infty$  is assigned in (3.5).

To *initialize* the *ES* population, the first-stage solutions of the *EV* and of the *WS* problem are used (see Sec. 3.4). As *termination* criterion for the *ES* the maximum number of fitness evaluations is provided by the user heuristically.

Since the object parameter of the *ES* are naturally defined by a mixed-integer string, the mixed-integer *ES* for bounded search spaces is used in this work [9, 17]. This *ES* employs the following recombination, mutation, and selection operators:

For the *recombination*, two of the  $\mu$  parents are randomly selected. The *dominant* recombination operator generates an offspring by randomly taking each of the offsprings parameters from one of the selected parents with equal probability. The *intermediate* recombination operator uses the arithmetic mean of both parents' parameters. The *mutation* operator with a constant mutation rate adds values from a random function scaled by the adaptive mean mutation step size  $\rho$  (mutation strength). The random functions used are a normal distribution for real valued parameters and a geometric distribution for integer parameters. The initial value of  $\rho$  is set to 10% of the range of the corresponding object parameter. The  $(\mu, \kappa, \lambda)$ -*selection* chooses the best individuals from a set of  $\mu$  parents and  $\lambda$  offspring. The parent individuals which exceed the maximum age of  $\kappa$  generations are not further considered in the selection.

The standard parameters of the *ES* are:  $\mu = 10$ ,  $\kappa = 5$ ,  $\lambda = 70$ ; the object parameters are recombined by the dominant method, the endogenous strategy parameters (here: mutation strength) by the intermediate method.

The subproblems (3.3)-(3.4) are solved using the state-of-the-art solver CPLEX [8] which is based on modern branch-and-bound algorithms with cuts and heuristics [14]. Its termination criterion is the gap between the objective of the best integer solution found and the best lower bound (integrality gap).

### 3.2 Exact approaches based on stage decomposition

Exact methods based on stage decomposition address (3.1)-(3.2) and construct some approximations or models of the value function  $Q(\mathbf{x}, \omega)$  as function of the first-stage decisions  $\mathbf{x}$ . The approximation requires the generation of optimality and feasibility cuts. The models are then used in a master problem to generate approximations of the first-stage solution. The differences between the different methods mainly lie in the way the master problem is constructed and solved [18].

In case of pure continuous second-stage variables, the value function is convex in  $\mathbf{x}$  and this property is exploited for efficient cuts, like in the L-shaped decomposition [4]. Laporte and Louveaux [12] proposed a specific L-shaped decomposition based approach for *SIPs* when the first-stage variables are pure binary. Carøe and Tind [6] have generalized this algorithm to cases with mixed-integer first- and second-stage variables. The resulting master problem consists of nonlinear and possibly discontinuous cuts and no practical method for its solution is currently known [19].

In contrast to exact (L-shaped) approaches, the non convexity of the second-stage does not prohibit the use of the stage decomposition in *ES/SIP*.

### 3.3 An exact approach based on scenario decomposition

Carøe and Schultz [5] used the scenario (or dual) decomposition to develop a branch-and-bound algorithm for general *SIPs* called *dual decomposition* algorithm. The

focus of this approach is mainly on generating lower bounds (of minimization problems) than improving the upper bounds (feasible solutions). The dual decomposition algorithm has been successfully applied to chemical batch scheduling problems [10, 20, 22] and is thus used as benchmark in this work.

For the scenario decomposition, the *SIP* (2.1)-(2.4) is equivalently reformulated by adding copies of the first-stage variables  $\mathbf{x}_\omega$  for each scenario  $\omega$  as follows:

$$z^* = \min_{\mathbf{y}_1, \dots, \mathbf{x}_\omega, \mathbf{y}_1, \dots, \mathbf{y}_\Omega} \sum_{\omega=1}^{\Omega} \pi_\omega (\mathbf{c}^T \mathbf{x}_\omega + \mathbf{q}_\omega^T \mathbf{y}_\omega) \quad (3.6)$$

$$s.t. \quad \mathbf{A} \mathbf{x}_\omega \leq \mathbf{b}, \quad (3.7)$$

$$\mathbf{T}_\omega \mathbf{x}_\omega + \mathbf{W}_\omega \mathbf{y}_\omega \leq \mathbf{h}_\omega, \quad (3.8)$$

$$\mathbf{x}_1 = \mathbf{x}_2 = \dots = \mathbf{x}_\Omega, \quad (3.9)$$

$$\mathbf{x}_\omega \in X, \mathbf{y}_\omega \in Y, \omega = 1, \dots, \Omega. \quad (3.10)$$

When the nonanticipativity constraints (3.9) are dropped, the *SIP* decomposes into its scenarios. The dual decomposition algorithm of Carøe and Schultz [5], which is regarded to be one of the state-of-the-art mathematical algorithm for *SIPs* [19], is based on Lagrange relaxation of the nonanticipativity constraints and a standard branch-and-bound algorithm on the first-stage decisions  $\mathbf{x}$  to reestablish the relaxed constraints.

The design of the dual decomposition algorithm is focussed on generating tight lower bounds to make the pruning of the branch-and-bound tree efficient. In the *ES/SIP* approach, the focus is on generating good feasible solutions by evolutionary search which corresponds to search on the leafs of the branch-and-bound tree.

To formulate the Lagrangian dual of the primal *SIP* (3.6)-(3.10), the constraints (3.9) are written as  $\sum_{\omega=1}^{\Omega} \mathbf{H}_\omega \mathbf{x}_\omega = 0$ ,  $\mathbf{H} = (\mathbf{H}_1, \dots, \mathbf{H}_\Omega)$ . With the vector of Lagrangian multipliers  $\boldsymbol{\lambda} \in \mathbb{R}^l$  the Lagrangian dual is written as

$$z_{LD} = \max_{\boldsymbol{\lambda}} D(\boldsymbol{\lambda}), \quad (3.11)$$

$$D(\boldsymbol{\lambda}) = \min_{\mathbf{x}_1, \dots, \mathbf{x}_\Omega, \mathbf{y}_1, \dots, \mathbf{y}_\Omega} \sum_{\omega=1}^{\Omega} \pi_\omega (\mathbf{c}^T \mathbf{x}_\omega + \mathbf{q}_\omega^T \mathbf{y}_\omega) + \boldsymbol{\lambda}^T \mathbf{H}_\omega \mathbf{x}_\omega \quad (3.12)$$

$$s.t. \quad \mathbf{A} \mathbf{x}_\omega \leq \mathbf{b}, \mathbf{T}_\omega \mathbf{x}_\omega + \mathbf{W}_\omega \mathbf{y}_\omega \leq \mathbf{h}_\omega, \quad (3.13)$$

$$\mathbf{x}_\omega \in X, \mathbf{y}_\omega \in Y, \omega = 1, \dots, \Omega. \quad (3.14)$$

The Lagrangian dual provides a lower bound of the primal for arbitrary  $\boldsymbol{\lambda}$ . This lower bound is provided to the branch-and-bound algorithm at each node. In contrast, the *ES/SIP* approach does not provide lower bounds or an optimality guarantee during the course of computation.

The Lagrangian dual (3.11) is a nonlinear concave maximization problem. Its solution requires subgradient methods which are computationally expensive. The subproblem (3.12)-(3.14) of the Lagrangian dual decomposes in sub-subproblems,

which correspond to single scenario problems. As in the *ES/SIP* approach, this sub-subproblems are separately solved by a standard MILP solver.

A candidate solution for the upper bound of the branch-and-bound algorithm is heuristically generated from the possibly different first-stage solutions  $\mathbf{x}_\omega$  of the dual subproblem at each node. The first-stage candidate solutions are evaluated by solving the remaining second-stage subproblems in the same way as the *ES* does by solving (3.3)-(3.4). Note that the candidate solutions may be infeasible in the primal *SIP* when the recourse is not relatively complete. In contrast, the *ES/SIP* approach uses an *ES* to generate candidate solutions. An *ES* represents an advanced self-adapting meta-heuristic, which has the potential to overcome the drawbacks of the simple heuristics.

In contrast to the *ES/SIP* approach, which terminates heuristically, the dual decomposition algorithm provides conservative bounds and thus may terminate when a certain optimality gap (distance between best solution found and best bound) is reached. *DDSIP* does not use any specific initialization scheme.

### 3.4 The expected value and the wait-and-see problems

A lower bound  $z_{WS}$  of the *SIP* is provided by the *wait-and-see* (*WS*) problem which considers the perfect knowledge of the future such that the first-stage decisions of the *SIP* can be optimally chosen with respect to the realization of the randomness [4]. It is obtained by solving (3.6)-(3.10) with dropped nonanticipativity constraints (3.9) and provides  $\Omega$  possibly different first-stage solutions  $\mathbf{x}_\omega$ .

In the *expected value* (*EV*) problem the random variables of the *SIP* are replaced by their expected value to generate a single scenario problem [4]. Plugging its first-stage solution  $\mathbf{x}_{EV}$  into the *SIP* results in the *expected result of using the EV solution* (*EEV*) and provides an upper bound  $z_{EEV}$  of the *SIP*.

The advantage of using a *SIP* instead of a deterministic approach is measured by the *value of the stochastic solution* (*VSS*):  $VSS = z_{EEV} - z^*$ . A similar measure is defined for the best expected result the *WS* solutions:  $K_{WS} = z_{WS} - z^*$ . In *SIP*s without complete recourse, an expected result is possibly infeasible.

## 4 The chemical batch scheduling test problem

### 4.1 Scheduling of the the EPS-plant

The flow sheet of the multi-product batch plant for the production of polymers is shown in Fig. 1.  $P = 2$  types ( $p \in \{A, B\}$ ) of the polymer *expandable polystyrene* (*EPS*) in  $F = 5$  grain size fractions each are produced from a number of raw materials (*E*). The availability of raw materials and the product storage capacity are assumed to be unlimited. The preparation stage is not limiting the production process and is thus neglected in the sequel. The polymerization stage operates in batch mode. The production of each batch is controlled by a recipe. For each *EPS*-type,  $R = 5$  recipes exist ( $r_p \in \{1, \dots, R\} \forall p$ ) which determine the grain size distribution

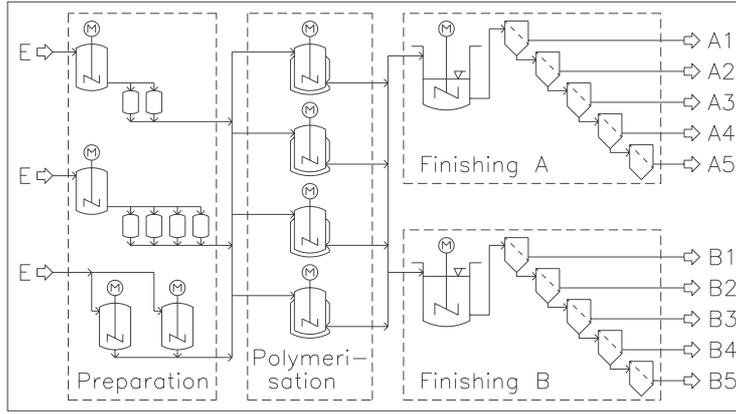


Figure 1: The flow sheet of the EPS-process.

such that each batch yields a main product and  $F - 1$  coupled products. The duration of a polymerization is the same for all recipes. After the polymerization of a batch is finished, this batch is directly transferred to the corresponding mixer of the finishing stage  $A$  or  $B$ . The mixers are semi-continuous storage tanks, the finishing lines operate continuously. If a mixing tank runs empty, the corresponding finishing line has to be shut down temporarily. After a shutdown, the line has to be stopped for a certain period of time. The objective is to maximize the profit calculated from sales revenues, production costs, storage costs and penalties for lateness and for finishing line start-ups and shut-downs. The main production decisions are the discrete choice of recipes  $r_p$  and their starting times.

The scheduling problem has uncertainties in the demand and in the capacity of the polymerization stage. To allow a large number of numerical studies, the original scheduling problem was reduced from  $F = 5$  grain size fractions of each product and  $R = 5$  recipes to  $F = 3$  and  $R = 3$ . The main characteristics of the original problem, the coupled production, remains in the reduced size problems.

## 4.2 The aggregated scheduling model

The scheduling problem is to find the optimal production decisions for a scheduling horizon of ten days. This problem is addressed using the aggregated scheduling model of the EPS-process which is due to Sand and Engell [20]. This multi-period model considers  $i \in \{1, \dots, I = 5\}$  periods, each period corresponds to two days.

To represent the production decisions in the multi-period model, the choice of recipes  $r_p$  and their starting times are modelled by variables which denote the numbers of batches according to recipes  $r_p$  to be started in period  $i$ :  $N_{i,r_p} \in \{0, \dots, N^{max} = 12\}$ . This variable is a bounded integer variable. Once all  $N_{i,r_p}$  of period  $i$  are fixed, the state of the finishing lines of product  $p$  in the same period  $i$  is uniquely determined and described by the binary variable  $z_{i,p} \in \{0, 1\}$ .

In the *SIP*, the first-stage decisions are all production decisions of the periods  $i \in \{1, \dots, I_1\}$ . The resulting vector of first-stage decisions  $\mathbf{x}$  consists of all  $N_{i,r_p}$  with  $i \in \{1, \dots, I_1\}$ . The remaining production decisions together with all remaining continuous variables including sales, storages etc. from the set of mixed-integer second-stage variables.

Instance	A	B	C
$I_1$	4	3	2
$n'_1$	32	24	16
$n''_1$	0	0	0
$n'_2$	131	131	131
$n''_2$	8	16	24
$m_1$	36	27	18
$m_2$	85	94	103
$\Omega$	8	8	8
uncertain demand	$i = 1, \dots, 5$	$i = 2, \dots, 5$	$i = 1, \dots, 5$
uncertain capacity	–	–	$i = 1, \dots, 5$
recourse	rel. complete	rel. complete	not rel. complete
cost coefficients	constant	dynamic	constant
optimal solution $z^*$	-9.25	-10.74	-14.21
rel. $VSS$	42%	109%	$+\infty$
rel. $K_{WS}$	89%	84%	33%

Table 1: Properties of the test instances of the EPS-problem.

The feasibility of the first-stage decisions is determined by the following constraints: (1) the capacity of the polymerization stage limits the number of batches that can be produced in each period  $i$ , (2) the feed into the mixing tanks of the finishing lines is defined by the number of polymerization batches produced per period and is required to be either zero or between the minimum and the maximum capacity depending on the state of the finishing lines, (3) a minimum distance of two periods between two consecutive state changes of the finishing lines must be maintained.

## 5 Test instances and experimental setup

Three test instances of the scheduling model were investigated. All instances are characterized by pure-integer first-stage, mixed-integer second-stage, and uncertainties only in the right-hand side  $\mathbf{h}_w$ . The properties of the instances are summarized in Tab. 1, the values of  $z^*$ ,  $VSS$ , and  $K_{WS}$  were obtained by CPLEX.

Instance A considers a relatively high amount of uncertain parameters with a high number of first-stage decisions ( $I_1 = 4$ ); the demand is uncertain in all periods.

Instance B considers a medium amount of uncertain parameters with a medium number of first-stage decisions ( $I_1 = 3$ ). In contrast to both other instances the costs are dynamic, i. e. they vary over time. The polymerization costs decrease over time while the storage costs increase over time. By this, the objective is more sensitive to first-stage decisions, since early decisions will be penalized later e.g. by high storage costs. The demand is uncertain in all the periods with  $i > 1$ .

Instance C considers a relatively high amount of uncertain parameters with a low number of first-stage decisions ( $I_1 = 2$ ). The demand is uncertain in all periods. In addition, the polymerization capacity is uncertain in all periods. In contrast to instance A and B, the instance C has no relatively complete recourse.

The scheduling problem for the EPS-process has to be solved within limited computing time, thus the quality of an algorithm is defined by the solution found at  $t = 200s$ . In contrast to the dual decomposition, *ES/SIP* is a randomized algorithm, thus the median objective of five single runs with identical parameter settings is considered.

The *ES* part of *ES/SIP* was implemented in C/C++ using *TEA* [9], a library for the design of standard and non-standard *EAs*. For the MILP subproblems GAMS/CPLEX 8.1 [8] is used. The calculations were performed on a 2.4 GHz Linux machine. The termination criterion for the MILP in *ES/SIP* subproblems is set to a relative integrality gap of 1% which is reasonable with respect to the range of the objective function.

More than twenty variations of the exogenous strategy parameters of the *ES* were tested including  $\mu \in \{5, 10, 35, 70, 140\}$ ,  $\kappa \in \{5, 10000\}$ ,  $\lambda \in \{1, 2, 3.5, 7\}\mu$ , and variants with and without recombination of strategy and object parameters. The best setting for instance A and B within this variations is ( $\mu = 10, \kappa = 10000, \lambda = 70$ ) without recombination, for C it is ( $\mu = 10, \kappa = 5, \lambda = 70$ ) with recombination of strategy and object parameters.

The dual decomposition algorithm was used in its C implementation *DDSIP* [13]. The Lagrangian dual problem is solved by a conic bundle algorithm. The CPLEX 9.03 callable library [8] is used to solve the MILP subproblems. The calculations were performed on a 3.06 GHz Linux machine.

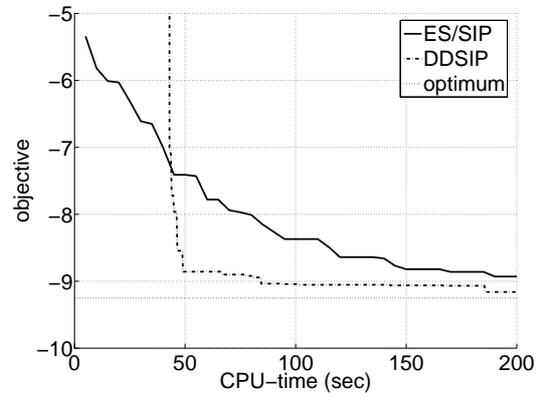
The algorithmic parameters of *DDSIP* were also tuned to obtain the best results. The Lagrangian dual is solved in the root node of the branch-and-bound tree to obtain the multipliers  $\lambda$  by the conic bundle algorithm. In all other nodes, only the dual subproblems are solved using this  $\lambda$ . The integrality gap of the primal is set to 0%, that for the dual subproblems to 0.3%. The best setting of the iteration limit of the conic bundle algorithm is different for each problem, the best settings for the problems (A;B;C) are (26/16/14). The best heuristics for instance A is *take the solution with best objective*, for B and C it is *take the average of the sub-solutions rounded to next integer*.

## 6 Numerical results

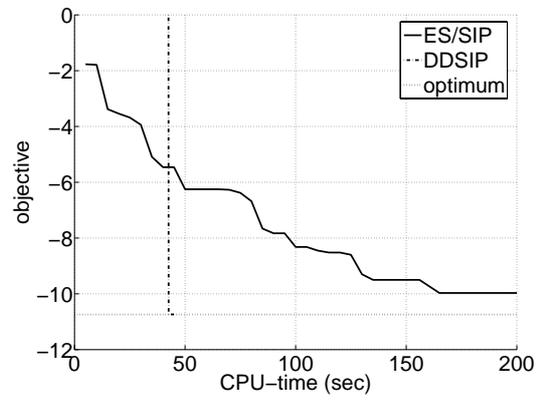
The results of both algorithms after  $t = 200$  sec. are shown in Tab. 2. For instance A, *DDSIP* approximates the exact solution better than *ES/SIP*. For problem B, *DDSIP* performs better, the exact optimum is already found after 42 sec., whereas *ES/SIP* is still more than five percent away at  $t = 200$  sec.; In problem C, *ES/SIP* performs better, no feasible upper bound was found by *DDSIP* within the given time limit. In contrast *ES/SIP* has found a solution which is closer to the exact optimum than the integrality gap of its MILP subproblems.

To further analyze the course of computation, the evolution of the median of the best solutions found by *ES/SIP* and of the upper bound of *DDSIP* are plotted over the CPU-time in Fig. 2. In addition, the optimal solutions are depicted.

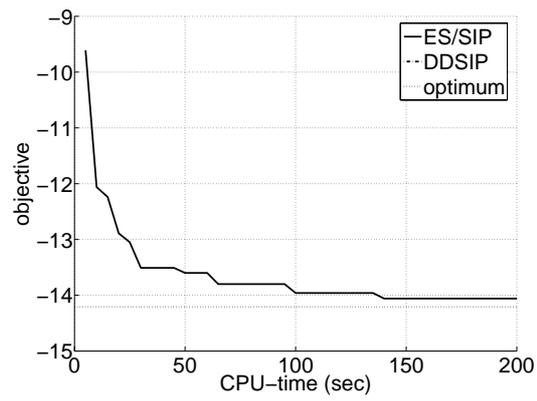
For instance A the solution of the Lagrangian dual at the root node takes 34 sec., for instance B 39 sec., and for instance C 34 sec., which are all in the same range.



(a) Instance A.



(b) Instance B.



(c) Instance C.

Figure 2: The course of the computations for the test instances.

<b>Instance</b>	<b>A</b>	<b>B</b>	<b>C</b>
exact solution of the <i>SIP</i> $z^*$	-9.25	-10.74	-14.21
<b><i>ES/SIP</i></b>			
best objective $z_E$ (median of five runs)	-8.91	-9.97	-14.12
relative gap $((z^* - z_E)/z^*)$	3.68%	7.17%	0.63%
<b><i>DDSIP</i></b>			
best objective $z_D$	-9.16	-10.74	not available
relative gap $((z^* - z_D)/z^*)$	0.97%	0 %	$\infty$ (not available)

Table 2: Results of the numerical studies: best objective after  $t = 200sec..$

*ES/SIP* already improves its feasible initial solution while *DDSIP* is still solving the computational expensive Lagrangian dual at the root node. After the solution of the Lagrangian dual has been found, *DDSIP* quickly improves its upper bounds in instances A and B. In instance C (without relative complete recourse), *DDSIP* is not able to generate any feasible upper bound within 200 seconds of CPU-time, but after 300 seconds (not depicted here).

The *ES* shows a similar evolution for all instances, in the average it evolves about fifteen to twenty generations within 200 seconds.

## 7 Conclusion and Perspectives

In this paper the hybrid evolution strategy based approach *ES/SIP* for two-stage stochastic integer programs was presented and conceptually and numerically compared to an alternative exact dual decomposition algorithm. From numerical experiments using chemical batch plant scheduling problems, it was found that *DDSIP* performs better in finding the best objective in limited time for *SIPs* with relative complete recourse. For the problem without relative complete recourse, the hybrid evolution strategy of *ES/SIP* leads to upper bounds much faster than the simple heuristics of *ES/SIP*. The performance of the algorithms depends strongly on properties of the problem.

The *DDSIP* approach can be used for general *SIPs*, but it requires a high effort for the tuning. The *ES/SIP* as tested here is suitable for any *SIP* with integer first-stage and can be extended to mixed-integer first-stage easily since the variation operators for a real valued *ES* do already exist. For both approaches, the relation between the model properties and the algorithmic performance is subject to current research.

## 8 Acknowledgement

We gratefully acknowledge the fruitful cooperation with Rüdiger Schultz and Eva Clostermann of the Department of Mathematics, Universität Duisburg-Essen, Germany. The financial support by the German Research Foundation (DFG) for the Collaborative Research Center (Sonderforschungsbereich) *Design and Management*

of *Complex Technical Processes and Systems by Means of Computational Intelligence Methods* (SFB 531) at the Universität Dortmund, Project C10, is gratefully acknowledged.

## References

- [1] G. Applequist, O. Samikoglu, J. Pekny, and G. Reklaitis. Issues in the use, design, and evolution of process scheduling and planning systems. *ISA Transactions* 36, 81–121, 1997.
- [2] Th. Bäck, D. B. Fogel, and Z. Michalewicz, editors. *Handbook of Evolutionary Computation*. Oxford University Press, New York, 1997.
- [3] H. Beyer and H. Schwefel. Evolution strategies. *Natural Computing* 1, 3–52, 2002.
- [4] J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 1997.
- [5] C. C. Carøe and R. Schultz. Dual decomposition in stochastic integer programming. *Operations Research Letters* 24, 37–45, 1999.
- [6] C. C. Carøe and J. Tind. L-shaped decomposition of two-stage stochastic programs with integer recourse. *Mathematical Programming* 83(3), 451–464, 1998.
- [7] C. A. Coello Coello. Theoretical and numerical constraint handling techniques used with evolutionary algorithms: A survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering* 191(11-12), 1245–1287, 2002.
- [8] CPLEX. *Using the CPLEX Callable Library*. ILOG Inc., Mountain View, CA, 2002.
- [9] M. Emmerich, M. Schütz, B. Groß, and M. Grötzner. Mixed-integer evolution strategy for chemical plant optimization with simulators. In I.C. Parmee, editor, *Evolutionary Design and Manufacture (ACDM)*, 55–67, Springer, New York, 2000.
- [10] S. Engell, A. Märkert, G. Sand, and R. Schultz. Aggregated scheduling of a multiproduct batch plant by two-stage stochastic integer programming. *Optimization and Engineering* 5, 335–359, 2004.
- [11] J. Kallrath. Planning and scheduling in the process industry. *OR Spectrum* 24, 219–250, 2002.
- [12] G. Laporte and F.V. Louveaux. The integer L-shaped method for stochastic integer programs with complete recourse. *Operations Research Letters* 13, 133–142, 1993.
- [13] A. Märkert. *User's guide to DDSIP - A C Package for the Dual Decomposition of Stochastic Programs with Mixed-Integer Recourse*. Department of Mathematics, Universität Duisburg-Essen, Duisburg, 2004. (<http://www.uni-duisburg.de/FB11/disma/ddsip-eng.shtml>, March 23, 2005).

- [14] G. Nemhauser and A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, New York, 1999.
- [15] G. R. Parija, S. Ahmed, and A. J. King. On bridging the gap between stochastic integer programming and MIP solver technologies. *INFORMS Journal on Computing* 16(1), 73–83, 2004.
- [16] J. Puchinger and G. Raidl. Combining metaheuristics and exact algorithms in combinatorial optimization: a survey and classification. In *Proceedings of the First International Work-Conference on the Interplay Between Natural and Artificial Computation*, Vol. 3562 of *LNCS*, 41–53, Springer, Berlin, 2005.
- [17] G. Rudolph. An evolutionary algorithm for integer programming. In Y. Davidor, H.-P. Schwefel, and R. Männer, editors, *Parallel problem Solving from Nature - PPSN III*, Vol. 866 of *LNCS*, 193–197, Springer, Berlin, 1994.
- [18] A. Ruszczyński and Shapiro A., editors. *Stochastic Programming*, Vol. 10 of *Handbooks in Operations Research and Management Science*. Elsevier, Amsterdam, 2003.
- [19] N. V. Sahinidis. Optimization under uncertainty: state-of-the-art and opportunities. *Computers and Chemical Engineering* 28, 971–983, 2004.
- [20] G. Sand and S. Engell. Modelling and solving real-time scheduling problems by stochastic integer programming. *Computers and Chemical Engineering* 28, 1087–1103, 2004.
- [21] N. Shah. Single- and multisite planning and scheduling: Current status and future challenges. In I. Grossmann and C. McDonald, editors, *Fourth International Conference on Foundations of Computer-Aided Process Operations*, 75–90, CACHE Corp., Austin, TX, 2003.
- [22] J. Till, S. Engell, G. Sand, and E. Cloostermann. Rigorous vs. stochastic algorithms for two-stage stochastic integer programming applications. In *Proceedings on the International Conference on Intelligent Computing (ICIC 2005)*, 2269–2278, Hefei, China, 2005.
- [23] J. Till, G. Sand, S. Engell, M. Emmerich, and L. Schönemann. A hybrid algorithm for solving two-stage stochastic integer problems by combining evolutionary algorithms and mathematical programming methods. Vol. 20A of *Computer-Aided Chemical Engineering*, 187–192, Elsevier, Amsterdam, 2005.
- [24] S. W. Wallace and W. T. Ziemba, editors. *Applications of Stochastic Programming*. MPS-SIAM series in optimization, Philadelphia, 2005.

# Strukturelle Evolution von Fuzzy-Reglern am Beispiel des inversen Rotationspendels

Frank Hoffmann und Daniel Schauten

Lehrstuhl für Elektrische Steuerung und Regelung  
Fakultät für Elektrotechnik und Informationstechnik

Universität Dortmund, 44221 Dortmund

Tel.: (0231) 755-2760

Fax: (0231) 755-2752

E-Mail: {Hoffmann, Schauten}@esr.e-technik.uni-dortmund.de

## Zusammenfassung

Diese Arbeit stellt ein neues Verfahren zur evolutionären Struktur- und Parameteroptimierung von TSK-Fuzzy-Reglern vor. Ausgehend von einem einfachen linearen Regler wird im Laufe der evolutionären Optimierung die Anzahl der Regeln und damit einhergehend die Komplexität des Reglers sukzessive erhöht. Diese Vorgehensweise ermöglicht eine graduelle Anpassung des nichtlinearen Reglerkennfeldes an die Regelungsziele. Die Praxistauglichkeit des Verfahrens wird am Beispiel des automatisierten Reglerentwurfs für ein Rotationspendel demonstriert. Der inkrementelle Ansatz erweist sich sowohl hinsichtlich der Identifikation eines stabilisierenden Reglers als auch der von ihm erzielbaren Regelgüte überlegen gegenüber der direkten Optimierung eines TSK-Reglers mit einer adäquaten fest vorgegebenen Reglerstruktur.

## 1 Einleitung

Methoden zur evolutionären Optimierung von Fuzzy-Systemen im Allgemeinen [1] und Fuzzy-Reglern im Besonderen [2, 3] wurden in zahlreichen Arbeiten vorgeschlagen, verglichen und eingehend analysiert. Dabei kodiert ein evolutionärer Algorithmus Parameter des Fuzzy-Systems wie Skalierungsfaktoren, Zugehörigkeitsfunktionen und Regeln. In einem genetischen Fuzzy-System automatisiert der evolutionäre Algorithmus die Aufgabe des Reglerentwurfs durch das Erlernen oder die Anpassung dieser Parameter. Bei der *Fuzzy-Modellierung* liegt das Hauptaugenmerk der Optimierung auf einer hohen Modellgenauigkeit für einen Satz von Trainingsbeispielen. In der *Fuzzy-Regelung* liegen keine direkten Trainingsbeispiele vor, vielmehr bewertet die Gütefunktion das Gesamtverhalten des geschlossenen Regelkreises im Zeit- oder Frequenzbereich.

Iterative Ansätze zur evolutionären Generierung von Fuzzy-Regeln sind in der Fuzzy-Modellierung weit verbreitet [4, 5, 6]. Einer dieser Ansätze basiert auf einem hierarchischem Fuzzy-Modell, das in unterschiedlichen Auflösungsgenauigkeiten mit jeweils mehr oder weniger Regeln operiert [4]. Diese Abstufung ermöglicht ein systematisches Abwägen zwischen Genauigkeit und Interpretierbarkeit des Fuzzy-Modells. Falls die Modellgüte von übergeordneter Bedeutung ist, lässt sich durch zusätzliche verfeinerte Regeln die Genauigkeit in Regionen mit hohem Fehler erhöhen. Das Modell wird inkrementell generiert, indem auf der nächsten Ebene die bestehenden Regeln entweder erweitert oder durch verfeinerte Regeln ersetzt werden.

Der sogenannte LOLIMOT-Ansatz verwendet eine Baumhierarchie lokaler linearer Modelle, wobei die Modellgüte durch Unterteilung dieser Modelle in Regionen mit großem residualen Fehler verbessert wird [5]. Die Prämisse der Regel, die den größten lokalen Fehler aufweist, wird ersetzt durch ein Paar von Regeln, indem eine der eingangsseitigen Fuzzy-Mengen in zwei überlappende Mengen unterteilt wird und die zugehörigen linearen Modelle mittels der Methode des kleinsten quadratischen Fehlers an die Trainingsdaten angepasst werden. Das Verfahren unterteilt dabei genau jene Regel und Zugehörigkeitsfunktion, für welche die daraus resultierende Partitionierung den geringsten Gesamtfehler aufweist.

Beim Entwurf von TSK-Fuzzy-Reglern wird das nichtlineare Verhalten der Strecke durch einen Satz linearer Modelle, gültig am jeweiligen Arbeitspunkt, approximiert. An jedem Arbeitspunkt wird dann für das lineare Modell ein stabilisierender Zustandsregler entworfen und als TSK-Fuzzy-Regel implementiert. Das Gesamtverhalten ergibt sich aus der unscharfen Überlagerung der Regeln in Form einer gewichteten Interpolation der Ausgabe der linearen Regler über dem Zustandsraum. Dabei bestimmt der Erfüllungsgrad der Prämisse für einen konkreten Eingangszustand den Beitrag dieser Regel zur Ausgangsgröße. Die Stabilität des Gesamtsystems lässt sich nachweisen, falls für alle linearen Regler eine gemeinsame Lyapunovfunktion existiert. Angenommen  $A_{ij}$  sei die Zustandsmatrix des geschlossenen Kreises für das  $i$ -te Modell unter Verwendung des  $j$ -ten Reglers. Das Gesamtsystem ist stabil, falls eine positiv definite Matrix  $P$  existiert, sodass alle Matrizen  $M_{ij}$  gemäß

$$M_{ij} = A_{ij}^T P + P A_{ij} < 0 \quad (1)$$

negativ definit sind. Für das Auffinden einer Lösung für Gleichung (1) existieren effiziente LMI-Algorithmen. Lineare TSK-Fuzzy-Modelle lassen sich ebenfalls für die modellprädiktive Regelung nichtlinearer Systeme verwenden [7]. Die optimale Stellgröße der modellprädiktiven Regelung lässt sich anhand des lokalen linearen Modells der TSK-Fuzzy-Regel herleiten.

Das Entwurfsziel einer optimalen Regelung besteht in der Minimierung des gewichteten Zustandsfehlers und Stellgrößenverbrauchs. Für ein lineares Streckenmodell

$$\dot{x} = Ax + Bu \quad (2)$$

erweist sich das Reglergesetz, welches das Gütefunktional über den unendlichen Zeithorizont

$$J = \int_{t=0}^{\infty} x(t)Qx(t) + u(t)Ru(t)dt \quad (3)$$

minimiert, als linear und lässt sich durch

$$u = -k^T x = -R^{-1}B^T P x \quad (4)$$

angeben, wobei die Matrix  $P$  die Lösung der algebraischen Riccatigleichung

$$PBR^{-1}B^T P - PA - A^T P - Q = 0 \quad (5)$$

darstellt. Der Entwurf eines optimalen Reglergesetzes für eine nichtlineare Strecke ist weitaus schwieriger und beruht auf einer vorherigen Eingangs-/Ausgangslinearisierung der Strecke. Der LQR(Linearer Quadratischer Regler)-Entwurf beurteilt den

quadratischen Stellgrößenverbrauch, obwohl aus praktischer Sicht die Verletzung der Stellgrößenbeschränkung von Bedeutung ist. Falls im Reglergesetz aus Gleichung (4) die Stellgrößenbeschränkung für eine Störauslenkung  $x_0$  verletzt wird, muss der LQR-Entwurf mit einem größerem Gewichtungsfaktor  $R$  solange wiederholt werden, bis die Stellgröße den Anforderungen entspricht.

Die Grundidee unseres Verfahrens besteht darin, das Konzept der graduellen Verfeinerung bei der Fuzzy-Modellierung auf den schrittweisen Entwurf von TSK-Fuzzy-Reglern für nichtlineare Systeme zu übertragen. Den Ausgangspunkt stellt ein optimaler linearer Regler dar, welcher aus einem am gewünschten Arbeitspunkt linearisierten Modell hergeleitet und dann durch eine einzelne globale Fuzzy-Regel

$$\text{IF } X \text{ is } A \text{ THEN } u = k_0 + \sum_i k_i x_i, \quad \text{mit } \mu_A(x) = 1, \forall x \in X$$

implementiert wird. Die Verstärkungsfaktoren  $k_i$  werden dabei mit Hilfe des LQR-Verfahrens anhand von Gleichung (4) bestimmt. Das ursprünglich lineare Kennfeld wird durch eine Partitionierung des Zustandsraumes in Fuzzy-Mengen verfeinert, sodass dieser in verschiedenen Regionen durch Regeln mit unterschiedlichen Verstärkungsfaktoren überdeckt wird. Da das optimale Kennfeld im vorhinein nicht bekannt ist, lässt sich nicht wie bei der Fuzzy-Modellierung ein lokaler Modellfehler bewerten, anhand dessen sich die Koeffizienten optimieren ließen. Es lässt sich lediglich das Gesamtverhalten des geschlossenen Regelkreises im Zusammenspiel aller Regeln nach Gleichung (3) bestimmen. Darüber hinaus ist nicht klar, welche Partitionierung für das Regelungsproblem optimal ist, also entlang welcher Zustandsgrößen der Zustandsraum zu partitionieren ist und in welche Fuzzy-Mengen diese zu unterteilen sind. Zur Lösung dieses komplexen Entwurfsproblems kommt daher eine Evolutionsstrategie zum Einsatz, welche sowohl die Struktur und Parameter der Zustandsraumpartitionierung als auch die Verstärkungsfaktoren in der Konklusion der Regeln in Bezug auf das Gütefunktional (3) optimiert.

## 2 Rotationspendel

Das inverse Rotationspendel in Abbildung 1 besteht aus zwei senkrecht zueinander angeordneten Achsen, wobei die erste vertikale Achse von einem Motor angetrieben wird. Das an der zweiten horizontalen Achse befestigte Pendel kann dagegen frei schwingen. Das Regelungsziel besteht im Aufschwingen des Pendels aus der unteren Ruhelage und der anschließenden Stabilisierung des freien Pendels in der oberen instabilen Ruhelage. Das dynamische Verhalten des inversen Rotationspendels ist beschrieben durch ein nichtlineares Differentialgleichungssystem vierter Ordnung

$$\ddot{\varphi} = -c_1 \dot{\varphi} + c_2 u \quad \text{und} \quad (6)$$

$$\ddot{\theta} = c_3 \left[ -c_4 \ddot{\varphi} \cos(\theta) + c_5 \dot{\varphi}^2 \cos(\theta) \sin(\theta) + c_6 \sin(\theta) - c_7 \dot{\theta} \right], \quad (7)$$

wobei  $\theta$  die Winkelstellung des inversen Pendels,  $\varphi$  die Winkelstellung des horizontalen Armes und  $\tau$  das vom Motor erzeugte Drehmoment darstellen. Die Koeffizienten

$c_i$  lassen sich aus den Systemparametern

$$c_1 = \mu_{gear} \quad c_3 = \frac{1}{J_p + m_p \left(\frac{l_p}{2}\right)^2} \quad c_5 = m_p \left(\frac{l_p}{2}\right)^2$$

$$c_2 = \eta \quad c_4 = m_p \frac{l_p l_b}{2} \quad c_6 = m_p g \frac{l_p}{2} \quad c_7 = \mu_{bearing}$$

berechnen, wobei  $l_p, m_p, J_p$  die Länge, die Masse und das Trägheitsmoment des vertikalen Armes,  $\mu_{gear}$  und  $\mu_{bearing}$  die Reibungskoeffizienten des Motorgetriebes und des Pendellagers und  $\eta$  die statische Verstärkung zwischen Motorspannung und Motorgeschwindigkeit beschreiben.

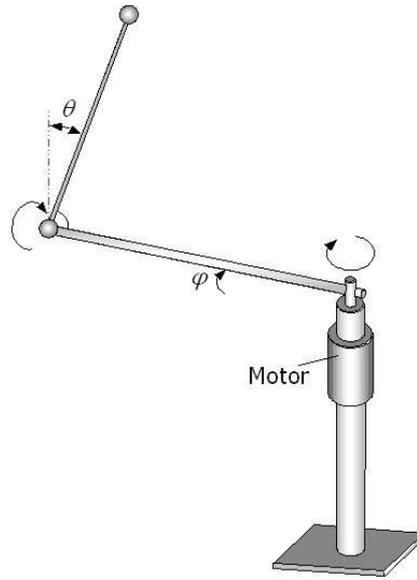


Abbildung 1: Inveres Rotationspendel

Die Regelungsaufgabe wird durch zwei separate Fuzzy-Regler gelöst: Ein Regler soll das Pendel in möglichst kurzer Zeit aufschwingen, wohingegen ein zweiter Regler das Pendel für kleinere Auslenkungen um die obere Ruhelage stabilisiert. Die maximal mögliche Umschaltgrenze ergibt sich aus dem maximalen Motordrehmoment, das gerade noch ausreicht, um das durch die Schwerkraft verursachte Drehmoment auszugleichen. Die Güte eines stabilisierenden Reglers wird durch das zeitliche Integral des quadratischen Zustandsgrößenfehlers quantifiziert. In den ersten Generationen gelingt es den meisten Reglern selbst für kleine Anfangsauslenkungen nicht, das Pendel zu stabilisieren. Eine Reglerbewertung dauert längstens  $t_{end}^{max} = 15s$ , wird aber im Falle eines instabilen Reglers vorzeitig abgebrochen, sobald das Pendel überkippt ( $|\theta| > \pi/2$ ). In diesem Fall ist die Fitness des Reglers umso kleiner, je später das Umkippen in  $t_{end}$  erfolgt:

$$J_0 = (15 - t_{end}). \quad (8)$$

Es sei darauf hingewiesen, dass der evolutionäre Algorithmus in Analogie zu Gütefunktionalen beim klassischen optimalen Reglerentwurf die Fitness minimiert. Für einen stabilisierenden Regler mit  $t_{end} = 15$  verschwindet  $J_0 = 0$ . In diesem Fall

berechnet sich die Fitness aus dem Integral des quadratischen Zustandsfehlers zu

$$J_1 = \frac{1}{t_{end}} \int_0^{t_{end}} \frac{\varphi(t)^2}{10} + \theta(t)^2 dt. \quad (9)$$

Der Gewichtungsfaktor für Fehler im Winkel  $\varphi$  ist kleiner, da das primäre Ziel in der Stabilisierung des vertikalen Pendels besteht, wofür größere Auslenkungen des horizontalen Armes toleriert werden können. Um der Stellgrößenbegrenzung Rechnung zu tragen, wird ein Überschreiten der maximalen Stellgröße  $u_{max}$  durch das Funktional

$$J_2 = \frac{1}{t_{end}} \int_{t:|u(t)|>u_{max}} \frac{(|u(t)| - u_{max})^2}{10} dt \quad (10)$$

bestraft, wobei sich das Integral auf diejenigen Zeitintervalle beschränkt, in denen die berechnete Stellgröße in Sättigung geht ( $|u(t)| > u_{max}$ ). Dies liegt darin begründet, dass in der Simulation die effektive Stellgröße für Werte außerhalb des Intervalls  $[-u_{max}, u_{max}]$  abgeschnitten wird, sodass darüberhinaus gehende Stellgrößensignale keinen weiteren Beitrag zur Minimierung von  $J_1$  leisten, obwohl sie in  $J_2$  bestraft werden. Die Gesamtfitness eines Reglers ergibt sich aus der Summe der drei Gütefunktionale:

$$J = J_0 + J_1 + J_2. \quad (11)$$

Für den automatisierten Entwurf eines Aufschwingreglers wird als Zielvorgabe das Erreichen einer maximalen Restauslenkung von  $|\min(\theta(t))| = 0,2$  für das inverse Pendel gewählt. Dieser Wert ist so festgelegt, dass durch überlappende Arbeitsbereiche später ein Umschalten über ein einfaches Hystereseglied zwischen den beiden Reglern ermöglicht wird.

$$J_{Auf} = (t_{max} - t_{end}) \left\{ \begin{array}{l} + t_{max} \\ + \min|\tilde{\theta}(t)| \\ + \frac{1}{t_{end}} \int_0^{t_{end}} \tilde{\theta}^2(\tau) d\tau \\ + \frac{0,1}{t_{end}} \int_0^{t_{end}} \varphi^2(\tau) d\tau \\ + \frac{0,1}{t_{end}} \int_0^{t_{end}} \tilde{u}^2(\tau) d\tau \end{array} \right. \quad \text{wenn } |\min(\tilde{\theta}(t))| > 0,2 \quad (12)$$

$$\left\{ \begin{array}{l} + t_{0,2} \\ + |\dot{\theta}(t_{0,2})| \\ + \frac{0,1}{t_{0,2}} \int_0^{t_{0,2}} \varphi^2(\tau) d\tau \\ + \frac{0,1}{t_{0,2}} \int_0^{t_{0,2}} \tilde{u}^2(\tau) d\tau \end{array} \right. \quad \text{wenn } |\min(\tilde{\theta}(t))| \leq 0,2$$

Das vollständige Gütefunktional zum Erlernen des Aufschwingreglers beinhaltet gemäß Gleichung (12) eine Fallunterscheidung. Zunächst soll der zweite Fall der Fitnessbewertung (im unteren Teil der Gleichung) erläutert werden. Die Gütebewertung hierfür setzt sich aus vier Termen zusammen. Der erste Term  $t_{0,2}$  gibt den

Zeitpunkt an, zu dem der Zielwinkel  $\tilde{\theta}_{0,2}$  zum ersten Mal durchlaufen wird. Damit wird die benötigte Zeit zum Erreichen von  $\tilde{\theta}_{0,2}$  bestraft. Der zweite Term  $|\dot{\theta}(t_{0,2})|$  berücksichtigt die Forderung an eine minimale Winkelgeschwindigkeit des Pendels im Zielwinkel  $\tilde{\theta}_{0,2}$ . Da eine kürzere Zeit  $t_{0,2}$  notwendigerweise größere Winkelgeschwindigkeiten erfordert, stehen die beiden genannten Gütekriterien in Konflikt. Hier gilt es einen Kompromiss zu finden. Die Forderungen an den Armwinkel  $\varphi$  und an den Stellgrößenverbrauch  $u$  werden wie beim stabilisierenden Regler mit Hilfe des ISE-Kriteriums angegeben. Da später bei Erreichen des Zielwinkels auf den stabilisierenden Regler umgeschaltet wird, wird jedoch nur bis zum Zeitpunkt  $t_{0,2}$  integriert. Dabei berücksichtigt  $\tilde{u}$  wiederum nur das Überschreiten der gegebenen Stellgrößenbeschränkung  $u_{max}$ .

Die Regler, die den Punkt  $\theta_{0,2}$  nicht innerhalb der gegebenen Simulationszeit  $t_{max}$  erreichen, werden deutlich schlechter bewertet. Aus diesem Grund erhalten sie zunächst den konstanten Bestrafungsterm  $t_{max}$ . Die folgenden beiden Terme dienen dazu, Individuen mit schlechten Reglerparametern zum Zielwinkel  $\theta_{0,2}$  hinzuführen. Durch den konstanten Bestrafungsterm  $t_{max}$  und das Integral über dem Quadrat von  $\theta$  ist gewährleistet, dass Individuen, die  $\theta_{0,2}$  nicht erreichen, in jedem Fall schlechter bewertet werden, als Individuen, die  $\theta_{0,2}$  erreichen. Wie bereits zuvor werden auch hier große Armwinkel und die Stellgröße mit Hilfe des ISE-Kriteriums bewertet.

### 3 Evolutionäre Strukturweiterung von TSK-Fuzzy-Reglern

Dieser Abschnitt beschreibt die schrittweise Strukturweiterung und Parameteroptimierung von TSK-Fuzzy-Reglern durch eine Evolutionsstrategie. Das Übertragungsverhalten eines TSK-Reglers mit  $M$  Regeln über einem  $N$ -dimensionalen Zustandsraum führt zu  $(N + 1)M$  direkten Reglerparametern, auf die zusätzlich Verstärkungen  $k_i$  und zusätzliche Parameter für die Definition der Zugehörigkeitsfunktionen im Eingangsraum entfallen. Wegen der hohen Dimension des Optimierungsraums wird die Wahrscheinlichkeit, zufällig einen stabilisierenden Regler zu finden, mit wachsender Anzahl von Regeln und damit der Parameter geringer. Um die Komplexität des Optimierungsproblems zu reduzieren wird daher ein Ansatz vorgeschlagen, in welchem die Anzahl der Regeln nicht statisch ist, sondern erst im Laufe der Evolution sukzessive und mit zunächst neutraler Auswirkung erhöht wird.

#### 3.1 Strukturelle Evolution

Die Strukturweiterung erfolgt durch Einfügen zusätzlicher Fuzzy-Mengen entlang jeweils einer Eingangsgröße, einhergehend mit einer Duplizierung der Regeln. Dabei kann die neue Fuzzy-Menge auf zwei unterschiedliche Arten generiert werden, durch Teilung einer Fuzzy-Menge oder durch Einfügen einer neuen Fuzzy-Menge zwischen zwei bestehenden Fuzzy-Mengen. Im Fall des Rotationspendels sind alle Fuzzy-Partitionierungen symmetrisch zum Ursprung, daher wird eine Erweiterungsoperation immer simultan auf die entsprechende Fuzzy-Menge im negativen und im positiven Teilraum angewandt.

Abbildung 2 illustriert den Erweiterungsschritt für einen Regler für das inverse Rotationspendel. Aus Gründen der Übersichtlichkeit wird eine Projektion des vierdimen-

sionalen Zustandsraumes auf die beiden Zustandsgrößen  $\theta$  und  $\dot{\theta}$  betrachtet, obwohl Partitionierungen entlang der beiden übrigen Variablen  $\varphi$  und  $\dot{\varphi}$  ebenso möglich sind. Die Anfangspopulation enthält Individuen, die die Koeffizienten  $\{k_0, k_\varphi, k_{\dot{\varphi}}, k_\theta, k_{\dot{\theta}}\}$  einer einzelnen TSK-Regel  $R_1$  (Abbildung 2 oben) der Form

IF  $\theta$  is  $A$  AND  $\dot{\theta}$  is  $B$  THEN  $u = k_0 + k_\varphi\varphi + k_{\dot{\varphi}}\dot{\varphi} + k_\theta\theta + k_{\dot{\theta}}\dot{\theta}$ ,  
mit  $\mu_A(\varphi) = 1$  und  $\mu_B(\dot{\varphi}) = 1 \quad \forall \varphi, \dot{\varphi} \in X$

enthalten. Die Koeffizienten werden entweder zufällig initialisiert, oder durch das LQR-Verfahren anhand von Gleichung (4) ermittelt und mit einer zusätzlichen zufälligen Störung versehen, um die Diversität der Anfangspopulation zu gewährleisten.

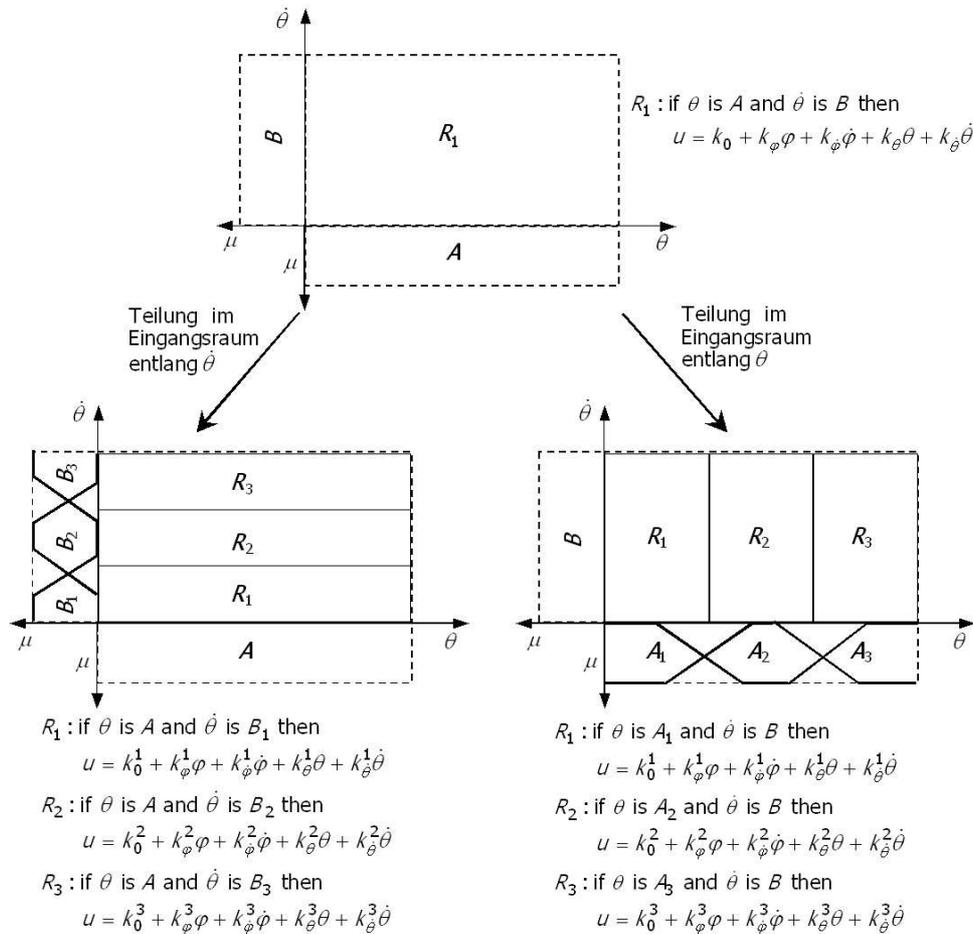


Abbildung 2: Zwei mögliche Partitionierungen des Zustandsraums für einen TSK-Fuzzy-Regler.

Die Verfeinerung der Zustandsraumpartitionierung kann entlang jeder der vier Zustandsgrößen erfolgen. Abbildung 2 zeigt zwei mögliche Unterteilungen entlang der Zustandsgrößen  $\dot{\theta}$  (unten links) und  $\theta$  (unten rechts). Im ersten Fall wird die globale Zugehörigkeitsfunktion  $B$  durch drei überlappende trapezförmige Fuzzy-Mengen  $B_1, B_2, B_3$  ersetzt. Die ursprünglich einzige Regel  $R_1$  wird dabei durch drei Regeln  $R_1, R_2, R_3$  mit unterschiedlichen Koeffizienten  $k_i^1, k_i^2, k_i^3$  substituiert. Wie bereits erwähnt, bleibt die Symmetrie der Partitionierung zum Ursprung gewährleistet. Ein

zur Kodierung der  $k_i$  ursprünglich aus fünf Objektvariablen bestehendes Individuum wird durch ein fünfzehnparametriges Individuum ersetzt, das zunächst aus drei identischen Kopien der ursprünglichen Koeffizienten  $k_i$  besteht. Zusätzliche Parameter kodieren die Stützstellen des Überlappungsbereichs der benachbarten Fuzzy-Mengen  $A_1/A_2$  und  $A_2/A_3$  bzw.  $B_1/B_2$  und  $B_2/B_3$ .

Bei der Fuzzy-Modellbildung ist es möglich, die optimalen Koeffizienten in der linearen Konklusion und den Restfehler durch lokale oder globale Regression zu bestimmen. Als optimale Erweiterung der Fuzzystruktur ergibt sich aus den Alternativen bezüglich der zu unterteilenden Eingangsvariable und Position der zusätzlichen Fuzzy-Menge diejenige Partition mit dem geringsten verbleibenden Fehler über die Trainingsdaten.

Dieser Ansatz des überwachten Lernens ist für den Reglerentwurf nicht anwendbar, da keine Trainingsdaten in Form von Ein/-Ausgangspaaren vorliegen, sondern das dynamische Verhalten des geschlossenen Regelkreises zu bewerten ist. Daher lässt sich erst retrospektiv entscheiden, welche Partitionierung mit einer nachgeschalteten Optimierung der Koeffizienten durch den evolutionären Algorithmus letztendlich die größte Verbesserung erzielt. Zu diesem Zweck verzweigt der evolutionäre Algorithmus zeitweise in mehrere voneinander isolierte Subpopulationen, mit strukturell unterschiedlichen Eingangsraumpartitionierungen. Alle Lösungen einer Subpopulation partitionieren den Eingangsraum entlang der gleichen Eingangsgröße. In den nachfolgenden Generationen entwickeln sich die Subpopulationen parallel und unabhängig voneinander, wobei der evolutionäre Algorithmus jeweils die ihrer Partitionierung des Fuzzy-Reglers entsprechenden Parameter optimiert. Zusätzlich zu den Koeffizienten der Konklusion kodiert jedes Individuum die Schnittpunkte benachbarter Fuzzy-Mengen. Durch deren Adaption kann der evolutionäre Algorithmus lokal die Partitionierung des Zustandsraums optimieren, um beispielsweise die Auflösung in den Regionen, die ein stark nichtlineares Verhalten erfordern, zu erhöhen.

Nach einigen Generationen werden die besten Lösungen der Subpopulationen untereinander verglichen. Die Population mit dem größten Fortschritt überlebt, alle übrigen Subpopulation werden verworfen. Dieser Kreislauf aus struktureller Erweiterung des Fuzzy-Reglers und Optimierung der zusätzlichen Freiheitsgrade wird solange fortgesetzt, bis eine feinere Partitionierung zu keiner nennenswerten Verbesserung des Regelverhaltens mehr führt.

Der wesentliche Vorteil dieser Methoden besteht darin, dass eine strukturelle Anpassung und Parameteroptimierung kombiniert angewendet wird. Anstatt eine komplexe Struktur mit einer Vielzahl von Parametern direkt zu optimieren, wird die Dimension des Optimierungsraumes schrittweise erhöht. Dabei wird die Population der nächsten Stufe durch die zunächst neutrale Erweiterung des Reglers bereits mit einer gut angepassten Lösung der vorherigen Stufe vorinitialisiert.

### 3.2 Unterteilung einer Fuzzy-Menge

Im linken Teil des in Abbildung 3 dargestellten Beispiels wird eine ursprüngliche Fuzzy-Menge  $A_r$  durch die beiden Nachfolger  $\hat{A}_{r_1}$  und  $\hat{A}_{r_2}$  substituiert. Gleichzeitig wird diese Operation auf das nicht dargestellte Gegenstück  $A_l$  von  $A_r$  in der linken

Hälfte angewandt zur Generierung der beiden Nachfolger  $\tilde{A}_{l_1}$  und  $\tilde{A}_{l_2}$ . In der Folge werden diejenigen Regeln

$$R_i : (A_r^{(1)}, A_{i_2}^{(2)}, A_{i_3}^{(3)}, A_{i_4}^{(4)} \Rightarrow f_i),$$

die Bezug auf die ursprüngliche Fuzzy-Menge  $A_r$  nehmen, durch zwei neue Regeln

$$\tilde{R}_{k1} : (\tilde{A}_{r_1}^{(1)}, A_{i_2}^{(2)}, A_{i_3}^{(3)}, A_{i_4}^{(4)} \Rightarrow f_i)$$

und

$$\tilde{R}_{k2} : (\tilde{A}_{r_2}^{(1)}, A_{i_2}^{(2)}, A_{i_3}^{(3)}, A_{i_4}^{(4)} \Rightarrow f_i)$$

ersetzt, wobei die Konklusion  $f_i$  von der ursprünglichen Regel übernommen wird. Diese Strukturserweiterung erfolgt somit zunächst neutral, das heißt ohne Veränderung des Ein-/Ausgangsverhalten des Fuzzy-Systems. Dennoch erfolgt eine Erweiterung der genetischen Repräsentation des TSK-Fuzzy-Systems, in dem das Individuum um zusätzliche Parameter für die hinzugefügten Verstärkungsfaktoren sowie für die Stützstellen des Überlappungsbereichs zwischen  $A_{r_1}/A_{r_2}$  erweitert wird. Zur Beibehaltung des Ein-/Ausgangsverhaltens werden diese Objektvariablen mit den Werten der Vorgängerregel vorinitialisiert. Die Mutationsraten der entsprechenden Objektvariablen werden zudem vergrößert, damit die Evolutionsstrategie im Anschluss an diesen Erweiterungsschritt die zusätzlichen Freiheitsgrade zur Optimierung des Regelverhaltens nutzen kann.

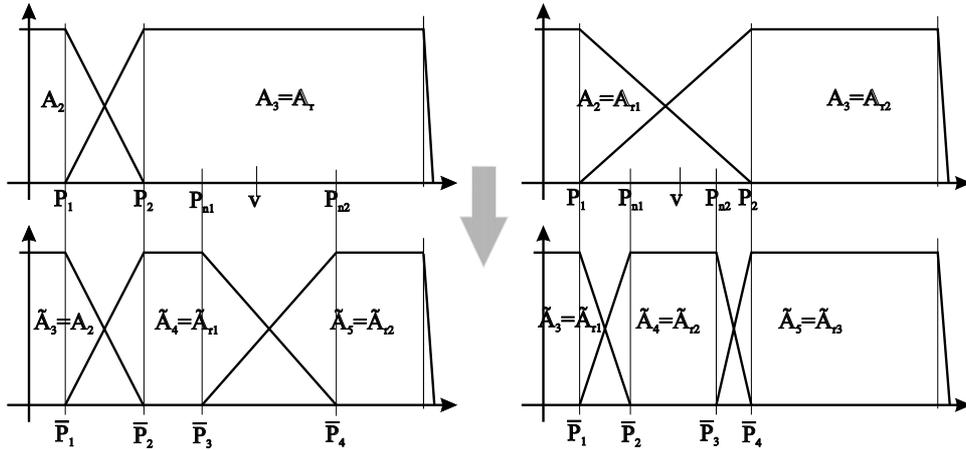


Abbildung 3: links: Einfügen einer neuen Fuzzy-Menge in den Kern, rechts: Einfügen einer Fuzzy-Menge zwischen zwei benachbarten Fuzzy-Mengen

### 3.3 Einfügen einer Fuzzy-Menge im Überlappungsbereich

Dieser Fall ist im rechten Teil der Abbildung 3 dargestellt, wobei aus Gründen der Übersichtlichkeit wieder nur der positive Wertebereich abgebildet ist. Die neue Zugehörigkeitsfunktion wird in den Überlappungsbereich zwischen den beiden benachbarten Fuzzy-Mengen  $A_{r_1}$  und  $A_{r_2}$  bzw.  $A_{l_1}$  und  $A_{l_2}$  eingefügt. Die Kernregionen der beiden ursprünglichen trapezförmigen Mengen  $A_{r_1}$  und  $A_{r_2}$  bleiben erhalten, dafür werden ihre Randbereiche verkleinert. Als Ergebnis der Erweiterung entstehen drei neue Zugehörigkeitsfunktionen  $\tilde{A}_{r_1}$ ,  $\tilde{A}_{r_2}$  und  $\tilde{A}_{r_3}$ .

$$R_i : (A_{r_1}^{(1)}, A_{i_2}^{(2)}, A_{i_3}^{(3)}, A_{i_4}^{(4)} \Rightarrow f_i)$$

sei wiederum eine Regel, deren Prämisse Bezug auf  $A_{r_1}$  nimmt. Es ist naheliegend, der neuen Regel

$$\tilde{R}_{k1} : (\tilde{A}_{r_1}^{(1)}, A_{i_2}^{(2)}, A_{i_3}^{(3)}, A_{i_4}^{(4)} \Rightarrow f_i),$$

mit Bezug auf die neue Fuzzy-Menge  $\tilde{A}_{r_1}$  die gleiche Konklusion  $f_i$  wie ihre Vorgängerregel zuzuordnen, da die ursprüngliche und die modifizierte Fuzzy-Menge sich den gleichen Kern teilen. Entsprechendes gilt für die Regel

$$\tilde{R}_{k3} : (\tilde{A}_{r_3}^{(1)}, A_{i_2}^{(2)}, A_{i_3}^{(3)}, A_{i_4}^{(4)} \Rightarrow f_k),$$

die wiederum aus der ursprünglichen Regel

$$R_k : (A_{r_2}^{(1)}, A_{i_2}^{(2)}, A_{i_3}^{(3)}, A_{i_4}^{(4)} \Rightarrow f_k)$$

gebildet wird. Da sich die verbleibende Regel

$$\tilde{R}_{k2} : (\tilde{A}_{r_2}^{(1)}, A_{i_2}^{(2)}, A_{i_3}^{(3)}, A_{i_4}^{(4)} \Rightarrow \tilde{f}_i)$$

auf die neu eingefügte Fuzzy-Menge im Übergangsbereich  $\tilde{A}_{r_2}^{(1)}$  bezieht, ergibt sich ihre Konklusion  $\tilde{f}$  aus der Interpolation der beiden Nachbarregeln. Hierzu lassen sich die Koeffizienten  $\tilde{k}$  von  $\tilde{f}$  als Mittelwerte der Koeffizienten  $k_i$  und  $k_k$  der ursprüngliche linken und rechten Nachbarkonklusion  $f_i$  und  $f_k$  berechnen:

$$\tilde{k}_i = \frac{1}{2}(k_i + k_k).$$

## 4 Ergebnisse

Das inkrementelle Entwurfsverfahren wird am Beispiel des TSK-Fuzzy-Reglerentwurfs für einen Aufschwingregler und einen Stabilisierungsregler für das inverse Rotationspendel erprobt. Die erzielten Ergebnisse werden im Folgenden am Beispiel des Stabilisierungsreglers detailliert beschrieben. Abschließend werden die Ergebnisse für den Aufschwingregler und für resultierende Gesamtregelungssystem zusammengefasst.

### 4.1 Stabilisierung

Die zur Optimierung eingesetzte Evolutionsstrategie verwendet eine (15,100)-Selektion, wobei die Nachkommen sowohl durch Mutation als auch durch Rekombination von Eltern entstehen. Die Anfangspopulation besteht aus TSK-Reglern mit einer einzigen global gültigen linearen Regel, deren Koeffizienten zufällig initialisiert und über zunächst 15 Generationen ohne Strukturveränderung optimiert werden. Die Anfangspopulation enthält noch keine stabilisierende Regler. Diese treten erst gegen Ende der ersten fünfzehn Generationen auf. Bei jedem Erweiterungsschritt verzweigt der evolutionäre Algorithmus, wie in Abschnitt 3.1 beschrieben, in vier Subpopulationen, die jeweils Zustandspartitionierungen entlang  $\varphi, \dot{\varphi}, \theta, \dot{\theta}$  entsprechen und sich für wiederum 15 Generationen parallel entwickeln. Die danach beste Subpopulation von Reglern überlebt und wird erneut einem der vier möglichen Erweiterungen unterzogen.

Stufe	# Gen.	Partition	$J$	$J_1$	$J_2$	# Regeln
0	15	—	instabil 0,15 ) <sup>1</sup>	— 0,09 ) <sup>1</sup>	— 0,06 ) <sup>1</sup>	1
1	30	$\theta$	349,6 4,6 ) <sup>2</sup>	17,3 2,9 ) <sup>2</sup>	332,3 1,7 ) <sup>2</sup>	3
2	45	$\varphi$	35,6	29,0	6,6	9
3	60	$\theta$	34,0	28,6	5,4	15
4	75	$\varphi$	33,2	27,8	5,4	25
5	90	$\theta$	31,4	26,2	5,2	35
6	105	$\theta$	23,5	15,5	8,0	105

Tabelle 1: Reihenfolge der Zustandsraumpartitionierung und Entwicklung der Gütefunktionale

In den ersten 15 Generationen der Stufe 0 werden die TSK-Regler für eine Anfangsauslenkung von  $\theta_0 = \pm 0.1$  bewertet und hinsichtlich des Gütefunctionals optimiert. Am Ende einer jeden Stufe  $S$  wird untersucht, bis zu welcher maximalen Anfangsauslenkung der beste gefundene Regler das Pendel noch stabilisieren kann. Diese Auslenkung dient in der nachfolgenden Stufe  $S + 1$  dann als Anfangsauslenkung für die Bewertung aller Regler. Der mit einer einzigen Regel arbeitende beste Regler aus Stufe 0 ist in der Lage, das Pendel für Anfangsauslenkungen aus dem Bereich  $\theta_0 \in [-0.3, 0.3]$  zu stabilisieren. Da in den folgenden Erweiterungsstufen sich alle gefundenen Regler als stabil erweisen und damit der Bestrafungsterm für vorzeitigen Abbruch  $J_0$  verschwindet, wird in der folgenden Analyse nur die Entwicklung der beiden Beiträge  $J_1$  und  $J_2$  betrachtet. Es zeigt sich, dass bereits die Regler der zweiten Erweiterungsstufe in der Lage sind, das Pendel für die dem maximalen Drehmoment entsprechende größtmögliche Anfangsauslenkung von  $\theta_0 = 0.6$  zu stabilisieren. Für größere Auslenkung reicht das vom Motor über den horizontalen Arm generierte Drehmoment nicht aus, um die Schwerkraft auszugleichen.

Nach Tabelle 1 führt im ersten Erweiterungsschritt die Partitionierung entlang der Eingangsgröße  $\theta$  zur größten Verbesserung des Regelverhaltens. Dies ist insofern verständlich, als dass diese Zustandsgröße den signifikantesten Beitrag zur Nichtlinearität der Dynamik liefert. Der optimierte Fuzzy-Regler mit drei TSK-Regeln stabilisiert das Pendel für Auslenkungen im Bereich  $\theta_0 \in [-0.6, 0.6]$ . Die nachfolgenden Erweiterungsstufen entlang der Zustandsgrößen  $\theta$  und  $\varphi$  führen zu einer Reduzierung der Werte des Gütefunctionals. Der wesentliche Beitrag zur Verbesserung der Güte stammt aus der Reduzierung des Stellgrößenverbrauchs  $J_2$  und weniger des quadratischen Zustandsfehlers  $J_1$ . Es sei darauf hingewiesen, dass die Fitnesswerte für  $J_1$  und  $J_2$  sich jeweils auf eine Anfangsauslenkung  $\theta_0 = 0.6$  beziehen, welche der Regler der ersten Stufe noch nicht stabilisieren kann. Bis hin zur sechsten Stufe reduzieren sich Stellgrößenverbrauch und Zustandsfehler mit wachsender Anzahl von Regeln. In der letzten Stufe nimmt dann der Zustandsfehler signifikant ab, bei einem gleichzeitig anwachsendem Stellgrößenverbrauch. Dies wird in in Abbildung 4 dargestellten Regelverhalten deutlich, in welchem die dunkle Linie dem finalen Regler mit dem geringsten Zustandsfehler entspricht.

<sup>1</sup>optimiert für  $\theta_0 = \pm 0.1$ , <sup>2</sup> optimiert für  $\theta_0 = \pm 0.3$

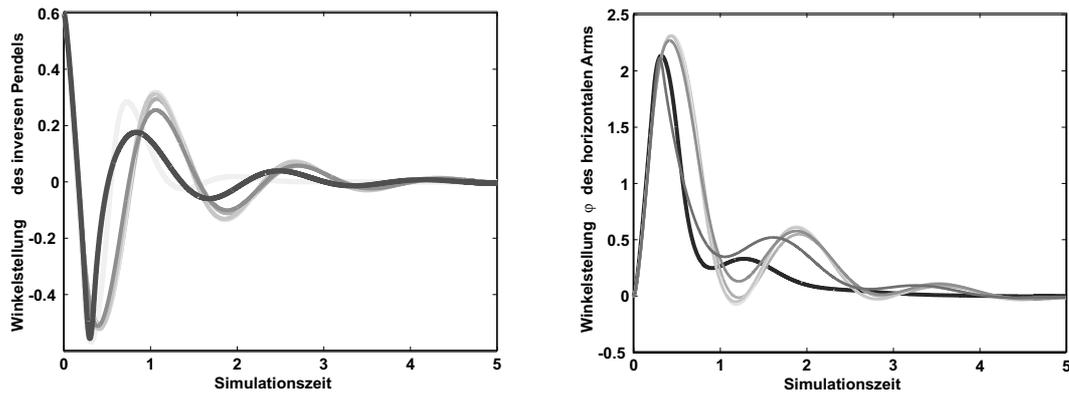


Abbildung 4: Entwicklung des Regelverhaltens für verschiedene Strukturweiterungen des TSK-Fuzzy-Reglers.

Abbildung 4 stellt den Verlauf der beiden Winkel für das Pendel (oben) und den horizontalen Arm (unten) dar, bei einer Störauslenkung von  $\theta = 0.6$ . Jede Abbildung enthält sechs zeitliche Verläufe welche den optimierten Reglern der Erweiterungsstufen 1-6 aus der Tabelle entsprechen. Dunklere Linien korrespondieren mit komplexeren Reglern mit einer größeren Anzahl von Regeln. Die Ergebnisse bestätigen eine kontinuierliche Verbesserung des Regelverhaltens mit zunehmender Komplexitätsstufe und damit verbundenen zusätzlichen Freiheitsgraden des Reglers.

Die Entwicklung des Kennfeldes über die zwei Erweiterungsstufen ist in Abbildung 5 zu sehen. Es sei angemerkt, dass die Graphiken jeweils die Projektion des vierdimensionalen Kennfeldes auf die Zustandsgrößen  $\varphi$  und  $\theta$  enthalten. In der linken Abbildung ist das beste lineare Reglergesetz abgebildet. In den nachfolgenden Erweiterungen wird deutlich, wie die Nichtlinearität des Kennfeldes mit wachsender Anzahl von Regeln zunimmt.

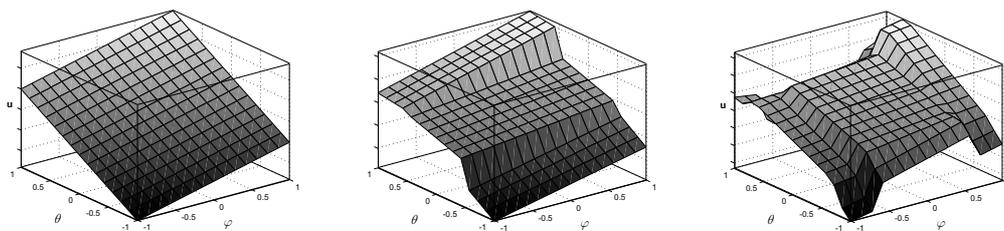


Abbildung 5: Entwicklung des Kennfeldes für die ersten zwei Erweiterungsstufen

Im Folgenden wird das Konvergenzverhalten des inkrementellen Reglerentwurfs verglichen mit der direkten Optimierung von Beginn an vollständigen TSK-Reglern verschiedener Komplexität. Zu diesem Zweck wurden die Zustandsraumpartitionierungen der Entwicklungsstufen in Tabelle 1 zugrunde gelegt, indem TSK-Regler mit 15, 25 und 35 Regeln einer direkten Optimierung unterworfen wurden. Dabei zeigen die jeweils besten gefundenen Lösungen ein deutlich schlechteres Verhalten, als die äquivalenten mit dem inkrementellen Ansatz entworfenen Regler der Stufen 3 bis 5. Diese Ergebnisse unterstreichen, dass die inkrementelle Erweiterung der Struktur und simultane Anpassung der zusätzlichen Freiheitsgrade das Optimierungsproblem vereinfacht.

## 4.2 Aufschwingen

Die Anwendung des inkrementellen evolutionären Entwurfsverfahrens für einen TSK-Aufschwingregler erfolgt analog der vorher beschriebenen Vorgehensweise unter Verwendung des Gütefunktionals aus Gleichung (12). Dabei zeigt sich, dass schon die optimierten Regler der Stufe 0 das inverse Pendel erfolgreich aufschwingen können. Die nachfolgenden Optimierungsstufen reduzieren lediglich die zum Aufschwingen benötigte Zeit  $t_{0,2}$ . Das in Abbildung 6 dargestellte zeitliche Verhalten des Pendels illustriert den erzielten Zeitgewinn von Stufe 5 gegenüber auf Stufe 0.

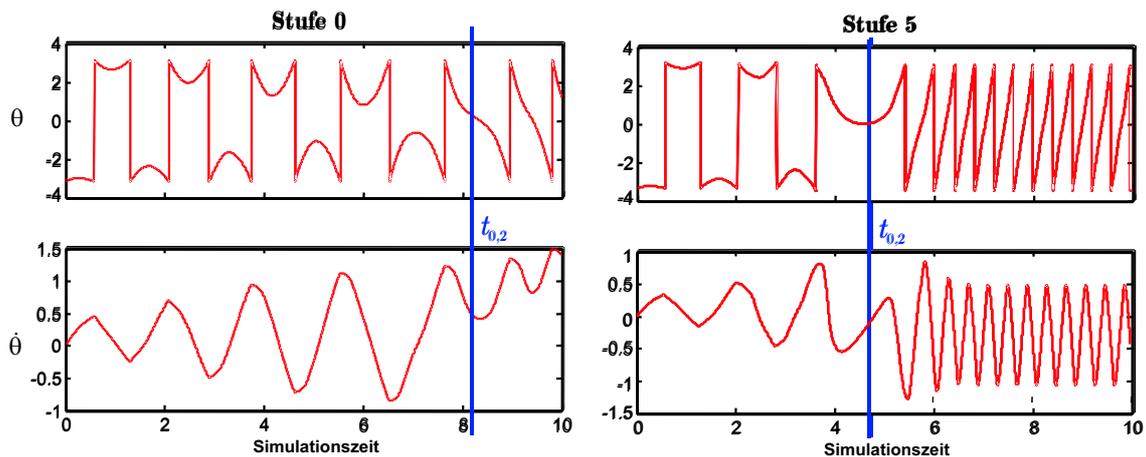


Abbildung 6: Ausgangsverhalten des Aufschwingreglers zu Beginn (Stufe 0) und Ende (Stufe 5) der strukturellen Evolution.

## 4.3 Gesamtregler

Die beiden generierten TSK-Regler werden über ein Hystereseglied derart verknüpft, dass der Zustandsraum des inversen Rotationspendels vollständig gemäß Abbildung 7 abgedeckt wird. Während des Aufschwingvorgangs wird bei Erreichen des

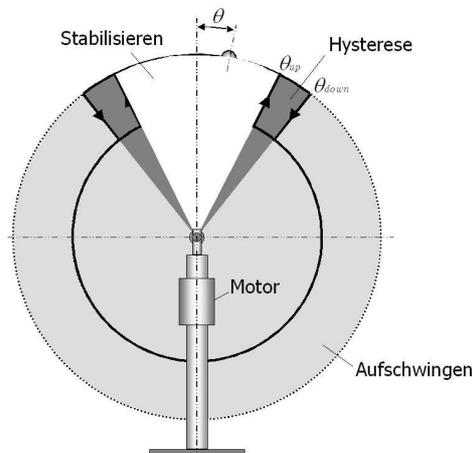


Abbildung 7: Einteilung der Aktivierung von Aufschwing- und Stabilisierungsregler in Abhängigkeit von  $\theta$ .

oberen Umschaltwinkels  $\theta_{up} = 17^\circ$  auf den stabilisierenden Regler umgeschaltet. Unterschreitet das Pendel z. B. durch eine Störung den unteren Umschaltwinkel  $\theta_{down} = 23^\circ$ , wird wieder der Aufschwingregler aktiviert.

Das erreichte gute Regelungsverhalten des so gewonnenen Gesamtsystems ist in Abbildung 8 als gestrichelte Linie dargestellt. Zur Analyse der Robustheit des Regelungssystems werden zwei verschiedene Störgrößen auf den Eingang des Pendelmodells überlagert. Das Störsignal besteht einerseits aus einem dauerhaften gaußverteilten Rauschen mit einer Varianz von  $\sigma^2 = 0.0005$ . Zusätzlich wird zum Zeitpunkt  $t^* = 8s$  dem bereits stabilisierten Pendel ein Rechteckimpuls mit einer Pulsweite von  $T = 200ms$  und einer Amplitude von 0.1 aufgeschaltet. Diese Störung ist so groß, dass der stabilisierende Regler diese nicht ausregeln kann, so dass vorübergehend der Aufschwingregler aktiviert wird. Als Ergebnis zeigt sich insgesamt ein sehr robustes Regelungsverhalten, welches nach dem Störimpuls zu einem schnellen Wiederaufschwingen und dauerhaftem Stabilisieren des inversen Rotationspendels führt (durchgezogene Linien in Abbildung 8).

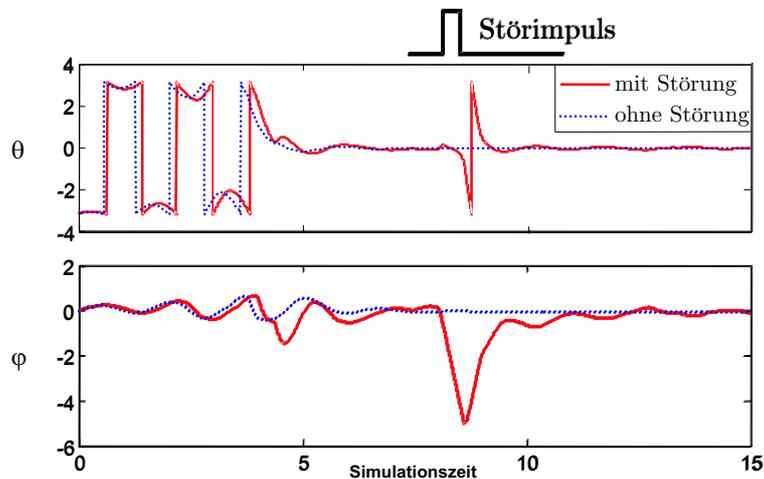


Abbildung 8: Ausgangsverhalten des Regelungssystems mit und ohne Störsignale.

## 5 Zusammenfassung

Der vorgeschlagene Ansatz zur gleichzeitigen Struktur- und Parameteroptimierung von TSK-Fuzzy-Reglern bietet zwei Vorteile gegenüber der herkömmlichen evolutionären Optimierung von Fuzzy-Reglern. Zum einen wird die für die Regelungsaufgabe am besten geeignete Struktur automatisch identifiziert und muss nicht im Vorhinein vom Anwender festgelegt werden. Zum zweiten wird die Optimierungsaufgabe in für den evolutionären Algorithmus leichter zu lösende Teilprobleme unterteilt. Die Population von Reglern der nächst höheren Komplexitätsstufe setzt sich bereits mit guten Lösungen der vorherigen Stufe zusammen, sodass anstelle einer globalen Suche lediglich eine lokale Anpassung der zusätzlichen Freiheitsgrade notwendig ist. Diese graduelle Optimierung wird dadurch möglich, dass neue Regeln auf zunächst einmal neutrale Weise der vorhandenen Regelbasis hinzugefügt werden und dadurch das bereits teiloptimierte Regelverhalten erhalten bleibt. Die Praxistauglichkeit der Methode wurde am Beispiel des TSK-Reglerentwurfs für ein inverses

Rotationspendel demonstriert. In diesem Zusammenhang wurden mit der gleichen Methode ein Aufschwingregler und ein stabilisierender Regler generiert. Die dabei erzielten Regelverhalten sind gegenüber denjenigen, die mittels direkter Optimierung eines TSK-Fuzzy-Reglers mit adäquater vollständiger Struktur erzielbar sind, überlegen.

Bisher wurde der evolutionäre Algorithmus nur für den Entwurf von TSK-Reglern innerhalb einer dynamischen Simulation des Rotationspendels durchgeführt. Um eine erfolgreiche Übertragung auf das reale Pendel zu ermöglichen, ist es erforderlich, einen robusten Regler zu entwerfen, bei welchem während der Optimierung die Parameterunsicherheiten der Strecke berücksichtigt werden. Dies kann beispielsweise dadurch geschehen, dass die Gütefunktion nicht nur für das nominale Pendelmodell ausgewertet, sondern die Gütefunktionale über ein breites Spektrum von Pendellängen und -massen gemittelt werden.

## Literatur

- [1] CORDÓN, O. ; HERRERA, F. ; HOFFMANN, F. ; MAGDALENA, L.: *Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*. Singapore : World Scientific, July 2001 (Advances in Fuzzy Systems)
- [2] MAGDALENA, L. ; VELASCO, J.R.: Evolutionary Based Learning of Fuzzy Controllers. In: PEDRYCZ, W[itold] (Hrsg.): *Fuzzy Evolutionary Computation*. Kluwer Academic Publishers, 1997, S. 249–268
- [3] KARR, C[harles]: Genetic algorithms for fuzzy controllers. In: *AI Expert* 6 (1991), Februar, Nr. 2, S. 26–33
- [4] CORDON, O. ; HERRERA, F. ; ZWIR, I.: Linguistic Modeling by Hierarchical Systems of Linguistic Rules. In: *IEEE Transactions on Fuzzy Systems* 10 (2002), S. 2–20
- [5] HOFFMANN, F. ; NELLES, O.: Genetic Programming for Model Selection of TSK-Fuzzy Systems. In: *Journal of Information Sciences* 136 (2001), August, Nr. 1-4, S. 7–28
- [6] ANGELOV, P. ; BUSWELL, R.: Identification of evolving fuzzy rule-based models. In: *IEEE Transactions on Fuzzy Systems* 10 (2002), Nr. 5, S. 667–677
- [7] ROUBOS, J. ; BABUSKA, R. ; BRUIJN, P. ; VERBRUGGEN, H.: Predictive control by local linearization of a takagi-sugeno fuzzy model. In: *Proceedings of FUZZ-IEEE-98*. 1998, S. 37–42

# Auf dem Weg zum industrietauglichen Evolutionären Algorithmus

**Wilfried Jakob**

Institut für Angewandte Informatik, Forschungszentrum Karlsruhe  
Postfach 3640, 76021 Karlsruhe  
Tel. (07247) 82-4663  
Fax (07247) 82-2602  
E-Mail: wilfried.jakob@iai.fzk.de

## 1 Motivation

Evolutionäre Algorithmen (EA) zeichnen sich durch ihre allgemeine Anwendbarkeit aus und sind dazu geeignet, Lösungen oder zumindest Näherungslösungen für Aufgaben zu finden, für die es z.B. auf Grund von Nichtlinearitäten keine mathematischen Lösungsverfahren gibt. Zu den in diesem Sinne erfolgreich bewältigten Aufgaben gehören auch viele NP-vollständigen Probleme. Diesem Vorteil steht der Nachteil der EA gegenüber, dass ihre Leistungsfähigkeit stark von einer geeigneten Wahl von zum Teil verfahrensspezifischen Strategieparametern wie z.B. der Populationsgröße oder Mutationsraten abhängt. Dazu kommt das Problem, dass EA in der Nähe eines Optimums schlecht konvergieren. Letzteres wird bei praktischen Anwendungen meist durch Hybridisierung gelöst, d.h. man unterstützt die evolutionäre Suche durch ein in der Regel anwendungsspezifisches (lokales) Suchverfahren (LSV) [1-5]. Dem Vorteil einer erheblichen Steigerung der Konvergenzgeschwindigkeit stehen zwei Nachteile gegenüber: Zum einen wird aus dem allgemein anwendbaren EA ein problemspezifisches Werkzeug und zum anderen kommen weitere Strategieparameter, nämlich die des lokalen Verfahrens hinzu. Zumindest muss die Intensität der lokalen Suche und damit die Aufteilung der Rechenleistung zwischen den beiden Algorithmen festgelegt werden. All das macht den (hybriden) EA zu einem Werkzeug für Spezialisten und man muss leider feststellen, dass sich Evolutionäre Algorithmen zwar durch ein breites Anwendungsspektrum, aber nicht durch eine breite Anwendung auszeichnen.

An ein industrietaugliches Optimierungs- oder Planungsverfahren können folgende Forderungen gestellt werden:

1. Klare Abgrenzung des Einsatzgebiets
2. Robustheit: Das Verfahren muss auch bei „ungünstigen“ Parametern der Anwendung oder Anwendungssituationen funktionieren.
3. Einfachheit: Möglichst wenig Verfahrensparameter, die nicht zu sensibel auf schlechte Einstellungen reagieren. Man sollte nur wenig Vorwissen für einen erfolgreichen Einsatz benötigen.
4. Es sollte möglich sein, existierende Lösungen in das Verfahren einzubeziehen. Es erhöht die Akzeptanz beim Anwender erheblich, wenn ein neues Verfahren zumindest nicht schlechter ist als das bisherige, da es dessen Lösungen verbessert.
5. Geschwindigkeit: Die beste Lösung nützt nichts, wenn sie zu spät kommt.

Für EA kann allgemein gesagt werden, dass sie als robust gelten und dass bei EA mit elitärer Akzeptanz der Nachkommen durch die Aufnahme existierender Lösungen in die Startpopulation zumindest deren Qualität als Ergebnis erreicht wird. Die Frage der Geschwindigkeit kann nur anwendungsabhängig beantwortet werden: Zum einen legt die Anwendung den verfügbaren Zeitrahmen für eine Optimierung fest und zum anderen bestimmt ihre Komplexität und die Anzahl ihrer Parameter den Zeitbedarf für einen Optimierungslauf genauso wie die Dauer der Evaluation einer vorgeschlagenen Lösung (Fitnessberechnung).

Der vorliegende Beitrag konzentriert sich auf die Einfachheit der Handhabung des Verfahrens unter Eingrenzung des Anwendungsgebiets auf *Parameteroptimierung*, wobei je nach Anwendungsfall von bis zu einigen Dutzend Parametern ausgegangen werden kann. Ziel ist eine für EA allgemein anwendbare Methode zur adaptiven Hybridisierung und dadurch eine deutliche Verringerung der Strategieparameter des resultierenden Verfahrens. Vom EA wird dabei gefordert, dass die Population größer als eins ist und dass die Nachkommen mit den Eltern um die Aufnahme in die Nachfolgeneration konkurrieren (Plus-Strategie im Sinne der ES-Terminologie [6]). Außerdem ist es hilfreich, wenn pro Paarung mehrere Nachkommen erzeugt werden.

Kapitel 2 gibt einen Überblick über relevante Strategieparameter der hier behandelten Klasse hybrider evolutionärer Algorithmen, die auch unter der Bezeichnung *Memetische Algorithmen* bekannt sind. Außerdem werden die für die experimentellen Überprüfungen verwendeten Verfahren kurz vorgestellt. Kapitel 3 behandelt das neue Konzept zur adaptiven Hybridisierung und vergleicht es mit anderen Ansätzen. Über die zur Überprüfung der Tauglichkeit der neuen Methode durchgeführten Experimente wird schließlich im 4. Kapitel berichtet, bevor der Beitrag mit einer Zusammenfassung der Ergebnisse endet.

## 2 Evolutionäre und memetische Algorithmen

In Bild 1 ist der Pseudocode eines typischen EA dargestellt. Die kursiv hervorgehobene Integration eines LSVs in die Produktion der Nachkommen macht ihn zum Memetischen Algorithmus [3]. In der Regel werden durch Anwendung genetischer Operatoren wie Crossover und verschiedene Mutationen mehrere Nachkommen erzeugt, von denen entweder nur das Beste oder alle lokal verbessert werden. In der Literatur

```

Initialize and evaluate start population
REPEAT UNTIL stop condition is satisfied (generational loop)
  FOR all individuals of the population
    Choose partner (within neighborhood, ranking-based selection)
    FOR all genetic operations (set of genetic operators)
      Produce offspring and evaluate it or them
    IF improve best only
      Improve best offspring by preselected local searcher
    ELSE
      Improve all offspring by preselected local searcher
    IF Lamarckian evolution
      Update chromosome of the best offspring to its improved version
    Accept or reject best offspring according to the acceptance rule
  Deliver best individual (and other, if required) as result
  
```

**Bild 1:** Typischer Ablauf eines EA mit integrierter lokaler Verbesserung (*kursiv*). Die Besonderheiten des nachfolgend beschriebenen EA sind durch den Times-Font hervorgehoben.

kommt auch noch die zufällige Auswahl eines Teils der zu verbessernden Nachkommen vor. Anschließend wird das Chromosom im Falle von Lamarckscher Evolution an die gefundene Verbesserung angepasst oder nicht, so dass nur der Fitnesszuwachs wirksam ist (Baldwin-Evolution). Die Frage, welcher der beiden Evolutionsarten der Vorzug zu geben ist, wird in der Literatur kontrovers diskutiert [7,8]. Meist wird der Baldwin-

Evolution der Vorzug gegeben, da sie eine größere genetische Vielfalt über einen längeren Zeitraum bewirkt. Eigene Untersuchungen haben jedoch gezeigt, dass bei Verwendung des von Gorges-Schleuter [9] vorgeschlagenen Nachbarschaftsmodells, das der Gefahr vorzeitiger Konvergenz wirksam entgegentritt, die Lamarcksche Evolution zum Teil erheblich besser abschneidet [4,5]. Dies gilt übrigens auch für die Evolutionsstrategie [10]. Daher wurden die hier beschriebenen Experimente mit Lamarckscher Evolution durchgeführt.

## 2.1 Strategieparameter

Zu den wichtigsten EA-Strategieparametern zählen die Populationsgröße, Mutations- und Crossoverraten, die Anzahl der pro Paarung erzeugten Kinder und weitere Parameter, die von der konkreten Ausgestaltung des EA abhängen, wie z.B. der Rankingfaktor bei rankingbasierter Selektion. Es wird davon ausgegangen, dass sich diese Parameter bis auf die Populationsgröße EA-spezifisch so einstellen lassen, dass sie für ein breites Spektrum von Parameteroptimierungsaufgaben geeignet sind.

Die Integration lokaler Suche in einen EA kann als Vorooptimierung der Startpopulation, als Nachoptimierung der EA-Ergebnisse oder in Form eines Memetischen Algorithmus erfolgen. Da eigene Untersuchungen die Überlegenheit der Memetischen Variante deutlich gezeigt haben [4,5], wird im Folgenden nur noch diese Art der Hybridisierung betrachtet. Hierbei sind folgende Strategieparameter von Bedeutung:

1. Auswahl des LSVs sofern mehrere zur Verfügung stehen
2. Auswahl der Anzahl zu verbessernder Nachkommen pro Paarung: nur das beste, alle, zufallsbestimmt
3. Genauigkeit der lokalen Suche (Anzahl der Iterationen, LSV-Abbruchschranke, ...)

Die beiden letzten Parameter bestimmen letztlich die Aufteilung der Rechenzeit zwischen evolutionärer und lokaler Suche, eine Frage, auf deren grundlegende Bedeutung Goldberg und Voessner bereits 1999 hingewiesen haben [11]. Der in [12] weiterentwickelte systemtheoretische Ansatz kann leider für praktische Anwendung keine verwertbaren Empfehlungen geben, wie die Autoren selber einräumen.

## 2.2 GLEAM und HyGLEAM

Das von Blume 1990 vorgestellte GLEAM (**General Learning Evolutionary Algorithm and Method**) [13] kann als repräsentativer EA-Vertreter angesehen werden, da GLEAM verschiedene Eigenschaften der klassischen EAs in sich vereint und durch seine flexible Codierung ein breites Anwendungsspektrum abdeckt. Geeignete lokale Suchverfahren sollen wegen der Bewahrung der allgemeinen Anwendbarkeit des resultierenden Hybrids möglichst wenig Anforderungen an den Suchraum stellen und Beschränkungen, die bei praktischen Problemen fast immer vorkommen, berücksichtigen können. Daher wurden zwei als robust und leistungsfähig bekannte Algorithmen ausgewählt, nämlich das Rosenbrock- [14] und das Complex-Verfahren [15], im Folgenden mit *R* und *C* abgekürzt. Da sich der vorliegende Beitrag auf eine adaptive Hybridisierung konzentriert, kann aus Platzgründen nur kurz auf die beteiligten Verfahren eingegangen werden und es wird auf die angegebene Literatur verwiesen. Eine ausführliche Beschreibung von GLEAM kann in [5] und von den beiden LSV in [16] gefunden werden.

GLEAM ist ein eigenständiger Evolutionärer Algorithmus, der Elemente der Evolutionsstrategie [6,16] und der (reellcodierten) Genetischen Algorithmen [2,17] mit Konzepten der Informatik (abstrakte Datentypen) verbindet. Die Codierung der zu optimie-

renden Parameter erfolgt in den so genannten Aktionen, die aus einem oder mehreren Parametern vom Typ Bool, Integer oder Real entsprechend ihrem anwendungsspezifisch definierten Aktionstyp bestehen. Eine Aktion drückt damit eine bestimmte Eigenschaft einer möglichen Lösung aus, z.B. bei einem Design die Anwesenheit einer bestimmten Komponente, die durch die Aktionsparameter näher bestimmt wird oder die Relationen zu anderen Komponenten entsprechend den Aktionsparametern eingeht. Im einfachsten Fall codieren die Aktionen nur bestimmte einzelne Parameter einer Lösung. Die Aktionen bilden eine Aktionskette (AK), die den Chromosomen des biologischen Vorbilds entspricht. Der hier interessierende AK-Typ ist der einfachste von GLEAM, bei dem die Aktionsreihenfolge keine Bedeutung trägt und jeder Aktionstyp mit genau einer Aktion in der AK vertreten ist. Bild 1 zeigt den grundsätzlichen Ablauf von GLEAM mit Reproduktion in der lokalen Nachbarschaft und ranking-basierter Partnerwahl. Als Akzeptanzstrategie für das beste Nachkommen wird die Konkurrenz mit dem Elter benutzt, wobei es entweder besser als das Elter sein muss oder besser als das lokal schlechteste und keine Verschlechterung der Nachbarschaft stattfinden darf (elitäre Ersetzungsstrategie). Das Codierungskonzept von GLEAM erlaubt eine einfache Abbildung von kombinatorischen Problemen auf die Parameteroptimierung und macht sie so einer Bearbeitung mit HyGLEAM zugänglich. Dazu wird jede Aktion um einen Permutationsparameter erweitert, der bestimmt, mit welcher anderen Aktion der Platz bei der AK-Auswertung getauscht wird. Der neue Parameter bestimmt so die Reihenfolge der Interpretation der Aktionen und die Aktionen müssen nun nicht mehr durch entsprechende Mutationen im Chromosom vertauscht werden.

Das Rosenbrock-Verfahren ist eine modifizierte Koordinatenstrategie und der Complex-Algorithmus ein Polyederverfahren, das für Restriktionen erweitert wurde. Beide Verfahren kommen ohne Ableitungen aus und werden gemäß der Implementierung von Schwefel [16] benutzt. Das Rosenbrock-Verfahren bricht ab, wenn ein Maß für die Länge des zurückgelegten Weges und die Richtungsänderung einen vorgegebenen Wert unterschreitet. Der Complex-Algorithmus terminiert, wenn entweder fünf mal hintereinander keine Verbesserung eintrat oder wenn die gleiche implizite Beschränkung fünf mal hintereinander eine unzureichende Kontraktion veranlasst hat. Die Implementierung beider Verfahren erlaubt außerdem die Vorgabe einer Iterationsgrenze.

In memetischen Teil von HyGLEAM (**H**ybrid **G**eneral Purpose **E**volutionary **A**lgorithm and **M**ethod) werden die beiden LSV wie in Bild 1 dargestellt in die Produktion der Nachkommen integriert. Dabei kann eingestellt werden, ob nur der beste oder alle Nachkommen lokal verbessert werden und ob der Genotyp angepasst werden soll oder nicht.

### **3 Adaption in memetischen Algorithmen**

Hart [18] und Krasnogor [19] haben gezeigt, dass die Wahl eines geeigneten LSVs anwendungsabhängig ist und einen wesentlichen Einfluss auf die Suchgeschwindigkeit hat. Eine weitere wesentliche Größe zur Parametrierung ist die bereits erwähnte Aufteilung der Rechenzeit zwischen globaler und lokaler Suche. Im Folgenden wird ein Konzept zur adaptiven Einstellung beider Parameter vorgestellt und mit anderen Ansätzen verglichen.

#### **3.1 Konzept der Kosten-Nutzen basierten Adaption**

Die adaptive Steuerung beruht auf dem beobachteten Erfolg ausgedrückt durch den erzielten Fitnesszuwachs und den dazu notwendigen Kosten in Form von Fitnessberech-

nungen. Sie wird zunächst am Beispiel der Aufteilung zwischen den beiden lokalen Verfahren beschrieben. Anfänglich ist die Wahrscheinlichkeit für ihre Anwendung gleich. Die Anzahl der Anwendungen wird je Verfahren gezählt und der jeweils erreichte relative Fitnessgewinn  $rfg$  wird zusammen mit den dazu benötigten Evaluationen  $eval$  aufsummiert. Da ein bestimmter Fitnesszuwachs (Fitness nach der lokalen Suche ( $f_{LS}$ ) abzüglich der Fitness nach Anwendung der genetischen Operatoren ( $f_{evo}$ )) mit zunehmender Ausgangsfitness  $f_{evo}$  höher zu bewerten ist, wird statt des absoluten Wertes der Fitnessdifferenz ein relatives Maß benutzt, das sich am noch möglichen Fitnessgewinn orientiert. Dazu wird die Fitnessfunktion im Bereich  $0 \dots f_{max}$  normiert und  $rfg$  wie folgt berechnet:

$$rfg = \frac{f_{LS} - f_{evo}}{f_{max} - f_{evo}} \quad (1)$$

Die Ausführungswahrscheinlichkeiten der lokalen Verfahren werden neu justiert, wenn entweder jedes Verfahren mindestens  $usage_{min}$  mal benutzt wurde oder nach spätestens  $matings_{max}$  Paarungen. Die neue Relation zwischen den beiden lokalen Verfahren berechnet sich wie folgt:

$$\frac{\sum rfg_{i,compl}}{\sum eval_{i,compl}} : \frac{\sum rfg_{j,rosen}}{\sum eval_{j,rosen}} \quad (2)$$

Die Summen werden nach der Anpassung zurückgesetzt, um eine schnellere Anpassung zu erreichen. Wenn die Anwendungswahrscheinlichkeit eines der Verfahren dreimal hintereinander weniger als ein  $P_{min}$  (0.1 in den Experimenten) beträgt, wird es abgeschaltet. Um einer voreiligen Abschaltung entgegenzuwirken, wird bei erstmaliger Unterschreitung von  $P_{min}$  die berechnete Wahrscheinlichkeit auf  $P_{min}$  heraufgesetzt.

Der Ansatz kann leicht auf weitere Suchverfahren oder Parameter wie die Genauigkeit der lokalen Suche erweitert werden. Dazu wird der Wertebereich in eine Anzahl von Bereichen unterteilt, was am Beispiel der Iterationsgrenze erläutert werden soll. Hierfür mögen 10 *Level* genannte Bereiche genügen, für die beispielsweise Werte zwischen 100 und 2000 zu Grunde gelegt werden, siehe auch Tabelle 1. Es sind immer genau drei Level gleichzeitig aktiv, d.h. sie haben eine Wahrscheinlichkeit, ausgewählt zu werden, von  $p > 0$ . Die Anpassung der

...	<b>Level 2</b> $p = 0$ $v = 200$	<b>Level 3</b> $p = 0.15$ $v = 350$	<b>Level 4</b> $p = 0.25$ $v = 500$	<b>Level 5</b> $p = 0.6$ $v = 750$	<b>Level 6</b> $p = 0$ $v = 1000$	...
...	<b>Level 2</b> $p = 0$ $v = 200$	<b>Level 3</b> $p = 0$ $v = 350$	<b>Level 4</b> $p = 0.4$ $v = 500$	<b>Level 5</b> $p = 0.6$ $v = 750$	<b>Level 6</b> $p = 0$ $v = 1000$	...
...	<b>Level 2</b> $p = 0$ $v = 200$	<b>Level 3</b> $p = 0$ $v = 350$	<b>Level 4</b> $p = 0.32$ $v = 500$	<b>Level 5</b> $p = 0.48$ $v = 750$	<b>Level 6</b> $p = 0.2$ $v = 1000$	...

**Bild 2:** Parameteranpassung am Beispiel der Überschreitung des Wahrscheinlichkeitsgrenzwertes von Level 5. Die jeweils momentan aktiven Levels sind grau hinterlegt. Die Levelwerte sind mit  $v$  angegeben.

Wahrscheinlichkeiten erfolgt wie bei der LSV-Auswahl beschrieben. Wenn dabei der niedrigste oder höchste aktive Level eine Wahrscheinlichkeit von mehr als 50% erhält, erfolgt eine Levelanpassung, wie in Bild 2 dargestellt. Dazu wird zunächst der Level am anderen Ende deaktiviert und seine Wahrscheinlichkeit dem Nachbarlevel zugeschlagen (mittlere Zeile von Bild 2). Der neue Level (Level 6 im Bild) erhält 20% der Wahrscheinlichkeit der anderen beiden und wird damit aktiviert (letzte Zeile in Bild 2). Damit wird eine Bewegung der aktiven Levels auf der Skala möglicher Level entspre-

chend dem erreichten Fitnesszuwachs und der dazu erforderlichen Evaluationen erreicht. Da die geringste Levelwahrscheinlichkeit auf 10% festgesetzt wird, bleiben auch bei Erreichen des Skalenendes drei Level aktiv, wodurch die Mobilität dauerhaft gewährleistet bleibt. Mit Hilfe dieses Ansatzes können auch weitere Strategieparameter der Hybridisierung angepasst werden; konkret im vorliegenden Fall sind das neben den Iterationsgrenzen die normierte Abbruchschranke des Rosenbrock-Verfahrens  $th_R$  und die Wahrscheinlichkeit  $all-impr_R$  und  $all-impr_C$ , mit der weitere Nachkommen einer Paarung außer dem besten lokal optimiert werden (*all-Verbesserung*).

Strategieparameter	Werte									
$limit_R, limit_C$	100	200	350	500	750	1000	1250	1500	1750	2000
$th_R$	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$	$10^{-6}$	$10^{-7}$	$10^{-8}$	$10^{-9}$	
$all-impr_R, all-impr_C$	0	0.2	0.4	0.6	0.8	1.0				

**Tabelle 1:** Adaptiv kontrollierte Strategieparameter mit Anzahl und Werten ihrer Levels.

Der Gedanke liegt nahe, die lokale Suche anfänglich nur recht grob zu betreiben und erst im Verlaufe der Evolution zu präzisieren. Schließlich genügt die ungefähre Bestimmung eines lokalen Optimums, solange sie genau genug ist, um zwischen den lokalen Optima korrekt differenzieren zu können. Daher werden die beiden Levels der Iterationsgrenzen ( $limit_R$  und  $limit_C$ ) beginnend mit dem niedrigsten mit Wahrscheinlichkeitswerten von 50%, 30% und 20% initialisiert, während die beiden anderen Levels jeweils mit einem Drittel vorbelegt werden. Tabelle 1 gibt die adaptiv angepassten Strategieparameter und ihre Level-Werte wieder.

Bild 3 zeigt den für das adaptive HyGLEAM/A erweiterten Pseudocode von Bild 1, der noch zwei extern einzustellende Optionen aufweist: Die Entscheidung über die Verbesserung nur des besten Nachkommen (*improve best only, best-Verbesserung*) oder aller und die über die Anpassung des Genotyps (*Lamarckian evolution*). Alle anderen Entscheidungen (*Choose ...*) erfolgen durch den beschriebenen adaptiven Mechanismus. Dabei gibt es bei der all-Verbesserung folgende Besonderheit: Wenn nach erfolgter lokaler Verbesserung ein anderer Nachkomme als der Evolutionsbeste die größte Fitness hat, es also richtig war, nicht nur den Evolutionsbesten lokal zu verbessern, dann wird die Differenz zwischen der besten Fitness der Nachkommen und des verbesserten Evolutionsbesten als Belohnung zum Fitnesszuwachs hinzuad-

```

Initialize and evaluate start population
REPEAT UNTIL stop condition is satisfied (generational loop)
  FOR all individuals of the population
    Choose partner (within neighborhood, ranking-based selection)
    FOR all genetic operations (set of genetic operators)
      Produce offspring and evaluate it or them
    IF improve best only
      Choose local searcher and its parameters
      Improve best offspring by selected local searcher
      Record effort and fitness gain
    ELSE
      Choose local searcher and LS_probability
      FOR all offspring
        IF best evo_offspring OR improve according to LS_probability
          Choose parameters of the LS
          Improve offspring by selected local searcher
          Record effort and fitness gain for LS selection and
            parameterization
        IF best improved offspring other than best evo_offspring
          Add difference to fitness gain as reward
          Record effort and fitness gain for LS_probability
      IF Lamarckian evolution
        Update chromosome of the best offspring to its improved version
      Accept or reject best offspring according to the acceptance rule
  Deliver best individual (and other, if required) as result

```

**Bild 3:** Um die Adaption erweiterter Pseudocode von Bild 1.

diert. Dieser Anteil wird damit praktisch zweimal gezählt.

Erste Experimente haben gezeigt, dass die gegenseitige Beeinflussung der adaptiven Regelung aus unterschiedlichen Fitnessbereichen negative Auswirkungen haben kann. So kann es z.B. sinnvoll sein, anfänglich beide LSV zu verwenden und in einer späteren Phase der Evolution verstärkt auf nur eines der beiden zu setzen. Mit anderen Worten: *was gut für schlechte Individuen ist, muss nicht geeignet für gute sein und umgekehrt*. Um dieser Beobachtung gerecht zu werden, wurde eine getrennte Adaption für unterschiedliche Fitnessklassen eingeführt. Konkret wurden die Experimente mit den drei in Tabelle 2 wiedergegebenen Klassen durchgeführt. Wenn ein Individuum eine Fitness erreicht, deren Klasse noch unbenutzt ist, werden die Level-Einstellungen der nächst niedriger genutzten kopiert, um bereits gelernte Lektionen nicht noch einmal lernen zu müssen. Bei all-Verbesserung bestimmt die Fitness des Evolutionsbesten die Wahl des lokalen Verfahrens und von *all-impr<sub>R</sub>* bzw. *all-impr<sub>C</sub>*, während die Parametrierung des gewählten LSVs (*limit<sub>R</sub>*, *limit<sub>C</sub>* und *th<sub>R</sub>*) entsprechend der Fitness eines jeden Nachkommen eingestellt wird.

Fitnessklasse	Fitnessbereiche
fc1	0.4 0.7 1.0
fc2	0.35 0.65 0.85 1.0
fc3	0.3 0.55 0.75 0.9 1.0

**Tabelle 2:** Fitnessklassen und ihre Wertebereiche als Anteil an  $f_{max}$ .

Neben der Frage, ob mit all- oder best-Verbesserung gearbeitet werden soll, spielt noch die Anpassungsgeschwindigkeit der Adaption eine wichtige Rolle. Tabelle 3 zeigt die Werte für  $usage_{min}$  und  $matings_{max}$  der drei untersuchten Anpassungsgeschwindigkeiten *schnell*, *mittel* und *langsam* für die LSV-Auswahl und die Parameteranpassungen. Nur wenn es gelingt, für alle untersuchten Testanwendungen eine gemeinsame Parametrierung dieser neuen Strategieparameter der Adaption zu finden, kann von einem Fortschritt gesprochen werden und nur dann ist man dem eingangs genannten Ziel einer Verfahrensvereinfachung näher gekommen.

Geschwindigkeit	LSV-Auswahl		Parameteranp.	
	$usage_{min}$	$matings_{max}$	$usage_{min}$	$matings_{max}$
schnell	3	15	3	12
mittel	5	20	4	15
langsam	8	30	7	25

**Tabelle 3:** Die Parametrierungen der drei Anpassungsgeschwindigkeiten für die LSV-Auswahl und die Parameteranpassungen.

### 3.2 Andere Verfahren zur Adaption von Strategieparametern

Ong und Keane [20] benutzen einen ähnlichen Kosten-Nutzen basierten Ansatz mit dem Ziel „to promote cooperation and competition among the different LSs working together to accomplish the shared optimization goal”.<sup>1</sup> Sie ermitteln eine Belohnung  $\eta$  an Hand von Formel (3) und berechnen daraus die Wahrscheinlichkeitsverteilung zur Auswahl eines ihrer insgesamt 9 LSV, wobei  $f_{Best}$  die Fitness des bisher besten Individuums ist.

$$\eta = \beta \frac{f_{LS} - f_{evo}}{eval_{LS}} \quad \text{mit} \quad \beta = \frac{f_{LS}}{f_{Best}} \quad (3)$$

Der Faktor  $\beta$  relativiert den Fitnesszuwachs bezogen auf die Fitness des besten Individuums was ähnlich zur vorgestellten relativen Fitness ist: Der Zuwachs wird nicht

<sup>1</sup> Die Ideen von Ong und Keane entstanden offenbar parallel zur vorliegenden Arbeit, siehe auch [5].

absolut genommen sondern am bisher erreichten gemessen. Ong und Keane vergleichen diesen Ansatz mit einem heuristischen, der auf dem Fitnesszuwachs genotypisch ähnlicher Individuen aufbaut, und kommen zu dem Ergebnis, dass das auf der Belohnung  $\eta$  basierende Verfahren besser ist und dass es erheblich von der Anwendung abhängt, welche LSV besser oder schlechter geeignet sind. Im Unterschied zu HyGLEAM/A wird die Intensität der lokalen Suche durch eine Begrenzung auf 100 Iterationen für alle LSV beschränkt. Da Ong und Keane sich auf die LSV-Auswahl konzentrieren, während die vorliegende Arbeit stärker die Adaption der Rechenzeitverteilung zwischen globaler und lokaler Suche zum Ziel hat, ergänzen sich beide Arbeiten.

Bisher war von einem Mechanismus zur Adaption die Rede, der außerhalb des bereits selbst Adaption bewirkenden evolutionären Mechanismus steht. Warum also nicht die Strategieparameter zum Chromosom hinzufügen und die Adaption der Evolution überlassen, wie es Krasnogor et al. [21] getan haben? Schließlich hat die Selbstadaption beispielsweise bei der Schrittweitensteuerung der ES gute Ergebnisse gebracht. Der wesentliche Unterschied ist hierbei, dass unterschiedliche Schrittweiten den Aufwand für die Mutation nicht vergrößern und es tatsächlich nur auf den Fitnessgewinn ankommt. Aber die Auswahl zwischen unterschiedlichen LSV bewirkt unterschiedliche Kosten, sofern sie nicht einheitlich vorzeitig gestoppt werden. Krasnogor et al. sagen selbst: „The rationale is to propagate local searchers that are associated with fit individuals, as those individuals were probably improved by their respective memes.“ Oder aber sie wurden durch die vorangegangene Evolution verbessert! Dass man dies der Fitnesssumme nicht mehr ansehen kann, rechtfertigt die Untersuchung des vorgeschlagenen Ansatzes zur Kosten-Nutzen basierten Adaption.

Mit Zitzler et al. [22] und Bambha et al. [23] wird der Gedanke des adaptiv gesteuerten Zuwachses der Genauigkeit der lokalen Suche im Verlauf der Adaption geteilt. Der wesentliche Unterschied besteht in der vorher festzulegenden Dauer der Optimierung, die ihrem Verfahren zu Grunde liegt. Insbesondere bei neuen noch nicht bearbeiteten Aufgabenstellungen ist die Abschätzung einer ausreichenden Laufzeit dagegen problematisch. So sehen die Autoren auch den wesentlichen Einsatzbereich ihres Ansatzes bei Aufgaben, die einer festen zeitlichen Restriktion unterliegen.

## 4 Experimente und ihre Ergebnisse

### 4.1 Benchmarkaufgaben

Die zur experimentellen Überprüfung des HyGLEAM/A-Ansatzes notwendigen Testaufgaben müssen repräsentativ hinsichtlich realer Anwendungen sein, aber auch eine schnelle Optimierung erlauben, damit statistische Untersuchungen in vertretbarer Zeit möglich sind. Tabelle 4 gibt einen Überblick über wichtige Eigenschaften der sieben benutzten Testaufgaben, von denen zwei Anwendungsprobleme sind, während die fünf anderen der GENEsYs-Sammlung [24] entnommen wurden und in der Tabelle mit ihren dortigen Funktionsnummern angegeben sind. Shekel's Foxholes und die verallgemeinerte Rastrigin Funktion wurden um  $30^\circ$  im Raum gedreht, um sie schwieriger zu machen, siehe auch [5]. Ein Problem wird in der Tabelle als *stark multimodal* (*stark mm.*) bezeichnet, wenn es mehr als das 20-fache seiner Dimension an Suboptima aufweist. Tabelle 4 gibt für die Testfunktionen auch die Wertebereiche und die Genauigkeit der Zielwerte an, wobei lediglich im Falle von Shekel's Foxholes das exakte Optimum gefordert wird. Bei der fraktalen Funktion ist es unbekannt.

Testaufgabe	Parameter	Komb. Optim.	Modalität	Implizite Restrikt.	Wertebereich	Zielwert
Schwefels Kugel (f1)	30 real	nein	unimodal	nein	$[-10^{10}, 10^{10}]$	0.01
Shekel's Foxholes (f5)	2 real	nein	multimod.	nein	$[-500, 500]$	0.998004
Verallg. Rastrigin F. (f7)	5 real	nein	stark mm.	nein	$[-5.12, 5.12]$	0.0001
Fletcher & Powell (f16)	5 real	nein	multimod.	nein	$[-3.14, 3.14]$	0.00001
Fraktale F. (f13)	20 real	nein	stark mm.	nein	$[-5, 5]$	-0.05
Designoptimierung	3 real	nein	stark mm.	nein		
Ressourcenoptimierung	87 int.	(ja)	multimod.	ja		

**Tabelle 4:** Wichtige Eigenschaften der sieben Testaufgaben.

Aus Platzgründen können die beiden Anwendungsaufgaben hier nur kurz vorgestellt werden und der interessierte Leser wird auf die angegebene Literatur verwiesen. Alle sind ausführlich in [5] beschrieben.

Bei der Designoptimierung geht es um den Entwurf eines mikrooptischen Bauteils (Heterodynempfänger) bestehend aus zwei Kugellinsen und einer Photodiode [25]. Der Abstand zwischen dem Lichtleiter und erster Kugellinse sowie die beiden Brechungsindizes sollen so bestimmt werden, dass nicht nur eine optimale Ausleuchtung der Photodiode im Kollimationspunkt erreicht wird, sondern die ganze Anordnung auch möglichst unempfindlich gegenüber Fertigungstoleranzen innerhalb vorgegebener Grenzen ist. Letzteres macht die ansonsten einfache Aufgabe zu einem trotz der nur drei Parameter schwieriger, stark multimodalen Optimierungsproblem.

Die Ressourcenoptimierung erfolgt in Zusammenhang mit einer Scheduling-Aufgabe aus der Verfahrenstechnik [26]. Dabei sind 87 Chargen mit jeweils unterschiedlichem Mitarbeiterbedarf während der Abarbeitung jeder Charge so zu starten, dass der resultierende Mitarbeiterbedarf (Ressource menschliche Arbeitskraft) pro Schicht ein vorgegebenes Maximum nicht überschreitet und möglichst noch darunter bleibt. Gleichzeitig sind natürlich die üblichen Randbedingungen wie Lieferfristen, die Verfügbarkeit gemeinsam genutzter Anlagen und die Verfügbarkeit von durch Vorgängerchargen produzierten Vorprodukten zu beachten. Da Belegungskonflikte, die durch die Reihenfolge der Chargen in den Aktionsketten gelöst werden, auch durch geeignete Verschiebung der Starttermine beseitigt werden können, ist der kombinatorische Charakter der Aufgabenstellung begrenzt (eingeklammertes „ja“ in Tabelle 4). Das Optimum ist nicht bekannt und für die Erfüllung der Testaufgabe wird ein Wert nahe dem besten Ergebnis von [26] gefordert: Eine Reduktion des maximalen Mitarbeiterbedarfs von 12 auf 9 pro Schicht bei gleichzeitiger Verkürzung der ursprünglich manuell geplanten Produktionszeit auf 70%.

## 4.2 Ergebnisse für HyGLEAM

GLEAM oder ein Hybrid wird zusammen mit konkreten Werten seiner Strategieparameter als *Job* bezeichnet. Die Bewertung basiert auf dem durchschnittlichen Aufwand gemessen in Evaluationen von in der Regel 100 Läufen pro Job, wobei nur solche Jobs verglichen werden, die erfolgreich sind, d.h. bei denen alle Läufe das vorgegebene Ziel erreicht haben. Der Vergleich basiert dann auf der erreichten Verbesserung gegenüber GLEAM bzw. im nächsten Abschnitt auch gegenüber HyGLEAM. Tabelle 5 vergleicht die besten Jobs von GLEAM und HyGLEAM zusammen mit ihren Strategieparametern. Die Tabelle zeigt die durchgängige Überlegenheit der Lamarckschen Evolution und die enorme Spreizung geeigneter Populationsgrößen beim reinen EA (GLEAM) sowie

deren deutliche Verringerung bei allen Hybriden. Ansonsten bestätigt die Tabelle die Notwendigkeit einer adaptiven Auswahl des LSVs, seiner Genauigkeit und der Entscheidung zwischen best- oder all-Verbesserung.

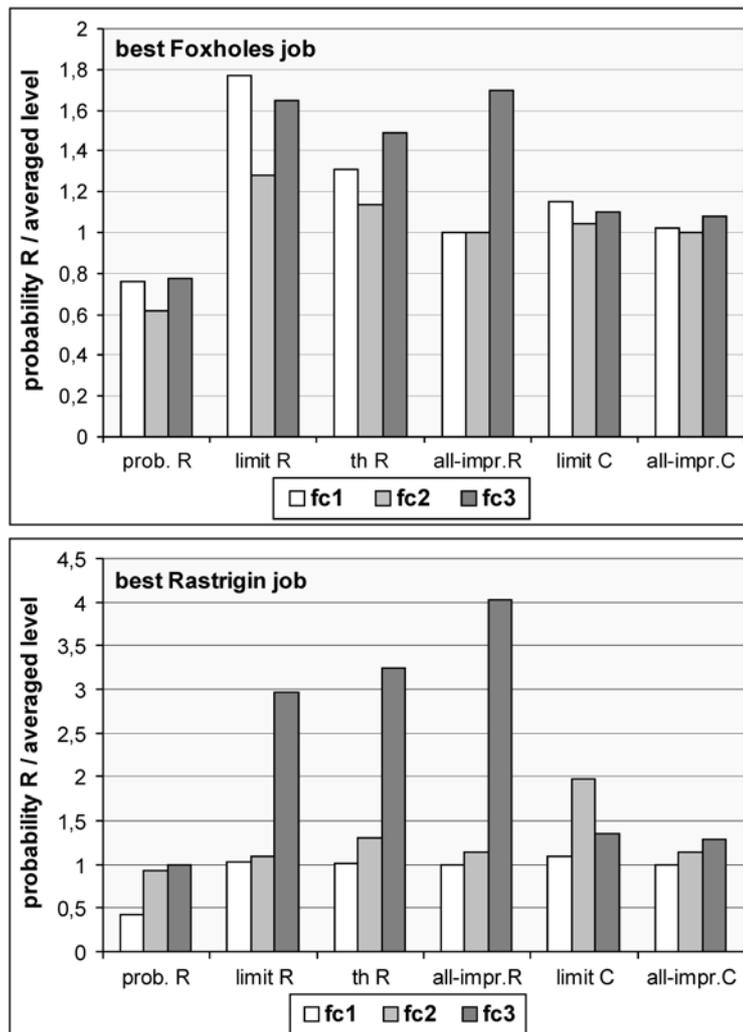
Testaufgabe	GLEAM		HyGLEAM						
	Eval.	Pg	LSV	Eval.	Pg	th <sub>R</sub>	b/a	Lamarck	Verb.
Fletcher	483 566	600	C	4 684	5		best	ja	103
Ressourcenopt.	5 376 334	1 800	R	69 448	5	10 <sup>-1</sup>	best	ja	77,4
Foxholes	103 192	350	C	8 831	20		best	ja	11,7
Rastrigin	3 518 702	11 200	R	315 715	70	10 <sup>-2</sup>	all	ja	11,2
Fraktal	195 129	20	R	30 626	5	10 <sup>-2</sup>	best	ja	6,4
Designopt.	5 773	210	C	1 041	5		best	ja	5,6
Kugel	kein Erfolg		R	29 595	5	10 <sup>-6</sup>	best	ja	

**Tabelle 5:** Ergebnisse der sieben Testaufgaben in absteigender Reihenfolge der erreichten Verbesserung. Abkürzungen: Eval: Evaluationen, Pg: Populationsgröße, R: Rosenbrock-Verfahren, C: Complex-Algorithmus. b/a: best- oder all-Verbesserung, Lamarck: Lamarcksche Evolution, Verb: Verbesserungsfaktor gegenüber dem besten GLEAM-Job

### 4.3 Ergebnisse für adaptives HyGLEAM (HyGLEAM/A)

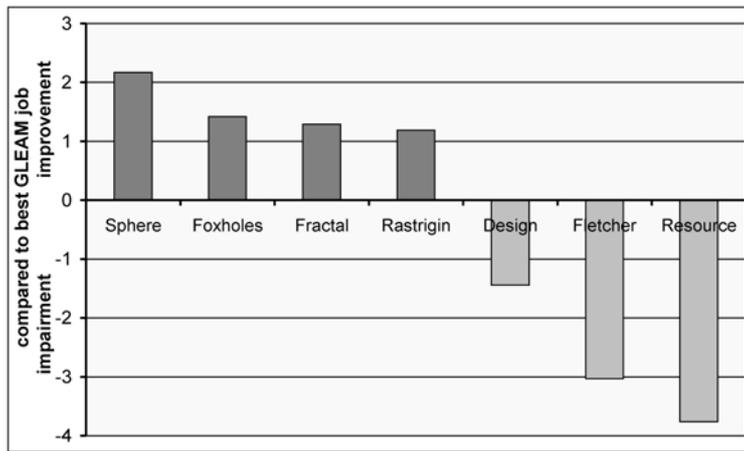
Bild 4 zeigt exemplarisch, dass sich je nach Anwendung im Durchschnitt unterschiedliche Werte für die Rosenbrockwahrscheinlichkeit (*prob. R*) und die Levels am Ende eines Laufes für die verschiedenen Fitnessklassen ergeben. Es unterstreicht zusammen mit weiteren Beobachtungen die in Kapitel 3 gegebene Begründung für die Einführung der nach Fitnessklassen getrennten Adaption. Aus technischen Gründen sind die Indices der Größen zur LSV-Unterscheidung in den Bildern nicht tief gestellt.

Beim Vergleich zwischen dem per Hand optimal eingestellten HyGLEAM und der Standardeinstellung von HyGLEAM/A (d.h. mit beiden LSV und all-Verbesserung) kann man beides erwarten, eine Verbesserung auf Grund der Adaption oder eine Verschlechterung



**Bild 4:** Exemplarische Entwicklung der Rosenbrockwahrscheinlichkeit (*prob. R*) und der Levels.

insbesondere dann, wenn ein nicht- oder schlecht geeignetes LSV abgeschaltet werden muss. So funktioniert der Complex-Algorithmus weder mit der Kugel-Funktion noch mit der Ressourcenoptimierung und mit der Rastrigin und der fraktalen Funktion schlechter als reines GLEAM. Letzteres trifft auch auf das Rosenbrock-Verfahren und die Designoptimierung zu. In der Tat lassen sich beide Effekte beobachten, wie

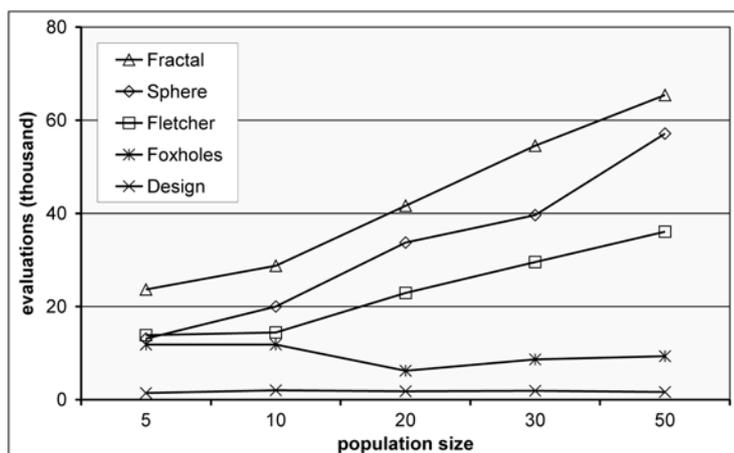


**Bild 5:** Verbesserungen und Verschlechterungen der besten Standard-HyGLEAM/A-Jobs verglichen mit den besten Jobs des per Hand optimierten HyGLEAMs.

Bild 5 zeigt, jedoch nicht bei allen Testfällen entsprechend ihrer Eignung für das eine oder andere LSV. So konnte z.B. im Falle der Kugel-Funktion trotz des ungeeigneten Complex-Verfahrens eine Verringerung der Evaluationen erreicht werden, während dies bei der Ressourcenoptimierung nicht möglich war. Offenbar ist es anwendungsabhängig, wie schnell ein ungeeignetes LSV verdrängt werden kann und ob es sozusagen nur „stört“ oder etwas zum Erfolg beiträgt, wenn auch mit höherem Aufwand.

Bei vier der sieben Testaufgaben (Foxhole, fraktale und Rastrigin Funktion sowie die Designoptimierung) bewirkt die Adaption keine relevante Leistungsänderung gegenüber HyGLEAM mit manueller Optimierung der Strategieparameter. Der beschleunigten Optimierung bei der Kugel-Funktion steht die zum Teil deutliche Verlangsamung bei Fletchers Funktion und bei der Ressourcenoptimierung gegenüber. Die beiden letzten fielen in Tabelle 5 durch herausragende Performancesteigerungen gegenüber GLEAM auf (Faktor 104 bzw. 77) und sie erreichen bei Standard-HyGLEAM/A immerhin noch eine Leistungssteigerung gegenüber GLEAM von einem Faktor von 35 bzw. 21 (siehe auch Tabelle 6), die ohne den bei HyGLEAM sonst notwendigen erheblichen manuellen Anpassungsaufwand für die Strategieparameter erreicht wurde.

Tabelle 6 fasst die Ergebnisse für die besten Jobs von GLEAM, manuell optimiertem HyGLEAM, Standard-HyGLEAM/A und dem besten HyGLEAM/A zusammen und vergleicht relevante Strategieparameter. Wenn Standard-HyGLEAM/A bereits am besten abschnitt, wurde in der Tabelle die Zeile für das beste HyGLEAM/A weggelassen. Aus der Tabelle lassen sich zwei Ergebnisse ablesen. Erstens fällt die deutliche und durchgängige Verringerung der Populationsgröße gegenüber GLEAM auf. Bild 6 zeigt ihren meist moderaten Einfluss auf den Aufwand für die fünf Test-



**Bild 6:** Einfluss der Populationsgröße auf den Aufwand für alle Testaufgaben, die auch bei kleinen Populationsgrößen eine Erfolgsrate von 100% aufweisen.

aufgaben, bei denen auch kleine Populationsgrößen eine Erfolgsrate von 100% liefern. Die Rastrigin Funktion konvergiert nicht mehr sicher ab einer Populationsgröße von 50 und weniger und die Ressourcenoptimierung ab 10 und weniger. Aber es kann als ein wesentliches Ergebnis festgehalten werden, dass der Bereich sinnvoller Populationsgrößen deutlich eingengt werden konnte: Statt einem Wertebereich von 20 bis 11200 wie bei GLEAM kommt bei HyGLEAM/A nur noch ein Bereich zwischen 5 und 70 in Frage. Auf Grund des moderaten Einflusses steigender Populationsgrößen auf den Aufwand sollte man sicherheitshalber bei einem unbekanntem Problem mit einer Populationsgröße von 20 beginnen und prüfen, ob eine Verringerung möglich ist. Wenn von einer erheblichen Komplexität auszugehen ist, empfehlen sich größere Populationen.

Testaufgabe	Algorithmus	Evaluationen	Populationsgröße	Adaptionsgeschw.	Fitnessklasse	b/a	LSV
Funktion nach Fletcher & Powell	GLEAM	483 566	600				
	HyG mit C	4 684	5				
	Stand.-HyG/A	13 841	5	schnell	fc2	all	beide
	Best HyG/A	11 808	10	schnell	fc2	best	beide
Ressourcenoptimierung	GLEAM	5 376 334	1 800				
	HyG mit R	69 448	5				
	Stand.-HyG/A	251 917	20	mittel	fc1	all	beide
	Best HyG/A	235 419	30	langsam	gc2	best	beide
Shekel's Foxholes	GLEAM	103 192	350				
	HyG mit C	8 831	20				
	Stand.-HyG/A	6 209	20	langsam	fc1	all	beide
verallg. Rastrigin Funktion	GLEAM	3 518 702	11 200				
	HyG mit R	315 715	70				
	Stand.-HyG/A	265 892	90	langsam	fc1	all	beide
Fraktale Funktion	GLEAM	195 129	20				
	HyG mit R	30 626	5				
	Stand.-HyG/A	23 649	5	schnell	fc2	all	beide
	Best HyG/A	20 753	5	mittel	fc1	all	beide
Designoptimierung	GLEAM	5 773	210				
	HyG mit C	1 041	5				
	Stand.-HyG/A	1 433	5	langsam	fc2	all	beide
	Best HyG/A	652	5	mittel	fc1	best	C
Schwefels Kugel	HyG mit R	29 595	5				
	Stand.-HyG/A	13 090	5	schnell	fc1	all	beide
	Best HyG/A	8 436	5	schnell	fc3	best	R

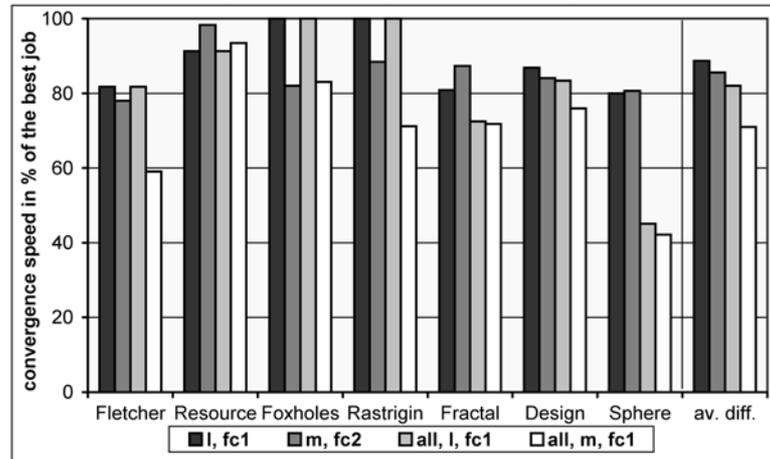
**Tabelle 6:** Ergebnisse der sieben Testfunktionen in absteigender Reihenfolge der Aufwandseinsparungen. Abkürzungen: HyG: HyGLEAM, Stand.-HyG/A: Standard-HyGLEAM/A, b/a: best- oder adaptive all-Verbesserung, weitere Abkürzungen siehe Tabelle 5.

Zweitens zeigt Tabelle 6 die Anwendungsabhängigkeit einer geeigneten Wahl der am Ende von Abschnitt 3.1 vorgestellten Strategieparameter der Adaption *Adaptionsgeschwindigkeit*, *Fitnessklasse* und *best/all-Verbesserung*. Obwohl auch die die *Menge der anwendbaren LSV (beide, Rosenbrock- oder Complex-Verfahren)* eine Rolle spielen kann, wie das Kugel-Problem und die Designoptimierung zeigen, soll aus Gründen der allgemeinen Anwendbarkeit die Wahl des LSVs der Adaption vorbehalten bleiben. Aus einer Analyse des gesamten Zahlenmaterials der Experimente kann folgendes abgeleitet werden:

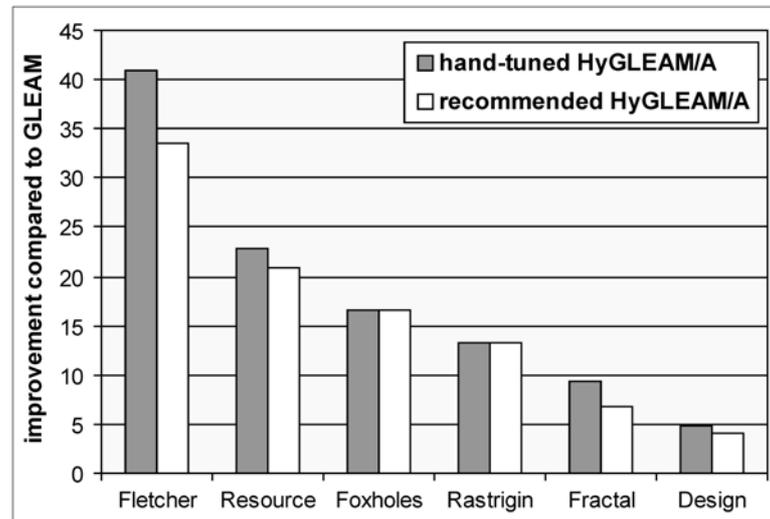
1. Die Wahl der Fitnessklasse hat einen eher geringen Effekt.

2. Langsame und mittlere Adaptionsgeschwindigkeit funktioniert immer, während die schnelle Fehladaptation vor allem hinsichtlich der LSV-Wahl mit sich bringen kann.
3. Die lokale Verbesserung nur des besten Nachkommen (best-Verbesserung) liefert zwar meist etwas bessere Resultate, ist aber im Falle der Rastrigin-Funktion fatal, da in diesem Falle nur all-Verbesserung zu einer Erfolgsrate von 100% führt.

Angesichts dessen und den moderaten Abweichungen bei den resultierenden Evaluationswerten in Tabelle 6 stellt sich die Frage, ob ein geeigneter „guter Mittelwert“ gefunden werden kann. Bild 7 vergleicht die besten zwei Parametrierungen jeweils für variable best/all-Verbesserung und für reine all-Verbesserung an Hand der erreichten Konvergenzgeschwindigkeit verglichen mit dem besten per Hand eingestellten HyGLEAM/A-Job. Der durch die vier Säulen rechts dargestellte durchschnittliche Leistungsverlust zeigt, dass die manuelle Wahl zwischen best- und all-Verbesserung die geringsten Performanceverluste bringt. Es wird aber auch deutlich, dass der Abstand zur besten Standard-HyGLEAM/A-Parametrierung gering ist und mit 82% der Leistung der besten manuell eingestellten HyGLEAM/A-Jobs nur geringe Leistungsverluste in Kauf genommen werden müssen.



**Bild 7:** Vergleich der besten zwei Parametrierungen jeweils für manuell eingestellte best-/all-Verbesserung und für reine all-Verbesserung auf der Basis der erreichten Konvergenzgeschwindigkeit im Vergleich zum besten Job. Rechts der Durchschnitt für alle Testfälle (*av. diff.*). Abkürzungen: l, m: langsame, mittlere Adaptionsgeschwindigkeit



**Bild 8:** Vergleich des Leistungsgewinns gegenüber reinem GLEAM zwischen dem manuell eingestellten HyGLEAM/A und der empfohlenen Parametrierung von Standard-HyGLEAM/A.

Damit kann Standard-HyGLEAM/A mit langsamer Adaption, Fitnessklasse *fc1* bestehend aus drei Klassen und adaptiver all-Verbesserung sowie Verwendung beider LSV als die gesuchte günstige Parametrierung angesehen werden.

Bild 8 vergleicht die gegenüber GLEAM erreichten Performancegewinne der empfohlenen Parametrierung von Standard-HyGLEAM/A mit der besten manuellen Einstellung

von HyGLEAM/A für sechs der sieben Testaufgaben.<sup>2</sup> Das Bild verdeutlicht noch einmal den moderaten Leistungsverlust, der als Preis dafür zu zahlen ist, dass bis auf die Populationsgröße keine manuellen Einstellungen am resultierenden memetischen Algorithmus für Parameteroptimierungen mehr notwendig sind.

## 5 Zusammenfassung und Ausblick

Es konnte an Hand der sieben Testfälle gezeigt werden, dass das Ziel eines hybriden evolutionären Algorithmus mit möglichst wenig Verfahrensparametern durch den vorgeschlagenen Mechanismus zur Kosten-Nutzen basierten Adaption für memetische Algorithmen zur Parameteroptimierung nahezu erreicht werden konnte. Lediglich die Populationsgröße muss noch aufgabenspezifisch eingestellt werden, wobei der Bereich sinnvoller Größen deutlich eingeschränkt werden konnte, und zwar auf Werte zwischen 5 und 70 anstelle von 20 und 11200. Als beste Parametrierung der Adaption konnte folgendes ermittelt werden:

1. Verwendung beider lokaler Suchverfahren
2. Lokale Verbesserung eines adaptiv bestimmten Anteils aller Nachkommen einer Paarung (adaptive all-Verbesserung, siehe auch Abschnitt 3.1 und Tabelle 1)
3. Getrennte Adaption in drei Fitnessklassen (siehe auch Tabelle 2)
4. Langsame Adaptionsgeschwindigkeit (siehe auch Tabelle 3)

Die so eingestellte Adaption kontrolliert folgende Strategieparameter des Memetischen Algorithmus pro lokalem Suchverfahren: die Auswahl des lokalen Verfahrens, die Iterationsgrenze, eine konvergenzabhängige Abbruchschranke (nur beim Rosenbrock-Verfahren) und die Wahrscheinlichkeit, weitere Nachkommen lokal zu verbessern als nur das Beste einer Paarung (adaptive all-Verbesserung), siehe auch Tabelle 1. Mit der vorgeschlagenen Parametrierung konnte im Durchschnitt der Testfälle 82% der Leistungsfähigkeit (gemessen in Anzahl der Fitnessberechnungen) des besten manuell eingestellten adaptiven Hybriden erreicht werden.

Da der vorgestellte Mechanismus zu Adaption so konzipiert ist, dass er auch auf andere EA als dem hier verwendeten, anwendbar ist, ist man dem Ziel eines allgemein anwendbaren, robusten EA-Hybriden für Parameteroptimierung, der wenig Vorkenntnisse erfordert, ein großes Stück näher gekommen. Natürlich ist damit kein universelles Allzweckwerkzeug geschaffen, wie die eingangs gemachten Anwendungseinschränkungen zeigen und wie es auch im „no free lunch“-Theorem [27] zum Ausdruck kommt.

Da die Ergebnisse auf fünf Testfunktionen und zwei realen Anwendungen basieren, wäre eine Ausweitung der Testfälle zur besseren Absicherung der Resultate sicher sinnvoll. Wie die Arbeiten von Ong und Keane [20] gezeigt haben, ist auch von der Hinzunahme weiterer lokaler Suchverfahren eine Verbesserung zu erwarten.

## 6 Literatur

- [1] Davis, L. (ed): *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991
- [2] Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolutionary Programs*. 2nd ed., Springer, New York, USA, 1994

---

<sup>2</sup> Schwefels Kugel ist hier nicht enthalten, da GLEAM alleine das Problem nicht lösen konnte und damit kein Vergleichmaßstab verfügbar ist.

- [3] Hart, W. E., Krasnogor, N., Smith, J. E. (eds), *Recent Advances in Memetic Algorithms*. Springer, Berlin, 2005
- [4] Jakob, W.: *HyGLEAM – An Approach to Generally Applicable Hybridization of Evolutionary Algorithms*. In Conf. Proc. Parallel Problem Solving from Nature (PPSN VII), LNCS 2439, Springer, Berlin, 2002, S.527-536
- [5] Jakob, W.: *Eine neue Methodik zur Erhöhung der Leistungsfähigkeit Evolutionärer Algorithmen durch die Integration lokaler Suchverfahren*. Dissertation, Fak. f. Maschinenbau, Universität Karlsruhe, FZKA 6965, Forschungszentrum Karlsruhe, März 2004. Siehe auch: <http://www.iai.fzk.de/~jakob/HyGLEAM/>
- [6] Rechenberg, I.: *Evolutionsstrategie '94*. Frommann-Holzboog, Stuttgart-Bad Cannstatt, 1994
- [7] Gruau, F., Whitley, D.: *Adding Learning to the Cellular Development of Neural Networks: Evolution and the Baldwin Effect*. Evolutionary Computation, Vol.1, 1993, S.213-233
- [8] Whitley, D., Gordon, V., Mathias, K.: *Lamarckian Evolution, The Baldwin Effect and Function Optimization*. In: Davidor, Y. et al. (eds): Conf. Proc. PPSN III, LNCS 866, Springer Verlag, Berlin, 1994, S.6-14
- [9] Gorges-Schleuter, M.: *Genetic Algorithms and Population Structures - A Massively Parallel Algorithm*. Dissertation, Dept. Comp. Science, University of Dortmund, 1990
- [10] Gorges-Schleuter, M.: *A Comparative Study of Global and Local Selection in Evolution Strategies*. In Schwefel, H.-P., Männer, R. (eds.): Conf. Proc. PPSN V, LNCS 1498, Springer, Berlin, 1998, S.367-377
- [11] Goldberg, D. E., Voessner, S.: *Optimizing Global-Local Search Hybrids,* in Conf. Proc. GECCO 99, Morgan Kaufmann, San Mateo, CA, USA, 1999, S.220-228
- [12] Shina, A., Chen, Y., Goldberg, D. E.: *Designing Efficient Genetic and Evolutionary Algorithm Hybrids*. In [3], 2005, S.259-288
- [13] 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
- [14] H.H. Rosenbrock: *An Automatic Method for Finding the Greatest or Least Value of a Function*. In: The Computer Journal, 3, 1960, S.175-184
- [15] M.J. Box: *A New Method of Constrained Optimization and a Comparison with Other Methods*. In: The Computer Journal, 8, 1965, S.42-52
- [16] H.-P. Schwefel: *Evolution and Optimum Seeking*. John Wiley & Sons, New York, USA, 1995
- [17] H.J. Holland: *Adaptation in Natural and Artificial Systems*. Dissertation, The University of Michigan Press, Ann Arbor, USA, 1975
- [18] Hart, W. E.: *Adaptive Global Optimization with Local Search*. Dissertation, Univ. California, San Diego, CA, USA, 1994
- [19] Krasnogor, N.: *Studies on the Theory and Design Space of Memetic Algorithms*. Dissertation, Faculty Comput., Math. Eng., Univ. West of England, Bristol, U.K., 2002
- [20] Ong, Y. S., Keane, A. J.: *Meta-Lamarckian Learning in Memetic Algorithms*. IEEE Trans. on Evolutionary Computation, Vol. 8, Nr. 2, April 2004, S.99-110, Zitat S.100
- [21] Krasnogor, N. , Blackburne, B. P., Burke, E. K., Hirst, J. D.: *Multimeme Algorithms for Protein Structure Prediction*. In Conf. Proc. PPSN VII, LNCS 2439, Springer, Berlin, 2002, S.769-778, Zitat S.772
- [22] Zitzler, E., Teich, J., Bhattacharyya, S. S.: *Optimizing the Efficiency of Parameterized Local Search within Global Search: A Preliminary Study*. In Proc. Conference on Evolutionary Computation (CEC 2000), Piscataway, NJ, USA, IEEE press, 2000, S.365-372
- [23] Bambha, N. K., Bhattacharyya, S. S., Teich, J., Zitzler, E.: *Systematic Integration of Parameterized Local Search Into Evolutionary Algorithms*. IEEE Trans. on Evolutionary Computation, Vol. 8, Nr. 2, April 2004, S.137-155
- [24] Bäck, T.: *GENEsYs 1.0*, Uni. Dortmund, 1992, <ftp://lumpi.informatik.uni-dortmund.de/pub/GA/>
- [25] Sieber, I., Eggert, H., Guth, H., Jakob, W., Scherer, K.-P., Ziegler, P.: *Design Optimization Considering Tolerance Effects of Microoptical Benches*. MicroSystem Technologies 98, VDE-Verlag, 1998, S.65-70
- [26] Blume, C., Gerbe, M.: *Deutliche Senkung der Produktionskosten durch Optimierung des Ressourceneinsatzes*. In: atp 36, Nr. 5, 1994, S.25-29
- [27] Wolpert, D. H., Macready, W. G.: *No free lunch theorems for optimization*. IEEE Trans. Evolutionary Computation, Vol. 1, April 1997, S. 67-82

# Partikel Schwarm Optimierung eines Regelungssystems für ein industrielles Hydraulikventil

**Renato Krohling, Martin Sternke,  
Daniel Schauten und Frank Hoffmann**

Lehrstuhl für Elektrische Steuerung und Regelung  
Fakultät für Elektrotechnik und Informationstechnik  
Universität Dortmund, 44221 Dortmund

Tel.: (0231) 755–2760

Fax: (0231) 755–2752

E-Mail: {Krohling, Sternke, Schauten, Hoffmann}@esr.e-technik.uni-dortmund.de

## Zusammenfassung

In diesem Beitrag werden die Ergebnisse einer automatisierten Hardware-in-the-Loop-Optimierung eines realen industriellen Anwendungsproblems mit Hilfe der Partikel Schwarm Optimierung beschrieben und mit den bisher mit einer Evolutionsstrategie erzielten Ergebnissen verglichen. Das betrachtete Anwendungsproblem besteht in der Optimierung der Reglerparameter für ein industrielles Hydraulikventil. Die Beurteilung und Diskussion der erzielten Ergebnisse erfolgt einerseits bezogen auf die industrielle Anwendung an sich und andererseits bezogen auf unterschiedliche Verhaltensweisen der eingesetzten Optimierungsverfahren.

## 1 Einleitung

In der industriellen Praxis erlangt die Hardware-in-the-Loop-Optimierung von Regelungssystemen immer mehr an Bedeutung. Gerade bei hochparametrischen Optimierungsproblemen werden dabei immer häufiger Evolutionäre Algorithmen zur Lösungsfindung eingesetzt. Der praktische Hintergrund dieses Beitrags liegt in der Optimierung von 24 Parametern eines Reglers für ein industrielles Hydraulikventil. Neben der hohen Anzahl zu optimierender Parameter liegt die besondere Schwierigkeit dieser Anwendung darin, dass zahlreiche, teilweise gegenläufige Gütekriterien simultan bei der Optimierung zu berücksichtigen sind.

Zum Ausgangspunkt einer mittlerweile langjährigen Industriekooperation wurde diese Regleroptimierung in der industriellen Praxis von Prozessexperten „von Hand“ vorgenommen. Da die vielzähligen einzelnen Reglerparameter nicht unabhängig voneinander eingestellt werden können, ist dieser Prozess zeitaufwändig und erfordert viel Erfahrung. Zur Einsparung von Zeit und Kosten einerseits, sowie zur Erhöhung der Qualität und Robustheit des Regelungssystems andererseits, war eine Automatisierung der Optimierung erwünscht. Hierzu wurde das Bewertungsverhalten der Experten durch ein Fuzzy-Gütemaß nachgebildet und die Optimierung der realen Reglerparameter in einem Hardware-in-the-loop-Versuchsaufbau mit Hilfe einer Evolutionsstrategie vorgenommen (Abschnitt 3).

Dabei zeigte sich an diesem Optimierungsproblem, dass eine herkömmliche Evolutionsstrategie zwar erfolgreich einsetzbar ist, aber dennoch ein nennenswertes Verbesserungspotenzial vorhanden ist. Hierzu wurden in [1] zahlreiche neue Strategieelemente entwickelt und evaluiert, durch die die Gefahr einer vorzeitigen Konvergenz der Optimierung in einem lokalen Minimum reduziert wird und die Diversität der Individuen über einen längeren Zeitraum der Optimierung aufrechterhalten bleibt. Die implementierten Strategien zielen insbesondere auf eine automatisierte dynamische Umschaltung von Breiten- und Tiefensuchenverhalten innerhalb der Evolutionsstrategie ab. Die damit erzielten Ergebnisse dienen als Referenz für die Einordnung der erzielten Ergebnisse mit einer Partikel Schwarm Optimierung (PSO).

PSO basiert auf dem dynamischen Verhalten mehrerer Partikel in einem Schwarm [2]. Die Partikel repräsentieren analog zu den Individuen bei der Evolutionsstrategie die zu optimierenden Reglerparameter. Für die Suche nach guten Lösungen bewegen sich die Partikel im Parameterraum mit unterschiedlichen Geschwindigkeiten. Die Geschwindigkeiten der Partikel werden in einem Lernprozess fortlaufend anhand ihrer eigenen Historie und der Historie anderer Partikel in der Population angepasst, wobei grundlegende Prinzipien des Verhaltens in einem sozialen System, wie Bewertung, Vergleich und Imitation, berücksichtigt werden (Abschnitt 2).

Der Einsatz der PSO soll einerseits die Gleichwertigkeit bezüglich der erzielbaren Güte am realen Optimierungsproblem aufzeigen. Andererseits sollen Unterschiede gegenüber dem Verhalten der Evolutionsstrategie bezüglich der Robustheit, des Konvergenzverhaltens und der Populationsdiversität aufgezeigt werden (Abschnitt 4).

## 2 Partikel Schwarm Optimierung (PSO)

In der Natur lässt sich an vielen Beispielen beobachten, dass Gruppen von Individuen in Interaktion miteinander Probleme lösen, die jedes einzelne von ihnen nicht lösen könnte und in einigen Fällen nicht einmal als Problem wahrnimmt. Als Beispiel hierfür können Vogelschwärme aufgeführt werden, die in der Gruppe Nahrungsquellen schneller finden. Auch der Mensch lässt sich als Schwarm betrachten, insbesondere, wenn man die Änderung seines Sozialverhaltens als eine Bewegung im multidimensionalen Raum betrachtet. Die Fähigkeit, durch Interaktion in der Gruppe zum Teil unbewusst Probleme zu lösen, bezeichnet man als *Schwarm Intelligenz*.

Der Algorithmus zur Partikel Schwarm Optimierung (PSO) wurde von KENNEDY und EBERHART im Jahr 1995 erstmalig veröffentlicht [2, 3]. Gegenstand der PSO ist die Repräsentation potenzieller Lösungen eines Optimierungsproblems durch einen *Schwarm*, einer Population von so genannten *Partikeln*  $\mathcal{P}_i$ . Im Gegensatz zu evolutionären Algorithmen, wird jedem Partikel  $\mathcal{P}_i$  neben seiner Position  $\mathbf{x}_i$  im Optimierungssuchraum eine Geschwindigkeit  $\mathbf{v}_i$  zugeordnet, nach dessen Betrag und Richtung sich das Partikel im Suchraum bewegt. Zusätzlich erhält jedes Partikel  $\mathcal{P}_i$  ein Gedächtnis, in dem die Koordinaten  $\mathbf{x}_{best_i}$  und die Güte  $P_{best_i}$  der bisher besten vom Partikel gefundenen Lösung gespeichert wird. Zusätzlich wird für die Implementierung eines *sozialen Verhaltens* die global über alle Partikelbewegungen ermittelte beste bekannte Lösung mit den Koordinaten  $\mathbf{x}_{G_{best}}$  und der Güte  $G_{best}$  gespeichert.

Die PSO ist ähnlich dem evolutionären Konzept ein iterativer Optimierungsprozess. Dabei wird zu jedem Iterationsschritt die Geschwindigkeit eines jeden Partikels  $\mathcal{P}_i$

individuell beschleunigt. Die Berechnung der Beschleunigung erfolgt unter einer stochastisch gewichteten Einbeziehung der Daten seines eigenen bekannten jemals erreichten lokalen Minimums  $P_{best_i}$  sowie der Daten des aktuellen globalen Minimums  $G_{best}$ . Abbildung 1 zeigt beispielhaft die Dynamik eines Schwarms aus vier Partikeln.

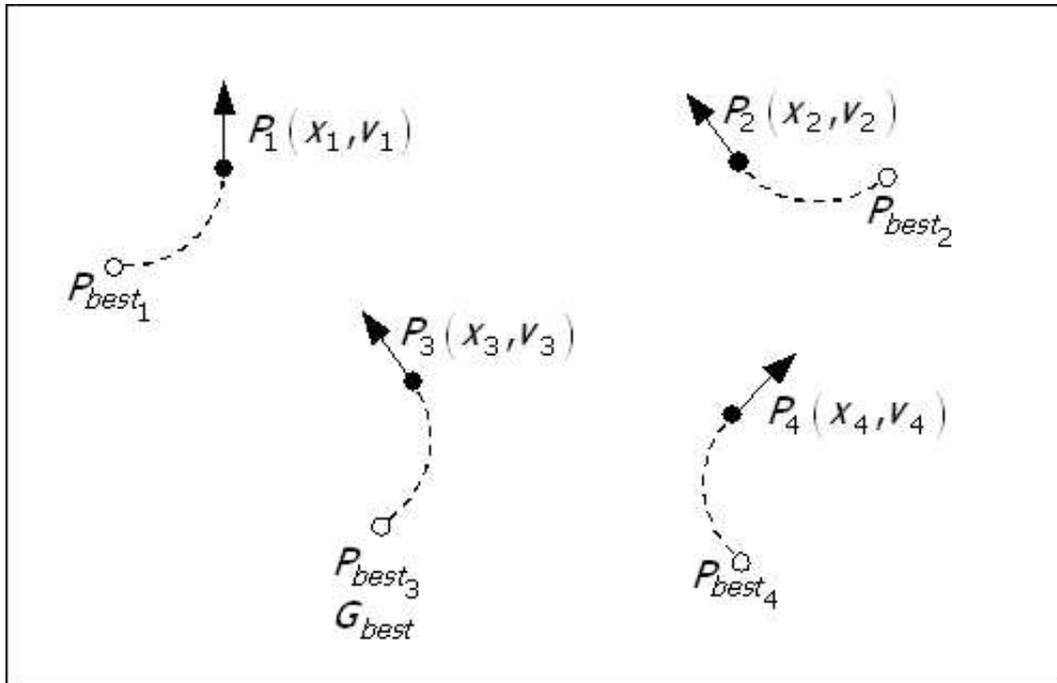


Abbildung 1: Beispiel eines Partikelschwarms von vier Partikeln.

Der prinzipielle Ablauf einer PSO lässt sich nach [3] gemäß Algorithmus 1 beschreiben. Der Algorithmus beginnt mit der Initialisierung eines Schwarms von Partikeln  $P_i$  mit gleichverteilt zufällig ermittelten Koordinaten  $\mathbf{x}_i$  und Geschwindigkeiten  $\mathbf{v}_i$  im Optimierungssuchraum. Die Berechnung der Gütewerte  $P_i$  für jedes Partikel  $P_i$  des Schwarms initialisiert sowohl die lokalen Minima nach Betrag  $P_{best_i}$  und Ort  $\mathbf{x}_{best_i}$  als auch das globale Minimum  $G_{best}$  in  $\mathbf{x}_{G_{best}}$  der ersten Iteration (*Epoche*).

Unter Verwendung dieser Extrema erfolgt individuell für jedes Partikel  $P_i$  eine stochastisch variierte Anpassung der Geschwindigkeit  $\mathbf{v}_i$ . Die dabei verwendete positive Konstante  $w$  stellt das Momentum eines Partikels dar. Der zweite Teil der Geschwindigkeitsanpassung  $c_1 \text{rand}(\mathbf{x}_{best_i} - \mathbf{x}_i)$  ist der sogenannte *Kognitionsteil*, der die unabhängige Bewegung des Partikels selbst darstellt. Der dritte und letzte Beitrag  $c_2 \text{Rand}(\mathbf{x}_{G_{best}} - \mathbf{x}_i)$  beschreibt als *Sozialteil* das kooperative Verhalten unter den Partikeln. Die Zufallszahlen  $\text{rand}(\dots)$  und  $\text{Rand}(\dots)$  werden nach einer gleichmäßigen Wahrscheinlichkeitsverteilung im Intervall  $[0, 1]$  erzeugt. Im Rahmen dieser Arbeit werden in Anlehnung an [4] hierzu die Optimierungsergebnisse für drei unterschiedliche Verteilungsarten vorgestellt.

Generell wird die Geschwindigkeit eines jeden Partikels noch auf eine maximale Geschwindigkeit  $\bar{\mathbf{v}}_i$  beschränkt, mit der gewissermaßen die Auflösung bestimmt wird, mit der die Regionen um die aktuelle Lösungen erforscht werden. Hohe Werte von

---

**Algorithmus 1** Standardvariante PSO

---

**Gegeben:** Schwarmgröße  $n$ , Gewichtungsparmeter  $w$ ,  $c_1$ ,  $c_2$

```
for  $i = 1 \dots n$  do
   $\mathbf{x}_i = \underline{\mathbf{x}}_i + (\underline{\mathbf{x}}_i - \bar{\mathbf{x}}_i) * \mathcal{U}(0, 1)$ 
   $\mathbf{v}_i = \underline{\mathbf{v}}_i + (\underline{\mathbf{v}}_i - \bar{\mathbf{v}}_i) * \mathcal{U}(0, 1)$ 
   $\mathbf{x}_{best_i} = \mathbf{x}_i$ 
   $P_{best_i} = f(\mathbf{x}_i)$ 
end for
 $G_{best} = \mathit{argmin}\{f(\mathbf{x}_i)\}$ 
 $\mathbf{x}_{G_{best}} = \{\mathbf{x} | f(\mathbf{x}) = G_{best}\}$ 
while Abbruchbedingung nicht erfüllt do
  for  $i = 1 \dots n$  do
     $\tilde{\mathbf{v}}_i = w\mathbf{v}_i + c_1\mathit{rand}(\mathbf{x}_{best_i} - \mathbf{x}_i) + c_2\mathit{Rand}(\mathbf{x}_{G_{best}} - \mathbf{x}_i)$ 
     $\tilde{\mathbf{x}}_i = \mathbf{x}_i + \tilde{\mathbf{v}}_i$ 
     $\mathbf{x}_i \leftarrow \tilde{\mathbf{x}}_i, \quad \mathbf{v}_i \leftarrow \tilde{\mathbf{v}}_i$ 
    if  $f(\mathbf{x}_i) < P_{best_i}$  then
       $P_{best_i} \leftarrow f(\mathbf{x}_i)$ 
       $\mathbf{x}_{best_i} \leftarrow \mathbf{x}_i$ 
    end if
    if  $f(\mathbf{x}_i) < G_{best}$  then
       $G_{best} \leftarrow f(\mathbf{x}_i)$ 
       $\mathbf{x}_{G_{best}} \leftarrow \mathbf{x}_i$ 
    end if
  end for
end while
```

---

$\bar{\mathbf{v}}_i$  fördern ein exploratives Verhalten im Sinne einer globalen Breitensuche. Entsprechend bewirken kleine Werte für  $\bar{\mathbf{v}}_i$  ein exploitatives Verhalten im Sinne einer lokalen Tiefensuche. Für zu kleine Werte der maximalen Geschwindigkeit besteht somit die Gefahr einer unerwünschten vorzeitigen Konvergenz in einem lokalen Minimum, da aufgrund der fehlenden Charakteristik einer Breitensuche möglicherweise nicht genügend gute Regionen des Suchraums besucht werden.

### 3 Evolutionäre Regleroptimierung für ein industrielles Hydraulikventil

#### 3.1 Hydraulikventil

Der Einsatzbereich industrieller Hydraulikventile erstreckt sich über weite Bereiche des Maschinen- und Anlagenbaus, wie beispielsweise in der Holz- und Papier- und Verpackungsindustrie.

Bei dem hier untersuchten Hydraulikventil handelt es sich um ein Regelventil NG6 der Bosch Rexroth AG, ehemals Robert Bosch GmbH. Abbildung 2 zeigt schematisch das zur Gruppe der Schieberventile gehörende Regelventil in der Ansicht eines Querschnitts. Daraus ersichtlich enthält das Ventil mittig in einer Längsbohrung im Gehäuse einen Kolben (Ventilschieber), der durch magnetische Krafteinwirkung in Längsrichtung verschiebbar ist. In die Längsbohrung sind Ringkanäle eingearbeitet,

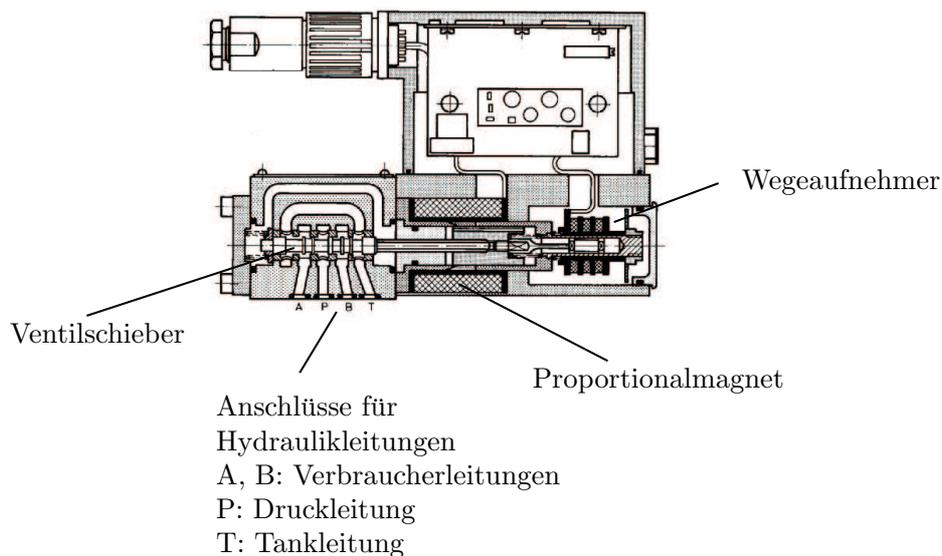


Abbildung 2: Querschnittsdarstellung des eingesetzten Hydraulikventils [5].

die jeweils mit einem äußeren Anschluss am Gehäuse verbunden sind (Anschlüsse A, B, P und T). Eine Bewegung des Ventilschiebers bewirkt eine Veränderung der Durchflusswege für eine Hydraulikflüssigkeit durch Schließung und Öffnung der einzelnen Ringkanäle. Die Position  $x(t)$  (auch als *Ventilhub* bezeichnet) des Ventilschiebers soll mit Hilfe der eingesetzten *Lageregelung* geregelt werden. Detailliertere Informationen zur Wirkungsweise und zum Einsatz von Hydraulikventilen sind in [6] zu finden.

### 3.2 Lageregelung

Die Dynamik hydraulischer Stetigventile wird wesentlich beeinflusst durch Nichtlinearitäten wie Strombegrenzung, Coulombsche Reibung, magnetische Hysterese, Sättigung der Flussdichte sowie durch Störgrößen in Form von Strömungskräften. Beim Einsatz als Stellglied in einem geschlossenen Regelkreis, z. B. zur Positionierung eines Zylinderantriebs, werden hohe Anforderungen an die dynamischen Eigenschaften, wie Dämpfung und Grenzfrequenz des Ventillageregelkreises, gestellt. Die Struktur des Reglers ist aus diesem Grund stark an die Nichtlinearitäten der Strecke angepasst und beinhaltet eine Vielzahl von einzustellenden Parametern.

Bei der hier eingesetzten Reglerstruktur handelt es sich dabei im Wesentlichen um eine Kombination aus einem erweiterten PI-Regler und einem Zustandsregler gemäß Abbildung 3. Dabei wird die auftretende Regelabweichung  $\Delta x(t) = x_{soll}(t) - x(t)$  im P- und im I-Zweig jeweils unterschiedlich im Klein-, im Mittel- und im Großsignalbereich<sup>1</sup> durch enthaltene Kennlinienglieder verarbeitet. Die Verstärkungsfaktoren beider Zweige  $p_p(\Delta x)$  und  $p_i(\Delta x)$  stehen demnach in einem nichtlinearen Zusammenhang mit der Größe der Regelabweichung.

<sup>1</sup>Die Bezeichnungen Klein-, Mittel- und Großsignal beziehen sich auf die Amplitudenänderung des jeweiligen Signals.

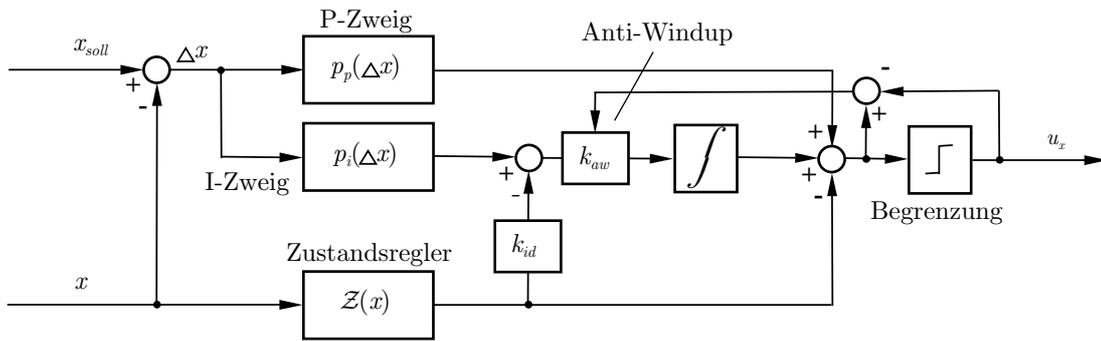


Abbildung 3: Struktur des verwendeten Lagereglers [1].

Der Zustandsregler erhält als Eingangsgröße den Ventilhub  $x(t)$  und weist unter anderem einen zweifach differenzierenden Charakter auf und bringt dadurch ein Maß für die Geschwindigkeit  $\dot{x}(t)$  und die Beschleunigung  $\ddot{x}(t)$  des Ventilschiebers in die Stellgröße des Lagereglers  $u_x$  ein. Im Zweig des Zustandsreglers sind ebenfalls Verstärkungsfaktoren und Verzögerungs-Zeitkonstanten enthalten.

Insgesamt enthält die hier verwendete Reglerstruktur 24 zu optimierende Parameter, wovon 8 Parameter auf den P-Zweig, 11 Parameter auf den I-Zweig und 5 Parameter auf den Zustandsregler entfallen.

### 3.3 Evolutionäre Regleroptimierung

Für die Anbindung der Lageregelung an einen automatisierten Optimierungsprozess war es zuerst notwendig, ein geeignetes Gütemaß zur Beurteilung des Regelungsverhaltens zu entwerfen. Hierzu wurde ein Fuzzy-Gütemaß entwickelt, das ausschließlich im *Zeitbereich* arbeitet. Grundlage hierfür ist die Analyse des Ventilverhaltens im Regelkreis als Reaktion auf eine Stimulusfunktion, die sich aus unterschiedlichen Sollwertänderungen zusammensetzt. Aus dem Regelungsverhalten des Ventils werden vier charakteristische Kenngrößen abgeleitet: *Überschwingen* ( $M_p$ ), *Anregelzeit* ( $T_{An}$ ), *Ausregelzeit* ( $T_{Aus}$ ) und *Schwingneigung*.

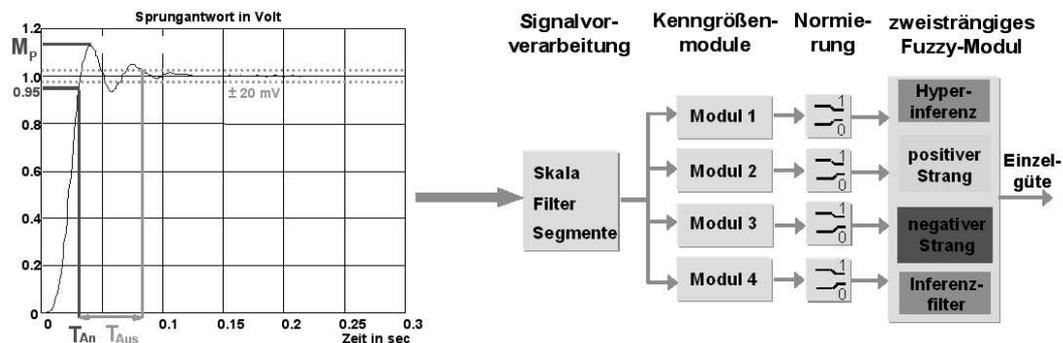


Abbildung 4: Prinzip der Gütebewertung eines Regelungsverhaltens.

Diese Kenngrößen dienen als Eingangsgrößen eines Fuzzy-Bewertungsmoduls (Abbildung 4). Als Fuzzy-Modul wird dabei eine zweisträngige Struktur mit Inferenzfilter

gewählt [7]. Dieses zweisträngige Konzept erlaubt auch die Verarbeitung von Warnungen in Form von negativen Regeln und liefert somit einen wesentlichen Beitrag zu der wunschgemäß starken Abwertung inakzeptablen Regelungsverhaltens. Darüber hinaus lassen sich mit dieser Fuzzy-Struktur die Anforderungen an das Monotonieverhalten des Gütemaßes in Bezug auf die Eingangsgrößen erfüllen. Zusätzlich ergibt sich der Vorteil einer objektiven Gütebewertung [8].

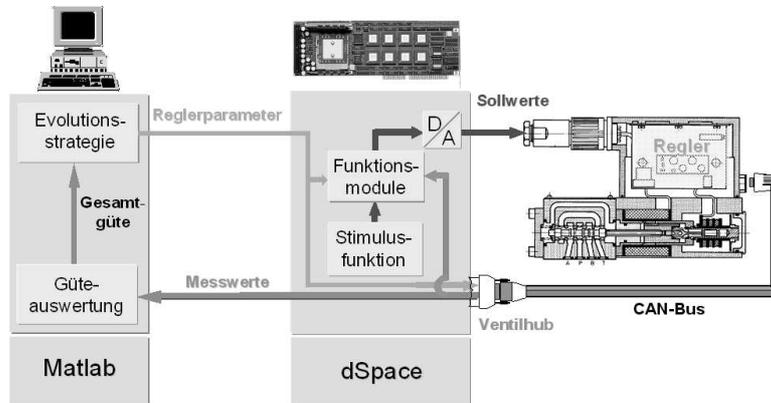


Abbildung 5: Aufbau der evolutionären Hardware-in-the-Loop Optimierung.

Mit Hilfe dieses entworfenen Gütemaßes erfolgte eine automatisierte Optimierung der Reglerparameter des digitalen Lagereglers anhand des realen Hydraulikventils (Hardware-in-the-Loop). Als Optimierungsalgorithmus wurde aufgrund der großen Anzahl der Reglerparameter eine Evolutionsstrategie eingesetzt. Abbildung 5 zeigt schematisch den Aufbau des Laborstands.

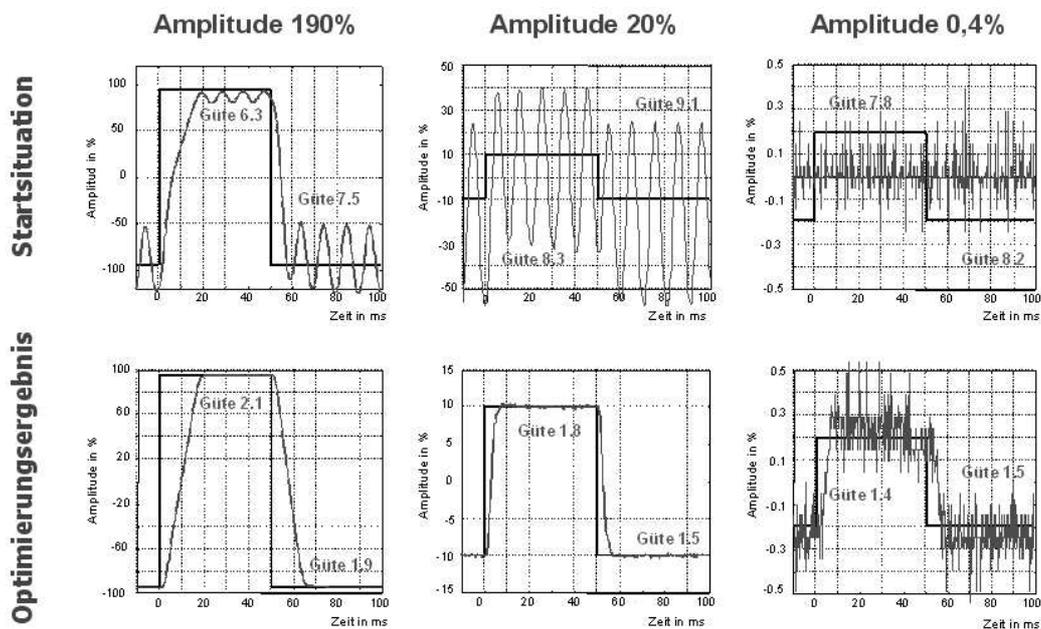


Abbildung 6: Sprungantworten im Groß- (links), Mittel- (Mitte) und Kleinsignalbereich (rechts) zu Beginn (oben) und Ende (unten) einer evolutionären Optimierung.

Die mit diesem Aufbau durchgeführten Optimierungen zeigten, dass die Evolutionsstrategie durchaus in der Lage ist, eine Ventilregelung mit relativ schlechter Anfangsgüte zu einem derart guten dynamischen Verhalten zu führen, wie es bis dahin lediglich mit großem Zeitaufwand durch interaktive Optimierung durch einen Prozessexperten erzielt werden konnte. Abbildung 6 zeigt exemplarisch die typische Entwicklung von relativ schlechten Sprungantworten zu Optimierungsbeginn hin zu gutem Regelungsverhalten zum Ende der Optimierung in den unterschiedlichen Signalbereichen. Dennoch konnte auch ein deutliches Verbesserungspotenzial eruiert werden, wozu in [1] zahlreiche neue Strategieelemente entwickelt und evaluiert wurden. Die dort erzielten Ergebnisse dienen als Referenz für die Untersuchungen zum Einsatz einer Partikel Schwarm Optimierung.

## 4 Einordnung der mit PSO erzielten Ergebnisse

Die Analyse der mit der Partikel Schwarm Optimierung (PSO) erzielten Ergebnisse verfolgt verschiedene Fragestellungen. Es ist zunächst von grundsätzlichem Interesse, ob dieser Optimierungsansatz erfolgreich für die hier vorliegende industrielle Anwendung eingesetzt werden kann. Weitergehend stellt sich die Frage, wie die mit der PSO erzielten Ergebnisse im Vergleich zu denen anderer Optimierungsverfahren, die an dem Hydraulikventil eingesetzt worden sind, einzuordnen sind.

Wie in Abschnitt 3 ausgeführt, erfolgt die Bewertung der Güte einer Reglerparametereinstellung  $\mathbf{p}$  anhand charakteristischer Merkmale verschiedener Sprungantworten, die mit dieser Einstellung erzielt werden. Diese Merkmale werden zunächst durch ein Fuzzy-Modul zu verschiedenen Einzelgüten  $q_i$  für die einzelnen Sprungantworten verrechnet. Diese Güten  $q_i$  orientieren sich in ihrem Wertebereich an dem Schulnotensystem. Die bestmögliche Güte von  $q_i = 1$  entspricht somit *sehr gut*, und eine Güte von  $q_i = 6$  entspricht *ungenügend*. Um eine differenziertere Bewertung schlechter Einstellungen vornehmen zu können, wird ein zweites Fuzzy-Modul eingesetzt, das bei einer Güte von  $q_i = 6$  ggf. zusätzlich einen weiteren Strafanteil hinzufügt. Die schlechtmögliche Güte liegt daher bei  $q_i = 26$ . Aus den genannten Einzelgüten wird dann eine Gesamtgüte  $Q$  mit  $1 \leq Q \leq 26$  für die gegebene Reglerparametereinstellung  $\mathbf{p}$  berechnet [1]. Ein von Experten als *akzeptabel* eingestuftes Regelungsverhalten setzt erfahrungsgemäß eine Güte  $Q \leq 3$  voraus. Dies kann als obere Grenze für die Performance angesehen werden, die ein wie auch immer geartetes Optimierungsverfahren für dieses Problem erzielen muss. Ein von Experten als *gut* eingestuftes Regelungsverhalten wird erfahrungsgemäß ab einer Güte  $Q \leq 2.5$  erreicht. Dies ist ein zweiter charakteristischer Referenzwert für die Beurteilung einer Optimierung.

Für die Optimierung der 24 Reglerparameter ist in vorangegangenen Arbeiten eine Evolutionsstrategie (ES) erfolgreich eingesetzt worden, wobei für diese eine Reihe neuer dezentraler Strategieelemente entwickelt worden sind. Eine detaillierte Aufstellung der so erzielten Ergebnisse findet sich in [1]. Abbildung 7 zeigt exemplarisch den Güteverlauf einer Optimierung mit der erweiterten Evolutionsstrategie.

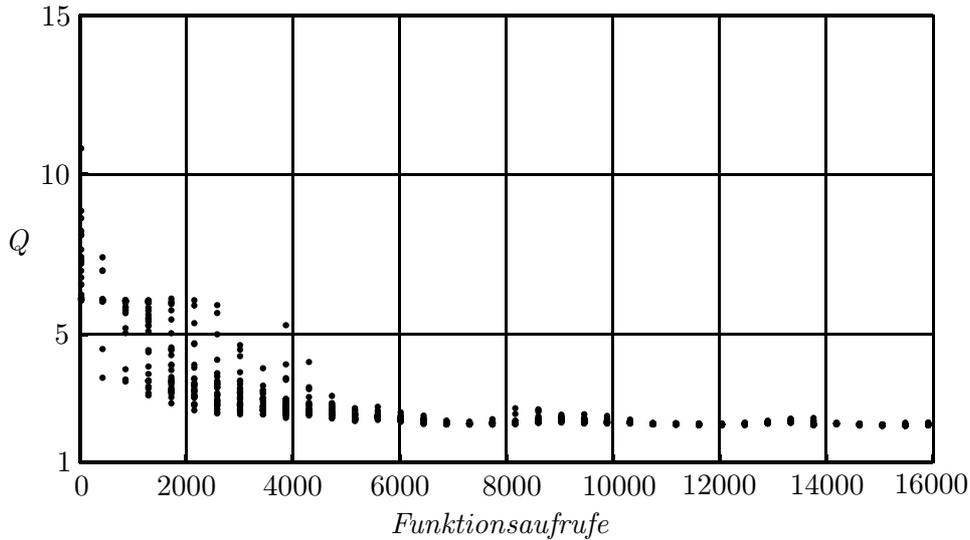


Abbildung 7: Typischer Güteverlauf einer Hardware-in-the-Loop-Optimierung mit der erweiterten Evolutionsstrategie aus [1].

Die Darstellung der Güteentwicklung in Abbildung 7 ist wie folgt zu verstehen. Es wird eine  $(\mu + \lambda)$ -Strategie verwendet, wobei die Anzahl der Eltern zu  $\mu = 30$  und die Anzahl der durch Reproduktion und Mutation erzeugten Nachkommen in einer Generation zu  $\lambda = 400$  gewählt worden ist. Die Optimierung hat sich in dem dargestellten Fall über 40 Generationen hingezogen<sup>2</sup>, wodurch sich die Gesamtanzahl der Individuen, für die Gütebewertungen (Funktionsaufrufe) vorgenommen werden mussten, zu  $D_{max} = 40 \cdot 400 = 16000$  ergibt<sup>3</sup>.

In Abbildung 7 sind zur besseren Übersicht nur die jeweils 30 Eltern einer jeden Generation übereinander dargestellt, nicht aber die Nachkommen. Die Abszisse ist zur besseren Einordnung des Aufwands auf die Gesamtanzahl der Funktionsaufrufe skaliert. Es ist zu erkennen, dass in den ersten zehn Generationen ein relativ breites Spektrum an Gütewerten ( $2.5 \leq Q_i \leq 6$ ) abgedeckt wird. In [1] wird ausgeführt, dass diese Diversität in den Gütewerten (auch als Phänotypen bezeichnet) auf eine entsprechende Diversität der Individuen (auch als Genotypen bezeichnet) schließen lässt. Der Parameterraum wird also zu Beginn der Optimierung vergleichsweise breit durchsucht (Breitensuche). In den nachfolgenden zehn Generationen konvergieren die Gütewerte relativ stark, was eine entsprechende Konzentration der Suche im Parameterraum vermuten lässt (Tiefensuche). In den letzten 20 Generationen setzt sich die Konvergenz fort und es tritt insgesamt keine signifikante Veränderung mehr ein. Der oben eingeführte Referenz-Gütwert  $Q_{ref,1} = 3$  wird bereits sehr früh erfüllt, der Referenz-Gütwert  $Q_{ref,2} = 2.5$  wird nach etwa 4000 Funktionsaufrufen erstmals erreicht. Der beste erreichte Gütwert liegt in etwa bei  $Q_{min} = 2.1$ .

Als Alternative zu der Evolutionsstrategie wird hier eine Partikel Schwarm Optimierung (PSO) eingesetzt. Die PSO in der Standardvariante (siehe Algorithmus 1) weist eine Reihe frei wählbarer Parameter auf, die eine Anpassung an das jeweilige Optimierungsproblem ermöglichen sollen. Auf Details wird an dieser Stelle verzichtet,

<sup>2</sup>Die Optimierung wird beendet, wenn ein geeignetes Abbruchkriterium erfüllt wird.

<sup>3</sup>Die Startpopulation mit 30 Individuen ist dabei nicht mitgerechnet.

von Bedeutung sind hier in erster Linie die Anzahl der Partikel und die Anzahl der maximal zulässigen Epochen. Diese Parameter entsprechen im Prinzip der Anzahl an Nachkommen bzw. der Anzahl an Generationen für die Evolutionsstrategie. Im weiteren Sinne kann zu den Parametern auch die Art und Weise gezählt werden, mit der die für die PSO benötigten Zufallszahlen generiert werden. Die Verwendung bestimmter Verteilungen für die Generierung von Zufallszahlen kann u. U. gewisse Vorteile bringen [4].

Für die ersten Experimente wurde die Anzahl der Partikel zu  $N_P = 100$  und die Anzahl der maximal zulässigen Epochen ebenfalls zu  $E = 100$  gewählt. Die maximale Anzahl an zu untersuchenden Reglerparametereinstellungen ergibt sich zu  $D_{max} = N_P \cdot E$  und beträgt in diesem Fall somit  $D_{max} = 10000$ . Orientiert man sich an dem typischen Verlauf der Güteentwicklung bei der Optimierung mit der Evolutionsstrategie aus Abbildung 7, so scheint dies ausreichend groß gewählt zu sein. Für alle weiteren Parameter der PSO wurden die Standardinstellungen mit  $c_1 = c_2 = 2.0$  [4] beibehalten. Für die Generierung der Zufallszahlen sind nacheinander verschiedene Verteilungen eingesetzt worden: Gleichverteilung, Normalverteilung und Cauchy-Verteilung. Tabelle 1 gibt eine Übersicht über die so erzielten Ergebnisse.

Verteilung	$Q_{min,mean}$	$Q_{min,best}$	$E(Q_{ref,1})$	$E(Q_{ref,2})$
Gleich	2.75	2.20	52/100	84/100
Normal	2.01	1.82	25/100	28/100
Cauchy	2.15	2.10	48/100	70/100

Tabelle 1: Ergebnisse der Optimierung mit PSO für verschiedene Varianten der Generierung von Zufallszahlen

Für jede Variante ist die Optimierung dreimal mit identischen Parametereinstellungen der PSO durchgeführt worden<sup>4</sup>. In Tabelle 1 sind der Mittelwert  $Q_{min,mean}$  der jeweils besten erzielten Güte (je ein Gütewert je Wiederholung der PSO) und das beste Einzelergebnis  $Q_{min,best}$  jeder Variante angegeben. Ferner die mittlere Anzahl an Epochen, die benötigt wird, um die Referenz-Gütwerte  $Q_{ref,1}$  bzw.  $Q_{ref,2}$  zu erreichen<sup>5</sup>. Die Ergebnisse zeigen, dass die PSO in allen drei Varianten die oben genannte Mindestgüte von  $Q_{ref,1} = 3$  erfüllt bzw. teilweise sogar deutlich bessere Ergebnisse liefert. Im Hinblick auf die Referenzgüte  $Q_{ref,2} = 2.5$  zeigt sich, dass dieser Wert bei der Verwendung gleichverteilter Zufallszahlen im Durchschnitt nicht erreicht wird, bei normalverteilten oder Cauchy-verteiltern Zufallszahlen aber immer übertroffen wird. Die besten Ergebnisse werden bei der Verwendung normalverteilter Zufallszahlen erzielt, sowohl bzgl. des besten Gütewertes als auch im Hinblick auf die Anzahl der benötigten Epochen zum Erreichen der Referenz-Gütwerte. In Abbildung 8 ist die Güteentwicklung für das Experiment dargestellt, das insgesamt das beste Ergebnis liefert (normalverteilte Zufallszahlen,  $Q_{min} = 1.82$ ). Die Art der Darstellung entspricht im Prinzip der Art, die auch schon in Abbildung 7 angewandt worden ist.

<sup>4</sup>Aus Zeitgründen war eine höhere Anzahl an Wiederholungen nicht möglich.

<sup>5</sup>Hierfür werden nur diejenigen Experimente betrachtet, bei denen die genannten Referenz-Gütwerte auch erreicht werden.

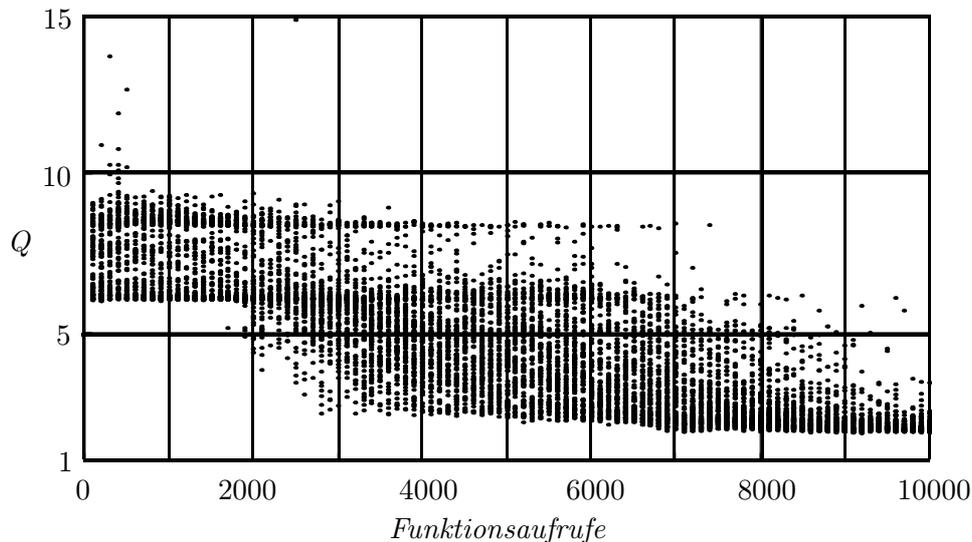


Abbildung 8: Güteverlauf der Optimierung mit der PSO (bei Verwendung normalverteilter Zufallszahlen)

Die Analyse der Ergebnisse aus Tabelle 1 zeigt, dass die PSO vergleichbare beste Güten  $Q_{min}$  erreicht wie die ES. Im Hinblick auf andere Bewertungskriterien wie beispielsweise eine anhaltende Diversität der Population und eine damit verbundene Abdeckung des Suchraumes scheint die PSO zudem Vorteile gegenüber der ES zu haben. Vergleicht man die Güteentwicklung bei dem Einsatz der ES (Abbildung 7) mit der bei dem Einsatz der PSO (Abbildung 8), so zeigt sich bei der PSO eine stärker ausgeprägte und zudem deutlich länger anhaltende Diversität der Gütewerte. Es sei jedoch noch einmal darauf hingewiesen, dass in Abbildung 7 nur die Eltern, nicht aber die Nachkommen dargestellt sind.

Um unterschiedliche Präferenzen industrieller Anwender möglichst gut, transparent und schnell berücksichtigen zu können, ist das in Abschnitt 3 vorgestellte Gesamtkonzept entwickelt worden. In diesem Zusammenhang ist es wünschenswert, dass auch das Optimierungsverfahren möglichst einfach einzusetzen ist und möglichst rasch Reglereinstellungen liefert, die den praktischen Anforderungen genügen. Die einfache Handhabbarkeit des eingesetzten Optimierungsverfahrens ist wichtig, damit industrielle Anwender ggf. selbst Optimierungen durchführen können, ohne sich zu sehr mit dem Verfahren an sich beschäftigen zu müssen. Aufgrund der anschaulichen und gut nachvollziehbaren zugrunde liegenden Wirkungsprinzipien und der vergleichsweise geringen Anzahl an interpretierbaren Parametern kann die PSO auch von Nicht-Experten eingesetzt werden. Weiterhin ist auch zu bedenken, dass industrielle Anwender i. d. R. eine Abwägung von Aufwand und Nutzen vornehmen und einen Kompromiss zwischen diesen beiden Aspekten suchen.

Bezogen auf die zeitintensive und damit u. U. auch kostenintensive Hardware-in-the-loop-Optimierung könnte dies bedeuten: Ein Anwender würde u. U. eine Reglereinstellung mit einer Güte von  $Q = 2.1$ , die er nach 4h Optimierung erreicht und die seinen praktischen Anforderungen voll genügt, in der Aufwand-Nutzen-Abwägung einer Reglereinstellung mit einer Güte von  $Q = 1.8$ , die er aber erst nach 12h erreicht,

vorziehen. Im Hinblick auf diesen Aspekt kann das praxisbezogene Ziel formuliert werden, mit hoher Sicherheit möglichst schnell eine gute Einstellung zu finden.

Unter Sicherheit wird hier verstanden, dass durch die Optimierung mit hoher Wahrscheinlichkeit auch tatsächlich ein entsprechender Gütewert erreicht wird. Unter Geschwindigkeit wird hier die Anzahl der hierfür benötigten Güteauswertungen verstanden, da diese den Zeit- und damit den Kostenaufwand bestimmen. Die beiden Teilziele Sicherheit und Geschwindigkeit sind oftmals gegenläufig. So könnte beispielsweise eine zufällige Erzeugung von Reglereinstellungen sehr rasch zum Ziel führen. Experimente an dem Hydraulikventil haben aber gezeigt, dass es oftmals auch nach mehreren zehntausend Versuchen so nicht gelingt, eine geeignete Reglereinstellung zu generieren. Eine andere Strategie kann darin bestehen, ein lokales gradientenbasiertes Suchverfahren einzusetzen. Die Suche wird dabei ggf. jeweils von einem zufällig generierten Startpunkt aus neu gestartet. Bei einer günstigen Wahl des Startpunktes und einer nicht zu komplexen Gütelandschaft kann so prinzipiell schnell ein gutes Resultat erzielt werden. Diese Strategie ist ebenfalls in [1] untersucht worden. Die Ergebnisse zeigen aber, dass so bestenfalls nur eine Güte von  $Q_{min} = 2.7$  erreicht werden konnte, zudem scheitert der Großteil der einzelnen lokalen Suchen gänzlich.

Unter dem Gesichtspunkt der Sicherheit ist diese Strategie für das hier betrachtete Problem daher eher ungeeignet. Die Ergebnisse sowohl der ES als auch der PSO haben gezeigt, dass beide Ansätze im Hinblick auf die Sicherheit gut geeignet sind, da sie nur in sehr wenigen Ausnahmefällen einmal nicht zu einem akzeptablen bzw. guten Ergebnis führen. Es stellt sich dann die Frage, wie sich die Ansätze unter dem Aspekt der Geschwindigkeit miteinander vergleichen. Es sind daher weitere Experimente durchgeführt worden, bei denen die Anzahl der Partikel auf  $N_P = 60$  reduziert worden ist. Die Anzahl der Epochen wurde zu  $E = 100$  gewählt, sodass insgesamt  $D_{max} = 6000$  Güteauswertungen zur Verfügung standen. Für alle übrigen Parameter der PSO wurden die Default-Einstellungen beibehalten, und es wurden normalverteilte Zufallszahlen verwendet. Die Optimierung wurde zehnmal wiederholt, die Ergebnisse der einzelnen Experimente sind in Tabelle 2 aufgeführt.

<i>Laufindex</i>	1	2	3	4	5	6	7	8	9	10
$Q_{min}$	2.71	2.04	2.60	1.98	2.13	2.06	2.17	2.06	3.31	2.78
$E(Q_{ref,1})$	42	31	64	29	50	39	33	31	-	52
$E(Q_{ref,2})$	-	46	-	33	67	39	65	40	-	-

Tabelle 2: Ergebnisse für die wiederholte Durchführung der PSO mit  $N_P = 60$

Es zeigt sich, dass einer der zehn Durchläufe negativ herausfällt, da nur eine Güte von  $Q_{min} > 3$  erzielt wird. Für drei der zehn Experimente ist  $2.5 < Q_{min} < 3$ . Sechs der zehn Experimente liefern eine Güte von  $Q_{min} \approx 2$ , also einen guten Gütewert. Tabelle 3 zeigt eine einfache statistische Auswertung der Experimente. Ein Vergleich mit den Ergebnissen aus Tabelle 1 ist aufgrund der unterschiedlichen Anzahl an Partikeln und Wiederholungen der Experimente nur bedingt möglich, aber es zeigt sich in der Tendenz, dass trotz des insgesamt geringeren Aufwandes nur eine verhältnismäßig kleine Verschlechterung der erzielten Güte  $Q_{min}$  eintritt. Es

zeigt sich aber eine relativ starke Streuung in der Anzahl der benötigten Epochen, um einen der Referenz-Gütwerte zu erreichen. Die Anzahl ist für dieses Beispiel im ungünstigsten beobachteten Fall in etwa doppelt so hoch wie im günstigsten beobachteten Fall. Auch bei vergleichbaren Endergebnissen  $Q_{min}$  zeigen sich zwischen den Experimenten teilweise deutliche Unterschiede.

$Q_{min,worst}$	3.31	$E(Q_{ref,1})_{worst}$	64	$E(Q_{ref,2})_{worst}$	67
$Q_{min,best}$	1.98	$E(Q_{ref,1})_{best}$	31	$E(Q_{ref,2})_{best}$	33
$Q_{min,mean}$	2.38	$E(Q_{ref,1})_{mean}$	41	$E(Q_{ref,2})_{mean}$	48
$Q_{min,median}$	2.15	$E(Q_{ref,1})_{median}$	39	$E(Q_{ref,2})_{median}$	43
$std(Q_{min})$	0.44	$std(E(Q_{ref,1}))$	12	$std(E(Q_{ref,2}))$	14

Tabelle 3: Statistik für die wiederholte Durchführung der PSO mit  $N_P = 60$

In Abbildung 9 sind für die zehn Experimente die Güten der Partikel in der jeweils letzten Epoche dargestellt. Der beste Gütwert  $Q_{min}$  eines Experiments wird nicht zwangsläufig in der letzten Epoche erreicht, zur besseren Vergleichbarkeit mit den Ergebnissen aus Tabelle 2 ist dieser für jedes Experiment gesondert als ‘+’ dargestellt.

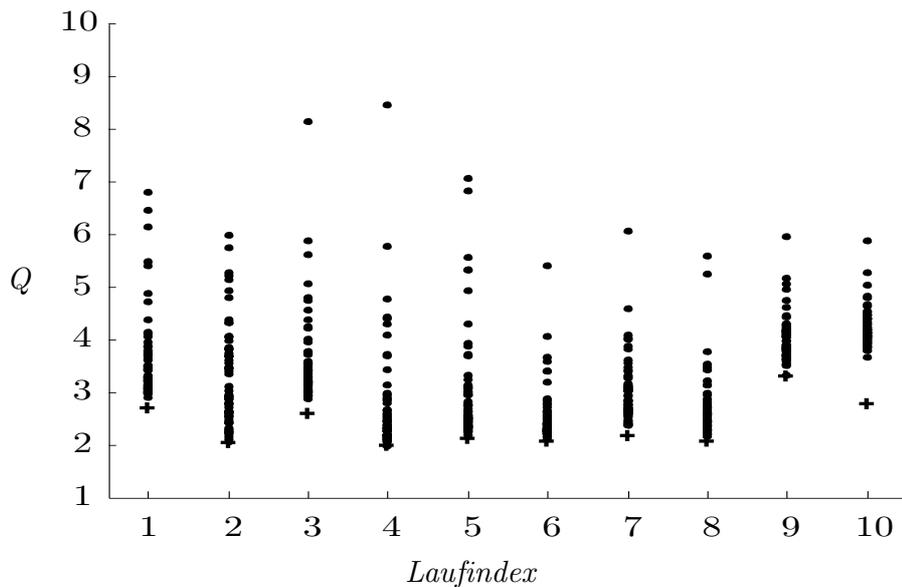


Abbildung 9: Gütediversität innerhalb der jeweils letzten Epoche

In allen Fällen ist in der letzten Epoche noch immer eine deutliche Diversität in den Gütwerten zu erkennen. Es kann daher vermutet werden, dass diejenigen Experimente, die bis zu dem betrachteten Zeitpunkt nur zu einer Güte  $Q_{min} > 2.5$  geführt haben, bei längerer Laufzeit durchaus noch bessere Resultate hätten erzielen können.

## 5 Zusammenfassung und Ausblick

In diesem Beitrag wird der erfolgreiche Einsatz einer Partikel Schwarm Optimierung zur Hardware-in-the-Loop-Optimierung eines hochdimensionalen realen industriellen Regelungssystems dokumentiert. Im Vergleich zu den mit einer Evolutionsstrategie erzielten Ergebnissen stellen sich aus Anwendungssicht mit PSO vergleichbar gute Lösungen ein. Weiterhin zeigt sich bei allen Optimierungen mit PSO eine deutlich erhöhte Populationsdiversität über den gesamten Optimierungszeitraum, welches auf eine geringere Wahrscheinlichkeit einer vorzeitigen Konvergenz in einem lokalen Minimum schließen lässt. Bei Vorgabe einer deutlich reduzierten Optimierungszeit im Sinne der Anzahl zugelassener Güteauswertungen am realen Prozess stellt sich in fast allen Fällen ein *akzeptables* Ergebnis und in etwa jedem zweiten Optimierungslauf ein *gutes* Ergebnis ein. Aus Sicht des Anwenders ist somit in akzeptabler Zeit mit der verwendeten Standardvariante der PSO ein zufriedenstellendes Ergebnis erzielbar. Bezüglich weiterer Verbesserungen der PSO im Sinne einer beschleunigten und sichereren Auffindung von *guten* Reglerparametereinstellungen  $\mathbf{p}$  wird zukünftig der Einfluss der Parameter  $\bar{v}_i$ ,  $w$ ,  $c_1$  und  $c_2$  analysiert, mit dem Ziel, entsprechende gewinnbringende Adaptionstrategien konzipieren zu können.

## Literatur

- [1] OTTERSBUCH, B.: *Dezentrale Strategieelemente für Evolutionsstrategien und Anwendung zum Reglerentwurf*. 2005 (Universitätsbibliothek Dortmund). – <http://hdl.handle.net/2003/21532>
- [2] KENNEDY, J. ; EBERHART, R.: Particle Swarm Optimization. In: *Proceedings of the IEEE International Conference on Neural Networks* Bd. IV. Piscataway, NJ, USA : IEEE-Press, 1995, S. 1941–1948
- [3] KENNEDY, J. ; EBERHART, R. ; Y., Shi: *Swarm Intelligence*. San Francisco, USA : Morgan Kaufmann Publishers, 2001
- [4] COELHO, L.d.S. ; KROHLING, R.A.: Predictive Controller Tuning using Modified Particle Swarm Optimisation Based on Cauchy and Gaussian Distributions. In: HOFFMANN, F. ET AL. (Hrsg.): *Advances in Soft Computing - Engineering, Design and Manufacturing*, Springer Engineering series, 2005, S. 287–298
- [5] BLUMENDELLER, W. ; LEUTNER, V. ; SCHWANE, U. ; KIENDL, H.: Regleroptimierung mit evolutionären Algorithmen. In: *2. Internationales Fluidtechnisches Kolloquium*. Dresden, 2000, S. 437–452
- [6] Robert Bosch GmbH, Geschäftsbereich Automationstechnik: *Elektrohydraulische Proportional- und Regelungssysteme*. 1999
- [7] KIENDL, H.: *Fuzzy Control methodenorientiert*. München : Oldenbourg, 1997
- [8] SCHWANE, U. ; NICOLAUS, B. ; KIENDL, H. ; BLUMENDELLER, W. ; LEUTNER, V.: Fuzzy-Gütemaß zur evolutionären Optimierung eines industriellen hydraulischen Regelventils. In: *Fachtagung Computational Intelligence, VDI-Berichte 1526*. Baden Baden, 2000, S. 229–234

# Inferenz mit Fuzzy-Zeit-Termen

**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

## 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. Jedoch ist diese Erweiterung noch nicht komplett, denn die Zeit an sich ist scharf formuliert. Deshalb ist es die Aufgabe dieser Arbeit, eine Möglichkeit anzugeben, um nun auch die Zeit unscharf darstellen zu können und als Nebeneffekt noch die Lesbarkeit zu erhöhen. Es existieren einige Ansätze, um Zeit in Fuzzy-Logik einzubringen. In Kapitel 2 werden fünf solcher Ansätze vorgestellt.

### 1.2. Problemstellung

Ein Problem bei Fuzzy-Reglern ist, dass es keine sachlich motivierten Ansätze gibt, durch welche klar ersichtlich wird, wie zeitliche Aspekte in Fuzzy-Logik gehandhabt werden können. Model Checking mit temporaler Logik ist eine in sich geschlossene Methodik. Die temporalen Prädikate können zur Beschreibung von Szenarien genutzt werden um Prozesszustände mit zeitlichen Abhängigkeiten zu modellieren. Das Problem, welches in [Schmidt04] gelöst ist, beschäftigt sich damit, Prädikate der temporalen Logik in die Fuzzy-Logik zu übersetzen und zwar so, 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. Gesucht ist demnach die Vereinigung der Temporal-Logik und Fuzzy-Logik zur *temporalen Fuzzy-Logik*. Was jetzt noch fehlt ist die Beschreibung der Zeit anstatt in scharfen Zeitintervallen in unscharfen Fuzzy-Zeit-Termen.

Im weiteren Verlauf dieses Kapitels ist eine Abgrenzung zu anderen Arbeiten gegeben, während das zweite Kapitel den Stand der Forschung angibt. Das dritte Kapitel gibt die

Semantik und den Syntax der Fuzzy-Zeit-Terme an. Auch wird beschrieben, wie und wo diese eingesetzt werden können. Das vierte Kapitel zeigt die Unterschiede in der Funktionsweise der Fuzzy-Regelung einmal mit normaler Fuzzy-Logik und einmal mit temporaler Fuzzy-Logik. Kapitel 5 fasst die Arbeit zusammen und beschreibt die daraus gewonnenen Schlussfolgerungen.

### 1.3. Abgrenzung

Ein *Überwachungssystem* ist ein von einem zu überwachenden Prozess unabhängiges System, welches diesen Prozess in seinem Verhalten mittels Sensoren überwacht. Prozesskennzahlen informieren dabei über die internen, nicht zwingenderweise bekannten, Zustände des Prozesses. Bei den Aktuatoren, ist nicht immer bekannt, welchen quantitativen Einfluss diese auf den Prozess haben, somit ist eine Steuerung des Prozesses nicht möglich. Ist jedoch der qualitative Einfluss bekannt, so kann der Prozess geregelt werden. Werden nun in dem Verhalten des Prozesses Abweichungen zu den gewünschten benutzerdefinierten Vorgaben erkannt, kann ein *Regler* in das Verhalten des Prozesses eingreifen und Parameter so verändern, dass das Verhalten des Prozesses sich dem Verhalten nähert, welches von einem Benutzer gewünscht wird [Castillo02]. Ein einfaches Beispiel für ein Überwachungssystem ist die Überwachung der Helligkeit in einem Raum. Sinkt die Helligkeit unter einen angegebenen Schwellwert, so erkennt dies das Überwachungssystem und gibt dem Prozess den Auftrag den Raum stärker zu beleuchten, indem mehr Lampen angeschaltet werden.

Ein *vorausschauendes Überwachungssystem* benutzt nicht nur aktuelle Sensordaten aus dem Prozess, sondern auch mögliche Sensordaten aus der Zukunft [Fantoni00], [Palit00]. Natürlich ist es nicht möglich diese zukünftigen Daten zu messen. Sie müssen mit geeigneten Methoden aus dem bekannten vergangenen Signalverlauf vorhergesagt werden. Werden diese zukünftigen Sensorwerte an ein Überwachungssystem gegeben, so kann dieses eine zukünftige Abweichung vom gewünschten, benutzerdefinierten Verhalten feststellen. Das Eintreten der Abweichung ist dabei nicht garantiert; sie muss nicht eintreten. Im Gegenteil, dadurch, dass dem Überwachungssystem bekannt ist, was bei den aktuellen Parameterwerten in der Zukunft passieren würde, können schon frühzeitig Maßnahmen ergriffen werden, um ein anderes Verhalten herbei zu führen. Wenn nun im obigen Beispiel Lampen verwendet werden, welche eine lange Zeit benötigen, um ihre maximale Helligkeit zu erreichen (Neonröhren mit einem sehr hohen Wirkungsgrad: ca. 15 Minuten, Energiesparlampen: ca. 10 Minuten oder ähnliche), so genügt ein nicht vorausschauendes Überwachungssystem nicht mehr. Das vorausschauende Überwachungssystem kann jedoch feststellen, dass es im Raum immer dunkler wird. Bevor es im Raum zu dunkel ist, also die Helligkeit den angegebenen Schwellwert unterschreitet, schaltet das Überwachungssystem weitere Lampen ein.

Ein *Wartungssystem* baut meistens, nicht immer, auf einem Diagnosesystem auf [Althoff92]. Das in dieser Arbeit vorzustellende Wartungssystem baut jedoch auf einem (vorausschauenden) Überwachungssystem auf. Dann, wenn ein Überwachungssystem durch Veränderung der Prozessparameter keine Verbesserung mehr erreichen kann und sich die Prozesskennzahlen nicht innerhalb eines tolerierbaren Bereiches befinden, liegt ein *Fehler* im System vor, welcher nicht ausgeglichen werden kann. Dieser Fehler kann zum Beispiel eine defekte Teilkomponente sein, welche ersetzt werden muss. Das Wartungssystem generiert in diesem Fall einen Wartungsauftrag für einen Benutzer und teilt diesem mit, welche Teilkomponente einen Fehler verursacht haben könnte. Durch den vorausschauenden Aspekt eines Wartungssystems können Ausfälle dieser Art frühzeitig vorhergesagt und hierfür Wartungsaufträge generiert werden. Die Wartungsaufträge können zeitlich in der Zukunft datiert sein, da der vorhergesagte Ausfall nicht unmittelbar, sondern in der Zukunft eintritt. Das System wird so lange be-

trieben, wie es funktionsfähig bleibt, also seine Prozesskennzahlen in einem tolerierbaren Bereich liegen. Die Abstände zwischen verschiedenen Wartungen, bei welchen ein Bediener die Maschine anhält und sie repariert sind maximal. Dadurch, dass die Wartungsaufträge auch in der Zukunft liegen können und es so möglich ist mehrere Wartungsaufträge zu sammeln und zu einem Zeitpunkt alle Wartungsarbeiten parallel auszuführen, wird der Prozess nur einmal angehalten und somit die Standzeiten (*Wartungszeiten*) verringert. So entsteht kein außerplanmäßiger Produktionsausfall, denn die Wartungsarbeiten können eventuell zu Zeiten geringer Auslastung durchgeführt werden. Im obigen Beispiel entspricht dies dem Überwachungssystem, welches versucht den Raum durch Einschalten weiterer Lampen zu erhellen. Da aber defekte Lampen existieren, liegt die Helligkeit auch dann noch unter dem gegebenen Schwellwert, wenn alle Lampen angeschaltet sind. In diesem Fall generiert das Wartungssystem einen Wartungsauftrag, in welchem es dem Benutzer mitteilt, defekte Lampen im Raum auszutauschen. Eine andere Möglichkeit, dass ein Wartungssystem einen Wartungsauftrag generiert, ist wenn festgestellt wird, dass Lampen Anzeichen für einen baldigen Defekt aufweisen, so dass dann nicht mehr genügend Licht produziert werden kann.

Als Endziel, welches über die temporalen Fuzzy-Prädikate und Fuzzy-Zeit-Terme dieses Papers 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.

## **2. Stand der Forschung**

In der Forschungsabteilung der Firma Flender Service GmbH werden spezielle Sensoren zur Überwachung entwickelt – siehe [Flender01] und [Flender02]. Diese Sensoren dienen der Überwachung von beispielsweise Fräsmaschinen, welche Lager, Zahnräder und andere Komponenten mit hohem Verschleiß beinhalten. Zur Beobachtung der Komponenten mit hohem Verschleiß werden Schwingungssensoren verwendet, denn die Erfahrung zeigt, dass sich das Schwingungsprofil dieser Komponenten charakteristisch mit dem Erreichen des Endes der Lebensdauer verändert. Das heißt die Schwingungen werden aufgenommen, zur Steuereinheit übertragen und dort fouriertransformiert. Aus den Signalverläufen werden unter Verwendung von, mit diesen Maschinen, gesammeltem Wissen Rückschlüsse auf die weitere Lebensdauer der Komponenten gezogen, so dass diese ausgetauscht werden können bevor es zu einem Ausfall kommt. Einsetzbar ist dieses System in allen Maschinen, an welchen über Jahre hinweg der Verschleiß dieser Komponenten gemessen und protokolliert wurde. Die Einschränkung bei diesem Ansatz liegt dabei, Einzelteile einer Maschine zu beobachten, welche sich bewegen beziehungsweise hohen mechanischen Kräften ausgesetzt sind und durch diese Bewegung verschleifen, beziehungsweise durch die Kräfte ermüden. Dies stellt den Überwachungsanteil des Systems dar. Wartungsarbeiten werden vorgenommen, wenn das Überwachungssystem eine Wartungsfirma über das Internet über einen baldigen Ausfall informiert. Dies ist ein Wartungssystem, welches gezielt auf eine Aufgabe zugeschnitten ist. Unser Wartungssystem soll aber allgemeiner und flexibel einsetzbar sein. Außerdem ist unser Ansatz intuitiver und näher an der menschlichen Ausdrucksweise. Durch Verwendung von Fuzzy-Logik ist er mathema-

tisch formalisiert und dadurch nachvollziehbar. Im Folgenden werden fünf Ansätze vorgestellt, welche den Begriff der Zeit in Fuzzy-Logik einbringen.

Unter anderem in [Fick00] werden Takagi Sugeno Regeln [Takagi85] in einem Fuzzy-Regler zeitlich abhängig gemacht, indem die Zeit  $t$  als weitere Eingabevariable in den Bedingungsteil der Regeln mit aufgenommen wird. Zum Beispiel „**IF** expression **AND**  $t_0 - t$  **IS** now **THEN** action“. Dies hat zur Folge, dass Regeln nur zu gewissen Zeiten feuern oder wie bei [Fick00] neu dargestellt, jede Bedingung einer Regel mit einem Zeitintervall konjunktiv verknüpft werden muss. Die Regelaktivierung wird um so höher sein, je näher der aktuelle Zeitpunkt am gegebenen Zeitpunkt liegt. Solch ein Konstrukt ist dann nützlich, wenn Regeln nur zu einer bestimmten Tageszeit feuern dürfen. Zum Beispiel „ $(t_0 - t)$  modulo Tag **IS** *Mittagszeit*“. In diesem Ansatz ist die Zeit zwar unscharf, aber es gibt keine Möglichkeit, Ereignisse untereinander auf zeitliche Abhängigkeiten zu vergleichen, da nur eine Zeitangabe pro Regel vorgesehen ist. Diese Vergleichbarkeit wird jedoch von uns für ein Wartungssystem gefordert.

In [Bovenkamp97] wird ein neuer Ansatz zum temporalen Schließen mit Fuzzy-Logik vorgestellt. So genannte Fuzzy-Zeit-Objekte werden definiert, um Unschärfe in Fakten und Zeit zu repräsentieren. Fuzzy-Zeit-Objekte sind zweistellige, einwertige Zugehörigkeitsfunktionen der Form  $\mu_{f \times z}(x, t) := \min(\mu_f(x), \mu_z(t))$ . Für diese Zugehörigkeitsfunktionen muss für den Fakt  $\max_x \mu_f(x) = 1$  und die Zeit  $\max_t \mu_z(t) = 1$  gelten.

Damit gilt auch die Separierbarkeit  $\forall x \forall t \in \{t | \max_t \mu_z(t) = 1\} : \mu_f(x) = \mu_{f \times z}(x, t)$  beziehungsweise  $\forall t \forall x \in \{x | \max_x \mu_f(x) = 1\} : \mu_z(t) = \mu_{f \times z}(x, t)$ . Man kann also ein

Fuzzy-Zeit-Objekt genau dann wenn die Separierbarkeit gilt in Fakt und Zeit zerlegen, so dass man die ursprünglichen Fuzzy Terme  $\mu_f(x)$  beziehungsweise  $\mu_z(t)$  wieder erhält, ohne dass die Zeit die Unschärfe des Fakt beeinflusst. Die Beziehung zwischen Fakt und Zeit wird durch das temporales Schließen gebildet. Auch bei zeitlich oder faktisch eingeschränkten Fuzzy-Zeit-Objekten gilt dieser Zusammenhang, wobei Fuzzy-Zeit-Objekte nicht gemischt werden dürfen. Die Fuzzy-Zeit-Objekte sind nicht intuitiv verwendbar und erlauben keine **UND** beziehungsweise **ODER** Verknüpfungen von Fuzzy-Zeit-Objekten, aber es ist möglich ein Fuzzy-Zeit-Objekt in einer Folgerung zu verwenden. Dieser Ansatz reicht noch nicht ganz um unsere geforderte Mächtigkeit, beliebige Verwendung von zeitlichen Abhängigkeiten in Bedingung und Folgerung einer Fuzzy-Regel, zu erfüllen.

In [Schmidt04] wird gezeigt, dass es möglich ist, die Fuzzy-Logik mit temporalen Prädikaten so zu erweitern, dass sich daraus die so genannte *temporale Fuzzy-Logik* ergibt. Diese ermöglicht die Modellierung von zeitlichen Abhängigkeiten von Ereignissen und kann in einem Fuzzy-Regler zur Überwachung, Regelung und Wartung eingesetzt werden. Dieser Einsatz ist beispielhaft in einem Wartungsbeispiel gezeigt, in welchem ein Benutzer über defekte Lampen informiert wird. Außerdem wird im Hauptteil die mathematische Basis zu den temporalen Fuzzy-Prädikaten geschaffen. Die temporalen Fuzzy-Prädikate sind eine eins-zu-eins Abbildung der temporalen Prädikate, welche nach [Karjoth87] vollständig sind, das heißt mit ihnen können Bedingungen über den gesamten Zeitbereich erstellt werden. Somit sind die temporalen Fuzzy-Prädikate ebenfalls vollständig. Des weiteren kann deren Auswertung effizient ausgeführt werden. Es ist möglich im Bedingungsteil einer Regel **AND** beziehungsweise **OR** verknüpfte Prädikate zu verwenden. Wie die Auswirkung von den angegebenen Zeiten auf die Schlussfolgerungen sind, war zu diesem Zeitpunkt noch nicht bekannt, denn der Einfluss der Zeitangaben der Regelbedingung auf die Regelfolgerung

war noch nicht untersucht, aber er ist dennoch zu beachten. Diese Lücke wird mit dieser Arbeit geschlossen.

In [Lamine01] wird die Lineare Temporal Logic (LTL) genutzt um zeitliche Abhängigkeiten in Programme einzubringen. Die Syntax der LTL ist ähnlich zu dem der Programmiersprache C. Die LTL ist unscharf in der Auswertung von Bedingungen, denn es wird mit Wahrscheinlichkeiten beziehungsweise Wahrscheinlichkeitsverteilungen für ihr Eintreten gerechnet. Aber es werden keine natürlichsprachlichen Konstrukte verwendet (auch keine Fuzzy-Inferenz). Somit steht die LTL also zwischen der Temporal-Logik und der Fuzzy-Logik. Die Zeit in einer Bedingung ist durch das Prädikat **always** mit dem Zeitintervall  $[0, ?]$  gegeben. Das angegebene Zeitintervall steht für „von Jetzt (0) bis in alle Ewigkeit (?)“. Das vorgestellte Anwendungsgebiet, ist ein Roboter, welcher auf einem Straßennetz fährt, wobei die Wahrscheinlichkeitsverteilungen für den weiteren Straßenverlauf bekannt sind. Die einzelnen Regeln werden dann dazu genutzt, um in einer Regelschleife die Programmteile zu aktivieren, welche bei den aktuell vorliegenden und vermuteten zukünftigen Bedingungen mit hoher Wahrscheinlichkeit das gewünschte Ergebnis liefern. Es sind UND beziehungsweise ODER Verknüpfungen von Zeitintervallen möglich, aber es wird hierzu keine Fuzzy-Logik beziehungsweise eine Inferenz darauf verwendet, da die Zeitintervalle nur darüber entscheiden, ob eine Regel überhaupt in Betracht gezogen wird.

In [Cárdenas02] wird die Fuzzy Temporal Constraint Logic (FTCL), eine Erweiterung von Prolog mit Horn-Klauseln um Fuzzy-Logik und temporale Prädikate, präsentiert. Die temporalen Prädikate **before**, **after** und **at the same time** ermöglichen es, zeitliche Abhängigkeiten von Ereignissen untereinander auszudrücken. Die Zeit ist dabei als unscharf anzusehen. Außerdem sollen alle Ereignisse, welche zu einer gegebenen Zeit eintreten, nur mit Ereignissen im gleichen Zeitintervall verglichen werden, da Relationen zu unterschiedlichen Zeiten nicht möglich sind. Ein Beispiel, welches zeigt, dass der in [Cárdenas02] vorgestellte Ansatz nicht für ein Wartungssystem verwendet werden kann, ist ein Ereignis, welches kausal von einem anderen Ereignis abhängt. Am

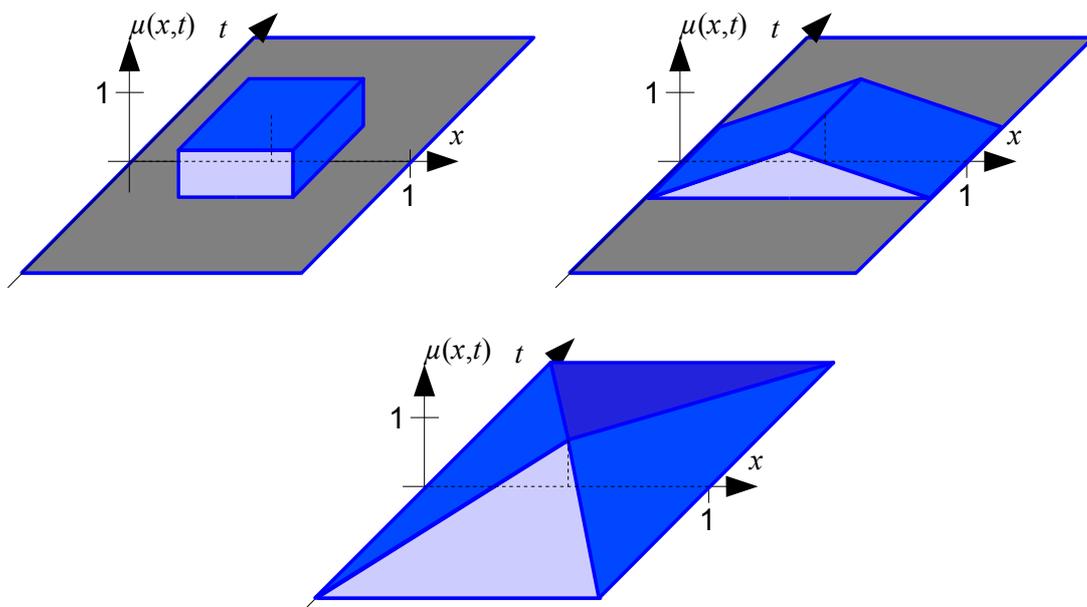


Abbildung 1: Vergleich einer scharfen und einer unscharfen Aktivierung der Funktion  $\mu(x, t)$  für einen gegebenen Fakt  $x$  zu einem Zeitpunkt  $t$ . Bei der unscharfen Aktivierung ist sowohl der Fakt als auch die Zeit unscharf.

folgenden Ereignis wird dies ersichtlich: „Gestern war es heiß und heute leckt das Rohr, dann existiert heute ein Folgeschaden durch die Hitzeeinwirkung“.

Wie an den gezeigten Beispielen zu sehen ist, gehen die verwandten Arbeiten in die von uns angestrebte Richtung, Zeit, beziehungsweise zeitliche Abhängigkeiten von Bedingungen in Fuzzy-Regler einzubinden. Jedoch gibt es noch keine Lösung, welche dies anhand von der schon ausgereiften Temporal-Logik gemacht hat. Auch wurde noch kein Fuzzy-Regler zur Wartung verwendet. Dies ist durch [Giron02] ersichtlich, wo mehr als 200 Fuzzy-Regler in mehr als 20 Kategorien klassifiziert werden. Ein Fuzzy-Regler mit temporaler Fuzzy-Logik oder ein Fuzzy-Regler als Wartungssystem existieren demnach noch nicht. Diese Entwicklung ist unser Ziel, welches wir schrittweise erreichen möchten. Ein Schritt für diese Entwicklung ist die hier vorgestellte Erweiterung der Fuzzy-Logik zur temporalen Fuzzy-Logik unter Verwendung von Fuzzy-Zeit-Termen.

### 3. Fuzzy-Zeit-Terme

#### 3.1. Semantik

Um Zeit in Fuzzy-Logik verwenden zu können, erweitern wir die Prädikate wie in [Schmidt04] beschrieben und erhalten so temporale Prädikate wie zum Beispiel **WILL-BE** oder **WAS**. Hinzu kommt, dass wir nicht nur aufgezeichnete Sensordaten aus der Vergangenheit oder vorhergesagte Sensordaten aus der Zukunft nutzen. Wir geben auch noch die Zeiten in einer fuzzymäßigen Art an. Dies kann ähnlich wie mit Fuzzy-Termen geschehen. Nur steht dann auf der  $x$ -Achse nicht ein Fakt sondern eine Zeitangabe. Auf der  $y$ -Achse bleibt die Zugehörigkeit – in diesem Fall zu einem Zeitpunkt und nicht zu einem Fakt – stehen. Eine solche Zugehörigkeitsfunktion nennen wir Fuzzy-Zeit-Term. Zuerst betrachten wir die Semantik dieser Makros um anschließend ihre Definition anzugeben. Abbildung 1 zeigt den Vergleich einer klassischen, scharfen Zugehörigkeitsfunktion für einen Fakt  $x$  und die Zeit  $t$  und einem Fuzzy-Zeit-Term mit unscharfem Fakt  $x$  und unscharfer Zeit  $t$ . In der klassischen Variante kann die Funktion sowohl für den Fakt als auch für die Zeit nur den Wert 0 oder 1 annehmen während das unscharfe Pendant überall Werte zwischen einschließlich 0 und 1 annehmen kann.

Fuzzy-Zeit-Terme können in einer Fuzzy-Regel an drei Stellen eingesetzt werden. Zuerst neben Prädikaten im Bedingungsteil einer Regel. In der Regel werden die Prädikate je nach ihrem Verwendungszweck auf alle vorhandenen Sensordaten in der Vergangenheit, Gegenwart oder Zukunft angewendet. Mit einem Fuzzy-Zeit-Term kann dieser Zeitraum jedoch eingeschränkt werden. dadurch werden die Prädikate nur auf Sensordaten aus einem ausgewählten Zeitraum angewendet. Zusätzlich wird das Ergebnis des Prädikates  $P(x)$  für jeden Sensorwert  $x$  zu einer gegebenen Zeit  $t$  mit dem Zugehörigkeitsgrad  $\mu(t)$  des Fuzzy-Zeit-Term multipliziert. So kann ein Fuzzy-Zeit-Term auch

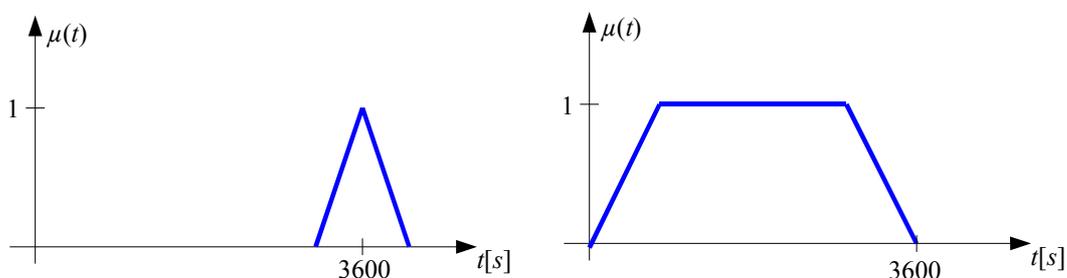


Abbildung 2: Fuzzy-Zeit-Term für das zukünftige Zeitintervall „in einer Stunde“ und „während der nächsten Stunde“. Die Zugehörigkeit nimmt jeweils an den Rändern der Intervalle linear ab.

dazu verwendet werden Sensordaten zu bestimmten Zeiten mehr oder weniger zu gewichten. Wie im Beispiel in Abbildung 2 zu sehen ist, können die Räder der Intervalle weniger gewichtet sein als deren Zentren.

Die zweite Stelle, an welcher Fuzzy-Zeit-Terme verwendet werden können, ist als alleinstehende Folgerung im Folgerungsteil einer Regel. Dort werden die Fuzzy-Zeit-Terme je nach der Regelaktivierung generiert. Daher repräsentieren sie für eine Regel den Intensitätsverlauf über die Zeit. Aus diesem Grund macht es auch nur Sinn die alleinstehenden Fuzzy-Zeit-Terme bei Regeln zu verwenden, welche in ihren Bedingungen zeitliche Prädikate verwenden. Eine Regel, welcher nur die aktuellen Sensordaten zu Grunde liegen, wird immer nur zum aktuellen Zeitpunkt feuern. Betrachten wir als Beispiel die Regel „**IF**  $x$  **WILLBE** *condition* **THEN**  $time$ “. Erhält man für  $time$  das linke Makro aus Abbildung 2, so besagt dies, dass die Regelbedingung mit den aktuellen Vorhersagen in etwa einer Stunde kurz zu 100% wahr sein wird und in etwas mehr beziehungsweise weniger als einer Stunde weniger als zu 100% wahr sein wird.

Nun kann man analog zu Ausgabevariablen, die auch als Eingabevariablen für andere Regeln verwendet werden können, auch den Intensitätsverlauf einer Regel als Eingabe einer anderen Regel verwenden.

Die dritte und letzte Stelle um Fuzzy-Zeit-Terme einzusetzen ist in dem Folgerungsteil einer Regel in Verbindung mit einem Prädikat. Zum Beispiel die Folgerung „ $x_s$  **WILLBE** *in\_one\_hour low*“ setzt die Ausgabevariable für den Aktuator  $x_s$  auf *low* mit der berechneten Regelaktivierung. Aber der Wert wird nur in dem durch den Fuzzy-Zeit-Term *in\_one\_hour* gegebenen Zeitintervall gesetzt. Zur Berechnung wird die Regelaktivierung zu einem Zeitpunkt  $t$  mit der Zugehörigkeit  $\mu(t)$  des Fuzzy-Zeit-Terms multipliziert.

Der Zeitpunkt  $T_0$  liegt in der Zukunft, aber für den Fuzzy-Regler ist  $T_0$  die Gegenwart. Wie in Abbildung 3 gezeigt, kann es deshalb auch Sinn machen, in den Folgerungen Fuzzy-Zeit-Terme in der Vergangenheit anzugeben. Denn wenn der Regler sich mit  $T_0$  in der Zukunft befindet und auf vorhergesagten Daten rechnet, ist der aktuelle Zeitpunkt  $T$  des Prozesses für den Regler in der Vergangenheit. Man kann nun Einfluss auf den Prozess im Zeitintervall  $[T, T_0]$  nehmen.

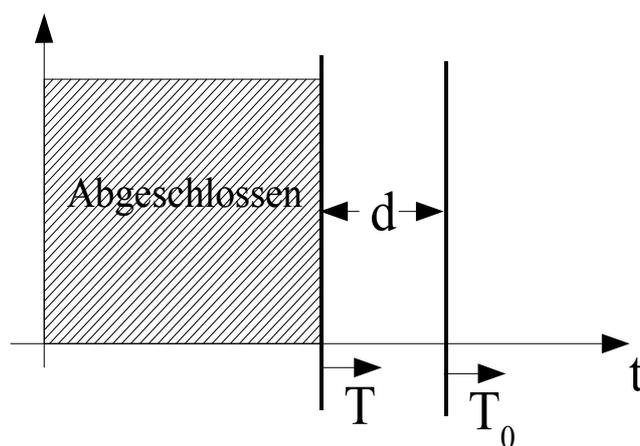


Abbildung 3: Regelung in der Zukunft zum Zeitpunkt  $T_0$ , welcher vom aktuellen Zeitpunkt  $T$  um  $d$  in der Zukunft liegt. Dadurch kann die Regelung Ausgaben in der Vergangenheit haben, welche aber für den zu regelnden Prozess noch in der Zukunft liegen.

### 3.2. Syntax

Im Folgenden ist die Definition von zeitlichen Fuzzy-Termen und deren zeitlichen Zugehörigkeitsfunktionen, den sogenannten Fuzzy-Zeit-Termen gegeben. Ein zeitlicher Fuzzy-Term ist ähnlich wie ein Fuzzy-Term für einen Fakt definiert, nur ist der Fakt  $x$  durch die Zeit  $t$  ersetzt. Er ist der Einfachheit wegen ein Polygonzug mit  $n$  Stützstellen, welcher durch die Punkte  $(t_i, y_i)$  des Polygonzugs der Länge  $n$  dargestellt wird. In den meisten Fällen wird  $n = 3$  oder  $n = 4$  und  $y_0 = y_{n-1} = 0$  gelten und so, mit dem zeitlichen Fuzzy-Term, ein gleichschenkliges Dreieck beziehungsweise ein Trapez dargestellt. Da ein Fuzzy-Term beliebig viele Stützstellen  $n$  besitzen kann, können beliebige stetige Funktionen durch lineare Interpolation approximiert werden. Ein zeitlicher Fuzzy-Term ist definiert als

$$ft = \{(t_i, y_i) | t_i < t_j, 0 \leq y_i \leq 1, i < j, 0 \leq i, j < n\}.$$

Der Fuzzy-Zeit-Term interpoliert nun die Stützstellen des zeitlichen Fuzzy-Terms  $ft$  linear und liefert so zu jedem Zeitpunkt einen Zugehörigkeitswert zwischen 0 und 1. Dabei ist die Zeit immer relativ zur aktuellen Zeit  $t_c$  angegeben. Wenn  $t < 0$  ist, dann liegt der Zeitpunkt in der Vergangenheit während  $t > 0$  in der Zukunft liegt. Die Gegenwart entspricht  $t = 0$ . Die Zugehörigkeitsfunktion eines Fuzzy-Zeit-Terms ist wie folgt definiert:

$$\mu_{ft}(t) = \begin{cases} y_i + (y_{i+1} - y_i) \cdot \frac{(t - t_i)}{(t_{i+1} - t_i)} & , t_i \leq t < t_{i+1} \\ y_0 & , t < t_0 \\ y_{n-1} & , t \geq t_{n-1} \end{cases}$$

Außer der bis jetzt vorgestellten Verwendung, können die Fuzzy-Zeit-Terme auch noch zu anderen Zwecken verwendet werden. Zum Beispiel in Verbindung mit dem im weiteren Verlauf erläuterten Prädikat **COUNT**. Mit dem Prädikat **COUNT** kann überprüft werden, wie oft ein Fuzzy-Zeit-Term aktiviert wurde. Als Anzahl der Aktivierungen zählen in diesem Falle die Peaks, welche einen definierten Schwellwert überschreiten. Wird der Schwellwert überschritten, so ist dies der Anfang eines Peaks. Wird der Schwellwert zu einem späteren Zeitpunkt wieder unterschritten, so ist dies das Ende eines Peaks. Die Anzahl der gefundenen Peaks wird dann mit einem Fuzzy-Term verglichen und die Zugehörigkeit zu diesem Fuzzy-Term gibt die Aktivierung der Bedingung an.

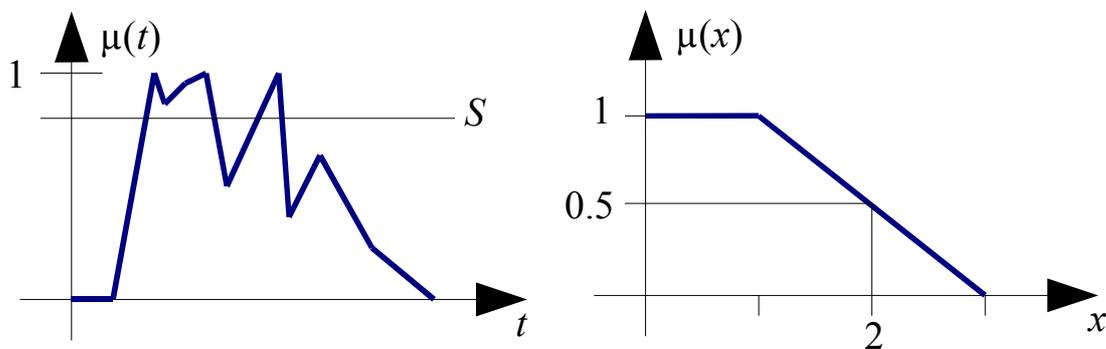


Abbildung 4: Fuzzy-Zeit-Term „time“ mit Schwellwert  $S = 80\%$  und zwei Peaks (links) und Fuzzy-Term „none“ mit einer Aktivierung von  $50\%$  bei  $x = 2$  (rechts).

Als Beispiel hierfür sei die Bedingung „*time COUNT none*“ gegeben. Hier werden die Peaks in *time* gezählt. Je nachdem, wie der Fuzzy-Term *none* definiert ist, wird die Aktivierung der Bedingung berechnet. In Abbildung 4 ist ein solches Beispiel gegeben. Das linke Bild zeigt ein Fuzzy-Zeit-Term, welcher in der Folgerung einer Regel gesetzt wurde. Bei einem Schwellwert von  $S = 80\%$  gibt es zwei Peaks. Das Rechte Bild zeigt den Fuzzy-Term *none*. In diesem Fall ergibt sich für  $x = 2$  eine Zugehörigkeit von 50%. Damit ist die Bedingung „*time COUNT none*“ zu 50% aktiviert.

#### 4. Fuzzy-Regelung mit Fuzzy-Zeit-Termen

Dadurch, dass es in der temporalen Fuzzy-Logik möglich ist zeitliche Einschränkungen bei den Prädikaten zu verwenden, ändert sich dementsprechend auch die Art, wie die Fuzzy-Regelung durchzuführen ist. In diesem Abschnitt wird nun gezeigt, wie sich dieser Einfluss bemerkbar macht. Vergleiche hierzu Abbildung 5 und 6.

Als Beispiel dienen je zwei Regeln ohne und mit Fuzzy-Zeit-Termen. Die Regeln ohne Fuzzy-Zeit-Term sind dabei wegen der Einfachheit in der Gegenwart formuliert, während die Regeln mit Fuzzy-Zeit-Term auf Sensordaten in der Zukunft arbeiten und die Ausgabe für einen Zeitpunkt in der Zukunft berechnen.

Erster Fall, Regeln ohne Fuzzy-Zeit-Terme

**IF**  $a$  **IS** *dunkel* **THEN**  $N_s = \textit{wenig}$

**IF**  $d$  **IS** *hell* **THEN**  $N_s = \textit{viel}$

und zweiter Fall, Regeln mit Fuzzy-Zeit-Termen

**IF**  $a$  **WILLBE** *in\_einer\_stunde dunkel* **THEN**  $N_s$  **WILLBE** *in\_einer\_stunde wenig*

**IF**  $d$  **WILLBE** *nächste\_stunde hell* **THEN**  $N_s$  **WILLBE** *nächste\_stunde viel*

Abbildung 5 zeigt die Fuzzifizierung der zwei Variablen  $a$  und  $d$  während Abbildung 6 die Fuzzifizierung der Variablen  $a$  und  $d$  mit Fuzzy-Zeit-Termen zeigt. Im ersten Fall interessieren wir uns wie dunkel  $a$  beziehungsweise wie hell  $d$  zum aktuellen Zeitpunkt sind. Im letzteren Fall sind wir jedoch nur daran interessiert, wie  $a$  und  $d$  in einer Stunde beziehungsweise in der nächsten Stunde sein werden. Im Beispiel ist  $a$  zu etwa 50% *dunkel* beziehungsweise wird  $a$  in einer Stunde zu 25% *dunkel* sein. die Variable  $d$  dagegen ist aktuell zu 25% *hell* und wird während der nächsten Stunde zu 50% *hell* sein.

Die Inferenz berechnet aus den Aktivierungen der Bedingungen einer Regel die Regelaktivierung, und daraus die Aktivierungen der Ausgabevariablen. Ob nun Fuzzy-Zeit-Terme verwendet werden oder nicht spielt für das Ergebnis keinerlei Rolle bei der Inferenz. Aus diesem Grund beinhaltet das Beispiel auch keine Und- ( $\wedge$ ) beziehungsweise Oder-Verknüpfungen ( $\vee$ ). Da die Regeln jeweils nur eine Bedingung beinhalten, entsprechen die Prozentangaben der oben genannten Aktivierungen auch den Regelaktivierungen. Das heißt im ersten Fall ist *wenig* zu 50% und *viel* zu 25% aktiviert. Im zweiten Fall dagegen ist *wenig* zu 25% und *viel* zu 50% aktiviert. Zu beachten ist jedoch, dass wir uns zu jeder Regel merken, wann diese wie stark aktiviert wurde.

Die Komposition berechnet für jede Ausgabevariable – hier gibt es nur eine mit dem Namen  $N_s$  – eine Zugehörigkeitsfunktion je nachdem wie stark die einzelnen Fuzzy-Terme aktiviert sind. Im ersten Fall erhalten wir eine Zugehörigkeitsfunktion  $\mu(N)$ , welche nur vom Fakt  $N$  abhängt, während wir im zweiten Fall eine Zugehörigkeitsfunktion  $\mu(N, t)$  erhalten, welche neben dem Fakt  $N$  auch noch von der Zeit  $t$  abhängt.

Für die Defuzzifizierung nutzen wir die weit verbreitete Schwerpunktmethodene aus [Watanabe86]. So berechnet sich im ersten Fall die Ausgabe  $N_s$  durch ermitteln des Schwer-

IF  $a$  IS *dunkel* THEN  $N_s = \text{wenig}$   
 IF  $d$  IS *hell* THEN  $N_s = \text{viel}$

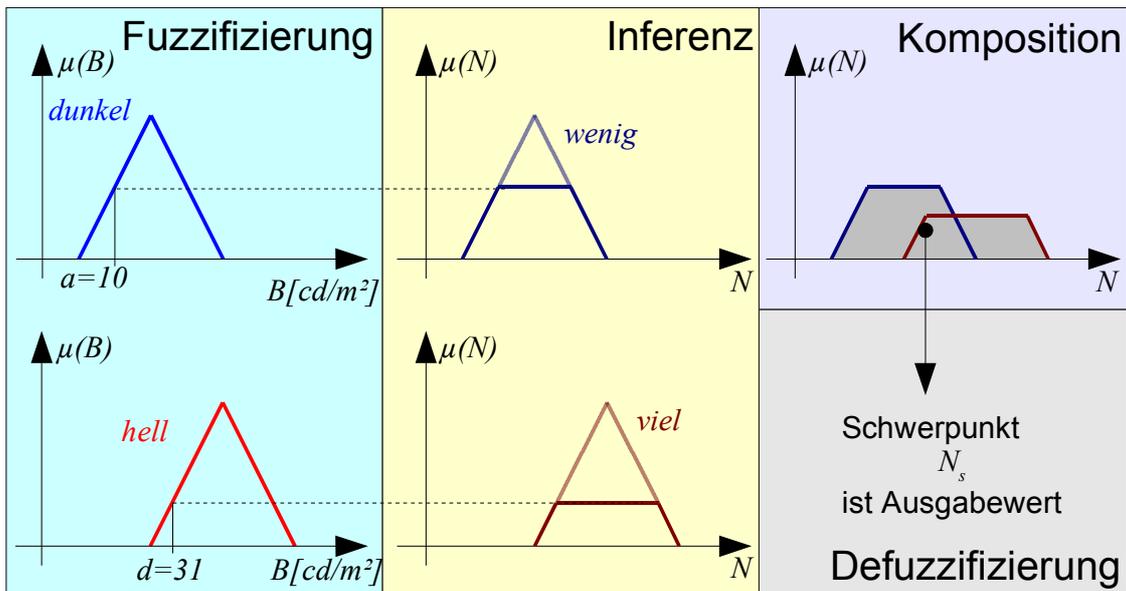


Abbildung 5: Beispiel für zwei Regeln mit zwei Eingabevariablen  $a$  und  $d$  und einer Ausgabeveriable  $N_s$ . Die Variablen werden fuzzifiziert und die Regeln über die Inferenz beziehungsweise die Komposition ausgewertet um anschließend durch die Defuzzifizierung den Ausgabewert  $N_s$  zu erhalten. Die Eingabevariablen sind auf  $a = 10$  und  $d = 31$  gesetzt.

IF  $a$  WILLBE in\_einer\_stunde *dunkel* THEN  $N_s$  WILLBE in\_einer\_stunde *wenig*  
 IF  $d$  WILLBE nächste\_stunde *hell* THEN  $N_s$  WILLBE nächste\_stunde *viel*

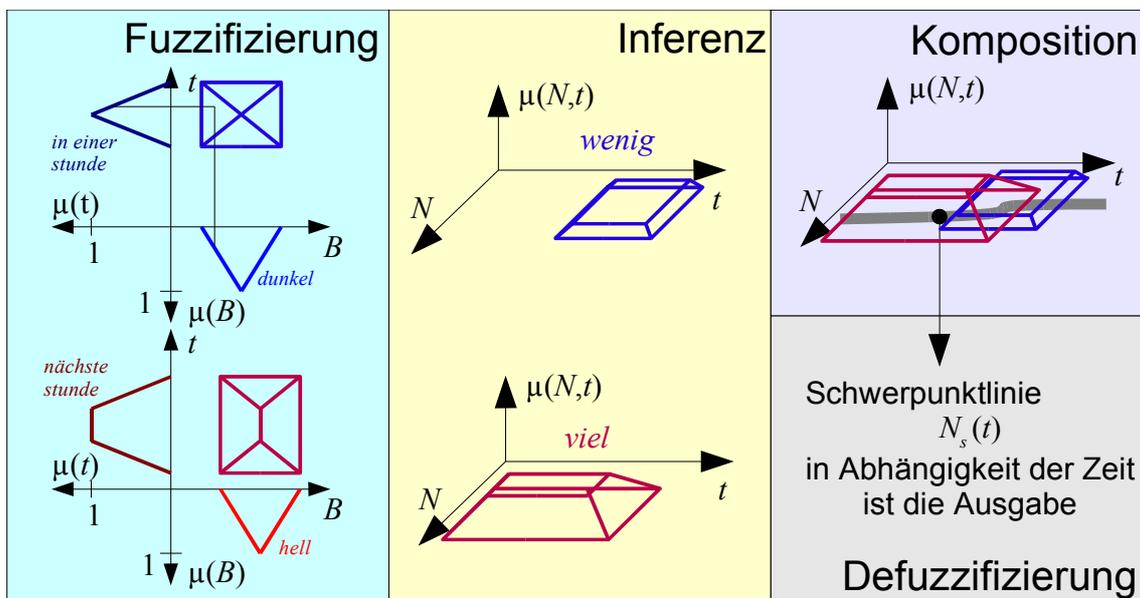


Abbildung 6: Beispiel für zwei temporale Regeln mit zwei Eingabevariablen  $a$  und  $d$  und einer Ausgabefunktion  $N_s(t)$ . Die Variablen werden fuzzifiziert und die Regeln über die Inferenz beziehungsweise die Komposition ausgewertet um anschließend durch die Defuzzifizierung die Ausgabefunktion  $N_s(t)$  zu erhalten. Die Eingabevariablen sind über dem gesamten Zeitraum auf die konstanten Werte  $a(t) = 10$  und  $d(t) = 31$  gesetzt.

punktes der Zugehörigkeitsfunktion  $\mu(N)$ . Im zweiten Fall wird nicht etwa der Schwerpunkt der Funktion  $\mu(N, t)$  bestimmt, denn dann würde das Ergebnis ein Skalar und nicht mehr abhängig von der Zeit sein. Vielmehr wird für jeden Zeitpunkt  $t_i$  der Schwerpunkt  $N_s(t_i)$  aus der Zugehörigkeitsfunktion  $\mu(N, t) | t = t_i$  berechnet. So erhält man die Schwerpunktsgerade  $N_s(t)$ , welche zu jedem Zeitpunkt einen Ausgabewert liefert. So kann zum Beispiel zum aktuellen Zeitpunkt schon das Regelverhalten für die Zukunft festgelegt werden.

## 5. Schlussfolgerung

Wir haben gezeigt, dass die in [Schmidt04] vorgestellten temporalen Fuzzy-Prädikate in Verbindung mit den hier vorgestellten Fuzzy-Zeit-Termen eine Entwicklung ist, welche die Formulierung der Bedingungen in Fuzzy-Regeln noch natürlichsprachlicher macht. Dadurch wird nicht nur das Übersetzen von Expertenwissen nach temporaler Fuzzy-Logik einfacher, auch wird die temporale Fuzzy-Logik wartbarer da sie weniger kryptisch ist.

Es gibt zwischen dem hier vorgestellten Modell und dem aus [Schmidt04] noch weitere Unterschiede. Um diese anzugeben, betrachten wir die erste Bedingung „ $x$  WILLBE [3550, 3650] low“ mit einem Intervall für die Zeitangabe im Vergleich zu der zweiten Bedingung „ $x$  WILLBE in\_one\_hour low“ mit einem Fuzzy-Zeit-Term.

Erstere Bedingung liest sich deutlich schwerer als die Zweite, denn man muss wissen, ob die Angabe der Zeit in Sekunden oder einer anderen Einheit ist. Durch das Verwenden eines natürlichsprachlichen Makros ist dies jedoch sofort klar. Auch widersprechen die Zeitangaben durch ein scharf begrenztes Intervall dem Wesen der Fuzzy-Logik. Es handelt sich dabei nämlich um eine scharfe Zeitangabe. Dagegen ist die Zeitangabe über ein Fuzzy-Zeit-Term unscharf. Die Intervallgrenzen verwischen dabei genauso wie bei den unscharfen Fuzzy-Termen, welche auch durch Zugehörigkeitsfunktionen beschrieben werden.

Des weiteren können mit den Fuzzy-Zeit-Termen auch die Zeiten protokolliert werden, zu welchen eine Regel mit den aufgezeichneten und vorhergesagten Sensordaten feuert. Diese protokollierten Zeiten können dann verwendet werden um das Aktivierungsprofil von anderen Bedingungen zu beeinflussen.

## 6. Literatur

- [Althoff92] K.-D. Althoff, *Eine fallbasierte Lernkomponente als integrierter Bestandteil der MOLTKE-Werkbank zur Diagnose technischer Systeme*, Dissertation, Kaiserslautern, September 1992
- [Bothe95] H.-H. Bothe, *Fuzzy-Logik – Einführung in Theorie und Anwendungen*, Springer-Lehrbuch, 2. Auflage, Berlin Heidelberg, 1995
- [Bovenkamp98] E. G. P. Bovenkamp, J. C. A. Lubbe, *Temporal Reasoning with Fuzzy-Time-Objects*, 4<sup>th</sup> 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
- [Castillo02] O. Castillo, P. Melin, *A New Approach For Plant Monitoring Using Type-2 Fuzzy Logic and Fractal Theory*, Proceedings of the 5th International FLINS Conference, Belgium, Belgium, September 2002
- [Fantoni00] P.F. Fantoni, M. Hoffmann, B. H. Nystad, *Integration of sensor validation in modern control room alarm systems*, Proceedings of the 4th International FLINS Conference, Belgium, August 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
- [Flender02] Flender Service GmbH, *Condition Monitoring for the highest Availability of Power Technology*, [http://www.flender-cm.de/images/pdf-files/Leistungskurzbeschreibung\\_GB.pdf](http://www.flender-cm.de/images/pdf-files/Leistungskurzbeschreibung_GB.pdf), Juni 2002
- [Giron02] J. M. Giron-Sierra, G. Ortega, *A Survey of Stability of Fuzzy Logic Control with Aerospace Applications*, IFAC Proceedings of the 15<sup>th</sup> Triennial World Congress, Barcelona, Spain, 2002
- [Haslum01] P. Haslum, *Models for Prediction*, IJCAI 2001 workshop on Planning under Uncertainty
- [IEC97] IEC TC65/WG 7/TF8, *Fuzzy Control Programming*, International Technical Electrical Commission (IEC), 1997
- [Karjoth87] G. Karjoth, *Prozeßalgebra und temporale Logik – angewandt zur Spezifikation und Analyse von komplexen Protokollen*, Diss. Mathematik/Informatik, Universität Stuttgart, 1987
- [Lamine01] K. B. Lamine, F. Kabanza, *Reasoning About Robot Actions: A Model Checking Approach*, Revised Papers from the International Seminar on Advances in Plan-Based Control of Robotic Agents, Springer Verlag, 2001
- [Palit00] A. K. Palit, *Artificial Intelligent Approaches to Times Series Forecasting*, Dissertation, Bremen, Januar 2000
- [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
- [Škrjanc02] I. Škrjanc, D. Matko, *Fuzzy Predictive Functional Control in the State Space Domain*, *Journal of Intelligent and Robotic Systems*, Kluwer Academic Publishers, 2002
- [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.
- [Watanabe86] H. Watanabe, *Schwerpunktmethode*

# Modellierung hochdimensionaler Zusammenhänge in einer Dampferzeugerfeuerung mittels künstlicher neuronaler Netze

**T. Förster; W. Kästner; R. Hampel  
U.-S. Altmann; S. Grusla**  
Hochschule Zittau/Görlitz (FH)  
Institut für Prozeßtechnik,  
Prozeßautomatisierung und  
Meßtechnik (IPM)

Theodor-Körner-Allee 16  
02763 Zittau  
**Tel.:** +49(0)3583 61 1547  
**Fax:** +49(0)3583 61 1288  
**E-Mail:** [ipm@hs-zigr.de](mailto:ipm@hs-zigr.de)

**M. Rubik**  
Vattenfall Europe  
Powerconsult GmbH

Kraftwerksstraße 22  
03226 Vetschau/Spreewald  
+49(0)35433 75 2812  
+49(0)35433 75 2731  
[mario.rubik@vattenfall.de](mailto:mario.rubik@vattenfall.de)

**D. Haake; T. Brunne**  
Vattenfall Europe  
Generation AG \& Co. KG

Vom-Stein-Straße 39  
03050 Cottbus  
+49(0)2887 3736/3033  
+49(0)2887 3186  
[dietmar.haake@vattenfall.de](mailto:dietmar.haake@vattenfall.de)  
[thomas.brunne@vattenfall.de](mailto:thomas.brunne@vattenfall.de)

## 1 Einleitung und Motivation

Für die Modellierung komplexer technischer Systeme stellen künstliche neuronale Netze (KNN) [1] im Vergleich zu herkömmlichen mathematischen Methoden eine effiziente Modellierungsmethode dar. Hierbei kommt in diesem Artikel als spezielles KNN das Multilayer Perzeptron (MLP) zum Einsatz. Ziel ist es, signifikante Zusammenhänge zwischen den Prozeßgrößen einer Dampferzeugerfeuerung nachzubilden. Im Ergebnis der Modellierung entstand ein MLP, welches den Zusammenhang zwischen ausgewählten Ein- und Ausgangsgrößen im 57-dimensionalen Zustandsraum mit guter bis sehr guter Güte abbildet.

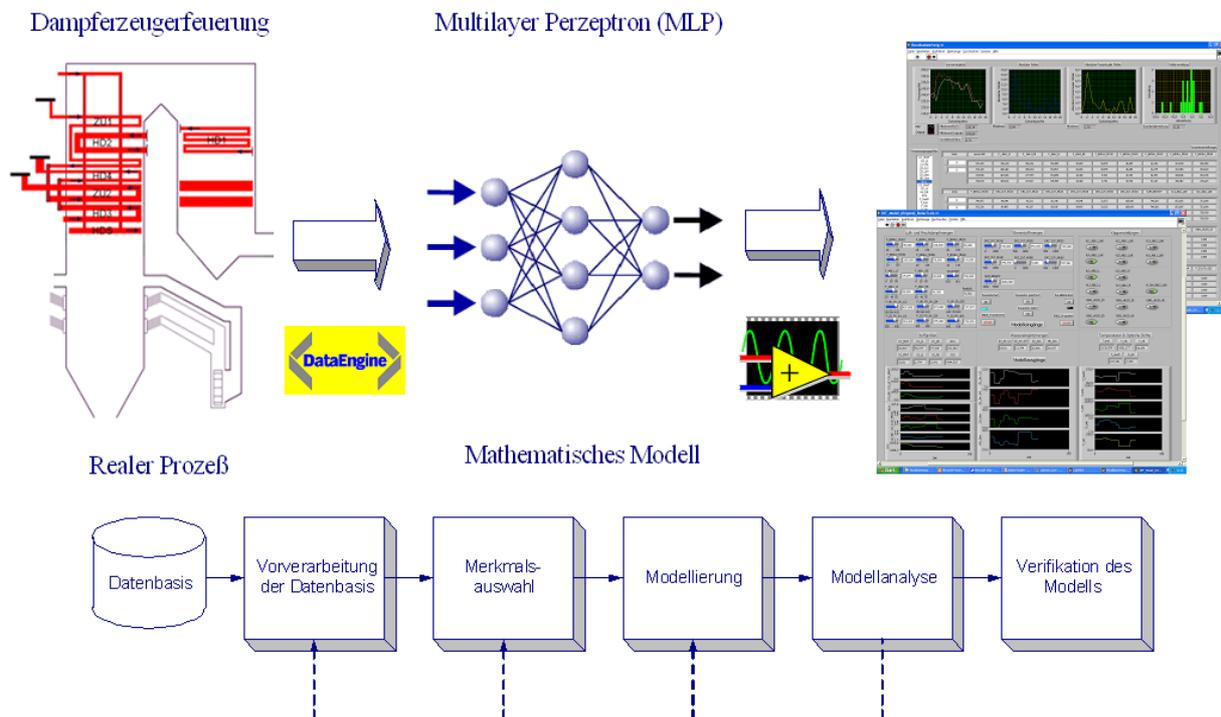


Abbildung 1-1: Einsatz von MLP zur Dampferzeugerfeuerung

## 2 Datenvorverarbeitung und Merkmalsauswahl

Den Ausgangspunkt der Modellierung bildete eine Datenbasis mit 337 Merkmalen und 30.238 Mustern. Eine qualitative Plausibilitätsprüfung, wissensbasierte Merkmalsreduktion, sowie eine Korrelationsanalyse [2] reduzierten die signifikanten Zusammenhänge auf 40 Eingangsgrößen und 17 Ausgangsgrößen. Als Eingangsgrößen wurden Merkmale gewählt, welche den Prozeßzustand sowie Stell- und Regeleingriffe charakterisieren. Als Ausgangsgrößen wurden Merkmale modelliert, welche sowohl den Wirkungsgrad als auch die Schadstoffemission beschreiben. Da sich die Abstraten der Merkmale teilweise unterschieden, wurde über die Zeitreihen der Einzelmerkmale eine kubische Splinefunktion gelegt [3]. Danach erfolgte die Bereinigung des Datensatzes um die Stillstands- und Ausfallzeiten einzelner Meßstellen. Letztlich entstand ein Datensatz mit 57 Merkmalen und 16.136 Mustern, welcher Lern- und Recalldaten beinhaltet.

## 3 Netzentwurf

Für die Optimierung der Netztopologie wurde das Weight-Pruning-Verfahren verwendet [4]. Ausgangspunkt ist ein voll vernetztes MLP mit überdimensionierten verdeckten Schichten in denen einzelne Gewichte  $w_{ij}$  aufgrund ihrer geringen Relevanz  $r_{ij}$  eliminiert wurden. Anschließend fand eine Sensibilitätsanalyse der Trainingsparameter statt. Die besten Ergebnisse erzielte letztlich eine parallele Netzstruktur von zwei MLP wobei das erste 3 und das zweite 14 Ausgangsmerkmale modellierte.

## 4 Modellgüte

Zur Bewertung der Modellgüte kamen 16 mathematische Kenngrößen wie z.B. betragsmäßiger absoluter Fehler, Konfidenzintervall oder betragsmäßiger prozentualer Fehler über dem Wertebereich zum Einsatz. Im Ergebnis besitzt das Modell mit paralleler Netzstruktur 14 Ausgangsmerkmale mit sehr guter und 3 Ausgangsmerkmale mit ausreichender Güte, wobei diese 3 Merkmale auf eine einzige physikalische Größe zurückzuführen sind. Die Hauptursache für die Abweichungen wurden lokalisiert und liegen im prozeßtechnischen Bereich (zeit- und ortsabhängige Effekte der Verbrennung).

Die Untersuchungen hinsichtlich der Plausibilität sowie der Generalisierungsfähigkeit des Modells wurden anhand des erstellten Recalldatensatzes realisiert und führten zu guten Ergebnissen.

## Literatur

- [1] A. Zell: "Simulation Neuronaler Netze", Addison-Wesley Publishing Company, Bonn, 1994
- [2] R. Storm: „Wahrscheinlichkeitsrechnung, mathematische Statistik und statistische Qualitätskontrolle“ Leipzig, Köln, Fachbuchverlag, 1995
- [3] H.-J. Bartsch: „Taschenbuch mathematischer Formeln“, Fachbuchverlag Leipzig im Carl Hanser Verlag, 1999
- [4] R. Reed: "Pruning algorithms a survey", IEEE Trans. On neural networks, Volume 4, Number 5, 1993, pp. 707-740

