

R. Mikut, M. Reischl (Hrsg.)

Institut für Angewandte Informatik /
Automatisierungstechnik



Proceedings

18. Workshop Computational Intelligence

Dortmund,
3. - 5. Dezember 2008



R. Mikut, M. Reischl (Hrsg.)

Proceedings 18. Workshop Computational Intelligence

Dortmund, 3.-5. Dezember 2008

Schriftenreihe des
Instituts für Angewandte Informatik / Automatisierungstechnik
an der Universität Karlsruhe (TH)
Band 24

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

Proceedings

18. Workshop Computational Intelligence

Dortmund, 3.-5. Dezember 2008

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 2008
Print on Demand

ISSN: 1614-5267
ISBN: 978-3-86644-282-5

Vorwort

Dieser Tagungsband enthält die Beiträge des 18. 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 3. – 5. Dezember 2008 im Haus Bommerholz bei Dortmund stattfindet.

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 PD Dr. R. Mikut (Forschungszentrum Karlsruhe GmbH), Prof. Dr. H.-G. Beyer (FH Vorarlberg), Dr. F. Hoffmann (TU Dortmund), Prof. Dr. A. Kistner (Universität Stuttgart), Prof. Dr. F. Klawonn (FH Braunschweig/Wolfenbüttel), Prof. Dr. A. Kroll (Universität Kassel) und PD Dr. Th. Runkler (Siemens AG).

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

Nähere Informationen zum GMA-Fachausschuss erhalten Sie unter

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

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

Ralf Mikut und Markus Reischl

Inhaltsverzeichnis

C. Stiller, I. Lulcheva (Universität Karlsruhe): Probabilistische Inferenz für Kognitive Automobile	1
R. Mustafa, T. Goernig, F. Svaricek (Universität der Bundeswehr, München, Continental Automotive Distance Control Systems GmbH, Lindau): Adaptive Fuzzy Path Planning for Semi–Autonomous Parking Maneuvers using a CV–Sensor	15
G. Rudolph, M. Preuss (Technische Universität Dortmund): Ein Evolutionsverfahren zur Approximation äquivalenter Urbilder von Pareto- optimalen Zielvektoren	30
C. Bauer, S. Braun, Y. Chen, W. Jakob, R. Mikut (Forschungszentrum Karlsruhe GmbH): Optimization of Artificial Central Pattern Generators with Evolutionary Algorithms	40
O. Kramer (Technische Universität Dortmund): Self-Adaptation in Constrained and Combinatorial Search Spaces	55
J. Krettek, J. Braun, F. Hoffmann, T. Bertram (Technische Universität Dortmund): Benutzerpräferenzmodellierung zur interaktiven multiobjektiven evolutionären Optimierung	74
H. Schulte (Bosch Rexroth AG): Robuster Beobachterentwurf für Takagi-Sugeno Fuzzy Systeme zur Überwachung hydrostatischer Fahrtriebe	90
N. Rosemann, J. Hülsmann, W. Brockmann (Universität Osnabrück): Disrupted Learning - Lernen bei harten Zustands- oder Strukturwechseln	105
M. Schneider, E. Kraft, U. Lehmann, J. Krone, P. Langbein, C. Ament, P. Otto (Fachhochschule Südwestfalen, Technische Universität Ilmenau): Geometrische Einflüsse auf das Training von Künstlichen Neuronalen Netzen bei nichtlinearem Systemverhalten	118
L. Schreiber, H. Sommer, A. Kroll (Universität Kassel): Lösung komplexer Probleme mittels heuristischer Verfahren	124

T. Bartz-Beielstein, T. Zimmer, W. Konen (Fachhochschule Köln): Parameterselektion für komplexe Modellierungsaufgaben der Wasserwirtschaft	136
M. Busch, B. Schweizer, H. Sommer, V. Sommer (Universität Kassel): Ein einfaches Berechnungsverfahren zu den Navier-Stokesschen Gleichungen – Die Struktur berechenbarer Modelle für komplexe Systeme	151
U. Priber, T. Päßler (Fraunhofer-Institut für Werkzeugmaschinen und Umformtechnik, Chemnitz): Fuzzy-basiertes System zur Optimierung eines Reglersystems für die Niederhalterkraft beim Tiefziehen	163
P. Gerland, H. Schulte, A. Kroll (Universität Kassel, Bosch Rexroth AG): Wahrscheinlichkeitsbasierte globale Situationserkennung bei komplexen Systemen	174
R. Mikut, M. Reischl, A. Ulrich, K. Hilpert (Forschungszentrum Karlsruhe GmbH): Data-Based Activity Analysis and Interpretation of Small Antibacterial Peptides	189
J. Beyer, K. Heesche, W. Hauptmann, C. Otte (Siemens AG, München, Otto-von-Guericke-Universität, Magdeburg): Combined Knowledge-based and Data-driven Modeling by Heterogeneous Mixture-of-Experts	204
C. Moewes, R. Kruse (Otto-von-Guericke-Universität, Magdeburg): Adjusting Monitored Experiments to Real-World Cases by Matching Labeled Time Series Motifs	214
E. Kraft, M. Schneider, U. Lehmann, J. Krone (FH Südwestfalen): Mehrdimensionales Systemverhalten mit evolutionär verbesserten neuronalen Netzen modelliert	224
T. Runkler (Siemens AG, München): A Single-Objective Fuzzy Clustering Model for Very High-Dimensional Data	234
C. Arnold, S. Nicolai, P. Bretschneider (Fraunhofer Anwendungszentrum Systemtechnik, Ilmenau): Anwendungsfälle für elektrische Energiespeicher und deren Dimensionierung mit Hilfe von Datenanalysen	244
S. Schäfer, S. Turck, M. Schneider, W. Roth (FH Südwestfalen): Electronic-Banking Kernel mit neuronalen Validierungsmechanismen auf Basis des FinTS 4.0 Standards	254

R. Bartz (Fachhochschule Köln): Client/Server-Architektur eines verteilten ANN mit XML-basierter Ablage	260
R. Seising (Friedrich-Schiller-Universität Jena): Daten, Information, Wissen – und die Computational Intelligence. Ein Beitrag zu 60 Jahren Informationstheorie	275
S. Hartmann, T. Runkler, R. Grothmann, J. Bamberger (Siemens AG, München): Ant Colony Optimization for Cash Distribution	276
M. Grube, R. Mikut, Th. Alber, M. Jagiella, G. Bretthauer (Forschungszentrum Karlsruhe GmbH, Endress+Hauser Conducta GmbH, Gerlingen): Ein selbstanpassender und prozessspezifischer Ansatz in der Sensordiagnose	286

Probabilistische Inferenz für Kognitive Automobile

Christoph Stiller, Irina Lulcheva

Institut für Mess- und Regelungstechnik, Universität Karlsruhe

Engler-Bunte-Ring 21, 76131 Karlsruhe

Tel.: (0721) 608-2334

Fax: (0721) 66 18 74

E-Mail: stiller, lulcheva@mrt.uka.de

Kurzfassung: Kognitive Automobile nehmen ihre Fahrumgebung sensorisch wahr, strukturieren dieses Wissen und vermögen auf der Grundlage dieser Information geeignetes Fahrverhalten zu generieren. Die Unsicherheit, die mit jedem Verarbeitungsschritt der Information verbunden ist, lässt sich durch probabilistische Modelle darstellen und durch die gesamte Kognitionskette propagieren. Über metrische Information hinaus, sind symbolisches und relationales Wissen für die Repräsentation einer Verkehrssituation von Bedeutung. Zur Inferenz dieses Wissens werden mit Markov-Logik-Netzen und Bayes-Logikprogrammen Methoden des Probabilistisch Logischen Lernens vorgestellt, die sich für Kognitive Automobile besonders eignen. Am Beispiel der Inferenz von Bewegungsbeziehungen zwischen Fahrzeugen wird die Leistungsfähigkeit der Methodik anhand realer Verkehrsdaten aufgezeigt.

1 Motivation

Im Automobilbereich gewinnen Systeme zur aktiven Vermeidung von Unfällen oder zur Minderung der Unfallschwere sowie zur Erhöhung von Komfort und Effizienz zunehmend an Bedeutung. Der Begriff *Fahrerassistenzsysteme mit maschineller Umfeldwahrnehmung* steht für ein sich rasant entwickelndes Forschungsgebiet, das mit der Einführung der sensorbasierten automatischen Abstandsregelung in den 90er Jahren begann [1].

Aktuelle Systeme unterstützen den Fahrer in unterschiedlichen Bereichen seiner Fahraufgabe. Diese umfassen eine komfortorientierte Längs- und Querassistenz, teilautomatisches Einparken und reichen bis hin zur automatischen Notbremsung in einer Gefahrensituation. In konsequenter Fortsetzung dieser Entwicklung werden mittelfristig Funktionen verfügbar sein, welche die Fahraufgabe in spezifischen Situationen vollständig übernehmen, z.B. Stau-Folgefahrt oder Kolonnen-Folgefahrt. Am Ziel der Entwicklung stehen unfallfreie Automobile, die selbst das Wissen komplexer Umgebungen strukturieren, die jeweilige Situation richtig einschätzen und dadurch zu zielgerichtetem sicheren Handeln fähig sind.

Der stetig steigende Automatisierungsgrad der Fahrerassistenz hat zur Folge, dass die Systeme zunehmend komplexeren Verkehrsszenarien ausgesetzt werden. Während sich das Systemverhalten bisheriger Systeme mit einem klassischen Zustandsautomaten abbilden lässt, verschließen sich diese bereits mittelfristig einer manuellen Erstellung. Für eine flexible, effiziente Situationsanalyse und Verhaltensentscheidung rücken explizite Methoden zur Wissensrepräsentation und Verhaltensmodellierung sowie Verfahren der künstlichen Intelligenz zunehmend in den Fokus aktueller Forschung.

Der Sonderforschungsbereich (SFB) Transregio *Kognitive Automobile* der Deutschen Forschungsgemeinschaft widmet sich der systematischen und interdisziplinären Erforschung

der maschinellen Kognition mobiler Systeme als Grundlage maschinellen Handelns. Das Kognitive Automobil soll dabei sowohl zu individuellem als auch zu kooperativem Wahrnehmen und Handeln fähig sein. Für diese Ziele haben sich Wissenschaftler der Universität, des Forschungszentrums und des Fraunhofer IITB in Karlsruhe sowie der Technischen Universität und der Universität der Bundeswehr in München miteinander vernetzt [2].

Aus der Vielfalt der Methoden der künstlichen Intelligenz kommen im Automobilbereich für sicherheitsrelevante Funktionen nur solche Ansätze in Frage, die sich einer einfachen Absicherung, Verifikation und Testbarkeit ihres Systemverhaltens öffnen. Entsprechend finden Black-Box-Verfahren, wie zum Beispiel neuronale Netze, trotz ihrer potentiellen Eignung zur Abbildung und Generierung von Systemverhalten an dieser Stelle kaum Beachtung. Probabilistische Modelle bieten sich hingegen in der automotiven Sensordatenanalyse besonders an, da sie unmittelbar die Ermittlung von Konfidenz- und Unsicherheitsmaßen ermöglichen. Um regelbasiertes Verhalten abzubilden, verwendet der vorgestellte Ansatz formale Beschreibungssysteme zur maschinellen Inferenz, d.h. Logiken. Damit werden funktionelle und relationale Beschreibungen der Verkehrsszene möglich. Lernverfahren lassen sich für die Ermittlung relevanter Regeln und deren Gültigkeit nutzen. Die gemäß Bild 1 durch Verknüpfung der einzelnen Elemente entstehenden Methoden des *probabilistisch logischen Lernens* bieten für den Diskursbereich Kognitiver Automobile attraktive Eigenschaften. So ließen sich beispielsweise im Straßenverkehr kaum deterministische Regeln formulieren, die absolut allgemeingültig sind, da nahezu stets Situationen denkbar sind, in denen einzelne Regeln gebrochen werden. Während selbst seltene Ausnahmen von Regeln klassische Logiksysteme inkonsistent machen, erlauben probabilistische Logiken die Formulierung von Regeln, die mit hoher Wahrscheinlichkeit gültig sind.

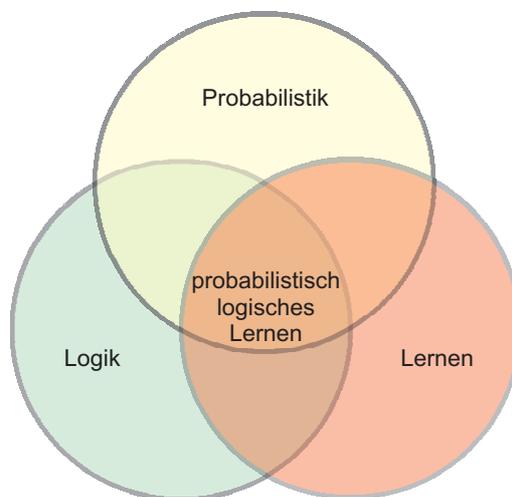


Bild 1: Probabilistisch logisches Lernen

Dieser Beitrag widmet sich entsprechend in Abschnitt 2 zunächst formalen Methoden zur Wissensrepräsentation für kognitive Automobile unter besonderer Berücksichtigung des Umgangs mit unsicherem und unvollständigem Wissen. Abschnitt 3 stellt mit *Markov-Logik-Netzen* und *Bayes-Logikprogrammen* alternative Ansätze des *probabilistisch logischen Lernens* gegenüber. Erste Ergebnisse aus realen Fahrsituationen werden in Abschnitt 4 vorgestellt.

2 Wissensrepräsentation

2.1 Ontologie

Der Diskursbereich kognitiver Automobile in einer Verkehrsszene ist von großer Komplexität geprägt. Um Wissen darin adäquat zu repräsentieren und zu kommunizieren, wird ein gemeinsames Vokabular für die Bedeutung relevanter Begriffe und deren Relationen benötigt. Dieses Vokabular kann durch eine Ontologie festgelegt werden.

Die bekannteste Definition bezeichnet eine *Ontologie* als eine “explizite formale Spezifikation einer gemeinsamen Konzeptualisierung” [3]. Diese beschreibt demnach einen Diskursbereich mit Hilfe von formal definierten Begriffen sowie Regeln über deren Zusammenhänge. Die Regeln ermöglichen die Ableitung von Rückschlüssen anhand des vorhandenen Wissens mittels Inferenz (logisches Folgern).

Die Begriffe innerhalb einer Ontologie sind in der Regel in Form einer hierarchischen Untergliederung, der sogenannten Taxonomie, gegeben und werden auch als Konzepte bezeichnet (z.B. das Konzept *Fahrzeug* und *Straße*). Relationen repräsentieren Querbeziehungen unter den Baumstrukturen der Konzepte (z.B. *Fahrzeug fährt auf Straße*). Auch für Relationen kann eine Taxonomie definiert werden. Sowohl Konzepten als auch Relationen können Eigenschaften zugeordnet werden (z.B. Größe eines Fahrzeugs). Eigenschaften können entlang den Baumstrukturen vererbt werden. Ein weiterer Bestandteil einer Ontologie sind Axiome. Das sind Aussagen über den Diskursbereich, die sicher sind. Axiome repräsentieren in der Regel Wissen, das nicht aus anderen Begriffen abgeleitet werden kann. Instanzen der definierten Konzepte, auch als Individuen bezeichnet, stellen das zur Verfügung stehende Wissen dar (z.B. VW Golf als Instanz der Klasse *Fahrzeug* oder Kaiserstraße als Instanz der Klasse *Straße*). Eine Ontologie bildet zusammen mit ihren Individuen eine Wissensbasis.

Zusammenfassend lässt sich sagen, dass eine Ontologie die explizite Wissensrepräsentation in komplexen Diskursbereichen ermöglicht. Eine solche Repräsentation weist Vorteile im Bezug auf Wartung, Erweiterbarkeit und Flexibilität auf, sowie auch in der Kommunikation und Wiederverwendbarkeit von Wissen, kann allerdings in ihrer klassischen Form nicht ohne Weiteres mit Unsicherheiten umgehen.

2.2 Probabilistische Sichtweise

Die maschinelle Wahrnehmung kognitiver Automobile ist naturgemäß unsicherheitsbehaftet. Dies liegt sowohl in der sensorischen Erfassung der Fahrumgebung, die regelmäßig Störgrößen ausgesetzt ist, als auch in der Unvollkommenheit der zugrundeliegenden Umgebungsmodelle begründet. Eine wesentliche Information einer Fahrumgebungsbeschreibung liegt entsprechend in der expliziten Angabe der Messunsicherheiten.

Die Unsicherheiten in der Messung kontinuierlicher Parameter lassen sich im einfachsten Falle durch Angabe von Toleranzbereichen oder Kovarianzmatrizen spezifizieren [4]. Gerade für nicht gauß'sche Verteilungen bietet sich die Spezifikation in Form von Konfidenzintervallen und zugeordneten Konfidenzniveaus an. Die Unsicherheit wertdiskreter Information, wie beispielsweise dem Ergebnis einer Klassifikation oder einer Detektion, kann durch eine Wahrscheinlichkeitsverteilung über alle möglichen Ereignisse ausgedrückt werden.

Fasst man die relevante Information über die Welt in einem n -dimensionalen Zufallsvektor

$$\mathbf{X} = (X_1, X_2, \dots, X_n)^T \quad (1)$$

mit den Zufallsvariablen $X_i, i = 1, \dots, n$ zusammen, so lässt sich durch Modellierung der Verbundverteilungsdichtefunktion ein in statistischem Sinne mächtigstes Weltmodell erstellen

$$P(\mathbf{X} = \mathbf{x}), \quad (2)$$

das die Wahrscheinlichkeit beschreibt, mit der eine Welt eine bestimmte Konfiguration $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$; $x_i \in \mathbb{R}$ annimmt. Bezeichne $\mathbf{X}_{i:j} = (X_i, X_{i+1}, \dots, X_j)^T$, $1 \leq i < j \leq n$ einen Teil des Weltwissens \mathbf{X} , $\mathbf{x}_{i:j}$ die entsprechende Realisation und seien die ersten k Größen $\mathbf{X}_{1:k}$ unmittelbar beobachtbar, dann lässt sich die Verteilung der übrigen Größen $\mathbf{X}_{k+1:n}$ mit Hilfe des Satzes von Bayes formulieren

$$P(\mathbf{X}_{k+1:n} | \mathbf{X}_{1:k}) = \frac{P(\mathbf{X})}{\int \dots \int P(\mathbf{X}) dX_{k+1} \dots dX_n}. \quad (3)$$

Wird im Extremfall für eine gegebene Beobachtung $\mathbf{x}_{1:k}$ die (differenzielle) Entropie der *a posteriori* Verteilung $P(\mathbf{X}_{k+1:n} | \mathbf{X}_{1:k})$ zu Null, so kann man $\mathbf{X}_{k+1:n}$ sicher bestimmen. Im Allgemeinen wird in der Bayes'schen Inferenz häufig eine ausgezeichnete Realisation der bedingten Verteilung geschätzt.

Wenngleich die Bayes'sche Inferenz formal optimal ist, erscheint diese Art der Schlussfolgerung zunächst nicht dem menschlichen Vorgehen zu entsprechen. Vor allen Dingen erscheint die mentale Modellierung einer hochdimensionalen Verteilungsfunktion $P(\mathbf{X})$ wie in Gl. (3) zweifelhaft. Stattdessen erscheinen niedrigdimensionale Interaktionen und die Konzepte der (bedingten) Irrelevanz oder der kausalen Beeinflussung dem menschlichen Denken näher zu liegen. Durch die Modellierung von \mathbf{X} als Markov-Netz oder als Bayes-Netz lassen sich diese Konzepte jedoch zur impliziten Modellierung von $P(\mathbf{X})$ nutzen, wie im folgenden Kapitel gezeigt wird.

Wie in Bild 2 skizziert, lassen sich Unsicherheiten durch die unterschiedlichen Ebenen der Kognitionskette systematisch propagieren. Dadurch wird einerseits die Unsicherheit einer der Handlung auf beliebiger Ebene zugrunde liegenden Information bis in die Handlungsausführung hinein adäquat berücksichtigt. Andererseits lässt sich durch Nutzung des Rückkanals Ambiguität auf niedrigeren Ebenen durch Wissen höherer Ebenen reduzieren.

3 Probabilistisch logisches Lernen

Der Begriff *probabilistisch logisches Lernen* (*probabilistic logic learning*, PLL) beschreibt ein jüngeres KI-Forschungsgebiet, welches relationale oder logische Repräsentationen, probabilistisches Schließen und Prinzipien des maschinellen Lernens integriert¹. Dabei werde *probabilistische* Repräsentations- und Schlussfolgerungsmechanismen, wie Bayes Netze, Markov Netze oder probabilistische kontextfreie Grammatiken verwendet. Diese werden mit formaler *Logik* - meist mit Prädikatenlogik und relationalen Repräsentationen

¹In der Literatur werden auch die Begriffe *probabilistische induktive Logikprogrammierung* (*probabilistic inductive logic programming*), *statistisches relationales Lernen* (*statistical relational learning*) oder *multi-relationales Data-Mining* (*multi-relational data mining*) verwendet.

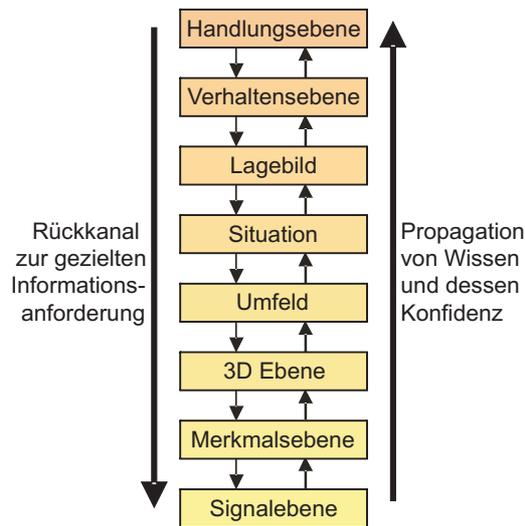


Bild 2: Bidirektionale Propagation von Unsicherheit durch die Kognitionskette

verknüpft. Diese ermöglichen eine elegante Beschreibung komplexer Situationen samt aller darin vorkommenden Objekte und deren Relationen. Die Motivation für das *Lernen* liegt darin, dass es oft einfacher ist, für einen komplexen Diskursbereich Trainingsdaten zu generieren und ein Modell zu lernen, als ein Modell mit konventionellen Methoden des Wissensengineerings zu erstellen. Somit hat PLL zum Ziel, probabilistische Modelle auf eine kompakte und intuitive Weise darzustellen, welche die relationale Struktur des Diskursbereichs widerspiegelt und, idealerweise, effizientes Lernen und Schlussfolgern unterstützt [5].

Die meisten dieser Verfahren basieren auf Kombinationen von probabilistischen Graphen oder probabilistischen kontextfreien Grammatiken und logischen Formeln oder Frames. Dazu gehören *Stochastic Logic Programs* (SLPs) [6], *PRogramming In Statistical Modeling* (PRISM) [7] und *Independent Choice Logic* (ICL) [8], *Probabilistic Relational Models* (PRMs) [9], *Bayesian Logic Programs* (BLPs) [10], und *Markov Logic Networks* (MLNs) [11]. Einen Überblick über alle bisherigen PLL-Ansätze gibt [5].

SLPs verbessern die Ausdrucksfähigkeit der stochastischen kontextfreien Grammatiken auf dem Level der Logikprogramme. Dabei kennzeichnen SLPs jede Klausel durch einen Wahrscheinlichkeitswert, wohingegen PRISM und ICL jedem instantiierten atomaren Satz eine Wahrscheinlichkeit zuordnen, aus welcher weiteres Wissen abgeleitet werden kann [5]. Die probabilistische Semantik der restlichen oben genannten Systeme basiert auf graphischen Modellen. PRMs und BLPs nutzen gerichtete Graphen in Kombination mit Frames bzw. Logikprogrammen. MLNs kombinieren dagegen ungerichtete Netze mit Prädikatenlogik. Inferenzalgorithmen werden in solchen Modellen auf einem anhand aller Evidenzvariablen instantiierten Graphen angewendet. Bei der Inferenz wird nur der für die Anfragevariablen relevante Teil des instantiierten Netzes konstruiert [12].

Im Folgenden wird auf Bayes-Logikprogramme (*Bayesian Logic Programs*) und Markov-Logik-Netze (*Markov Logic Networks*) detaillierter eingegangen.

3.1 Markov-Logik-Netze (MLN)

Ein Markov-Logik-Netzwerk (MLN) ist eine probabilistisch logische Repräsentation, welche Prädikatenlogik erster Stufe und Markov-Netze miteinander verknüpft [11]. Ein MLN besteht aus einer Menge prädikatenlogischer Formeln, die einen bestimmten Diskursbereich beschreiben, sowie einem jeder Formel zugeordnetem Gewicht. Das Gewicht wird gelernt und quantifiziert den Wahrheitsgehalt der durch die Formel ausgedrückten Behauptung. Ein MLN kann als ein Formalismus zur Erstellung eines Markov-Netzes mit logischen Variablen verstanden werden, d.h. zur Zerlegung einer Markov'schen Verbundverteilung binärer Zufallsvariablen in Faktoren, die jeweils nur von wenigen Variablen aus \mathbf{X} abhängen. Eine so definierte Verbundverteilung $P(\mathbf{X})$ erlaubt die Anwendung klassischer probabilistischer Inferenzalgorithmen.

Gegeben seien eine Menge prädikatenlogischer Formeln $F = \{F_1, \dots, F_n\}$, die einen bestimmten Diskursbereich beschreiben. Freie Variablen werden dabei als allquantifiziert betrachtet. Zusammen mit einer Menge reellwertiger Gewichte $w_1, \dots, w_n \in \mathbb{R}$, die der Formeln aus F zugeordnet werden, definieren sie ein MLN

$$L = \{(w_1, F_1), \dots, (w_n, F_n)\}.$$

Gegeben seien das MLN L und eine Menge logischer Konstanten $C = \{c_1, \dots, c_m\}$. Durch die Substitution der Konstanten anstelle der Variablen in den Formeln von L werden alle möglichen Grundatome $X = (X_1, \dots, X_q)$ und daraus alle möglichen Grundinstanzen aller Formeln $G = (G_1, \dots, G_p)$ aus L gebildet. Dabei bekommt jedes Element aus G das Gewicht der zu Grunde liegenden Formel aus L zugeordnet.

Alle Elemente X_i können auf einen Wahrheitswert x_i mit

$$(X_1 = x_1, \dots, X_q = x_q), x_i \in \{0, 1\}, i = 1, \dots, q$$

und alle Elemente G_k auf ein Feature $f_k(x)$ mit

$$(G_1 = f_1(x), \dots, G_p = f_p(x)), f_k(x) \in \{0, 1\}, k = 1, \dots, p$$

abgebildet werden.

Zur Erstellung eines Markov-Netzes $M(L, C)$ werden die Grundatome X_1, \dots, X_q als Elemente eines binären Zufallsfeldes betrachtet, die die Knoten eines Interaktionsgraphen bilden. Der Wert eines Knotens x_i ist genau dann 1, wenn das zu Grunde liegende Grundatom wahr ist. Es existiert genau dann eine ungerichtete Kante zwischen zwei Knoten aus $M(L, C)$, wenn beide Grundatome mindestens in einem Element aus G gemeinsam vorkommen. In der Terminologie eines Markov-Feldes sind solche Knoten *benachbart* und die Menge aller Nachbarschaftsbeziehungen bilden ein gültiges Nachbarschaftssystem. Somit bilden die Grundatome jeder Grundinstanz einer Formel eine Clique² von $M(L, C)$. Die Variablenbelegung der k -ten Clique wird durch das Feature $f_k(x)$ der entsprechenden Grundinstanz der Formel aus G und durch das ihr zugeordnete Gewicht w_k bewertet. Das Feature hängt nur von den Variablen in x ab, die zur betrachteten Clique gehören. Ein Feature $f_k(x)$ ist genau dann 1, wenn x eine mögliche Welt für G_k darstellt, d.h. wenn die Aussage von G_k über den Diskursbereich wahr ist. Als Verbundwahrscheinlichkeit möglicher Welten x , die das resultierende Markov-Netz beschreibt, wird

²Eine Clique ist ein Ensemble aus Zufallsvariablen, in dem jede beliebige zwei Variablen benachbart sind.

schließlich

$$P(X = x) = \frac{1}{Z} \exp \left(\sum_k w_k f_k(x) \right), \quad (4)$$

definiert, wobei Z ein reeller Normierungsfaktor ist und die Summation über alle Grundinstanzen aller Formeln erfolgt. Das resultierende Markov-Netz wird als *belegtes* Markov-Netz bezeichnet.

Tabelle 1 zeigt beispielhaft ein aus zwei Formeln bestehendes MLN $L1$, welches für eine gegebene Konstantenmenge, $C1 = \{O1, O2, R1\}$, das in Bild 3 dargestellte belegte Markov-Netz $M(L1, C1)$ ergibt. Für unterschiedliche Mengen logischer Konstanten

w_i	F_i
0.6	$\forall o \forall r \quad on(o, r) \wedge road(r) \wedge hasSpeed(o, Low) \Rightarrow car(o)$
1.4	$\forall o \quad hasDirection(o, Same) \Rightarrow car(o)$

Tabelle 1: Beispiel-MLN $L1$.

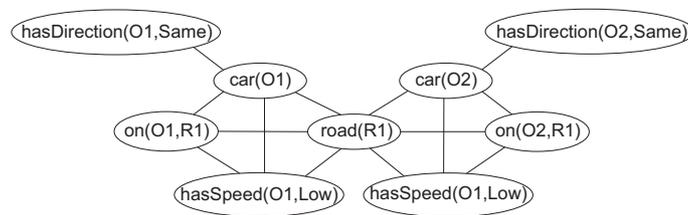


Bild 3: Das belegte Markov-Netz $M(L1, C1)$ für das MLN $L1$ und die Konstantenmenge $C1$ aus dem eingeführten Beispiel.

werden entsprechend unterschiedlich große belegte Netze konstruiert, die jedoch gewisse strukturelle Gemeinsamkeiten aufgrund der gleichen logischen Struktur des MLN aufweisen. Gl. (4) bewirkt, dass eine mögliche Welt, in der eine oder mehrere der Grundinstanzen der Formeln falsch sind, weniger wahrscheinlich aber nicht unmöglich wird. Auf diese Weise gelingt es, die harten Einschränkungen auf die Menge der möglichen Interpretationen aus der klassischen Prädikatenlogik zu schwächen. Das jeder Formel zugeordnete Gewicht ist ein Maß dafür, wie präzise die entsprechende Formel Wissen über den Diskursbereich vermittelt. Je größer das Gewicht, desto größer ist die Differenz der Logitwerte zwischen einer möglichen Welt, in der die Formel wahr ist, und einer, in der die Formel falsch ist (bei Festhalten aller anderen Werte). Im Sonderfall unendlich großer Gewichte degeneriert ein MLN zu einer klassischen prädikatenlogischen Wissensbasis. Im allgemeinen Fall werden die Gewichte anhand von Trainingsdaten gelernt oder von einem menschlichen Experten vorgegeben.

3.2 Bayes-Logikprogramme

Bayes-Logikprogramme (BLPs) stellen eine Verknüpfung von Bayes-Netzen und Logikprogrammen dar [10]. Hierbei werden Grundatome, die durch eine Menge definierter Klauseln definiert sind, als Zufallsvariablen aufgefasst. Es wird zwischen deterministischen - auch als *logisch* bezeichneten - und probabilistischen - auch als *Bayes'sch* bezeichneten - Prädikaten unterschieden. Entsprechend werden auch Atome entweder als logisch

oder als Bayes'sch bezeichnet. Ein BLP besteht aus einer Menge Bayes'scher definierter Klauseln der Form $(A|A_1, \dots, A_n)^3$, wobei A ein Bayes'sches Atom ist, $A_1, \dots, A_n, n \geq 0$ logische und Bayes'sche Atome sind. Jeder Bayes'schen Klausel c wird ihre bedingte Wahrscheinlichkeitsverteilung zugewiesen $cpd(c) = P(A|A_1, \dots, A_n)$, welche die probabilistische Abhängigkeiten unter den Grundatomen der Klausel quantifiziert.

Die Semantik der BLPs wird durch den Prozess der so genannten wissensbasierten Modellkonstruktion definiert. Bei diesem wird ein Bayes-Netz dadurch konstruiert, dass jede Bayes'sche definite Klausel durch die Menge aller möglichen Grundinstanzen ersetzt wird. Auf diese Weise werden alle möglichen Bayes'schen Grundatome des minimalen Herbrand-Modells gebildet. Das Bayes-Netz besitzt einen Knoten für jedes Bayes'sche Grundatom und eine gerichtete Kante zwischen zwei Knoten, falls eine direkte Abhängigkeit der beiden Grundatome durch das Logikprogramm ausgedrückt wird. Sitzt ein Bayes'sches Atom im Kopf einer Klausel, so bilden seine Instantiierungen Kindknoten, sitzt es im Rumpf – bilden sie Elternknoten. Zur Vervollständigung eines solchen Netzes sind den Knoten noch bedingte Wahrscheinlichkeitsverteilungen zuzuweisen. Diese Verteilungen können per Hand von einem Experten angegeben werden oder anhand von Trainingsdaten gelernt werden. Die für Forschungszwecke frei verfügbare BLP-Software⁴ enthält Algorithmen zum Lernen eines solchen Bayes-Netzes und zur anschließenden Inferenz in diesem konstruierten Netz. BLPs können jede Art von Bayes-Netzen sowie auch jede Art von gewöhnlichen Prolog-Programmen modellieren. Darüber hinaus wurde in [5] gezeigt, dass mit BLPs verborgene Markov-Modelle und probabilistische Grammatiken modelliert werden können.

Bild 4 zeigt eine Realisation einer Fahrstreifenwechselentscheidung durch ein Bayes-Logikprogramm. Durch die in der Abbildung vorgenommene Modellierung ist die Verbundwahrscheinlichkeit aller Netzknoten eindeutig definiert. Insbesondere lassen sich daraus beliebige bedingte Verteilungen für einen Fahrstreifenwechsel bestimmen und damit ein Maß dafür angeben, wie sinnvoll und durchführbar ein Fahrstreifenwechsel bei gegebenen Randbedingungen erscheint. Messunsicherheiten werden durch die Likelihood- und *a priori* Verteilungen berücksichtigt.

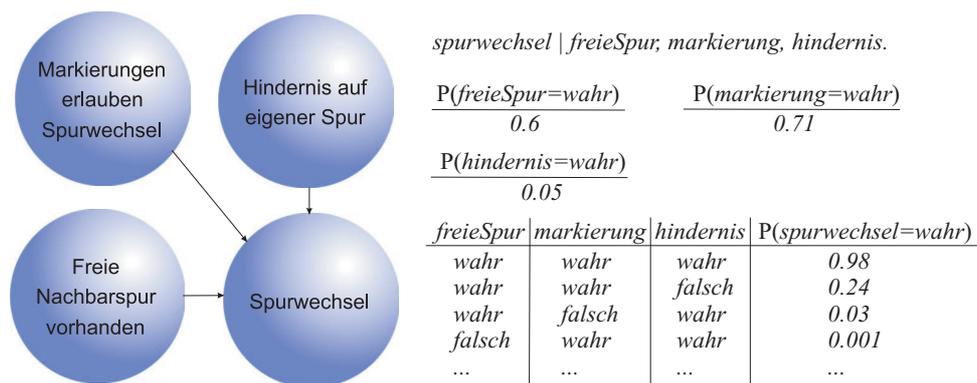


Bild 4: Modellierung eines Fahrstreifenwechsels in einem Bayes-Logikprogramm.

³In Bayes'schen Klauseln wird “|” anstelle eines Implikationspfeils verwendet und alle Variablen auf der rechten Seite sind (implizit) allquantifiziert.

⁴<http://www.informatik.uni-freiburg.de/~kersting/profile/>

4 Anwendungsbeispiele

4.1 Anwendung zur Klassifikation von Objekten der Verkehrsszene

Ein MLN zur Klassifikation von Objekten der Verkehrsszene wurde erstellt. Dabei wurde die Ontologie aus Bild 5 in Prädikatenlogik erster Stufe formalisiert. Einstellige Prädikate wurden dabei für die Objektklassen deklariert, zweistellige Prädikate für die Relationen, und Terme für die Merkmalsklassen und deren symbolischen Ausprägungen. Alle logischen Variablen sind hierbei typisiert. Innerhalb einer Spezialisierungsebene sind die Konzepte als paarweise disjunkt und in ihrer Vereinigung erschöpfend modelliert. Die Re-

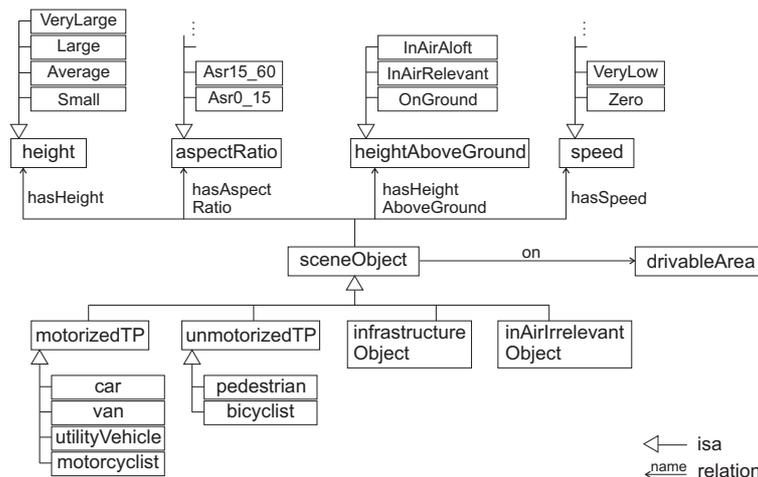


Bild 5: Ontologie für Objekte einer Verkehrsszene. Aus Platzgründen wurde für einige Merkmale nur ein Auszug aus der Menge der möglichen Ausprägungen aufgelistet.

lationen repräsentieren Beziehungen zwischen Objekten (z. B. $on(OI, Plane1)$) oder zu Objektmerkmalen (z. B. $hasHeight(OI, Average)$). Die quantitativen Wertebereiche der Merkmale – wie Form, Höhe oder Geschwindigkeit – werden zweckmäßig in eine Menge diskreter, symbolischer Ausprägungen unterteilt.

Mit Hilfe dieses Vokabulars wurden prädikatenlogische Formeln für ein MLN zur Klassifikation von Objekten der Verkehrsszene gebildet. Dabei ist zwischen harten und weichen Formeln zu unterscheiden. Die harten Formeln unterliegen keiner Trainingsphase, sondern gelten als stets wahr, d.h. ihnen wird ein unendliches Gewicht zugewiesen. Solche harten Formeln werden etwa zur Beschreibung der Taxonomie der Objektklassen verwendet. Die Formeln, die den Zusammenhang zwischen Objektklassen und Merkmalen beschreiben, sind dagegen weich. Für diese wird anhand von manuell segmentierten Szenenobjekten ein zugehöriges Gewicht gelernt. Das Gewicht spiegelt die Häufigkeit wider, mit der die Formel in den Trainingsdaten erfüllt wurde. Beispiele weicher Formeln sind:

$$\forall o \quad hasAspectRatio(o, +r) \Rightarrow car(o)$$

$$\forall o \quad hasAspectRatio(o, +r) \Rightarrow pedestrian(o)$$

Das Pluszeichen vor der Variable r bedeutet, dass während der Trainingsphase alle möglichen symbolischen Werte des Merkmals $aspectRatio$ automatisch nacheinander substituiert werden und für jede dieser Formeln ein eigenes Gewicht gelernt wird. Folgende Gewichte für die obigen Formeln wurden, z.B. für die symbolische Ausprägung der Objektform $Asr60_90$ gelernt:

- +1.7 $hasAspectRatio(o, Asr60_90) \Rightarrow car(o)$
- 3.7 $hasAspectRatio(o, Asr60_90) \Rightarrow pedestrian(o)$

Aus der Größe der Gewichte lässt sich schließen, dass diese bestimmte qualitative Formbeschreibung für ein Auto, aber nicht für einen Fußgänger charakteristisch ist.

Als Eingangsdaten für den so gebauten Klassifikator dienen die symbolischen Merkmalsausprägungen für alle Ontologie-Merkmale, die aus den vorhandenen quantitativen Sensorinformationen über ein konkretes Objekt abstrahiert werden. Diese dienen zur Erstellung einer Menge von wahren Grundatomen, die als Evidenzdaten bezeichnet werden.

Das MLN zur Klassifikation, gemeinsam mit den Evidenzdaten, wird automatisiert in ein belegtes Markov-Netz übersetzt. Für Anfragen nach einer oder mehrerer Objektklassen (z. B. *car*) werden mittels Inferenz anhand des belegten Markov-Netzes bedingte Wahrscheinlichkeiten für alle Grundatome der angefragten Objektklassen geschätzt (z. B. $P(car(O1))$, $P(car(O2))$), gegeben die Konstantenmenge $\{O1, O2\}$). Diese Wahrscheinlichkeiten liefern den Wahrheitsgrad der Aussage, die durch das jeweilige Grundatom gemacht wird (z. B. die Wahrscheinlichkeit, mit der die Konstante *O1* zu der Objektklasse *car* gehört). Dadurch wird eine Wahrscheinlichkeit für die Klassenzugehörigkeit jedes konkreten Objekts aus den Evidenzdaten bestimmt.

Das vorgestellte Verfahren wurde zur Klassifikation von manuell und automatisch bewegungsbasiert segmentierten Objekten eingesetzt. Zur Inferenz wird ein MC-SAT Algorithmus verwendet, welcher den MCMC-Algorithmus (Markov chain Monte Carlo) mit einem SAT-Solver kombiniert. Bild 6 stellt die Klassifikationsergebnisse für eine manuell segmentierte innerstädtische Verkehrsszene dar. Bild 8 veranschaulicht die Klassifikationsergebnisse für eine automatisch segmentierte Videosequenz. Mit dem in [13] beschriebenen Algorithmus werden Bildpunkte, die sich ähnlich bewegen, zu Regionen in der Bildebene zusammengefasst. Jede Region wird eindeutig durch eine starre Bewegung im 3d-Raum charakterisiert und durch ein rechteckiges Hypothesenfenster repräsentiert. Anhand des Hypothesenfensters werden quantitative Werte für alle interessierenden Merkmale bestimmt, welche anschließend zu Evidenzdaten abstrahiert werden. Bild 7 zeigt einige Bilder aus einer Videosequenz mit den segmentierten Hypothesenfenstern und Bild 8 die Klassifikationsergebnisse für diese Sequenz.

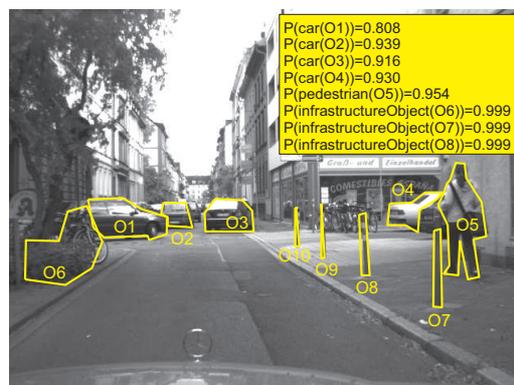


Bild 6: Klassifikationsergebnisse für manuell segmentierte Objekte einer innerstädtischen Verkehrsszene.



Bild 7: Bilder aus einer automatisch segmentierten Videosequenz mit den entsprechenden Hypothesenfenstern der Objekte *O1* und *O2*.

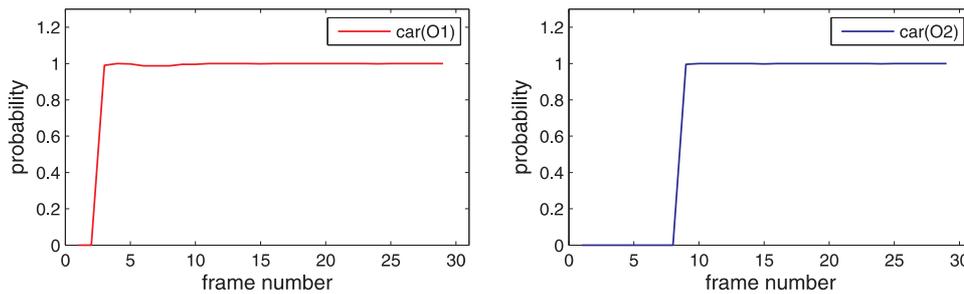


Bild 8: Klassifikationsergebnisse für die automatisch segmentierten Objekte *O1* und *O2* der Videosequenz aus Abb. 7

4.2 Anwendung zur Ermittlung von Geschehen in einer Verkehrsszene

Eine weitere Anwendung von MLN ist die Ermittlung begrifflicher Beschreibungen von elementaren Geschehen in Verkehrsszenen aus automatisch segmentierten Videosequenzen. Dabei werden die Geschehen durch Verben oder Verbalphrasen repräsentiert. Man kann zwischen Geschehen 1. Ordnung und Geschehen 2. Ordnung unterscheiden. Erstere beziehen sich nur auf ein einzelnes Objekt, wie z.B. *brake*, *accelerate*, *start*. Die Geschehen 2. Ordnung beschreiben Relationen zweier Objekte, wobei eine weitere Unterteilung in Abhängigkeit davon erfolgt, ob sich eines oder beide Objekte bewegen, z.B. *follow*, *moveAwayFrom*). Nachfolgend ist ein Auszug des entwickelten Vokabulars zur Ermittlung elementarer Geschehen aufgelistet, wobei die nachgestellten Ziffern die jeweilige Stelligkeit (Variablenzahl) der Prädikate kennzeichnen:

<i>sceneObject</i>	/1	<i>hasRelPosition</i>	/3
<i>accelerate</i>	/1	<i>hasRelDistance</i>	/3
<i>brake</i>	/1	<i>prevDist</i>	/2
<i>moveConstVel</i>	/1	<i>tendDist</i>	/2
<i>stand</i>	/1	<i>nextDist</i>	/2
<i>stop</i>	/1	<i>prevVel</i>	/2
<i>start</i>	/1	<i>tendVel</i>	/2
<i>moveAwayFrom</i>	/2	<i>nextVel</i>	/2
<i>follow</i>	/2	<i>prevDiffInOrientation</i>	/3
<i>driveInFrontOf</i>	/2	<i>nextDiffInOrientation</i>	/3

Aufgrund der höheren Stelligkeit der Prädikate ist es schwer die Ontologie sinnvoll als

Graph darzustellen. Harte Formeln werden aufgestellt, um die Abhängigkeiten unter diesen Begriffen festzuhalten.

Die elementaren Geschehnisse werden momentenbasiert modelliert, d.h. diese werden für jedes Einzelbild der Sequenz ausgewertet. Dazu werden Merkmalswerte des vorigen (z.B. $prevVel(O1, MovingPrev)$) und des nachfolgenden Bildes (z.B. $nextVel(O1, MovingNext)$), sowie die Tendenz der Merkmalsänderung (z.B. $tendVel(O1, IncreasingVel)$) berücksichtigt. Die Geschehen werden in Abhängigkeit von den Merkmalswerten durch prädikatenlogische Formeln definiert. Als Beispiel ist die Definition für $moveAwayFrom$ aufgeführt:

$$\begin{aligned} \forall o1, o2: \quad & moveAwayFrom(o1, o2) \quad <=> \quad &!(o1=o2) \quad \wedge \quad stand(o1) \\ & \wedge \quad prevVel(o2, MovingPrev) \quad \wedge \quad nextVel(o2, MovingNext) \quad \wedge \\ & tendDist(o1, o2, IncreasingDist) \quad \wedge \quad (hasRelDistance(o1, o2, CloseDist) \quad \vee \\ & hasRelDistance(o1, o2, VeryCloseDist) \vee hasRelDistance(o1, o2, MediumDist)). \end{aligned}$$

Diese besagt, dass das Geschehen $moveAwayFrom$ in einem Bild wahr wird, wenn das erste Objekt steht, das zweite sich bewegt, der relative Abstand zwischen den Objekten klein bis mittel ist und tendenziell größer wird.

Die Videosequenzen werden ebenfalls mit dem Algorithmus aus [13] bewegungssegmentiert. Für die segmentierten Objekthypothesen werden quantitative Werte für die Bewegung und Position im Raum geliefert. Diese Messreihen werden in einem anschließenden Abstraktionsschritt in begrifflichen Evidenzdaten überführt, die als Eingangsdaten für die Inferenz dienen. So eine Abstraktion aus verrauschten Signalen ist keine triviale Aufgabe und wird in der Literatur meistens durch die Verwendung synthetischer Daten umgangen.

In der aktuellen Implementierung werden die quantitativen Messreihen zuerst geglättet. Danach werden für jedes Bild Geschwindigkeitsbetrag und -richtung für jede Objekthypothese, sowie auch relative Position, relativer Abstand und Orientierungsdifferenz für jedes mögliche Paar Objekthypothesen berechnet. Dabei wird das Ego-Fahrzeug ebenfalls als Objekt in der Szene betrachtet, so dass dessen Relationen zu jeder segmentierten Objekthypothese ausgewertet werden. Zusätzlich wird pro Bild für bestimmte Merkmale die Differenz der umliegenden vier Bilder berechnet, um zu bestimmen, wie sich der Merkmalswert ändert. Schwellwertbasiert werden die quantitativen Wertebereiche der neu berechneten Merkmalsreihen geeignet in diskrete symbolische Ausprägungen unterteilt und auf wahre Grundatome abgebildet (z.B. $tendDist(Ego, O1, IncreasingDist)$). Dabei werden auch Merkmalswerte aus dem vorherigen und dem nächsten Bild einbezogen und

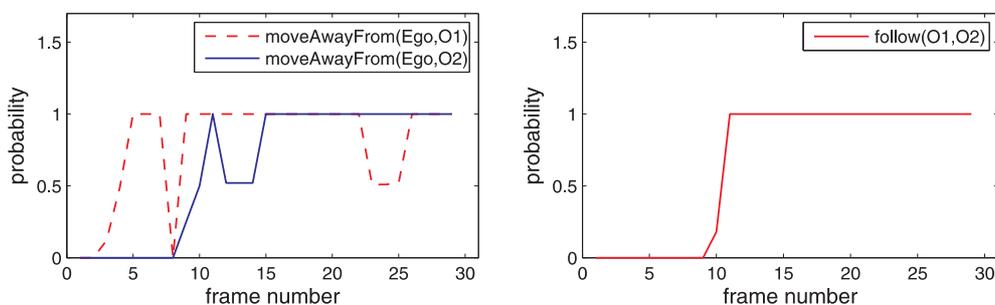


Bild 9: Ergebnisse der Ermittlung der Geschehen $moveAwayFrom$ (links) und $follow$ (rechts) für die Sequenz aus Bild 7.

auf wahre Grundatome abstrahiert. Pro Bild enthalten somit die Evidenzdaten eine Menge wahrer Grundatome pro Objekt, inklusive das Ego-Fahrzeug, und entsprechend viele wahren Grundatome für alle möglichen Paarrelationen. Als Beispiel sind die Evidenzdaten zu Bild 7 der Sequenz aus Abb. 7 nachfolgend aufgeführt:

<i>sceneObject(OI)</i>	<i>hasRelPosition(Ego,OI,NE)</i>
<i>prevVel(OI,MovingPrev)</i>	<i>hasRelDistance(Ego,OI,CloseDist)</i>
<i>tendVel(OI,ConstVel)</i>	<i>prevDist(Ego,OI,CloseDistPrev)</i>
<i>nextVel(OI,MovingNext)</i>	<i>tendDist(Ego,OI,IncreasingDist)</i>
<i>car(Ego)</i>	<i>nextDist(Ego,OI,CloseDistNext)</i>
<i>prevVel(Ego,ZeroPrev)</i>	<i>hasVelMagnitude(OI,ZeroVel)</i>
<i>tendVel(Ego,ConstVel)</i>	<i>prevDiffInOrientation(Ego,OI,EqualOrientPrev)</i>
<i>nextVel(Ego,ZeroNext)</i>	<i>nextDiffInOrientation(Ego,OI,EqualOrientNext)</i>

Das entwickelte MLN zur Ermittlung von Geschehen, gemeinsam mit den Evidenzdaten für ein Bild, wird automatisiert in ein belegtes Markov-Netz übersetzt. Durch Inferenz werden die bedingten Wahrscheinlichkeiten für alle Grundatome der definierten elementaren Geschehen für das entsprechende Bild geschätzt. Diese Auswertung wird für alle Bilder der Videosequenz nacheinander durchgeführt. Die Ergebnisse für die Sequenz aus Bild 7 für die Geschehen *moveAwayFrom* und *follow* sind in Abb. 9 veranschaulicht. Zur Inferenz wurde ebenfalls, wie in 4.1, der MC-SAT Algorithmus der frei verfügbaren Software Alchemy [14] verwendet.

5 Zusammenfassung und Ausblick

Kognitive Automobile nehmen ihre Fahrumgebung sensorisch wahr, strukturieren dieses Wissen und generieren daraus geeignetes Fahrverhalten. Probabilistische Ansätze erlauben die expliziten Berücksichtigung von Unsicherheit und deren konsistente Propagation durch die Kognitions-kette. Neben metrischem Wissen sind symbolisches und relationales Wissen für die Repräsentation von Verkehrssituationen unverzichtbar. Probabilistisch-logisches Lernen bietet zur dessen Einbeziehung mit Markov-Logik-Netzen und Bayes-Logikprogrammen besonders attraktive Ansätze für den Diskursbereich kognitiver Automobile. Insbesondere erlauben diese Methoden die konsistente Formulierung hochdimensionaler Verbundverteilungen durch niedrigdimensionale Funktionale, welche durch probabilistische Regeln mit erlernbaren Gewichten ausgedrückt werden. In Experimenten konnten Inferenzverfahren für die Klassifikation sowie für die Ermittlung von Bewegungsbeziehungen realisiert werden, die funktionale Eigenschaften und Relationen zwischen Objekten in Form von Regeln berücksichtigen. Laufende Arbeiten befassen sich mit Ansätzen für kognitives Sehen zur Beschreibung ganzheitlicher Verkehrssituationen.

6 Danksagung

Teile der berichteten Arbeiten werden durch die Deutsche Forschungsgemeinschaft, DFG unterstützt. Unser Dank gilt der DFG sowie den beteiligten Partnern des Sonderforschungsbereichs Transregio 28 Kognitive Automobile für die fruchtbare Zusammenarbeit.

Literatur

- [1] Maurer, M.; Stiller, C. (Hg.): *Fahrerassistenzsysteme mit maschineller Wahrnehmung*. Heidelberg: Springer-Verlag. 2005.
- [2] Stiller, C.; Färber, G.; Kammel, S.: Cooperative Cognitive Automobiles. In: *Proc. IEEE Intelligent Vehicles Symposium*, S. 215–220. Istanbul, Turkey. 2007.
- [3] Gruber, T.: A translation approach to portable ontology specifications. *Knowledge Acquisition* 5 (1993) 2, S. 199–220.
- [4] DIN Deutsches Institut für Normung e.V.: Leitfaden zur Angabe der Unsicherheit beim Messen (Deutsche Fassung). DIN V ENV 13005, Beuth Verlag GmbH. 1999.
- [5] Getoor, L.; Taskar, B.: *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press. ISBN 0262072882. 2007.
- [6] Muggleton, S.: Learning Structure and Parameters of Stochastic Logic Programs. In: *ILP*, S. 198–206. 2002.
- [7] Sato, T.: A Statistical Learning Method for Logic Programs with Distribution Semantics. In: *International Conference on Logic Programming*, S. 715–729. 1995.
- [8] Poole, D.: The Independent Choice Logic for Modelling Multiple Agents Under Uncertainty. *Artificial Intelligence* 94 (1997) 1-2, S. 7–56.
- [9] Friedman, N.; Getoor, L.; Koller, D.; Pfeffer, A.: Learning Probabilistic Relational Models. In: *IJCAI*, S. 1300–1309. 1999.
- [10] Kersting, K.; Raedt, L. D.: Bayesian Logic Programs. In: *Proceedings of the Work-in-Progress Track at the 10th International Conference on Inductive Logic Programming* (Cussens, J.; Frisch, A., Hg.), S. 138–155. 2000.
- [11] Richardson, M.; Domingos, P.: Markov logic networks. *Machine Learning* 62 (2006) 1-2, S. 107–136.
- [12] de Raedt, L.: *Logical and Relational Learning: From ILP to MRDM*. Springer-Verlag. ISBN 978-3-540-20040-6. 2008.
- [13] Bachmann, A.; Dang, T.: Improving Motion-Based Object Detection by Incorporating Object-Specific Knowledge. *International Journal of Intelligent Information and Database Systems (IJIIDS), Special Issue on: Information Processing in Intelligent Vehicles and Road Applications* 2 (2008) 2, S. 258–276.
- [14] Kok, S.; Sumner, M.; Richardson, M.; Singla, P.; Poon, H.; Lowd, D.; Domingos, P.: The Alchemy System for Statistical Relational AI. Techn. Ber., Department of Computer Science and Engineering, University of Washington, Seattle, WA. See <http://alchemy.cs.washington.edu>. 2006.

Adaptive Fuzzy Path Planning for Semi-Autonomous Parking Maneuvers using a CV-Sensor

¹Rashad Mustafa, ²Thomas Goernig, ¹Ferdinand Svaricek

¹Universität der Bundeswehr München, Fakultät für Luft-und Raumfahrttechnik

Institut für Steuer- und Regelungstechnik, 85577 Neubiberg

E-Mail: {rashad.mustafa, ferdinand.svaricek}@unibw.de

²Continental Automotive Distance Control Systems GmbH

88131 Lindau, Kemptener Straße 99

E-Mail: thomas.goernig@continental-corporation.com

Abstract: This paper introduces a new fuzzy approach for path planning during semi-autonomous vehicle parking maneuvers. A Pre Crash sensor, also known as Closing Velocity sensor (CV-sensor) is used for the first time in parking planning strategies. The reasons for using the CV-sensor are because it has a high detection range, is inexpensive and a side pre-crash sensing system will be offered in the near future to the market. In the scope of this work, it is assumed that a human performs the steering task and not an electrical drive. There are several important advantages using the proposed approach: For instance, emulation of the human driver behaviour and absence of a complex path planning. The effectiveness of the proposed method is verified by simulations and experimental results regarding the detection capabilities of the CV-sensor are given.

1 Introduction

Many researchers in the automotive industry have developed parking assistance systems to support the driver during parking maneuvers [1-11]. The planning approaches are usually based on environment detection and determination of a suitable trajectory. Classical control strategies as well as knowledge-based control algorithms can be implemented such that collision-free motion within the available parking space is ensured. The classical approaches are based on kinematic and dynamic models of the vehicle and its environment. However, these approaches rely on an imprecise mathematical description, since the vehicle's parameters are uncertain or unknown [1, 8, 9, 11, 5, 6, 10]. The knowledge-based parking control makes often use of a fuzzy logic approach. There are several important advantages using this approach: For example, a simple design and it is more easier to maintain the system since the rules can be linguistically interpreted and translated the human experience to appropriately automatic control strategies [2, 3, 4].

Kochem [5] used a 2D-local video camera to detect the parking gap. Four ultrasonic sensors for detecting obstacles and road curb are used by Lee [6], the range of the used ultrasonic sensor is less than 1.5 m. Furthermore, Lee [6] and Kochem [5] used a combination between arcs and clothoids for the parking trajectories. However, using many sensors (such as using many ultrasonic sensors) in a parking system leads to the problem of increasing the vehicle costs and misreadings are caused frequently by reflections from neighbouring sensors and they can cause the algorithm to falsely detect an obstacle [7]. In addition, the ultrasonic sensors have a low detecting range. This can cause problems by the detection of the parking gap. A video camera is not usually preferable due to high cost of both camera itself and the hardware for image processing. The CV-sensor considered in this study has a high detecting range and side pre-crash sensing systems will be available in the near future.

Several parking trajectories have been investigated in the literature [8, 1, 9, 10]. Based on a fixed initial and final position pure fifth-order polynomial paths, sinusoids and a combination of two arcs are considered. These profiles are not suitable for human parking, since they are far away from the human steering behavior. Pradalier et al. [11] implemented a steering angle control profile based on elementary movements. This means, that the steering angle and the velocity control profiles consist of lines. The drawback of using only lines in a steering angle control profile is, that corners in the profiles lead to jerks and hence can not be used in the case of human parking.

This paper introduces a new fuzzy approach for the generation of steering angle control profiles and shows results of a feasibility study on the capabilities of a CV-sensor for semi-autonomous parking. The proposed fuzzy parking algorithm uses the geometry data of the parking environment together with the data from the vehicle sensors to assist the steering of the vehicle during a parking maneuver. The system configuration for the simulation is shown in Figure 1.

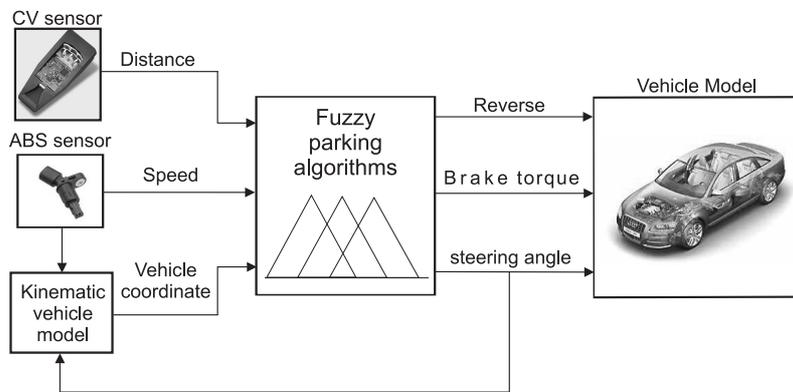


Figure 1: System configuration for simulation

The parking period depends on the vehicle speed during parking. Furthermore, the initial position is not fixed on a certain point, but changed relating to the parking environment. The final position is reached when the vehicle gets very close to the back vehicle in the parking gap or when the vehicle orientation becomes zero. The CV-sensor provides the fuzzy parking algorithm the distance between the vehicle and obstacles, while the ABS wheel sensors provide the velocity needed in the algorithm (Figure 1). Steering angle, reverse gear and braking torque are the outputs of the fuzzy parking algorithm and are the input signals of the vehicle model. The kinematic vehicle model determines the estimated position and orientation angle of the vehicle based on the signals of the ABS wheel sensors and the steering angle provided by the fuzzy parking algorithm. The reasons for using a CV-sensor are because it has a high detection range, is inexpensive and a side pre-crash sensing system will be offered in the near future to the market. Continental-TEMIC offers such kind of optical sensors to assist the driver in braking if there is a risk of a rear-end collision Furthermore to enhance the passive safety integration strategy (e.g. Airbag) as well as pedestrian protection.

This paper is structured as follows: The next section describes the models of the vehicle and the CV-sensor used in the simulation. Section 3 introduces the hierarchical structure of the proposed fuzzy system and discusses the design method, which makes use of the Matlab/Simulink Fuzzy Toolbox. Simulation results of parking maneuvers are included to illustrate the efficiency and robustness of the designed controller in section 4. Experi-

mentally testings of the effectiveness and the behaviour of the CV-sensor are discussed in section 5. Finally, conclusions are given in Section 6.

2 System Configuration

A kinematic model of the vehicle and a sensor model are required for the simulation of a CV-sensor based parking assistance system.

2.1 Vehicle Kinematic Simulation Model

A four wheeled vehicle is modelled as a rigid body moving in the plane as shown in Figure 2. The states and parameters used for the modeling are shown in Figure 2:

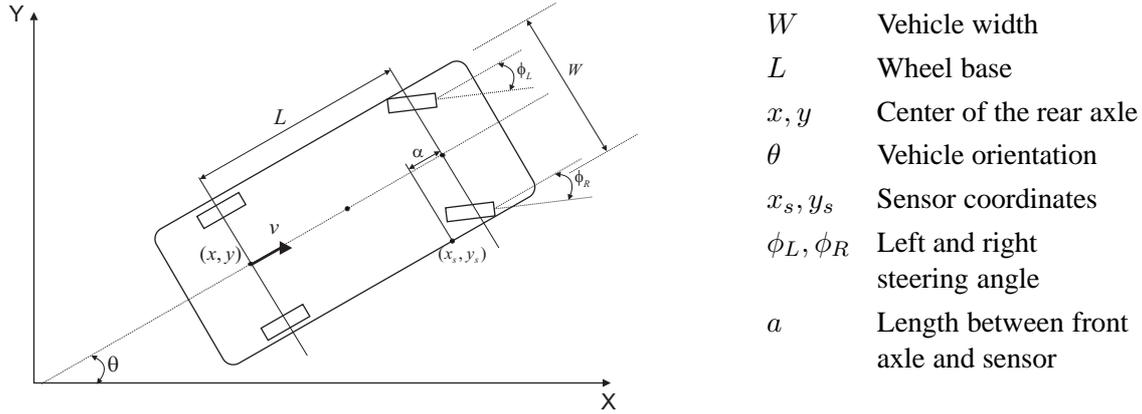


Figure 2: Vehicle model and location of the CV-sensor

The vehicle dynamic model can be represented by the coordinate (x, y) at the middle of the rear axle and the orientation θ . It will be assumed that centrifugal forces can be neglected when the speed is slow. Constraints on the system arise by allowing the wheels to roll and spin, but not to slip. The dynamic is described by the following equations:

$$\dot{x}(t) = v(t) \cos(\theta(t)) \quad \dot{y}(t) = v(t) \sin(\theta(t)) \quad \dot{\theta}(t) = \frac{v(t)}{L} \tan(\phi(t)) \quad (1)$$

where $\phi(t)$ is the average of the right and left front steering angle ϕ_R, ϕ_L respectively. In this work the angles ϕ_R and ϕ_L should be equal. In addition, the steering angle is mechanically limited: $|\phi| \leq \phi_{max}$ with $\phi_{max} = 35^\circ$. v denotes the speed of the vehicle.

2.2 CV-Sensor Simulation Model

The CV-sensor is mounted directly under the right side mirror of the vehicle. It scans the space at the right side of the vehicle up to approximately 8 meters ahead in three areas, i.e. to the right, left and straight ahead. The sensor has one laser beam, and three detectors as shown in Figure 3.

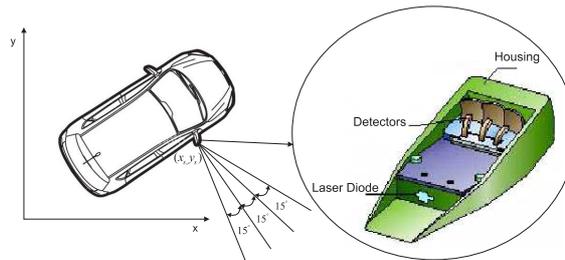


Figure 3: CV-sensor position

Each detector has a cover area of 15 degrees. The sensor emits one beam and captures their reflection with 3 detectors. The send and receive unit uses the signals to calculate the distance between the vehicle and obstacles as well as the closing speed [12]. The location (x_s, y_s) of the sensor is shown in Figure 2 and 3, and is described by the following equations:

$$x_s = x + L \cos(\theta) - \alpha \cos(\theta) + \frac{W}{2} \sin(\theta), \quad (2)$$

$$y_s = y + L \sin(\theta) - \alpha \sin(\theta) - \frac{W}{2} \cos(\theta). \quad (3)$$

The other sensors which are also incorporated in the system are the wheel sensors of the Anti-lock Brake System (ABS). These ABS-sensors supply information regarding the wheel speeds to the ECU which later on can be converted to the vehicle speed.

3 Parking Algorithm

Figure 4 shows the outline of the proposed hierarchical parking assistance system. The parking procedure can be divided into 3 phases, namely 1) Searching for a parking gap, 2) Stop forward movement and 3) Maneuvering phase.

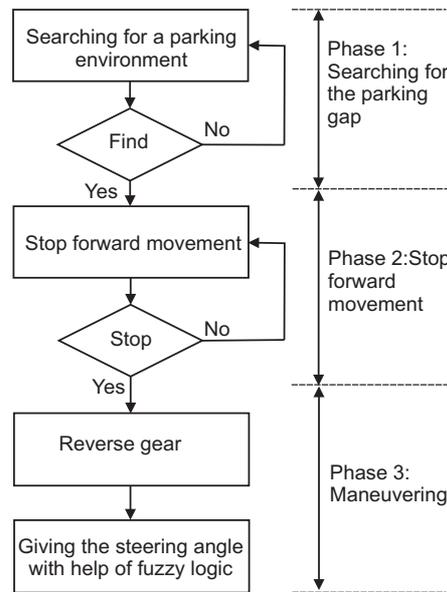


Figure 4: Parking algorithm flow chart

3.1 Parking Phases

3.1.1 Searching for the Parking Gap

The parking space is generally modelled as a rectangular gap between two other vehicles treated as rectangular obstacles as shown in Figure 5. At the beginning, the driver presses the park button to start the parking procedure and the coordinates (x, y, θ) are set to $(0, 0, 0)$. The vehicle has to move straight ahead while keeping an appropriate distance to the others vehicles until the required parking gap is detected. This stage depends on the signals coming from the CV- and the ABS-sensors. The middle detector "2" in the CV-sensor is used for the "searching for the parking gap" phase.

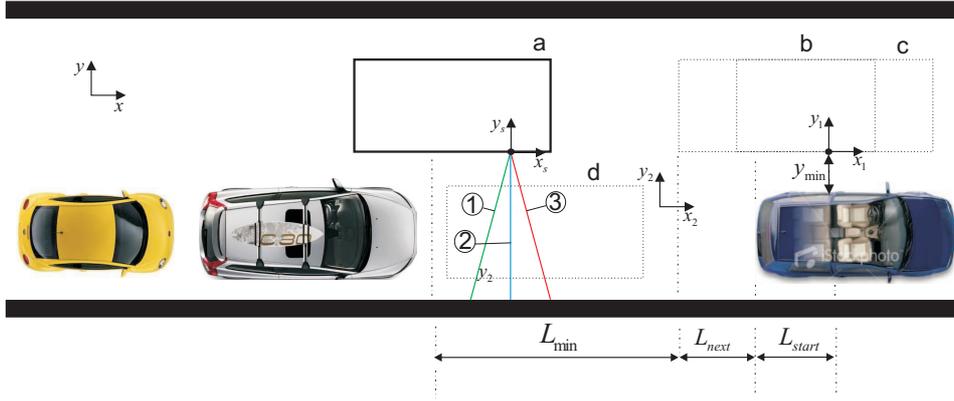


Figure 5: Parking phases

The required total parking distance L_{total} is divided into three parts:

$$L_{total} = L_{min} + L_{next} + L_{start} . \quad (4)$$

Let the minimum longitudinal gap distance required to park a vehicle be L_{min} , where L_{min} depends on the length of the parking vehicle. If the CV-sensor detects the minimum longitudinal distance L_{min} , the vehicle continues to move till it detects the next vehicle. The distance between the end of L_{min} and the next vehicle is called L_{next} . After detecting L_{min} and L_{next} , the vehicle continues to move till it detects a suitable distance in order to start the parking maneuver L_{start} . The determination of L_{start} is very important to prevent the collision between the parking vehicle and the side vehicle. L_{start} depends on the lateral distance y_{min} between the parking vehicle and the side vehicle measured by the CV-sensor. This relationship can be represented by a fuzzy logic system. The membership functions for the input y_{min} and the output L_{start} are shown in Figure 6. Each of them consists of three fuzzy sets, i.e. small, middle and large.

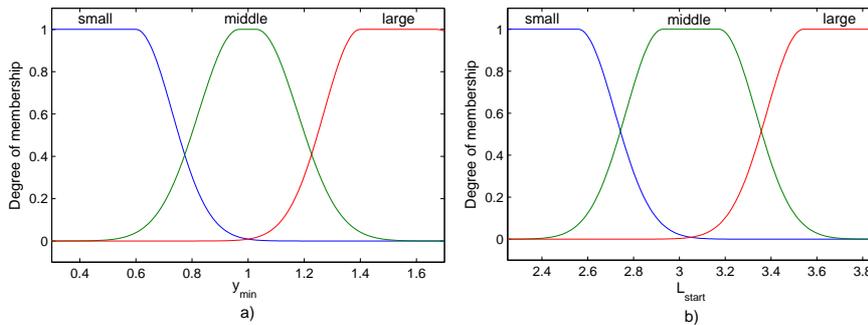


Figure 6: Fuzzy membership functions: a) input y_{min} , b) output L_{start}

The simple rule base for the determination of the starting position L_{start} is given by:

1. If y_{min} is *small* then L_{start} is *small*
2. If y_{min} is *middle* then L_{start} is *middle*
3. If y_{min} is *large* then L_{start} is *large*

For the defuzzification the Center of Gravity method (see e.g. [14]) is used.

During this phase the vehicle moves from a to b as shown in Figure 5.

3.1.2 Stop Forward Movement

Once the total parking length L_{total} is reached, the stop forward movement phase will be activated. In this phase, the vehicle is braked and the signals coming from the ABS-sensors are used to calculate the stopping distance by intergrating the vehicle speed. Moving from b to c represents this phase as illustrated in Figure 5 . After the vehicle stops, the velocity becomes zero and the maneuvering phase will be activated.

3.1.3 Maneuvering

The driver has to chosen the reverse gear and the fuzzy logic algorithm starts to calculate the steering angle by using the CV-sensor, the kinematic vehicle model and the vehicle speed. The kinematic vehicle model can be considered as an observer that calculates the coordinate position (x, y) and orientation θ of the vehicle based on the velocity v and the steering angle ϕ . As shown in Figure 5 moving from c to d represents the maneuvering phase.

3.2 Steering Angle Control Commands Planning

In this section the generation of the steering angle commands is described. These commands are designed by using only two parameters which are computed according to the actual situation in the environment to ensure a collision free path. The advantages of the proposed approach consist in an emulation of the human driver behaviour and in the absence of a complex path planning. The steering commands profile is a semi-smooth function consisting of trigeometric functions and constant segments to generate a feasible trajectory. The steering commands are specific for each step of the parallel parking maneuver as shown in Figure 7.

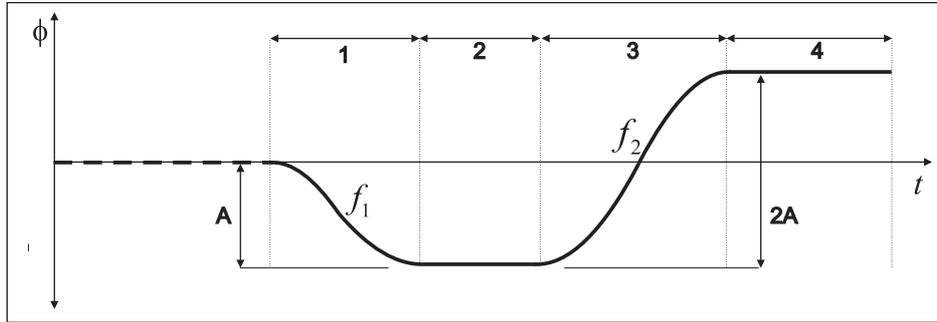


Figure 7: Elementary steering angle commands

The control profile in Figure 7 can be divided into four parts. The formal expression of these four parts is:

$$\phi(t) = \begin{cases} \frac{A}{2}(\cos(2\pi f_1 t) - 1) & 0 \leq t < t_1, \\ -A & t_1 \leq t < t_2, \\ A(\cos(2\pi f_2 t + \pi) + 1) & t_2 \leq t < t_3, \\ A & t_3 \leq t < t_4. \end{cases}$$

1. The first part, that starts at $t = 0$, is a cosine function with Amplitude $A/2$ and frequency f_1 . To start from zero steering angle, $-A$ is added to shift down the cosine function. The cosine curve is selected rather than e.g. a five order polynomial or circles curves since its properties are easier to adjust. The parameters f_1 and A

are determined by using a fuzzy logic approach as described in Section 3.2.1. The end of phase 1 is reached when the steering angle ϕ achieves $-A$. Then part 2 will be activated.

2. Part 2 starts at $t = t_1$ and consists of a straight line by $-A$. The constant value will be given until part three is activated. The activation condition (firing) is $(\theta_{switch} - \theta) < 0$ and it depends on the orientation of the vehicle that is computed by using the vehicle model of Section 2.1. The determination of the value θ_{switch} will be described in Section 3.2.2. .
3. The Part three function $A(\cos(2\pi f_2 t) + 1)$ is activated after the firing condition is satisfied . The function parameters are the amplitude A and the frequency f_2 . The frequency f_1 is multiplied by a factor K_f to get f_2 , ($f_2 = K_f f_1$). Part four will be activated when the steering angle achieves the value A .
4. Part 4 has a straight line by the value A . The constant value will be given until the orientation angle of the vehicle is equal to zero ($\theta = 0$) or when the parking vehicle is too close to the rear vehicle.

3.2.1 Frequency and Amplitude

The frequency f_1 is related to the speed of turning the steering wheel. As known, increasing the cosine frequency will increase the speed of rotation. The frequency of the cosine curve should lead to comfortable vehicle movements as well as should be comfortable for the driver, i.e., without turning fast if the parking gap is large. Furthermore, it is also not necessary to use the full steering wheel angle range in a large parking gap.

Fuzzification: Two fuzzy logic input sets are used for the determination of frequency and amplitude. One of them is the parking gap $X_{max} = L_{min} + L_{next}$ and the second input is the vehicle speed. The domain range values of each input are obtained from the human parking experience and the dimensions of the vehicle. Figures 8a and b shows the input membership functions . The linguistic values for X_{max} are *small*, *middle*, *large* and *vlarge*. The X_{max} values are between 6.1 and 11 meters. *Small*, *middle* and *fast* are the linguistic values for the vehicle speed. The speed range is between 1 and 6 km/h.

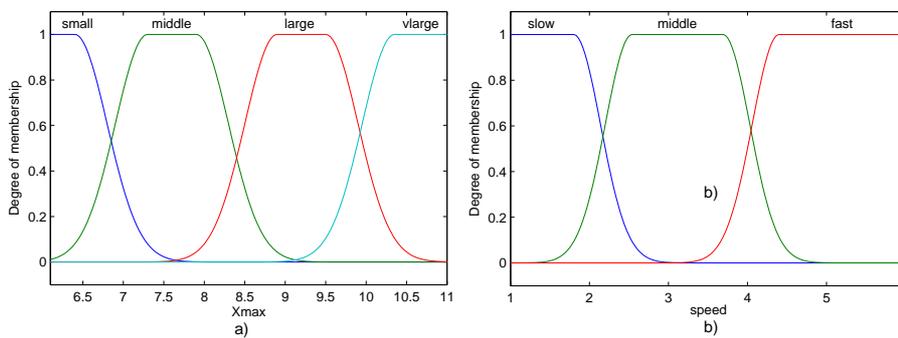


Figure 8: Input membership functions: a) X_{max} , b) Vehicle speed

The corresponding output membership functions for frequency and amplitude are shown in Figure 9.

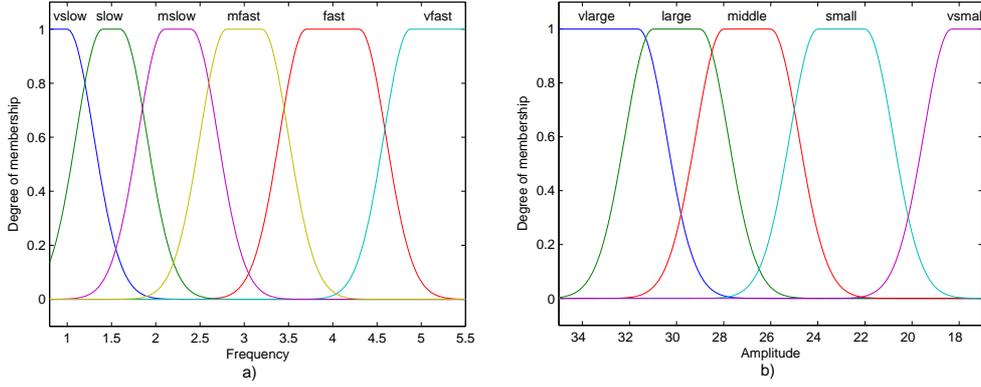


Figure 9: Output membership functions: a) Frequency, b) Amplitude

Fuzzy rule base: The rule base for realizing the turning behaviour of the steering wheel is given in Table 1.

	X_{max}				
	sets	small	middle	large	vlarge
Speed	slow	vfast	fast	slow	vslow
	middle	middle	slow	slow	vslow
	fast		vslow	vslow	vslow

Table 1: Rule base for determination of the Cosine-Frequency

The determination of the amplitude, realizing the behaviour of how far one should turn the steering wheel in different parking environments, is based on Table 2.

	X_{max}				
	sets	small	middle	large	vlarge
Speed	slow	vlarge	large	small	vsmall
	middle	vlarge	large	middle	vsmall
	fast		middle	middle	small

Table 2: Rule base for determination of the Cosine-Amplitude

A complete occupancy in the rule base is not always necessary. As can be seen for the empty cell in the Table 1 and 2, it is impossible to park a vehicle with fast speed and small parking gap.

Inference Mechanism: The membership functions of the outputs frequency and amplitude in each rule are interpreted by using the minimum method. Then, all output membership functions are accumulated using the maximum method [14].

Defuzzification: The resulting fuzzy set obtained from the fuzzy inference mechanism is converted into a numeric (crisp) value in the defuzzification unit. The Center Of Gravity method (COG) is used for the defuzzification [14].

3.2.2 Determination of the Vehicle Switching Orientation

The vehicle orientation angle θ_{switch} to activate the part 3 in Figure 7 is determined by taking into account the length of the existing parking gap $X_{max} = L_{min} + L_{next}$, the

vehicle speed and the distance y_{min} in Figure 5. The angle θ_{switch} defines the important moment to turn the steering wheel in the other direction. From the drivers experience simple rules can be derived: For example, if the parking gap is small and the distance y_{min} is large the vehicle switching orientation should be large.

Fuzzification: Three inputs (X_{max} , speed and y_{min}) are used for determining θ_{switch} . Figure 10a represents the input membership functions for y_{min} . The input membership functions for X_{max} and speed are the same as in Figure 8. The linguistic values for y_{min} are *small*, *middle* and *large*.

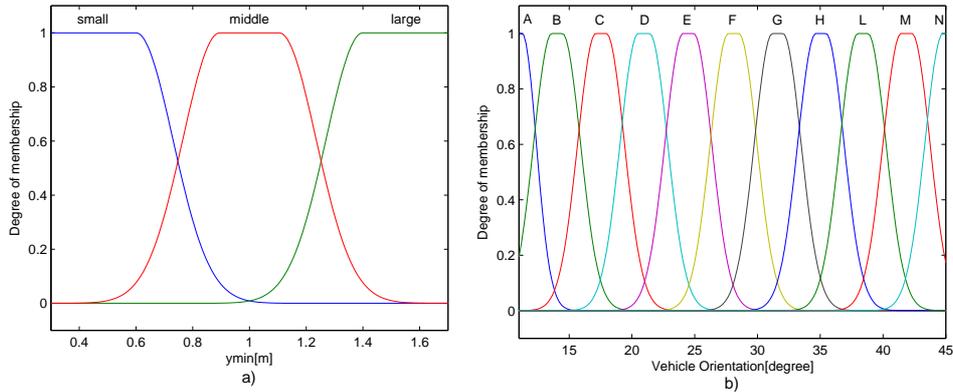


Figure 10: a) Input membership functions of y_{min} , b) Output membership functions of the vehicle switching orientation angle θ_{switch}

The output membership functions for θ_{switch} with their linguistic values (A, B, C, D, E, F, G, H, L, M and N) are shown in Figure 10b. The large number of the output membership functions are chosen to insure the precision and accuracy for determining this angle.

Fuzzy rule base: The rule base for the determination of the vehicle switch orientation is given in Table 3.

y_{min} is small						
		X_{max}				
		sets	small	middle	large	vlarge
speed	slow	H	G	E	C	
	middle	G	E	C	B	
	fast		A	A	A	
y_{min} is middle						
		X_{max}				
		sets	small	middle	large	vlarge
speed	slow	L	H	F	D	
	middle	H	G	E	C	
	fast		C	C	A	
y_{min} is large						
		X_{max}				
		sets	small	middle	large	vlarge
speed	slow	N	M	G	E	
	middle	L	H	F	D	
	fast		D	D	C	

Table 3: Rule base for determination of the switching angle θ_{switch}

Inference Mechanism: The inference mechanism method is the same method as mentioned in section 3.2.1.

Defuzzification: COG is used for the defuzzification of the switching angle θ_{switch} .

4 Simulation

4.1 Simulation Conditions

The properties of the proposed fuzzy parking algorithm based on a CV-sensor are examined by simulations. These simulations were performed using Matlab/Simulink and are carried out with the characteristics of an Audi A4, TDI 1.9l with automatic gear, the CV-sensor model of section 2.2 and the following assumptions:

- Vehicles and obstacles are modelled as rectangles and placed as shown in Fig. 12.
- The vehicle is moving at a low speed, i.e. not more than 6 km/h.
- The simulation is based on right side parking. In case of left side parking, it can be treated similarly by inserting another CV-sensor at the left side mirror.
- There must be one vehicle/obstacle in front of the parking gap. If there is no vehicle/obstacle, the parking algorithm is not able to start the parking maneuver.
- The vehicle operates on a flat ground where a pure rolling contact without slip between the wheels and the ground is assumed.

4.2 Simulation Results

Simulation results for the case ($X_{max} = 7.5$ m, $y_{min} = 0.485$ m, parking speed ≈ 2 km/h) are shown in Figure 11.

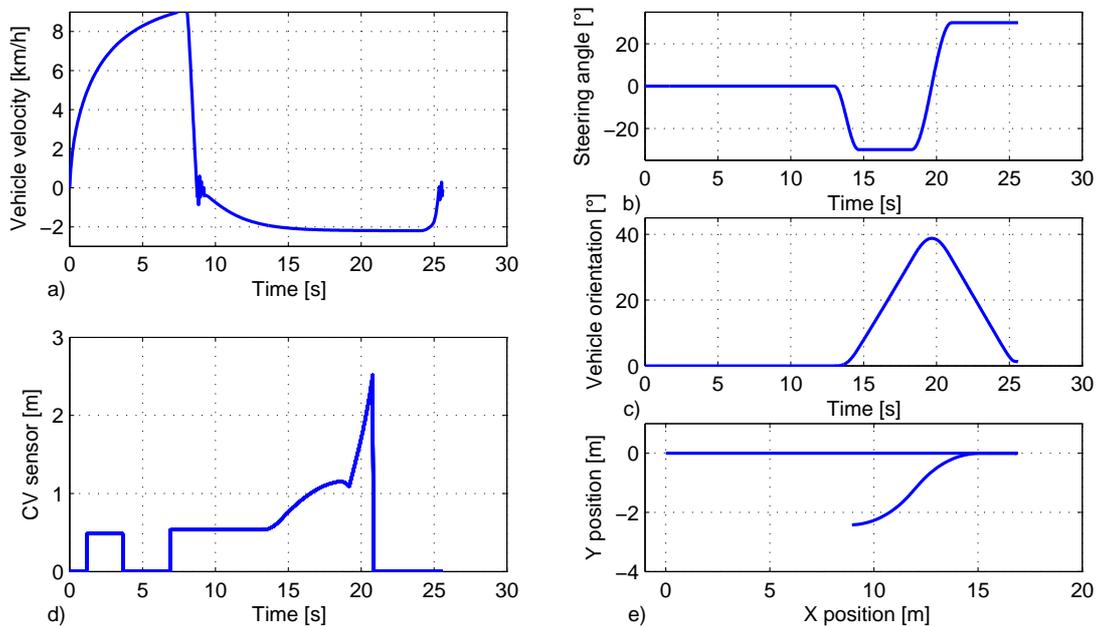


Figure 11: Simulation results for $X_{max} = 7.5$ m, $y_{min} = 0.485$ m: a) Vehicle's velocity (km/h), b) Steering angle (degree), c) Vehicle orientation angle (degree), d) CV-sensor output (m), e) Trajectory of the (x, y) -coordinate

It is assumed that the vehicle moves straight ahead ($y = \text{constant}$ and $\phi = 0$) during the forward movement. Figure 11a,b shows the velocity behaviour of the vehicle and the steering angle ϕ . The forward movement is stopped after 8 seconds. Then the reverse gear is chosen and the parking maneuver begins with the acceleration of the vehicle.

In Figure 11c the developing of the vehicle orientation angle θ is given. The steering wheel is turned right after 13 seconds and the vehicle orientation angle increases up to 40 degree. The turning of the steering wheel into the other direction starts after 19 seconds and the parking maneuver is finished after 26 seconds.

The simulated signal of the middle detector of the CV-sensor is shown in Figure 11d. Here we have the assumption that the CV-sensor gives a length of zero if no obstacle is detected. With this assumption the results of the simulation can directly be compared with the experimentally results. The trajectory of the center of the rear vehicle axle is shown in Figure 11e.

The snapshots of the vehicle motions in Figure 12 give a good overview about the properties of the proposed parking algorithm.

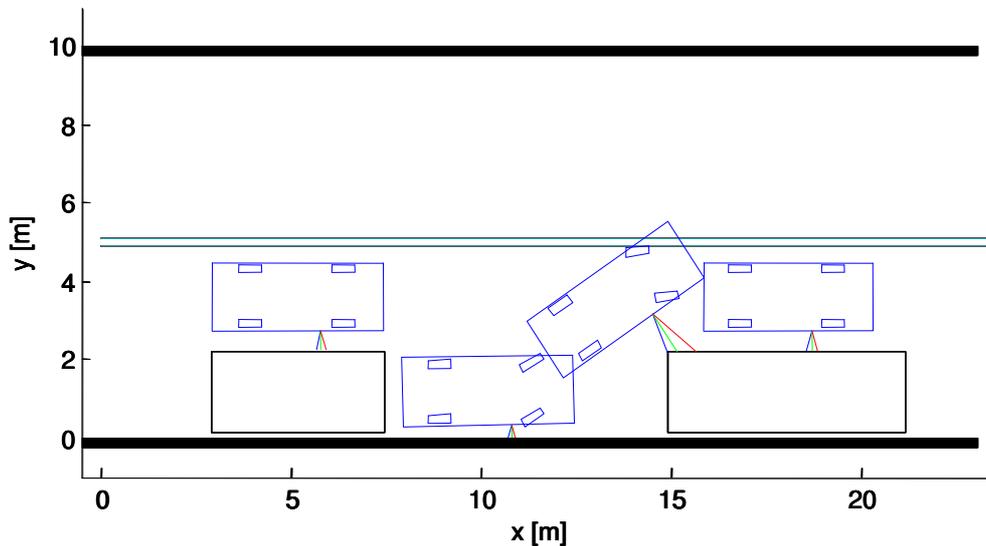


Figure 12: An example of a parking maneuver for $X_{max} = 7.5$ m, $y_{min} = 0.485$ m and a parking speed of ≈ 2.0 km/h

At next, the dependency of the steering control profile from the parking gap length is investigated: Two parking maneuvers are simulated, Figure 13a, with parking gaps of 7.5 and 10.5 meters, where y_{min} and the parking speed are not changed. For the greater parking gap, $X_{max} = 10.5$ m, the amplitude of the steering angle is smaller and therefore the control effort of the driver is lower. On the other hand, a faster rotation of the steering wheel by the driver is necessary when the gap is smaller.

The effect of changing the parking speed on the steering control profile is shown in Figure 13b. It can be seen that the angular speed of the steering wheel increases proportional with the vehicle speed.

Figure 13c shows the influence of changing the lateral distance y_{min} . In case of large values of y_{min} , such as 1.5 m, the constant parts of the steering profile are longer than for small values of y_{min} , as 0.5 m.

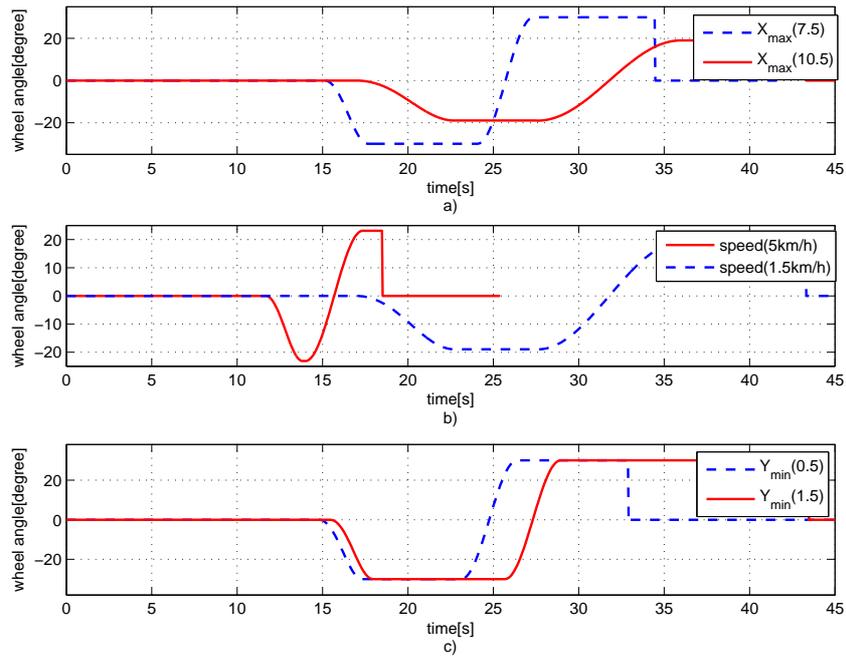


Figure 13: Steering angle control profiles when one parameter is changed. a) The length of the parking gap X_{max} . b) Parking speed. c) The lateral distance y_{min}

5 Experiment

The objective of the experiments is to test the effectiveness and the behavior of the CV-sensor in the parallel parking algorithm and maneuvering strategy. In the future, the parking controller should use the geometry data of the CV-sensor together with the data from the vehicle sensors to steer the vehicle inside the parking gap.

5.1 Experimental conditions

The overall experimental set-up of the parking system is composed of a host computer, a CV-sensor and a dSPACE MicroAutobox. The input datas are vehicle speed and steering wheel angle obtained from the CAN bus and the signals of the CV-sensor. The system possesses an interface on the host computer that display the signals coming from the MicroAutobox using Matlab/Simulink and ControlDesk. For detecting and parking, two



Figure 14: Overview of the experimental set-up

vehicles are placed at the right hand side of the test vehicle as can be shown in Figure 14a. The CV-sensor is mounted under the right side mirror as shown in Figure 14b.

5.2 Experimental Results

The complete detecting and parking experiment is shown in Figure 15. In Figure 15a the output of the CV-sensor during the maneuver is depicted. The outputs of the CV-

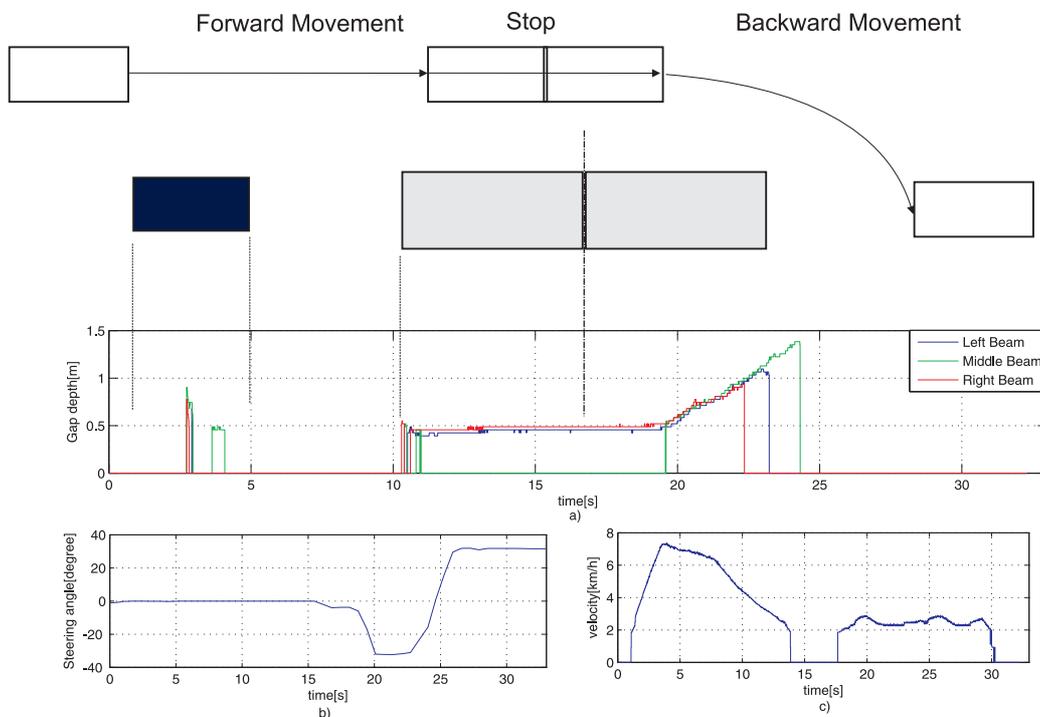


Figure 15: Experimental results corresponding to gap detecting and parking: a) CV-sensor channels. b) Steering angle (degree). c) Absolute value of the Vehicle speed (km/h)

sensor give the distance between the detectors and the next obstacle. A zero value says that no obstacle is in the detecting range of the sensor, that means there are no signals returning back to the detector. There are many noise sources affecting the measurements. They come from variations of the environment lighting, smoothness and the colour of the obstacles. The previous effects play a very important role in the sensor measurements. The first detected obstacle is a vehicle with a dark blue color, while the next obstacle is a silver bus. It can be seen that the vehicle with the dark colour has a poor reflecting of the emitted laser. For light colours, such as silver, the CV-sensor displays a good reflecting of the emitting signals.

The behaviour of the steering angle is illustrated in Figure 15b. The driver turns the steering wheel to the limit (approx. $520^\circ = 1.45$ rotation) after 18 seconds and the front wheels are right steered into their limits (approx. 32°). Three seconds later the wheels are turned left to their maximum when the vehicle reaches a suitable position inside the parking gap. Finally, the vehicle stops when it becomes parallel to the road. As can be seen from the vehicle speed in Figure 15c, the forward movement is from the time 0 to 14 s. Then the vehicle stops to change to reverse gear from the time 14 to 17.5 s. After that, at the time 17.5 s, the vehicle starts to move backward in the direction of the parking gap till it stops at 30 s.

6 Conclusions

This paper presents a new approach for the planning of steering angle control commands for semi-autonomous parking maneuvers. Furthermore, the feasibility of applying an ideal CV-sensor in the parallel parking problem is investigated. Both the dynamical behaviour of the vehicle and the characteristics of the CV-sensor have been analysed, and a planning algorithm has been developed to produce a collision free path which leads to an efficient parking maneuver. The path is planned by using the vehicle speed and the lateral distance to the parked vehicle beside the vehicle. This distance can be provided by a CV-sensor. The method was implemented with Matlab/Simulink, and the simulation results were discussed.

A drawback of the proposed planning approach regarding an online-realization in a vehicle is the estimation of the vehicle orientation θ with the kinematic model (1). A critical input of this model is the vehicle speed since the ABS-ECU provides no vehicle speed signal below 2 km/h (cp. Figure 15). Therefore, in further studies the signals of the CV-sensor should also be used in the parking phases to improve the robustness of path planning. Then the proposed algorithm should be validated in a vehicle by using a steering machine and a human driver.

Literatur

- [1] Paromtchik, I.E.: *Steering and Velocity Commands for Parking Assistance*. Proc. of the 10th IASTED International Conference Robotics and Application, Hawaii, USA, 2004, pp. 178–183.
- [2] Baturone, I.; Moreno-Velo, F. J.; Sanchez-Solano, S.; Ollero, A.: *Automatic Design of Fuzzy Controllers for Car-Like Autonomous Robots*. IEEE Trans. on Fuzzy Systems, Vol. 12, No. 4, 2004, pp. 447–465.
- [3] Li, T.-H.S.; Chang, S.-J.: *Autonomous Fuzzy Parking Control of a Car-Like Mobile Robot*. IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans, Vol. 33, No. 4, 2003, pp. 451–465.
- [4] Sugeno, M.; Murakami, K.: *An Experimental Study on Fuzzy Parking Control using a Model Car*, in: M. Sugeno (ed.) *Industrial Applications of Fuzzy Control*, North-Holland, Amsterdam, 1985, pp. 105–124.
- [5] Michael Kochem, M.: *Ein Fahrerassistenzsystem zur Unterstützung des rückwärtigen Parkvorgangs für Pkw*. VDI Fortschritt-Berichte, Reihe 12, Nummer 590, VDI-Verlag, Düsseldorf, 2005.
- [6] Lee, W.C.: *Beiträge zur Entwicklung eines Fahrerassistenz-Systems für Einparkvorgänge*. Shaker-Verlag, 2006.
- [7] Borenstein, J.; Koren, Y.: *The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots*. IEEE Journal of Robotics and Automation, Vol. 7, 1991, pp. 278–288.
- [8] Lyon, D.: *A Min-time Analysis of Three Trajectories with Curvature and Nonholonomic Constraints Using a Parallel Parking Criterion*. JSME International Journal, Series C, Vol. 46, No. 4. 2003, pp. 1523–1530.
- [9] Murray, R.M.; Sastry, S.S.: *Steering Nonholonomic Systems Using Sinusoids*. Proc. of the 29th IEEE Conference on Decision and Control, Honolulu. 1990, pp. 2097–2101.

- [10] Lo, Y.K.; Rad, A.B; Wong, C.W.; Ho, M.L.: *Automatic Parallel Parking*. Proc. of IEEE Intelligent Transportation Systems Conference, Shanghai, Vol.2, 2003, pp. 1190–1193.
- [11] Pradalier, C.; Vaussier, S.; Croke, P.: *Path Planning for a Parking Assistance System: Implementation and Experimentation*. Australasian Conference on Robotics and Automation, University of New South Wales, 2005, pp. 1240–1340.
- [12] Automotive Continental: *New type of precrash sensor is able to prevent many accidents in urban traffic*. http://www.conti-online.com/generator/www/com/en/continental/portal/themes/press_services/press_releases/products/automotive_systems/gt_pr_2007_04_18_cv_sensor_en.html, 2007.
- [13] Wada, M.; Yoon, K.; Hashimoto, H.; Matsuda, S.: *Parking Assistance System Based on High Accuracy Position Sensing*. Proceedings of the IEEE Int. Vehicle Electronics Conference, (IVEC '99), 1999, pp. 497–502.
- [14] Kiendl, H.: *Fuzzy Control methodenorientiert*. Oldenbourg–Verlag, München, 1997.

Ein Evolutionsverfahren zur Approximation äquivalenter Urbilder von Pareto-optimalen Zielvektoren

Günter Rudolph, Mike Preuss

Lehrstuhl für Algorithm Engineering, Fakultät für Informatik
Technische Universität Dortmund, 44221 Dortmund

E-Mail: {guenter.rudolph, mike.preuss}@tu-dortmund.de

1 Einleitung

In der mehrkriteriellen Optimierung liefert die Menge der Pareto-optimalen Zielvektoren genau solche Lösungen, die bezüglich eines Zieles nur dann verbessert werden können, wenn sie bezüglich eines anderen Zieles Verschlechterungen hinnehmen. Solche Lösungen sind z.B. für den Produktdesigner interessant, da die verschiedenen Ausprägungen hinsichtlich der Ziele auch unterschiedliche Produkte bedeuten. Die Urbilder dieser Zielvektoren sind dann eher für den Produktionstechniker von Bedeutung, da diese angeben, wie das Produkt mit den gewünschten Eigenschaften herzustellen ist. Wählt also der Produktdesigner einen Zielvektor aus, dann kann dieser möglicherweise durch verschiedene Urbilder erzeugt werden. In diesem Fall ergeben sich auch verschiedene Produktionsprozesse, wobei der eine oder andere für den Ingenieur etwa aus Erfahrungsgründen vorzuziehen ist oder weil Lieferengpässe den Einsatz anderer Materialien erfordern.

Üblicherweise erzeugen mehrkriterielle Optimierverfahren aber nur ein Urbild je Zielvektor, so dass äquivalente Urbilder im Bedarfsfall nicht zur Verfügung stehen. Deshalb haben wir in der Vergangenheit Spezialverfahren [1, 2] entwickelt, die nicht wie üblich die Paretofront sondern vielmehr die Paretomenge möglichst gleichmäßig approximieren. Tatsächlich ist es aber selten nötig, für jeden optimalen Zielvektor alle äquivalenten Urbilder zu kennen, da ja vom Produktdesigner auch nur ein einziger Zielvektor ausgewählt worden ist. Der Berechnungsaufwand kann also deutlich reduziert werden, wenn man nur für einen speziellen, vorgegebenen Pareto-optimalen Zielvektor weitere Urbilder findet, sofern sie überhaupt existieren.

Sei $x^* \in \mathcal{X} \subseteq \mathbb{R}^n$ eine Pareto-optimale Lösung des mehrkriteriellen Optimierungsproblems

$$f(x) \rightarrow \min! \quad \text{für } x \in \mathcal{X} \subseteq \mathbb{R}^n.$$

Wir sind an Lösungen aus \mathcal{X} interessiert, die ebenfalls Pareto-optimal und zugleich ungleich von x^* sind: $\{x \in \mathcal{X} : f(x) = f(x^*) \wedge \|x - x^*\| > \varepsilon\}$ für ein $\varepsilon > 0$.

Die Grundidee des hier vorgestellten Verfahrens ist es, nach einer Lösung im Entscheidungsraum zu suchen, die den Abstand zum vorgegebenen Zielvektor im Zielraum minimiert und zugleich den Abstand vom bekannten zugehörigen Urbild maximiert. In [3] wurden diese Aufgabe unter Verwendung von Straffunktionen angegangen, indem der Abstand der Lösungen im Zielraum minimiert und die Nähe zur ursprünglichen Lösung bestraft wurde:

$$F(x) = \|f(x) - f(x^*)\|^2 - \gamma \|x - x^*\|^2 \rightarrow \min!$$

für ein $\gamma > 0$. Hat man derart eine weitere Lösung gefunden, so muss auch die Nähe zu dieser Lösung bestraft werden, indem die Straffunktion und damit letztlich die Zielfunktion modifiziert wird. Insgesamt erhält man also eine Sequenz von Optimierungsproblemen.

Dieser Ansatz funktioniert zwar leidlich, birgt aber auch eine Reihe von Problemen. Die Idee, im Verlauf der Optimierung erhaltenes Wissen über gefundene Optima zu benutzen, um den weiteren Suchprozess durch Modifikationen an der Zielfunktion zu steuern, erscheint zunächst als relativ einfach und attraktiv. Allerdings ist es dazu nicht nur nötig, die ungefähre Position der Optima zu kennen. Auch die Lage, Ausdehnung, und die Form der zugehörigen Attraktionsbassins muss abgeschätzt werden, da ein Strafterm mit zu großer räumlicher Wirkung sonst benachbarte Bassins (und möglicherweise sogar deren Optima) teilweise überdeckt, während eine zu klein gewählte Korrektur lediglich den inneren Teil eines Bassins vor der Optimierung versteckt. Das „Hineinfallen“ in ein Bassin würde dadurch nicht verhindert, stattdessen würde lediglich frühzeitige Stagnation in diesem Bereich hervorgerufen. Wird ein additiver Korrekturterm mit unpassender Form gewählt, besteht außerdem die Gefahr, künstliche Optima zu erzeugen, die in der eigentlichen Zielfunktion gar nicht vorhanden sind. Eine gute Korrektur erfordert also ein sehr genaues Wissen über einen bestimmten Bereich der Zielfunktion, das „on-the-fly“ nur schwer zu gewinnen ist. Im Hinblick auf diese möglichen Schwierigkeiten erscheint es geraten, nach alternativen Ansätzen zu suchen. Ein solcher bietet sich sofort, wenn man die Maximierung des Abstandes zur ursprünglichen Lösung nicht durch Strafterme sondern auf natürliche Weise im mehrkriteriellen Sinne behandelt.

In Abschnitt 2 diskutieren wir noch einmal die inhärenten Probleme beim monokriteriellen Ansatz mit Straffunktionen, bevor wir in Abschnitt 3 den bikriteriellen Ansatz vorstellen und einer ersten empirischen Überprüfung unterziehen. Im letzten Abschnitt fassen wir unsere Ergebnisse zusammen und schließen mit einem Ausblick hinsichtlich des Ausbaus dieses Ansatzes.

2 Monokriterieller Ansatz mit Straffunktionen

In der Arbeit von Dickhöfer [3] wurde empirisch an einfachen Testfällen überprüft, ob sich der Optimierungsprozess durch die Modifikation der Zielfunktion mit additiven und multiplikativen Straftermen so steuern lässt, dass er weitere Lösungen einfacher auffinden kann. Als Optimierverfahren kam dabei ein generischer EA zum Einsatz.

Die eingesetzten additiven Strafterme basierten auf $\frac{1}{x}$ und $\cos x$ -Funktionen, die passend zur Lage der Optima parametrisierbar waren. Die multiplikativen Straffunktionen leiteten sich von den beiden ursprünglich für die eindimensionale Optimierung entwickelten Ansätzen der *Filled-Function* (nach Ge und Qin [4]) des *Tunneling* (nach Levy und Montalvo [5]) ab. In empirischen Untersuchungen ergab sich ein Vorteil für die multiplikativen Straffunktionen, allerdings mussten auch diese über die geschickte Wahl der Parameter an die Testfunktionen angepasst werden, um gute Ergebnisse zu erhalten. Bei den additiven Termen gestaltete sich diese Anpassung noch schwieriger, da sie auf unpassende Parameterwahlen wesentlich empfindlicher reagierten. Zusammenfassend lässt sich festhalten, dass die auf Straftermen basierenden Ansätze nur dann gut verwendbar sind, wenn schon relativ viel über die Zielfunktion bekannt ist. Ohne Wissen über die ungefähre Lage und

Zahl der Optima bzw. der gesuchten Lösungen lassen sie sich vor allem bei mehrdimensionalen Problemen kaum verwenden.

3 Bikriterieller Ansatz mit Nebenbedingungen

3.1 Grundidee der „Multikriterialisierung“

Der Begriff der „Multikriterialisierung“ (engl. multiobjectivization) von Problemen mit einfacher Zielsetzung wurde von Knowles et al. [6] geprägt. Die Hypothese bei diesem Ansatz besteht darin, dass das ursprüngliche monokriterielle Problem durch Hinzufügen weiterer Kriterien / Zielfunktionen einfacher zu lösen ist. Man könnte diese zusätzlichen Zielfunktionen dann als Helferfunktionen (engl. helper-functions) [7] interpretieren. Natürlich wird es darauf ankommen, wie diese Helferfunktionen beschaffen sind, um tatsächlich für ein mehrkriterielles Optimierverfahren hilfreich zu sein. Ob eine solche hilfreiche Helferfunktion immer existiert scheint eine offene Frage zu sein. Es sind Beispiele konstruiert worden, die zeigen, dass zusätzliche Zielfunktionen beweisbar hilfreich aber auch kontraproduktiv sein können [8, 9, 10]. Das ist allerdings noch kein Argument gegen die Existenz und Nützlichkeit von Helferfunktionen: Man muss sie wohl nur finden!

Wir vertreten die Auffassung, dass Helferfunktionen dann nutzbringend sind, wenn sie zusätzliches Problemwissen kodieren, das von einem Optimierverfahren verwertet werden kann. Bei einem praktischen Problem aus der Thermodynamik [11] haben wir diese Vermutung stützen können, weil unsere Helferfunktion die Bassins der lokalen Optima vergrößerte und zudem *implizit* kodierte, in welchen Bereichen des Suchraumes wir nicht suchen wollten. Dies hat das Auffinden der gewünschten Optima enorm erleichtert. Deshalb werden wir diesem Prinzip der Kodierung von Zusatzwissen in Helferfunktionen auch hier folgen.

3.2 Problemformulierung

In dem von uns betrachteten Szenario gehen wir davon aus, dass bereits eine Approximation der Paretofront eines mehrkriteriellen Problems berechnet wurde und dass der Produktdesigner einen Zielvektor dieser Approximation ausgewählt hat. Das verwendete Optimierverfahren liefert natürlich sofort das Urbild x^* dieses Zielvektors $f(x^*)$ im Entscheidungsraum \mathcal{X} .

Der Produktingenieur möchte nun wissen, ob es weitere Urbilder $x \neq x^*$ gibt, die auf den gleichen Zielvektor $f(x^*)$ abbilden. Eine notwendige Bedingung für ein solches Urbild $x \in \mathcal{X}$ lautet $\|f(x) - f(x^*)\| = 0$ für irgendeine Norm $\|\cdot\|$. Damit erhalten wir die erste Zielfunktion

$$F_1(x) = \|f(x) - f(x^*)\|^2 \rightarrow \min! \quad (1)$$

Durch einen Multistart-Absatz können sich nun möglicherweise weitere äquivalente Urbilder identifizieren lassen. Allerdings wird man auch die ursprüngliche Lösung finden — und dies umso wahrscheinlicher, je größer das Attraktionsgebiet des lokalen Optimums für das eingesetzte Optimierverfahren ist. Möchte man vermeiden, dass die bereits bekannte Lösung wiederholt angelaufen wird, dann muss dem Optimierverfahren geeignet „mitgeteilt“ werden, dass dort nicht gesucht werden soll. Dies kann man erreichen, wenn

dieser Bereich schlechtere Zielfunktionswerte bekommt. Das erreichen wir, indem wir Urbilder mit gleichem Zielfunktionswert in F_1 dadurch unterscheiden, dass solche mit größerem Abstand im Urbildraum zur ursprünglichen Lösung vorgezogen werden. Also muss $\|x - x^*\|$ maximiert werden. Wir formulieren eine äquivalente Zielfunktion für eine Minimierungsaufgabe:

$$F_2(x) = -\|x - x^*\|^2 \rightarrow \min! \quad (2)$$

Offensichtlich kann die Paretofront und -menge bei unbeschränkten Problemen auch unbeschränkt sein. Erfreulicherweise existiert aber eine natürliche Nebenbedingung, die eine Beschränkung der Mengen erzwingt. Wir sind ja an Urbildern interessiert, deren Zielvektoren möglichst ähnlich sind. Also können wir leicht eine maximale Schranke $\theta > 0$ bezüglich Zielfunktion (1) festlegen:

$$g(x) = F_1(x) - \theta < 0 \quad (3)$$

Durch (1), (2) und (3) ist das mehrkriterielle Hilfsproblem beschrieben, wobei (2) die Rolle der Helferfunktion spielt. Sie kodiert das Wissen, dass von x^* entfernte Punkte zu suchen sind.

3.3 Algorithmus

Das hier vorgeschlagene Verfahren ist eine hybride Metaheuristik: Zuerst wird ein evolutionärer mehrkriterieller Optimier-Algorithmus (EMOA) auf das Hilfsproblem (1) - (3) angewendet. Dieser stoppt nach einer zuvor festgelegten Anzahl von Zielfunktionsauswertungen. Die erhaltene Approximation der Paretomenge des Hilfsproblems wird nun bezüglich des ersten Zielkriteriums (1) aufsteigend sortiert. Für eine zuvor festgelegte Anzahl von Urbildern dieser sortierten Approximation wird ein skalares lokales Optimierverfahren bezüglich des ersten Zielkriteriums (1) gestartet, um

1. das Optimum genauer zu lokalisieren und um
2. festzustellen, ob man gegen ein bereits bekanntes Optimum konvergiert.

Konkret haben wir für unsere erste Validierung des Ansatzes den NSGA-II als EMOA und Hooke & Jeeves als lokales skalares Verfahren eingesetzt. Beim NSGA-II wurden die Strategieparameter $\eta_c = 15$, $\eta_m = 10$, $p_c = 0.6$ und $p_m = 1.0$ gewählt.

3.4 Ergebnisse

3.4.1 Testproblem TWO-ON-ONE

TWO-ON-ONE steht für den strukturell einfachsten möglichen Fall eines mehrkriteriellen Problems, bei dem die Paretofront mehrfach von einzelnen Teilen der Paretomenge abgedeckt wird (Abbildung 1). Diese werden durch die Multimodalität einer der beiden Zielfunktionen hervorgerufen, die in diesem Fall von einem Polynom 4. Ordnung gebildet wird, während die andere Funktion einfach aus einer in der Mitte angeordneten zweidimensionalen Parabel besteht. Die empirische Untersuchung in [1] kam zu dem Schluss, dass die getesteten mehrkriteriellen EA (NSGA-II, SPEA2, SMS-EMOA) nicht in der

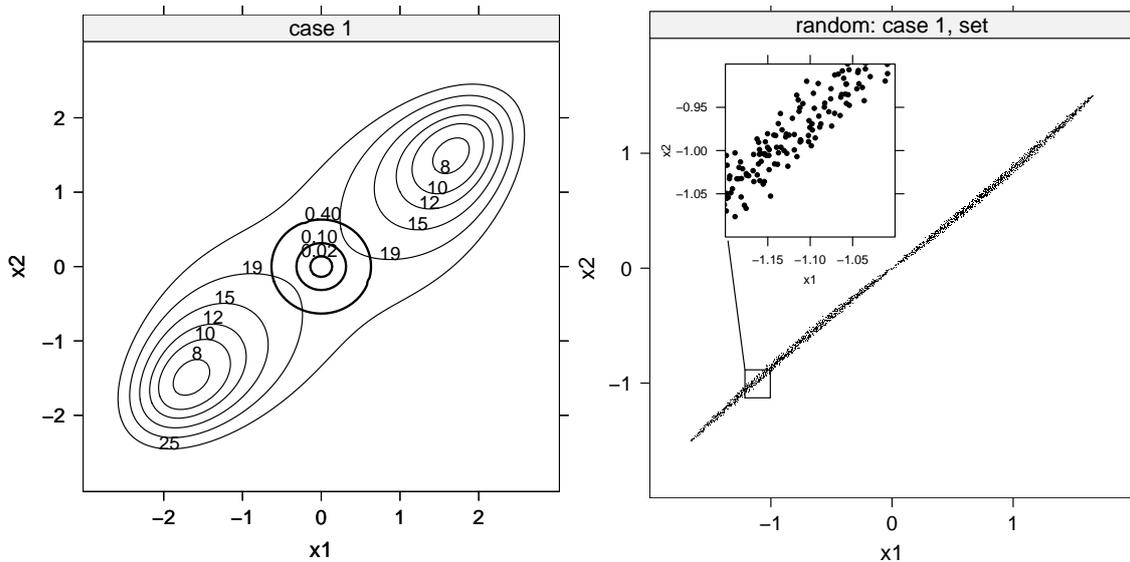


Bild 1: Links: Die Konturen der beiden Zielfunktionen des TWO-ON-ONE im Entscheidungsraum; die erste Funktion erzeugt die beiden äußeren Peaks, die zweite den in der Mitte liegenden. Rechts: Die Paretomenge besteht aus einem linken und einem rechten „Flügel“, wobei sie eigentlich nur eine Linie bildet, die aber durch (algorithmisch durchaus relevante) Ungenauigkeiten des Sampling-Prozesses verbreitert erscheint. Jeder „Flügel“ deckt die Paretofront vollständig ab. Diese ist konvex und sieht ähnlich aus wie die in Abbildung 4 für das SYM-PART Problem gezeigte.

Lage waren, beide Teilmengen der Paretomenge gut zu besetzen und eine starke Tendenz zum vollständigen Verlust einer Teilmenge bestand.

In der Abbildung 2 sieht man das Resultat eines typischen Laufes des NSGA-II mit 100 Individuen nach 250 Generationen: Augenscheinlich ist die Paretofront gut abgedeckt, so dass der Produktdesigner ausreichend Information für eine Entscheidung zur Verfügung hat. Angenommen, der Produktdesigner wählt aus der konkreten Approximation den Zielvektor $f(x^*) = (-17.4765, 4.97609)'$ mit dem vom NSGA-II gelieferten zugehörigen Entscheidungsvektor $x^* = (-1.595774, -1.558715)'$, dann wird aus Abbildung 1 und der Punktsymmetrie des Problems sofort ersichtlich, dass sich das zugehörige äquivalente Urbild nicht in der Approximationsmenge befindet.

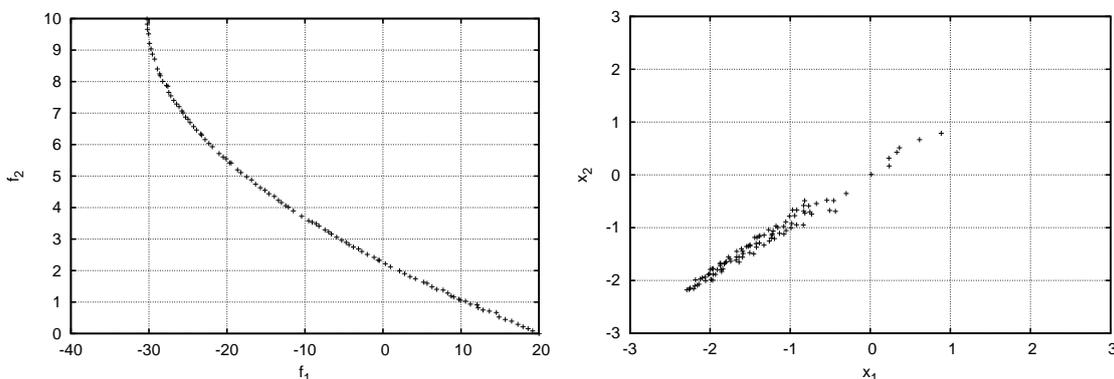


Bild 2: Resultat eines typischen Laufes des NSGA-II mit 100 Individuen nach 250 Generationen. Links: Gut abgedeckte Paretofront. Rechts: Unvollständig abgedeckte Paretomenge.

Um ein äquivalentes Urbild zu finden, starten wir unser hybrides Verfahren, indem zunächst der NSGA-II mit 20 Individuen über 100 Generationen im Bereich $\mathcal{X} = [-5, 5]^2$

und maximaler Abweichungsschranke $\theta = 100$ sucht. Das Ergebnis eines typischen Laufes ist in Abbildung 3 dargestellt.

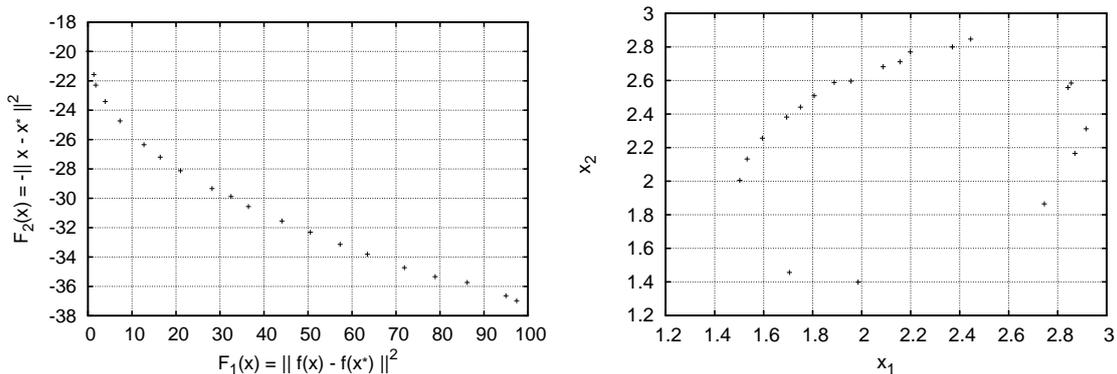


Bild 3: Typischer Lauf des NSGA-II für das Hilfsproblem bezüglich TWO-ON-ONE. Links: Nur die in der Grafik am weitesten links dargestellten Lösungen sind von Interesse, da für die übrigen die Abweichung vom gewünschten Zielvektor schon zu groß wird. Rechts: Zugehörige Urbilder des Hilfsproblems. Die Lösung mit kleinstem F_1 -Wert ist $x = (1.705102, 1, 456496)'$.

Wir schauen uns die 5 bezüglich F_1 besten Urbilder an, deren Zielvektoren in der linken Grafik der Abbildung 3 am weitesten links anzutreffen sind. Tabelle 1 fasst die Ergebnisse zusammen. Anschließend starten wir Hooke & Jeeves zur genaueren Lokalisation und stellen fest, dass alle Startlösungen aus Tabelle 1 zur gleichen Lösung konvergieren. Offensichtlich existiert nur ein äquivalentes Urbild. Beim nächsten Testproblem wollen wir die Eigenschaften unseres Ansatzes beleuchten, wenn mehrere äquivalente Urbilder existieren.

Rang	F_1	F_2	x_1	x_2
1	0.003433	-19.98722	1.705102	1.456486
2	1.326537	-21.57204	1.985328	1.398945
3	1.752043	-22.29680	1.502349	2.004770
4	3.937764	-23.40602	1.532254	2.132012
5	7.266334	-24.73223	1.594493	2.256312

Tabelle 1: Die 5 besten Lösungen des Hilfsproblems für TWO-ON-ONE. Werden die Urbilder (x_1, x_2) aus der Tabelle als Startwerte in einen lokalen Optimierer für F_1 verwendet, so wird stets das gleiche äquivalente Urbild lokalisiert.

3.4.2 Testproblem SYMPART

Dieses Testproblem ist insofern eine Weiterentwicklung von TWO-ON-ONE, als die Anzahl der gleichzeitig existierenden äquivalenten Lösungen für jeden Punkt auf der Paretofront auf 9 erweitert wurde (Abbildung 4). Außerdem sind die beiden einzelnen Zielfunktionen bis auf eine Translation vollständig identisch und lediglich quadratisch, also analytisch besser handhabbar. Das Verhalten herkömmlicher EMOA auf dieser Funktion wurde zum ersten mal in [2] untersucht, und erwartungsgemäß konnten die EMOA nicht sehr viele Teilmengen der Paretomenge finden und halten. Das ist allerdings auch nicht überraschend, da ihnen dazu die Mechanismen fehlen, sie sind für solch eine Situation schlicht nicht geeignet. Das Problem wurde für einen auf der CEC 2007 stattfindenden Wettbewerb auf eine größere Variablenzahl im Entscheidungsraum erweitert [12]. Keiner

der teilnehmenden Algorithmen war in der Lage, im Mittel deutlich mehr als eine der Teilmengen zu besetzen.

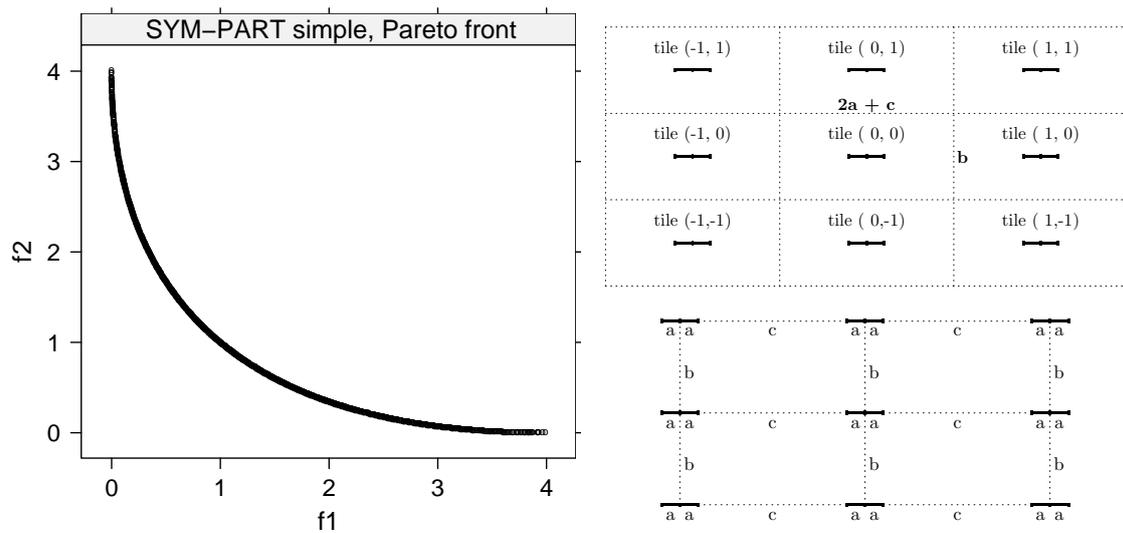


Bild 4: Links: Paretofront des SYMPART Problems. Rechts: Schematischer Aufbau im Entscheidungsraum, der hier aus 9 symmetrischen „Kacheln“ besteht. Jede „Kachel“ enthält einen Teil der Paretofront, der die Paretofront vollständig abdeckt und von 2 einfachen quadratischen Funktionen erzeugt wird. Die Teilmengen der Paretofront haben jeweils die Länge $2a$, die vertikalen bzw. horizontalen Abstände zur nächsten Teilmenge werden durch b bzw. c charakterisiert.

Wir wollen nun untersuchen, wie sich das hybride Verfahren bei multiplen äquivalenten Urbildern verhält. Abbildung 5 zeigt einen typischen Lauf eines NSGA-II mit 100 Individuen nach 250 Generationen im Bereich $\mathcal{X} = [-20, 20]^2$. Die Paretofront scheint gut repräsentiert zu sein; schließlich bildet jede der 9 Teilmengen der Paretofront vollständig auf die Paretofront ab. Die Approximation der Paretofront ist nur unvollständig gelungen. Lässt man den NSGA-II noch weitere Generationen laufen, so verschlechtert sich durch Drifteffekte die Approximation der Paretofront zusehends; insofern handelt es sich hier noch um eine gute Approximation.

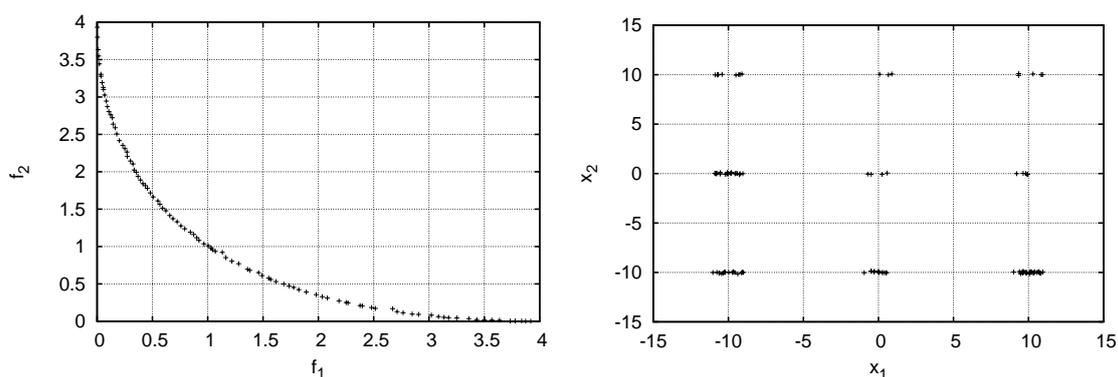


Bild 5: Typischer Lauf eines NSGA-II mit 100 Individuen nach 250 Generationen im Bereich $\mathcal{X} = [-20, 20]^2$. Links: Gut abgedeckte Paretofront. Rechts: Unvollständig abgedeckte Paretofront.

Angenommen, der Produktdesigner hätte den Zielvektor $f(x^*) = (0.757702, 1.27587)'$ mit zugehörigem Urbild $x^* = (9.870459, -10.00175)'$ ausgewählt. Wir starten von unserem hybriden Verfahren zunächst den NSGA-II mit diesmal 100 Individuen (da wir

mehrere äquivalente Urbilder erwarten) für maximal 300 Generationen und $\theta = 0.5$. Die nachfolgenden Grafiken in Abbildung 6 geben die Situation nach 100, 200 und schließlich 300 Generationen wieder.

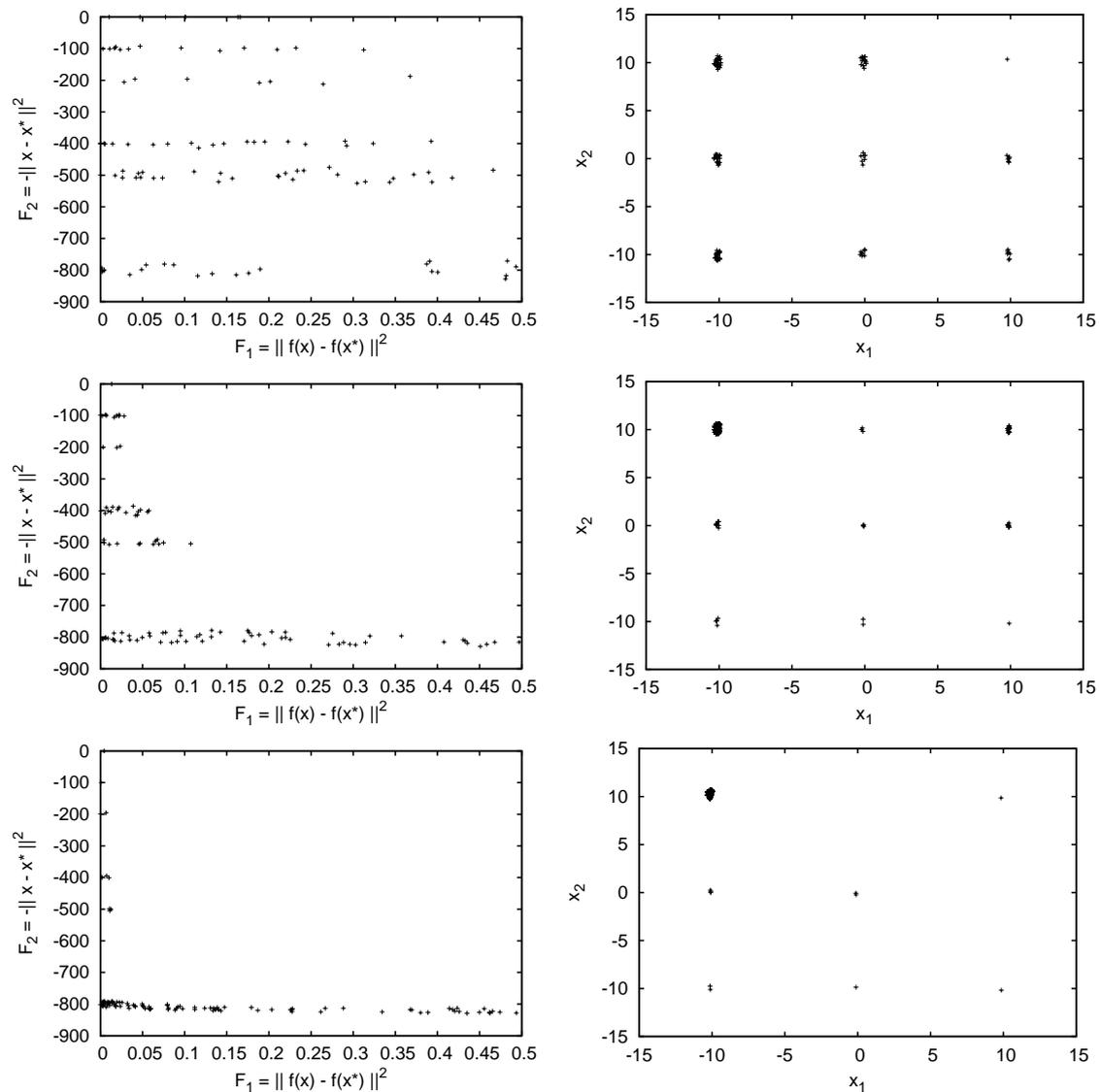


Bild 6: Typischer Lauf eines NSGA-II mit 100 Individuen nach 100 (oben), 200 (mitte) und 300 (unten) Generationen im Bereich $\mathcal{X} = [-20, 20]^2$. Linke Spalte: Paretofront-Approximationen des Hilfsproblems. Rechte Spalte: Zugehörige Urbilder.

Zunächst sind die Individuen noch weit von der Paretofront entfernt, aber es sind schon klare Tendenzen in Richtung äquivalenter Urbilder zu erkennen. Würde man nach 100 Generationen von den bezüglich F_1 besten Individuen lokale Suchen starten, so würden die 8 weiteren äquivalenten Urbilder identifiziert werden. Je länger der NSGA-II jedoch läuft, umso mehr äquivalente Urbilder „gehen verloren“, d.h., Individuen in diesen Bereichen werden von der Selektion verworfen. Das liegt zum einen an Drifteffekten und zum anderen an der Tatsache, dass bei genauer Approximation bzgl. F_1 (im Idealfall gilt $F_1 = 0$) Individuen mit kleineren F_2 -Werten (also größerer Entfernung zu x^*) solche mit geringerer Entfernung zu x^* dominieren. Dies ist ein methodisches Problem, das sich vermutlich durch den Einsatz eines ϵ -optimalen Archivs [13] lösen lassen könnte.

4 Zusammenfassung

Wir haben ein evolutionäres Verfahren vorgestellt, das das Auffinden äquivalenter Urbilder von Pareto-optimalen Zielvektoren eines mehrkriteriellen Problems mit Hilfe eines restringierten, bikriteriellen Hilfsproblems angeht. Es konnte exemplarisch gezeigt werden, dass der Ansatz bei Existenz genau eines weiteren äquivalenten Urbildes gut und effizient funktioniert. Sollen aber simultan mehrere äquivalente Urbilder gefunden werden, dann zeigen sich methodische Schwächen, die uns allerdings durch die Verwendung eines ϵ -optimalen Archivs heilbar erscheinen. Nichtsdestoweniger haben wir hier nicht mehr als ein „proof-of-principle“ vorstellen können. Es gilt noch abzuklären, was bei höherdimensionalen Suchräumen und/oder Zielräumen passiert. Die simultane Detektion von mehreren äquivalenten Urbildern dürfte jedenfalls noch einige Modifikationen erfordern, um diesen Ansatz praxistauglich zu machen.

Literatur

- [1] Preuss, M.; Naujoks, B.; Rudolph, G.: Pareto Set and EMOA Behavior for Simple Multimodal Multiobjective Functions. In: *Parallel Problem Solving from Nature (PPSN IX)* (Runarsson, T.; et al., Hg.), S. 513–522. Berlin: Springer. 2006.
- [2] Rudolph, G.; Naujoks, B.; Preuss, M.: Capabilities of EMOA to Detect and Preserve Equivalent Pareto Subsets. In: *Proceedings of the 4th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2007)* (Obayashi, S.; et al., Hg.), S. 36–50. Berlin: Springer. 2007.
- [3] Dickhöfer, T.: *Entwicklung und Vergleich von evolutionären Algorithmen zur Approximation äquivalenter Urbilder von Pareto-optimalen Zielvektoren*. Diplomarbeit, Technische Universität Dortmund, Fakultät für Informatik. 2007.
- [4] Ge, R. P.; Qin, Y. F.: A class of filled functions for finding global minimizers of several variables. *Journal of Optimization Theory and Applications* 54 (1987) 2, S. 241–252.
- [5] Levy, A. V.; Montalvo, A.: The Tunneling Algorithm for the Global Minimization of Functions. *SIAM Journal on Scientific and Statistical Computing* 6 (1985) 1, S. 15–29.
- [6] Knowles, J.; Watson, R.; Corne, D.: Reducing local optima in single-objective problems by multi-objectivization. In: *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization (EMO 2001)*, S. 269–283. Berlin: Springer. 2001.
- [7] Jensen, M.: Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimisation. *Journal of Mathematical Modelling and Algorithms* 3 (2004) 4, S. 323–347.
- [8] Neumann, F.; Wegener, I.: Minimum spanning trees made easier via multi-objective optimization. *Natural Computing* 5 (2006) 3, S. 305–319.

- [9] Brockhoff, D.; Friedrich, T.; Hebbinghaus, N.; Klein, C.; Neumann, F.; Zitzler, E.: Do additional objectives make a problem harder? In: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO 2007)*, S. 765–772. New York: ACM Press. 2007.
- [10] Handl, J.; Lovell, S.; Knowles, J.: Multiobjectivization by Decomposition of Scalar Cost Functions. In: *Parallel Problem Solving from Nature (PPSN X)* (Rudolph, G.; et al., Hg.), S. 31–40. Berlin: Springer. 2008.
- [11] Preuss, M.; Rudolph, G.; Tumakaka, F.: Solving Multimodal Problems via Multi-objective Techniques with Application to Phase Equilibrium Detection. In: *Proceedings of the 2007 IEEE International Congress on Evolutionary Computation (CEC2007)* (Tan, K.; et al., Hg.), S. 2703–2710. Piscataway (NJ): IEEE Press. 2007.
- [12] Huang, V. L.; Qin, A. K.; Deb, K.; Zitzler, E.; Suganthan, P. N.; Liang, J. J.; Preuss, M.; Huband, S.: Problem Definitions for Performance Assessment of Multi-objective Optimization Algorithms. Techn. Ber., Nanyang Technological University, Singapore. URL [CEC-07-TR-emoa-performance.pdf](#). 2007.
- [13] Schütze, O.; Laumanns, M.; Tantar, E.; Coello Coello, C.; Talbi, E.-G.: Convergence of stochastic search algorithms to gap-free Pareto front approximations. In: *Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation (GECCO 2007)*, S. 892–901. New York: ACM Press. 2007.

Optimization of Artificial Central Pattern Generators with Evolutionary Algorithms

**Christian Bauer, Sebastian Braun, Yang Chen,
Wilfried Jakob, Ralf Mikut**

Institute for Applied Computer Science, Forschungszentrum Karlsruhe GmbH
76021 Karlsruhe, Postfach 3640
Tel.: +49 7247 82 6672
Fax: +49 7247 82 5786
E-Mail: christian.bauer@iai.fzk.de

Abstract

In contrast to classical engineering approaches for the generation of movements in robots or prostheses, approaches to this subject inspired by neurophysiological circuits are in advance. One of the key structures of interest in this area is the Central Pattern Generator (CPG) which has been identified to be the source of movement generation in mammals. This neural circuit is capable of generating cyclic muscle activation patterns completely independent from the brain as shown by Brown in the early 20th century [1, 2]. In the past years this knowledge was adapted to the challenges in movement generation and control in robotics, for example for passive dynamic walkers by P. Manoonpong [3].

In this paper the results of a CPG implementation and its optimization to model the human movement generation are presented, which were aiming at a design to be as biologically realistic as possible. Due to the fact, that biological models are the fundamentals of the simulated CPG, a large number of parameters needs to be set, which results in a very complex configuration process to get an optimized CPG behavior. Based on previous works [4–7] an Evolutionary Algorithm was employed to optimize the rhythm-generation of the CPG. For this optimization the tool GLEAMKIT was used [8]. Electromyographic (EMG) data recorded from a human male during walking on a treadmill have been utilized for the fitness function as well as for evaluation purposes.

1 Introduction

In computer sciences, robotics research and development of prostheses, artificial neural networks are nowadays common tools to increase the possibilities and power of all different kinds of systems, e.g. object classification, learning, speech recognition or movement generation. To build these systems a lot of different approaches are possible which have several advantages and disadvantages depending on specific tasks for which they are developed. These approaches can be split into two main strategies, on the one hand side there is the 'classic technical' way of building an artificial neural network which evolved out of computer science. Few neurons are organized in layers and connected with each other by weighted data streams. On the other hand there is the biological inspired approach which relies on the findings in neural sciences and uses differential equations for simulating behavior of biological neurons during the data transmission in animals or humans. One of the most common models for such a biological realistic simulation is the Hodgkin-Huxley Model, published in 1952 [9].

One of the most popular concepts in humanoid robotics to generate coordinated leg movement is the one of Central Pattern Generators (CPGs). These are neural structures which have been discovered in the 20th century [10] by testing the locomotive capacities of decerebrated cats. The ability of cats, or mammals in general, to be able to walk or trot in an almost normal walking pattern, even with their cerebellum or spine surgically separated from the motor cortex, indicate, that basic gait patterns are encoded in the neural networks of the spine. These neural structures are called Central Pattern Generators and have been found to be responsible for generation of cyclic muscle activation patterns such as respiration, chewing or leg movement during walking. These CPGs are one part of the neural system which controls posture and movement, which is hierarchically organized and of a highly complex, nonlinear structure (see Figure 1).

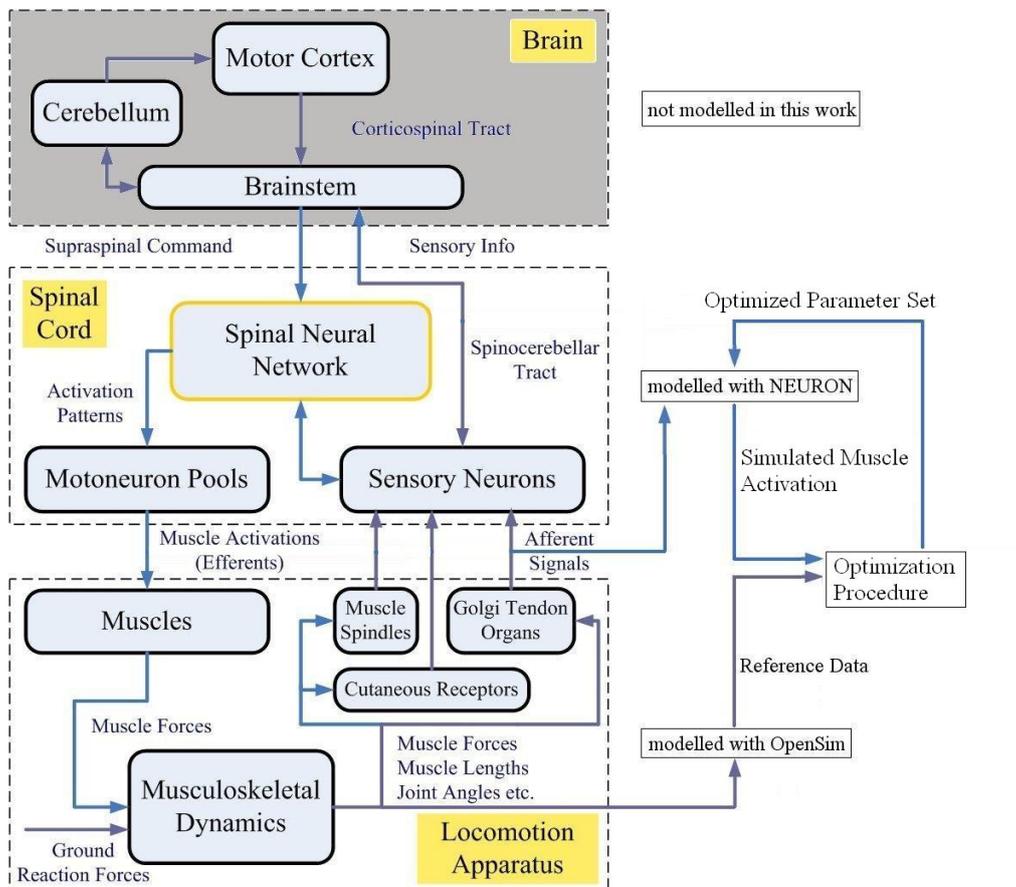


Figure 1: Locomotion control system in humans

This concept has been adapted by many scientists for movement generation in robotics. Matsuoka published his first concepts of artificial neural oscillators in 1985 and 1987 [11, 12]. The concept of CPG has been applied to passive dynamic walkers developed by F. Wörgötter and P. Manoonpong [3, 13] (RUNBOT) or Mori et al. [14]. These approaches have in common that the CPGs used are to a certain degree simplified artificial neural networks which just adapt the idea of CPGs for the task of bipedal locomotion.

In the research group at the Institute for Applied Computer Science (IAI) at the Forschungszentrum Karlsruhe the subject of interest is movement generation for prostheses and robots. The approach used here is of a different kind as the aforementioned and uses

biological realistic simulations of neurons and neural networks to simulate CPGs and to generate muscle activation patterns in a way which is biologically as realistic as possible. The first results have been presented in [4] containing simulation results of a CPG with a heuristically found parameter set, without the necessary coupling to a biomechanical simulation environment.

In this paper a more sophisticated simulation of a CPG is presented as well as the application of the Evolutionary Algorithm (EA) GLEAM to optimize the parameters of the CPG network so that the muscle activation pattern becomes as human like as possible. For this several different types of software tools had to be combined to operate in an integrated simulation and optimization platform. This platform has been evaluated regarding different sizes of the parameter sets (four or 14 parameters) and the scalability of the optimization process regarding distributed computing.

This paper is structured as follows. In Section 2 the used tools and principles are explained as well as how these are combined to form the system. In addition to that the methods used for optimizing the system are introduced in this section. Following this, the Evaluation section (Section 3) addresses the evaluation and test scenarios with which the system has been confronted. The results of these tests are presented and explained. Following this section, a discussion of the results can be found in Section 4 and future work is motivated. Section 5 concludes this paper with a short summary of the contents of this paper.

2 System and Methods

In this section the software tools used to build the simulation system are introduced as well as the proceedings and methods of simulating a CPG. Following that, it is explained how the CPG-generated activation patterns are compared to the reference data as well as the procedure to optimize these activation patterns.

2.1 Architecture for Simulation and Optimization

The platform developed consists of two main parts, the simulation and the optimization, and can be seen in Figure 2.

For the simulation of a CPG the neural simulation software NEURON published by a group of researchers at the University of Yale¹ is used. This tool is a programming environment specialized on generating and designing neural structures consisting of several neurons. The strength of NEURON lies in its ability to calculate the differential equations, which describe neural behavior. In principle, these neurons are laying at rest and the transmission of information along these neurons is done by action potentials, which means that specified currents of potassium and calcium ions are diffusing through the cell membranes sending a current along the axons. The duration and amplitude of these are identical to the according currents in biological neuron. The same is valid for the membrane and action potentials and the settling rate of the neurons. Using NEURON an artificial neural network was implemented which contains all parts that are responsible for movement generation in mammals. In Figure 3 the structure of this CPG network, the involved neurons and their interconnection is illustrated.

¹<http://www.neuron.yale.edu/neuron/>

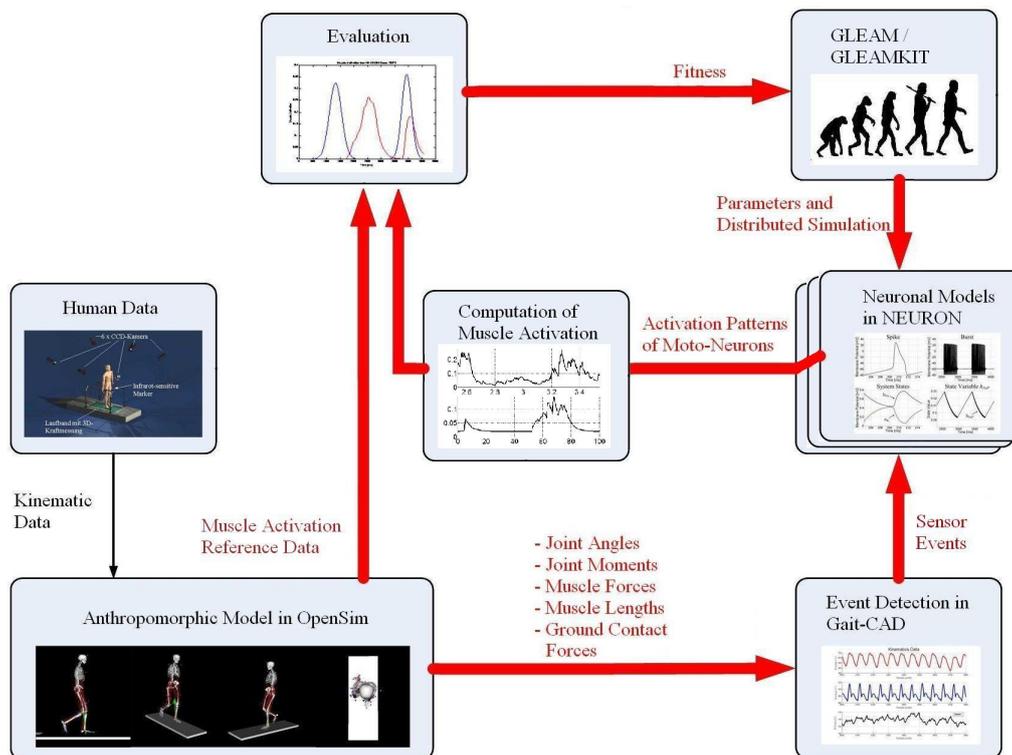


Figure 2: Overview of the software tools building the system

Former research results [10, 15, 16] prove that neural structures like this one exist in mammals and are the crucial part in the generation of cyclic motion patterns. Coupling these CPGs the generation of more sophisticated locomotion patterns is possible. To achieve this, each joint has to be controlled by one CPG, which means that several muscle activations for the flexors and extensors of this joint have to be generated by this CPG.

The biomechanics simulation Software OpenSim is a freeware tool for the analysis of muscle forces in motion capturing data recorded from patients or test persons to calculate direct and inverse kinematics and dynamics. It uses free available models from the commercial software SIMM and is available as open source from the projects homepage². Using OpenSim the motion data which has been recorded at the Orthopädische Klinik in Heidelberg, Germany, was used to calculate the muscle activation signals from the test person. The motion data for this calculation was recorded from a 23 years old human male via a commercially available 3D motion analysis system. For the marker placement the Helen Hayes marker set was used and the motion was performed on a treadmill running with different speeds. In addition to motion data, EMGs of eight prominent muscles in both thighs and shanks were recorded.

To use the benefit of biological realistic simulation the activation patterns generated by the CPG programmed in NEURON have to be as close to real human muscle activation patterns as possible. The pattern generation in NEURON can be adjusted by a large number of parameters which influence the output in a very complex way. To match this artificial activation to the recorded human muscle activation an Evolutionary Algorithm is employed to find a best matching result for the desired purpose.

²https://simtk.org/project/xml/downloads.xml?group_id=91

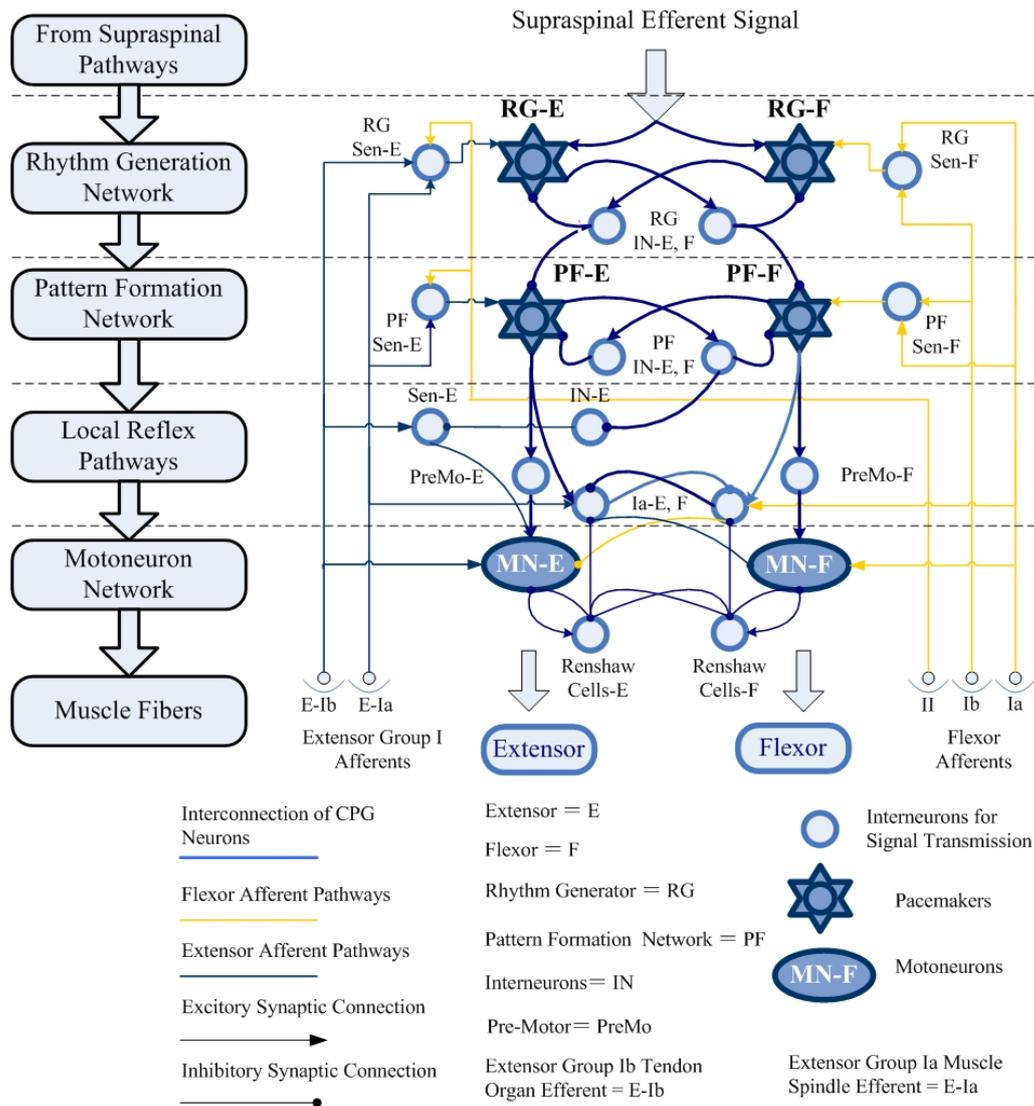


Figure 3: Structure of the neural connection of a CPG

EAs are able to deal with a large number of parameters and control figures like it is the case of the presented simulation. The simulation framework is extended by the GLEAM engine (General Learning Evolutionary Algorithm and Method [7, 17]) and GLEAMKIT which have been developed at the IAI [18, 19]. The former one is an open source implementation of the evolutionary algorithm GLEAM with interfaces to Windows and Matlab [18], the latter one is a Matlab toolbox forming a GUI for GLEAM for configuring the EA and the optimization runs. In addition to that an online visualization frontend is provided. These two toolboxes are used to apply the EA GLEAM to optimize the parameter sets of the simulation to match the generated artificial muscle activation patterns as close as possible to the muscle activation patterns of human locomotion.

The described simulation and optimization system is completed by the Matlab toolbox Gait-CAD for filtering and evaluation tasks. This toolbox has been developed at the IAI and is specialized for filtering, visualization and data mining in large datasets and time series [20].

Considering the fact that each simulation to evaluate one genome model of the EA takes

a certain amount of time to be completed (8 to 50 seconds, depending on the computer), the concept of distributed computing was used to enhance the performance and time effectiveness of the EA application. To achieve this the IAIDataShare software, which is a Client-Server-Architecture for assessing data based jobs for distributed computing was coupled with the system. IAIDataShare consists of one server instance which distributes the jobs to registered clients which calculate the simulation depending on the individuals of this EA-population. Each of these contains a NEURON parameter set and computes the resulting activation patterns of moto-neurons. If finished the results are written to a file which then is copied to the results directory by the server.

2.2 Methods - Generation of Muscle Activation

The problem of crosstalk between the recorded EMG signals made it impossible to differentiate exactly between the muscle activation of the targeted muscle and the activation of surrounding muscles involved in the locomotion. Therefore in our application, OpenSim has to be called to extract muscle activations and events such as ground contact from the kinematic data. This has to be done only once to get a reference dataset for the optimization process.

The CPG is built up of different neurons, implemented in NEURON. Former research has shown that a combination of neurons modeled with the Hodgkin-Huxley (HH) model [9] and Integrate-and-Fire neurons [21] is the way of choice in respect of computation efficiency and closeness to reality [5]. The HH-Model was used to build up the pacemakers and moto-neurons whereas the Integrate-and-Fire model was applied for building the interneurons. The advantage coming with the Integrate-and-Fire neuron model is that its computation time is proportional to the number of events and independent of the number of neurons involved, because no numerical integration is required but an analytical solution is used instead. This type of model is especially useful for discrete event simulation, which are characterized by only time relevant events such as spikes [21].

This model works in a very simple way. Mathematically it can be described by

$$\text{if spike comes, then } m_{IntF} = m_{IntF} + w \quad (1)$$

$$\text{if } m_{IntF} > 1, \text{ then } \begin{cases} \text{deliver a spike event} \\ \tau \frac{dm_{IntF}}{dt} + m_{IntF} = 0 \end{cases} \quad (2)$$

where the system dynamics is determined by a single membrane state variable m_{IntF} . If the neuron receives a spike event from the connected neuron, the membrane state value is increased by w , defined as the synaptic weight. If m_{IntF} exceeds 1, the neuron fires and a spike event is delivered. The membrane state variable decays exponentially with the time constant τ after firing. Refractory period can be easily added to the model by defining for how long the neuron can not receive new spike events after firing.

In contrary the HH neuron model is mathematically more complex and describes the membrane potentials of single neurons very realistic. Its general form is as follows:

$$C_m \cdot \frac{dV_m(t)}{dt} = - \sum_i^{n_{Ion}} I_{Ion,i}(t) - \sum_i^{n_{Syn}} I_{Syn,i}(t) - \sum_i^{n_{Ext}} I_{Ext,i}(t) \quad (3)$$

in which $C_m = 1\mu F/cm^2$ is the membrane capacity, and V_m the membrane potential. $I_{Ext,i}$ represents external influences like afferent inputs or flexor/extensor phases, $I_{Syn,i}$ is the synaptic current built as the product of the maximum conductance $\bar{g}_{Syn,i} = 0.05mS/cm^2$ of the synapse with the connection weight $w_{Syn,i}$ and time relevant variables ($t_{ls,i}$ for the time of the last spike and $\tau_{Syn,i} = 5ms$ as time constant for the synaptic current, n is the number of the connected source neurons).

$$I_{Syn,i}(t) = \bar{g}_{Syn,i} \cdot w_{Syn,i} \cdot \exp\left(\frac{-t - t_{ls,i}(t)}{\tau_{Syn,i}}\right) \cdot (V_m(t) - E_{Syn,i}) \quad (4)$$

The differentiation of inhibitory and excitatory synaptic currents is achieved by different values of reversal potentials $E_{Syn,i}$ which are about -40mV and -10mV respectively.

The ion currents $I_{Ion,i}$ are used to model the characteristic behaviors of a neuron by a combination of the three different types of involved ion channels, Sodium, Potassium and leakage. To model the CPG typical behavior the persistent sodium current I_{NaP} has been incorporated. Together with the three former currents of the HH model (I_{Na} , I_K and I_L for the sodium, potassium and leakage, respectively) the ion currents of a pacemaker neuron, key part of a CPG, can be modeled:

$$I_{Ion,Burster}(t) = I_{Na}(t) + I_K(t) + I_L(t) + I_{NaP}(t) \quad (5)$$

$$I_{Na}(t) = g_{Na,max} \cdot m_{Na}^3(t) \cdot h_{Na}(t) \cdot (V_m(t) - E_{Na}) \quad (6)$$

$$I_K(t) = g_{K,max} \cdot n_K^4(t) \cdot (V_m(t) - E_K) \quad (7)$$

$$I_L(t) = g_L(V_m(t) - E_L) \quad (8)$$

$$I_{NaP}(t) = g_{NaP,max} \cdot m_{NaP}(t) \cdot h_{NaP}(t) \cdot (V_m(t) - E_{Na}) \quad (9)$$

The constants E_x ($E_{Na} = 55mV$; $E_K = -85mV$; $E_L = -55mV$) are the reversal potentials of the ion currents, and g_x ($g_{Na,max} = 120mS/cm^2$; $g_{K,max} = 48mS/cm^2$; $g_{NaP,max} = 13mS/cm^2$; $g_L = 3mS/cm^2$) the maximum ionic conductances. m and n are the activation variables, h the inactivation variables of the neurons, representing the permeability of the ion channels in values ranging from 0 to 1.

To complete the CPG network the moto-neurons have to be modeled by using the following currents. The ion currents in soma include

$$I_{Ion,Moto,Soma}(t) = I_{Na}(t) + I_K(t) + I_L(t) + I_{CaN}(t) + I_{K(Ca)}(t) \quad (10)$$

$$I_{CaN}(t) = g_{CaN,max} \cdot m_{CaN}^2(t) \cdot h_{CaN}(t) \cdot (V_m(t) - E_{Ca}) \quad (11)$$

$$I_{K(Ca)}(t) = g_{K(Ca),max} \cdot m_{K(Ca)}(t) \cdot (V_m(t) - E_K) \quad (12)$$

$$g_{K(Ca),max} = \frac{C_a}{C_a + K_d} \quad (13)$$

$$\frac{dC_a}{dt} = f_{Ca}(-\alpha_{Ca} \cdot I_{Ca}(t) - k_{Ca} \cdot C_a) \quad (14)$$

and the dendrite current is composed of

$$I_{Ion,Moto,Dend}(t) = I_{CaN}(t) + I_{K(Ca)}(t) + I_{CaL}(t) \quad (15)$$

$$I_{CaL}(t) = g_{CaL,max} \cdot m_{CaL}(t) \cdot (V_m(t) - E_{Ca}) \quad (16)$$

where the constants are set by the expert to:

- $g_{CaL,max}$ $0.33mS/cm^2$
- $g_{CaN,max}$ $14mS/cm^2$ in soma, $0.3mS/cm^2$ in dendrite
- $g_{K(Ca),max}$ $1.1mS/cm^2$ in soma, $5mS/cm^2$ in dendrite

These are the newly added ion conductances of the L-like calcium current I_{CaL} , N-like calcium current I_{CaN} and the calcium-dependent potassium current $I_{K(Ca)}$.

To make the implemented CPG as biologically correct as possible the parameters to influence its behavior have to be chosen very carefully. Considering the fact that a very large number of parameters exists and in addition to that each change to one parameter affects the system and therefore the other parameters, an optimization of the system by hand comes not into account.

2.3 Methods - Computation of Muscle Activation

In NEURON the equations of Section 2.2 have been used to implement pacemaker and moto-neurons. Once implemented, neural behavior and muscle activation can be simulated with NEURON and frequency coded muscle activation patterns, comparable to those measurable by EMG, are produced. Because of the crosstalk effect observed when measuring EMGs in a clinical environment, reference data has to be computed using OpenSim. The relevant information is coded in the amplitude in the reference data and therefore the NEURON results have to be converted into an amplitude coded signal to make it comparable. From former clinical research it is known that the muscles, target of our investigation, are activated by two neural bursts during walking. This is equivalent to two spikes in the amplitude coded signal.

This leads to a calculation that has to be applied to get a more feasible representation of the signal to make it comparable with the recorded human data. For this a Gauss-representation algorithm is applied to the signal (see Figure 4) which encodes the frequency based information (upper part of Figure 4) into a data point consisting of three variables A , B and C (middle part of Figure 4). In the frequency encoded data the important values for calculation of the moto-neuron action potentials are the following:

- I : Average value of spike intervals of a burst
- D : Burst duration
- T_{MB} : Middle time of a burst

The applied Gaussian filter is defined as:

$$Y = \sum_i^N Y_i = \sum_i^N A_i \cdot \text{Exp}\left(-\frac{(t - B_i)^2}{C_i}\right) \quad (17)$$

The Gaussian variables A , B and C are $A = I/S_A$, $B = T_{MB}$ and $C = D/S_D$ with S_A and S_D as scaling factors for the amplitude and duration respectively and N as the number of bursts and i as the bursts index.

With this representation it is possible to compare the simulated CPG activation pattern to the human muscle activation from the clinical data of OpenSim (lower part of Figure 4).

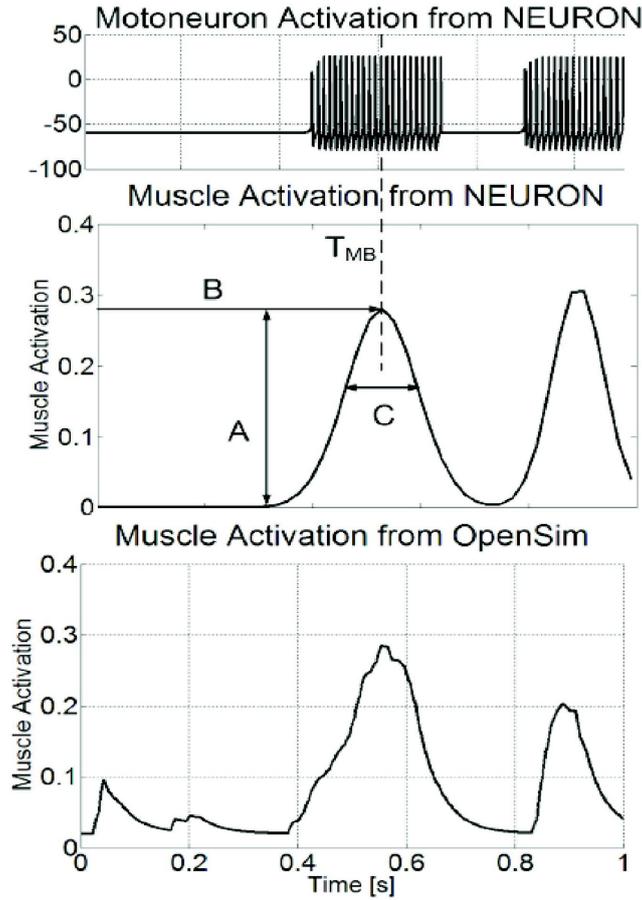


Figure 4: Making NEURON signals (upper third) comparable by applying a Gauss transformation (middle part) to the reference graph (lower third)

2.4 Methods - Evaluation Function

For the optimization process parameter sets with different numbers of parameters are considered to be the individuals of one population of the EA. It was decided to use one small parameter set consisting of the four most important parameters and one large parameter set consisting of 14 parameters. The most remarkable influence to the behavior of moto-neurons is considered to be the time constant $\tau_{Syn,i}$ and ionic conductance of the persistent sodium current g_{NaP} . By changing these two values for the flexor and extensor muscles in one joint the muscle activation and therefore the movement itself is changed. In the large parameter set values for potassium, sodium, chloride, N-like calcium and potassium controlled calcium conductances (g_{Na} , g_K , g_L , g_{CaN} and $g_{K(Ca)}$) are added for flexor and extensor muscles. These constants have been introduced in the equations of Section 2.2 and their values, set by an expert are given. Using distributed computing by IAIDataShare, a simulation of the CPG in NEURON is run with each parameter set.

Once the simulation of the CPG muscle activation is finished (left area of Figure 5) the results are written to a result file.

Comparing this result to the reference data, a fitness value is calculated and assigned to the according parameter set individuum.

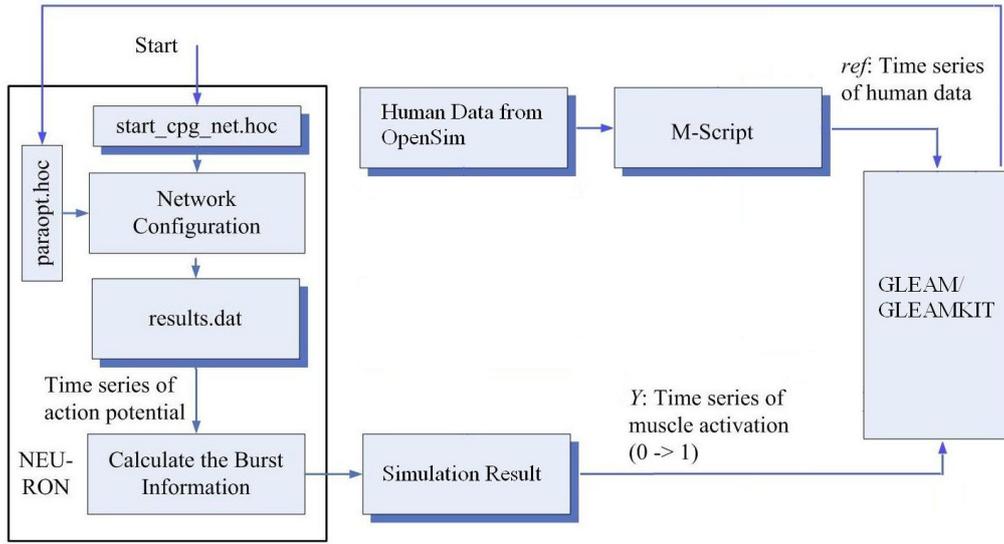


Figure 5: Connection between NEURON simulation and the EA

The fitness calculation is first done separately for flexor and extensor muscle activation. The results are then combined to get an overall fitness value for the actual simulated parameter set. First the sum of deviations of the values of each point of the simulated graph regarding the corresponding point in the reference graph is calculated:

$$Q_{sum} = \sum_{k=0}^m |x_N[k] - x_R[k]| \quad (18)$$

The location (X-axis: x_1 and x_2) and amplitude (Y-axis: y_1 and y_2) of the two maxima for the extensor and flexor muscles are the other four criteria which are used as a measurement for the matching of simulation result and reference graph. N and R in the indices assign the values to the simulation graph and reference graph, respectively. The developed fitness function takes into account that the importance of each of these four criteria influences the matching of the two graphs in different ways.

$$Q_{ges} = \alpha_1 \sum_{n=1}^2 \sum_{k=0}^m (|x_{n,N}[k] - x_{n,R}[k]|) + \alpha_2 |x_{1,n,N} - x_{1,n,R}| + \alpha_3 |x_{2,n,N} - x_{2,n,R}| + \alpha_4 |y_{1,n,N} - y_{1,n,R}| + \alpha_5 |y_{2,n,N} - y_{2,n,R}| \quad (19)$$

These criteria are weighted by weight factors α_i which leads to the fitness function for the whole optimization.

3 Evaluation and Results

For the evaluation of the complete system two different types of parameter sets have been used by the EA platform to generate activation patterns by the CPG. The first parameter

set consists of only four parameters and therefore the influence to the generated activation pattern is more limited than with the second parameter set which uses 14 parameters leading to a larger search space.

Each optimization run with the EA lasted nine to ten hours which corresponds to 160 generations and several different runs have been performed. Three examples shall be depicted in detail in this section:

- Run 1: small parameter set with weighting $\alpha_1 = 10, \alpha_2$ to $\alpha_5 = 1$
- Run 2: small parameter set with weighting α_1 to $\alpha_5 = 1$
- Run 3: large parameter set with weighting α_1 to $\alpha_5 = 1$

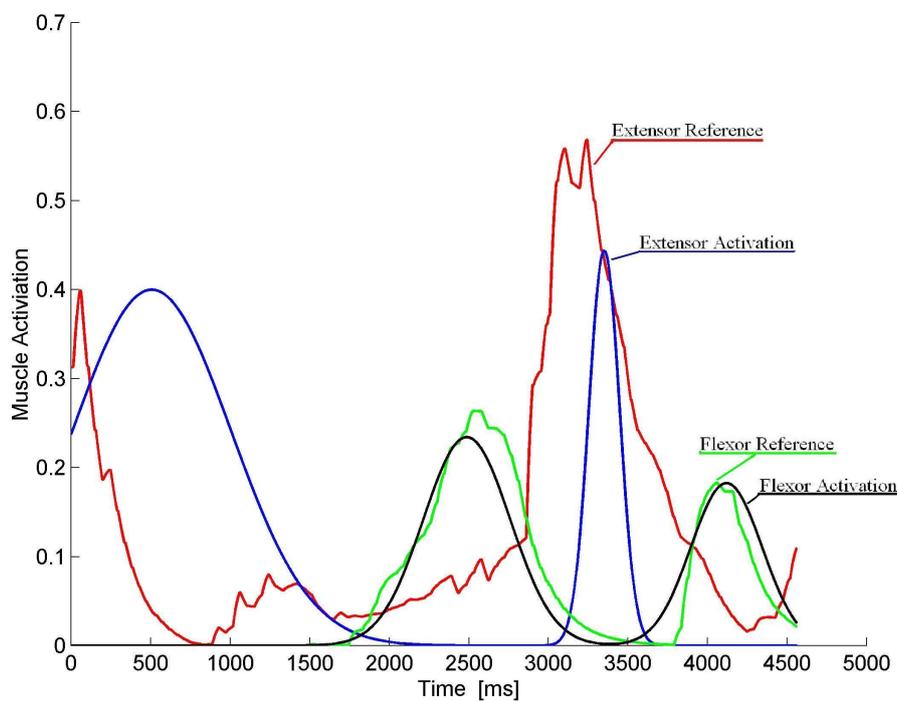


Figure 6: Result of optimizing 14 parameters

In addition to these three runs which were designed to test the quality of the optimization, the scalability of the EA was tested doing several optimization runs on different workstations and processors.

Table 1 shows the fitness values of the three mentioned runs. Two fitness values are shown, differing by the weights used for calculating the fitness function. The bold written ones are the ones of the weights used during the optimization process, the others are the results of the fitness function with the different weights applied to the result.

Comparing the first and second run the resulting fitness values are identical. Comparing these two to the third run, it is obvious that switching to the greater number of parameters improves the result significantly. In conclusion, no matter what parameter set is used

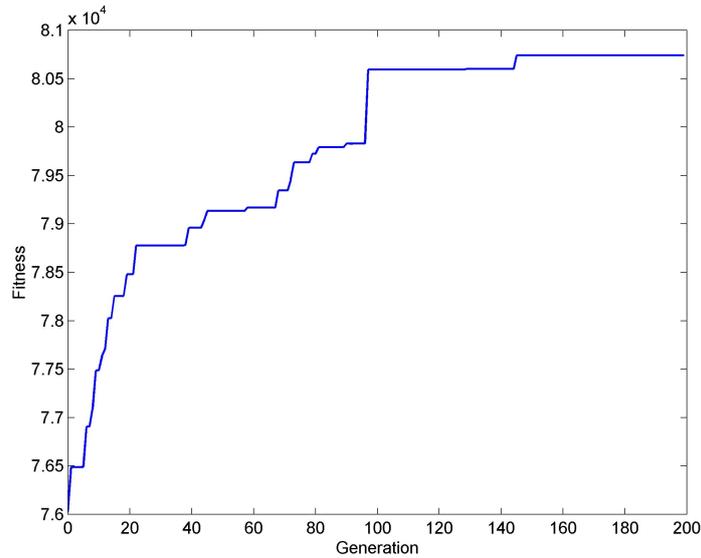


Figure 7: Fitness over Generation

Run	Weights	
	$\alpha_1 = 10, \alpha_2 \text{ to } \alpha_5 = 1$	$\alpha_1 \text{ to } \alpha_5 = 1$
Start by Expert	22323	40262
1. Run	43886	78937
2. Run	43886	78937
3. Run	54166	83907

Table 1: Runs with different weights

for optimization an improvement to the original parameter setting done by an expert is obvious.

For the test how the process is scaling three different configurations were used:

- A two processor machine with four cores each. Only one core used (Scenario A and C)
- A two processor machine with four cores each. All cores used (Scenario B, D and E)
- A heterogenous mixture of different office workstations (Scenario F)

The results are displayed in Table 2.

Comparing case A with B of Table 2, the scaling factor of 4.7, is not very good regarding the fact that eight times more cores have been used. The cause for this lies in two different circumstances. The first one is, that these two scaling tests have been performed on one machine with several cores, but other resources like the working memory had to be shared which led to waiting cycles during the computation. The other cause, which comes into

Scenario	Function	Number of machines/cores	Individuals per minute	Scaling factor
A	CPG	1/1	7.39	—
B	CPG	1/8	34.72	4.7
C	CPG	5/16	45.13	6.1

Table 2: Scaling

account in case C as well, leads to waiting cycles after finishing a job and conflicts when two processes want to write results to the result folder at the same time. This is rooted in the fact that IAIDataShare was designed for graphical applications and real time capacities have not been a design criteria in the first place. Taking this into consideration scalability can be improved dramatically by adding real time capabilities to IAIDataShare.

In Figure 7 the development of the fitness value over the generations during the optimization process is shown. This optimization run lasted several days and showed that at about 160 generations a stagnation occurs and lasts (this is why the last 200 generations have not been displayed in this graph). Further investigation of this fact and the influence of the amount of parameters involved as well as the weights in the fitness function to it is necessary.

The resulting graph of scenario C of Table 2 is shown in Figure 6. The matching of the flexor activation to the flexor reference is very good but the matching of the extensor activation curves bares potential for optimization. The cause for this can be found in oversimplification of the model which was part of the compromise between accuracy and computing effort.

4 Discussion

By employing NEURON for simulating a CPG the possibility is given to generate cyclic activation patterns in a very sophisticated and biological realistic way. Combining this simulation with the EA engine GLEAM and its GUI GLEAMKIT the optimization of the simulation despite its huge number of parameters has been made possible. Tests and evaluations have shown that there is still a lot of space for improvements which will be targeted in further research. Possible extensions could be tests targeting the robustness of the system against input from external sensors or multiple parallel execution.

The presented system is capable of generating biological correct and realistic actor activation patterns which, in the future, should be used for the motion generation or actor activation in robotics. To make this possible two challenges to get from simulation to application have to be targeted. First an interface has to be developed which is capable of translating the neural activation patterns into actuator signals for controlling several devices which function as 'muscles' in a robot. Then a new iteration of optimization processes is necessary. Apart from that, stepwise simplifications have to be performed to make the system suitable for real time applications in robots.

Apart from that, further potential for research lies in the setting of the weights in the calculation of the fitness. As depicted in Section 3 possibility exists, that despite different

weights in the fitness function the resulting parameter set leads to the same activation pattern generated. This fact motivates further investigation how this fact may be useful for designing more advanced optimization procedures.

In addition to that, as was shown by the scaling tests, there is still space for improvement, especially in the design of the IAIDataShare tool which takes responsibility for the distributed computing.

5 Conclusion

In this paper, an extended simulation environment for biological Central Pattern Generators has been presented and an optimization has been applied to match the generated muscle activation as close as possible to human locomotion data. For this optimization an Evolutionary Algorithm has been employed and several optimization runs have been performed which showed the influence of different prioritization of the parameters. Now a system exists which is capable of generating activation patterns for joint movements. This will be expanded and enhanced for applications in robotics and the development of prostheses in our future research.

6 Acknowledgements

Software tools used were the NEURON of N.T. Carnevale and M.L. Hines and their group at the University of Yale and the OpenSim toolbox for biological mechanics simulation. The project is part of the Collaborative Research Center (SFB) 588 of the German Research Foundation (DFG). Thanks go to the Orthopädische Klinik Heidelberg, especially Rüdiger Rupp, Christian Schuld and Joachim Schweidler, in Heidelberg Germany for assisting in recording human motion data and EMG signals.

References

- [1] Brown, T.: The Intrinsic Factors in the Act of Progression in the Mammal. *Proceedings of the Royal Society: London* 84 (1911), S. 308–319.
- [2] Brown, T.: The Factors in Rhythmic Activity of the Nervous System. *Proceedings of the Royal Society: London* 85 (1912), S. 278–289.
- [3] Manoonpong, P.; Geng, T.; Porr, B.; Woergoetter, F.: The RunBot Architecture for Adaptive, Fast, Dynamic Walking. In: *Proc., IEEE International Symposium on Circuits and Systems (ISCAS)*, S. 1181–1184. 2007.
- [4] Chen, Y.; Bauer, C.; Burmeister, O.; Rupp, R.; Mikut, R.: First Steps to Future Applications of Spinal Neural Circuit Models in Neuroprostheses and Humanoid Robots. In: *Proc., 17. Workshop Computational Intelligence*, S. 186–199. Universitätsverlag Karlsruhe. 2007.
- [5] Chen, Y.: *A Concept for the Application of Neural Oscillators and Spinal Reflexes to Humanoid Robots and Neuroprostheses*. Master thesis, Universität Karlsruhe (TH), Institut für Regelungs- und Steuerungstechnik, Forschungszentrum Karlsruhe GmbH. 2008.
- [6] Jakob, W.: Auf dem Weg zum industrietauglichen Evolutionären Algorithmus. In: *Proc., 15. Workshop Computational Intelligence*, S. 212–226. Universitätsverlag Karlsruhe. 2005.
- [7] Blume, C.: GLEAM - Ein EA für Prozessabläufe am Beispiel von Steuerungen für Industrieroboter. In: *Proc., 16. Workshop Computational Intelligence*, S. 11–24. Universitätsverlag Karlsruhe. 2006.

- [8] Blume, C.; Jakob, W.: GLEAM - an Evolutionary Algorithm for Planning and Control Based on Evolution Strategy. In: *Proc., GECCO*, Bd. Late-Breaking Papers, S. 31–38. L. Livermore National Laboratory. 2002.
- [9] Hodgkin, A.; Huxley, A.: A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve. *The Journal of Physiology* 117(4) (1952), S. 500–544.
- [10] Duysens, J.; Van de Crommert, H. W. A. A.: Neural Control of Locomotion; Part 1: The Central Pattern Generator from Cats to Humans. *Gait & Posture* 7(2) (1998), S. 131–141.
- [11] Matsuoka, K.: Sustained Oscillations Generated by Mutually Inhibiting Neurons with Adaptation. *Biological Cybernetics* 52 (1985), S. 367–376.
- [12] Matsuoka, K.: Mechanisms of Frequency and Pattern Control in the Neural Rhythm Generators. *Biological Cybernetics* 56 (1987), S. 345–353.
- [13] Manoonpong, P.; Geng, T.; Wörgötter, F.: Exploring the dynamic walking range of the biped robot RunBot with an active upper-body component. In: *Proc., IEEE-RAS International Conference on Humanoid Robots*, S. 418–424. 2006.
- [14] Mori, T.; Nakamura, Y.; Sato, M.; Ishii, S.: Reinforcement Learning for CPG-Driven Biped Robot. In: *Proc., National Conference on Artificial Intelligence*, S. 623–630. 2004.
- [15] Duysens, J.; Tax, A.; Murrer, L.; Dietz, V.: Backward and Forward Walking Use Different Patterns of Phase-Dependent Modulation of Cutaneous Reflexes in Humans. *Journal of Neurophysiology* 76 (1996) 1, S. 301–310.
- [16] Dietz, V.; Zijlstra, W.; Duysens, J.: Human Neuronal Interlimb Coordination during Split-Belt Locomotion. *Experimental Brain Research* 101 (1994), S. 513–20.
- [17] Blume, C.: GLEAM - A System for Simulated ‘Intuitive Learning’. In: *Proc. PPSN I*, Nr. 496 in LNCS, S. 48–54. Springer. 1990.
- [18] Braun, S.: *Softwareentwurf und Teilimplementierung einer Toolbox GLEAMKIT unter MATLAB*. Projektarbeit, Berufsakademie Karlsruhe, Forschungszentrum Karlsruhe. 2007.
- [19] Braun, S.: *Entwicklung einer Plattform für Evolutionäre Algorithmen mit verteilter Simulation am Beispiel der Parameteroptimierung von zentralen Mustergeneratoren für Roboterbewegungen*. Diplomarbeit, Berufsakademie Karlsruhe. 2008.
- [20] Mikut, R.; Burmeister, O.; Reischl, M.; Loose, T.: Die MATLAB-Toolbox Gait-CAD. In: *Proc., 16. Workshop Computational Intelligence*, S. 114–124. Universitätsverlag Karlsruhe. 2006.
- [21] Carnevale, N.; Hines, M.: *The NEURON Book*. Cambridge University Press. 2006.

Self-Adaptation in Constrained and Combinatorial Search Spaces

Oliver Kramer

Technische Universität Dortmund, Lehrstuhl XI

Otto-Hahn-Str. 14

Tel.: (0231) 755-7708

Fax: (0231) 755-7740

E-Mail: oliver.kramer@tu-dortmund.de

Abstract

The success of evolutionary algorithms depends on adequate parameter settings. How can evolutionary algorithms learn parameters automatically during the optimization? Evolution strategies gave an answer decades ago: self-adaptation. Their self-adaptive mutation control turned out to be exceptionally successful. But self-adaptation has not achieved the attention it deserves. We introduce various types of self-adaptive parameters. In constrained search spaces self-adaptation of Gaussian mutation may fail due to low success rates. Biased mutation turns out to be useful in these domains. Furthermore, we show that self-adaptation is also possible for combinatorial search. Self-adaptive inversion mutation accelerates the search on combinatorial TSP-like problems.

1 Introduction

Evolutionary algorithms are biologically inspired meta-heuristics. They perform stochastic search with the help of genetic operators. The success of evolutionary search is often a result of proper parameter settings and knowledge, the practitioner integrates into the search method. The adjustment of parameters and adaptive operator features is of crucial importance for reliable results to given problems and the efficiency of the evolutionary heuristics. Self-adaptation has become an important self-tuning method for Gaussian mutations. But in spite of its success in unconstrained continuous spaces, not much work has been spent on self-adaptation in constrained continuous or combinatorial domains. Most results of this article can also be found in the monograph [1] at full length.

1.1 The Concept of Self-Adaptation

Self-adaptation is usually associated with the step size control of evolution strategies (ES). It was originally introduced by Rechenberg and Schwefel [2], later by Fogel [3] for evolutionary programming (EP). Self-adaptive algorithms are characterized by the integration of strategy parameters, which influence the genetic operators or other parameters of the EA, into the individuals' chromosomes. Typical is the influence of strategy parameters on mutation rates, crossover probabilities and recently population sizes. The chromosome of each individual $\vec{a} = (O, \Sigma)$ contains a set of solutions encoding objective variables O and a set of strategy variables Σ . The strategy parameters $\sigma_i \in \Sigma$ are bound to the objective variables and evolved by means of the genetic operators recombination, mutation and selection. We summarize the concept of self-adaptation in the following definition.

Definition 1.1 (Self-Adaptation of EA Parameters) *Self-adaptation is the evolutionary control of strategy parameters Σ , bound to each individual \vec{a} and used to control parameters of the EA.*

Appropriate parameter settings have a positive effect on the individual's fitness and are consequently more likely inherited to the offspring. Hence, self-adaptation means an implicit control of the strategy parameters. Self-adaptation can be seen as a dynamical optimization process optimizing the strategy parameters during the walk through the search space. The *convergence* of the strategy parameters is a sign for a *good* working self-adaptive process, but it is no *necessary* condition. In ES and EP the self-adaptive strategy variables are usually parameterizations of the mutation distributions. For example, an individual \vec{a} of a (μ, λ) -ES with objective variable vector \vec{x} is mutated in the following way:

$$\vec{x}' := \vec{x} + \vec{z} \quad (1)$$

and

$$\vec{z} := (\sigma_1 \mathcal{N}_1(0, 1), \dots, \sigma_N \mathcal{N}_N(0, 1)) \quad (2)$$

The strategy parameter vector is mutated

$$\vec{\sigma}' := e^{(\tau_0 \mathcal{N}_0(0,1))} \cdot (\sigma_1 e^{(\tau_1 \mathcal{N}_1(0,1))}, \dots, \sigma_N e^{(\tau_N \mathcal{N}_N(0,1))}). \quad (3)$$

As the mutation step size has an important impact on the quality of the mutations and undergoes mutation itself, the evolutionary search controls the step sizes implicitly. Of course, self-adaptation is not restricted to steps sizes. Self-adaptive parameters also control the skewness of mutation probability functions or the probabilities of crossover.

1.2 Preliminary Work

Only few theoretical work on self-adaptation exists. Most of it concerns the continuous search domain, i.e. the analysis of the ES mutation strength self-adaptation. As Beyer and Schwefel [4] state, the analysis of EAs including the mutation control part is a difficult task. Beyer [5] gives a detailed analysis of the $(1, \lambda)$ -self-adaptation on the sphere function in terms of progress rate and self-adaptation response also considering the dynamics with fluctuations. The analysis reveals various characteristics of the SA- $(1, \lambda)$ -ES, e.g. it confirms that the choice for the learning parameter $\tau = c/\sqrt{N}$ is reasonable. Beyer and Meyer-Nieberg [6] present results for σ -self-adaptation on the sharp ridge for a $(1, \lambda)$ -ES without recombination. They use the evolution equation approximation and show that the radial and the axial progress rate as well as the self-adaptation response depend on the distance to the ridge axis. A further analysis reveals that the ES reaches a stationary normalized mutation strength. According to their analysis the function parameter d of the ridge function determines whether the ES concentrates on decreasing the distance to the ridge axis or increases the speed traveling along the ridge axis. If d is smaller than a population size dependent limit the stationary step size leads to an increase of the distance to the ridge. Auger [7] investigated the $(1, \lambda)$ -SA-ES on the one-dimensional sphere function. She proved sufficient conditions on the algorithm's parameters and showed that the convergence is $\frac{1}{t} \ln(\|X_t\|)$ with parent X_t at generation t . Her proof technique is based on Markov chains for continuous search spaces and the Foster-Lyapunov drift conditions which enable stability property proofs of Markov chains. The main result is the log-linear type of the convergence and an estimation of the convergence rate. Even though the proof

is very sophisticated, it does not show how the number of steps scales depending on the search space dimension. The work of Semenov and Terkel [8] is based on a stochastic Lyapunov function. A supermartingale (stochastic process with a relationship between a random variable and its conditional expectation) is used to investigate the convergence of a simple self-adaptive EA. A central result of their research is that the convergence velocity of the analyzed self-adaptive EA is asymptotically exponential. They use Monte-Carlo simulations to validate the confidence of the martingale inequalities numerically, because they cannot easily be proven. Rudolph [9] investigated the premature convergence of EAs using self-adaptive mutation strength control. He assumed that the (1+1)-EA is located in the vicinity \mathcal{P} of a local solution with fitness $f = \epsilon$. He proved that the probability to move from a solution in this set \mathcal{P} to a solution in a set of solutions \mathcal{S} with a fitness less than an $f < \epsilon$ (which in particular contains the global optimum, the rest of the search domain exhibits fitness values $f > \epsilon$) is smaller than 1. This result even holds for an infinite time horizon. Hence, an EA can get stuck at a non-global optimum with a positive probability. Rudolph also proposes ideas to overcome this problem. Rechenberg proved that the optimal step size of ES depends on the distance to the optimum and created the 1/5th adaptation rule [10]. Recently, Jägersküpfer [11] proved that the (1+1)-ES using Gaussian mutations adapted by the 1/5-rule needs a linear number of steps, i.e. $O(n)$, to halve the approximation error on the sphere function. The bound $O(n)$ holds with "overwhelming probability of $1 - e^{-\Omega(n^{1/3})}$ ". Later, Jägersküpfer [12] proved the bound $O(n \cdot \lambda \sqrt{\ln \lambda})$ for the $(1 + \lambda)$ -ES with Rechenberg's rule. He proposes a modified 1/5th rule for λ descendants as the regular algorithm fails for more than a single isotropic mutation. To the best of our knowledge, a runtime analysis for a self-adaptive ES has not been conducted yet with the exception of Auger's Markov chain result.

Eiben, Schut and Wilde [13] propose a method to control the population size and the selection pressure using tournament selection of an EA self-adaptively. The problem of such an adaptation is that the parameters *population size* and *tournament size* are global while self-adaptation is an approach for the individual or the component level. The idea of the approach is to derive the global parameters via aggregation of local information on individual or component level. The local information is still part of the individual's genome and participates in recombination and mutation. They introduce an aggregation mechanism which simply sums up all votes of the individuals for the global parameter. To the best of our knowledge this has been the only attempt to control global parameters self-adaptively.

2 Premature Step Size Reduction

ES on constrained optimization problems suffer from premature step size reduction in case of active inequality constraints. This results in premature convergence. Broadly speaking, the reason for the premature step size reduction is the fact that the constrained region cuts off the mutative success area. Consequently, the self-adaptation process favors smaller step sizes, whose success area is not cut off. At first, premature step size reduction is analyzed experimentally. Afterwards we prove theoretically that the step sizes tend to reduce at the constraint boundary.

2.1 Experimental Analysis

The premature step size reduction can be shown experimentally on problem 2.40 for the death penalty method and the dynamical penalty function by Joines and Houck [14], see Table 1. Problem 2.40 exhibits a linear objective function and an optimum with five active linear constraints. Each row of Table 1 shows the results of a (15,100)-ES after 50 runs. As termination condition fitness stagnation is chosen. If the difference between the fitness value of the best individual of a generation and the best of the following generation is smaller than a $\theta = 10^{-12}$, then the ES terminates as the magnitude of the steps sizes is too small to effect further improvements. Both constraint-handling methods are not able to approximate the optimum of the problem satisfactorily. The standard deviations *dev* show that the algorithms produce rather different results in the various runs.

	best	mean	worst	dev	ffc	cfc
DP	-4948.079	-4772.338	-4609.985	65.2	50624	96817
Dyn	-4780.554	-4559.129	-4358.446	85.0	31878	31878

Table 1: Experimental results of the death penalty method (DP) and the dynamic penalty function by Joines and Houck (Dyn) on problem 2.40. The parameter *ffc* counts the fitness function calls and *cfc* the constraint function calls. Both constraint-handling techniques are not able to approximate the optimum of the problem satisfactorily. The relatively high standard deviations *dev* show that the algorithms produce unsatisfactorily different results.

2.2 Theoretical Analysis

We consider the two dimensional case of a linear objective function and one linear constraint with the angle β between the latter and the contour lines of the fitness function. Figure 1 shows a typical situation of individual \vec{x} in the neighborhood of the constraint boundary. It shows the success rate situations of individual \vec{x} with distance d to the constraint boundary for the three cases: 1. $\sigma < d$, 2. $\sigma > d, \sigma < s$ and 3. $\sigma > d, \sigma > s$. We analyze the behavior of a (1+1)-EA with adaptive step sizes modeled¹ by the Markovian process $(X_t, \sigma_t)_{t \geq 0}$ generated by

$$X_{t+1} = \begin{cases} X_t + \sigma_t Z_t & \text{if } f(X_t + \sigma_t Z_t) < f(X_t) \wedge g(X_t + \sigma_t Z_t) = 0 \\ X_t & \text{otherwise} \end{cases} \quad (4)$$

and

$$\sigma_{t+1} = \begin{cases} \gamma \sigma_t & \text{if } f(X_t + \sigma_t Z_t) < f(X_t) \wedge g(X_t + \sigma_t Z_t) = 0 \\ \gamma^{-1} \sigma_t & \text{otherwise} \end{cases} \quad (5)$$

with step size σ_t and mutation parameter $\gamma > 1$. The function g measures the constraint violation. Each random vector $Z_t, t \geq 0$ is independent and identically distributed in the following way: We assume that mutations $\sigma_t Z_t$ are produced on the edge of the circle around X_t with radius σ_t . When a successful mutation is produced, the step length σ_t is increased and decreased otherwise. We are interested in the development of the step size σ_t and the distance d_t to the constraint boundary. For the sake of better readability we write σ instead of σ_t and d instead of d_t where possible. The following lemma 2.1 states the success probabilities for the three mentioned cases.

¹As Rudolph [9] states, this EA does not exactly match a (1+1)-EA with self-adaptive step size control, but it can be transferred to a broader class of EAs.

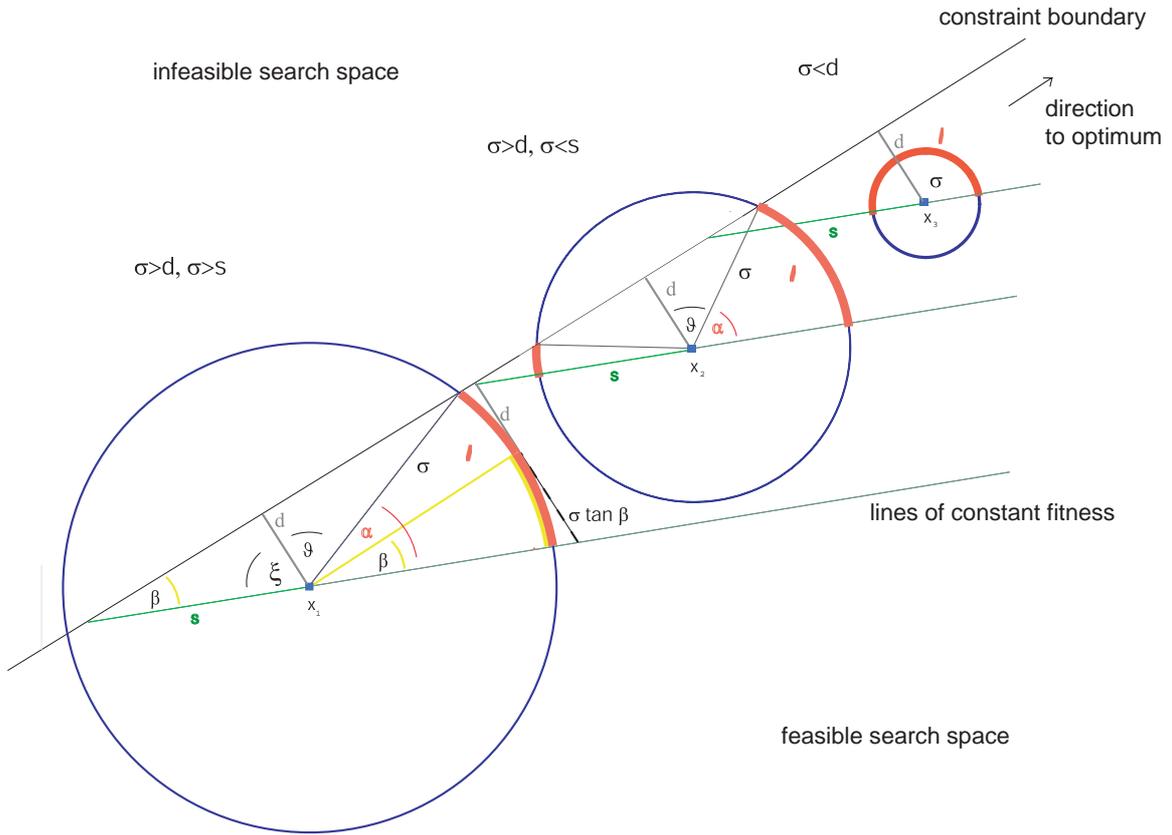


Figure 1: Success rates at the boundary of the feasible search space. Three cases have to be considered, i.e. 1. $\sigma < d$, 2. $\sigma > d, \sigma < s$ and 3. $\sigma > d, \sigma > s$. The bold circular arcs are the regions where successful mutations are produced.

Lemma 2.1 Let (p_s) be the success probability for individual X_t of the $(1+1)$ -EA, with step size σ and distance d to the constraint boundary. Then it holds $(p_s)_{\sigma < d} = 1/2$ and $(p_s)_{\sigma > d} < 1/2$. For $d/\sigma \rightarrow 0$ it holds $(p_s)_{\sigma > d} \rightarrow \beta/(2\pi)$.

The proof for lemma 2.1 can be found in [1].

Theorem 2.1 (Premature Step Size Reduction) Let f be a linear fitness function and g a linear constraint boundary with angle $\beta < \pi/2$ between f and g . In the vicinity of the constraint boundary $\sigma > d$, the above modeled $(1+1)$ -EA with $\gamma > 1 + (\tan \beta)$ reduces its step size σ in each iteration by $E(\gamma) = \gamma^{\frac{1}{1-p}} - \frac{1}{p} < 1$ with $p < 1/2$.

Proof 2.1 We distinguish two states Γ_1 and Γ_2 for an individual X_t in the neighborhood of the constraint boundary, $\sigma > d$.

1. Γ_1 denotes the situation that a successful mutation was produced, i.e. $f(X_t + \sigma_t Z_t) < f(X_t) \wedge g(X_t + \sigma_t Z_t) = 0$.
2. Γ_2 denotes the failure $f(X_t + \sigma_t Z_t) > f(X_t) \vee g(X_t + \sigma_t Z_t) > 0$. Lemma 2.1 shows that the probability for a success is small, at most smaller than $1/2$ and converges to $\beta/(2\pi)$ for $d/\sigma \rightarrow 0$.

The state transitions between the two states Γ_1 and Γ_2 are analyzed in the following.

- $\Gamma_1 \rightarrow \Gamma_1$: The probability for a success is p , which is relatively low, see lemma 2.1. A success results in a step size increase $\sigma_{t+1} = \gamma\sigma_t$. A successful mutation may lie arbitrarily close to the constraint boundary ($d/\sigma \rightarrow 0$) and therefore decrease the success probability p rapidly. But a success may also lead to an increase of the distance to the constraint boundary, at most by $\sigma \tan \beta$. The constrained case is not left, if the step size increase is higher than the distance increase to guarantee $d/\sigma < 1$:

$$\frac{d + \sigma \tan \beta}{\sigma \gamma} < 1 \equiv \frac{d}{\sigma} + \tan \beta < \gamma \quad (6)$$

As $d/\sigma < 1$ it must hold $\gamma > 1 + (\tan \beta)$ to fulfill the above condition for the proof of step size reduction. So, the probability for staying in state Γ_1 is

$$P(\Gamma_1 \rightarrow \Gamma_1 | \Gamma_1) = p, \quad \sigma_{t+1} = \gamma\sigma_t. \quad (7)$$

- $\Gamma_2 \rightarrow \Gamma_1$: The probability for a success if the last step was a failure is again p , the step size is increased and the constrained case is not left for $\gamma > 1 + \tan \beta$.

$$P(\Gamma_2 \rightarrow \Gamma_1 | \Gamma_2) = p, \quad \sigma_{t+1} = \gamma\sigma_t. \quad (8)$$

- $\Gamma_1 \rightarrow \Gamma_2$: The probability for a failure is $1 - p$. It results in a step decrease $\sigma_{t+1} = \gamma^{-1}\sigma_t$. If the step decrease leads to $d/\sigma_{t+1} > 1$, the constraint boundary is left. In this case step size decrease and increase occur with the same probability and the expected change of σ becomes $E(\gamma) = \gamma \cdot \gamma^{-1} = 1$. But the constraint boundary will be reached again within the following steps. Hence, we summarize

$$P(\Gamma_1 \rightarrow \Gamma_2 | \Gamma_1) = 1 - p, \quad \sigma_{t+1} = \gamma^{-1}\sigma_t. \quad (9)$$

- $\Gamma_2 \rightarrow \Gamma_2$: Similar to transition $\Gamma_1 \rightarrow \Gamma_2$ the probability to stay in the state Γ_2 is the probability $1 - p$ for a failure.

$$P(\Gamma_2 \rightarrow \Gamma_2 | \Gamma_2) = 1 - p, \quad \sigma_{t+1} = \gamma^{-1}\sigma_t. \quad (10)$$

This yields the following state transition probability matrix \vec{T} for states Γ_1 and Γ_2 :

$$\vec{T} = \begin{pmatrix} p & (1-p) \\ p & (1-p) \end{pmatrix} \quad (11)$$

It is worth to mention that in the case of a success with an overwhelming probability of $p' = (1 - \beta/(2\pi)) \rightarrow 1$ for $\beta \rightarrow 0$, the distance d to the constraint boundary is decreased. This condition also contributes to an iterative reduction of the distance to the constraint boundary before reaching the optimum. From the probability of each state transition and the step size change, an expected step size change can be derived. Figure 2 shows the probabilities for the state transitions of Γ_1 and Γ_2 , together with the change of step size σ for each transition.

In each state, there are only two possibilities: staying in the same state or leaving the state. Hence, the state transitions are geometrically distributed. We are able to determine

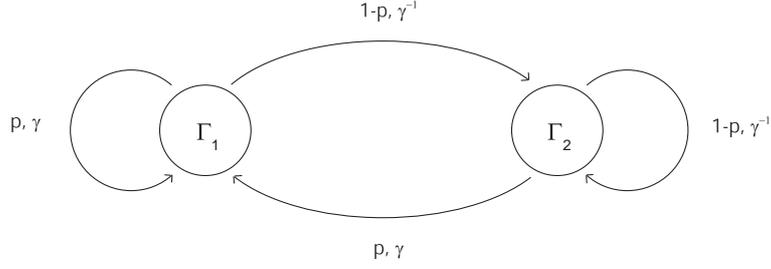


Figure 2: The state transitions of Γ_1 and Γ_2 , its probabilities at the constraint boundary and the influence on the step size σ .

the expected change of the step sizes. The probability to leave state Γ_1 is $1 - p$, so the expected number of iterations to stay is $1/(1 - p)$. Each iteration the step size is increased by γ . When leaving to state Γ_2 the step size is decreased by γ^{-1} . The probability to leave state Γ_2 is p , so the expected number of iterations to stay is $1/p$ with a step decrease by γ^{-1} . Returning to state Γ_1 leads to a step increase by γ . From these considerations we can now determine the expected development of γ :

$$E(\gamma) = \gamma^{\frac{1}{1-p}} \cdot \gamma^{-1} \cdot \gamma^{-\frac{1}{p}} \cdot \gamma = \gamma^{\frac{1}{1-p} - \frac{1}{p}} \quad (12)$$

Lemma 2.1 has proven that $p < 1/2$, so $\frac{1}{1-p} - \frac{1}{p} < 0$ and

$$\boxed{E(\gamma) = \gamma^{\frac{1}{1-p} - \frac{1}{p}} < 1} \quad (13)$$

The expected change of step size σ is $E(\gamma) < 1$ in each generation, so the steps are decreasing at the constraint boundary. For small success probabilities, see lemma 2.1, the decrease becomes quite high.

We proved the step size reduction theoretically for the (1+1)-EA and experimentally for a (15,100)-ES in section 2.1. In the next sections we present two heuristics and a self-adaptive mutation operator to overcome the problem of premature step size reduction.

2.3 Minimum Step Size Reduction Mechanism

As mentioned the death penalty method suffers from premature step size reduction. The death penalty step control evolution strategy (DSES) is based on death penalty, i.e. rejection of infeasible solutions. For the initialization feasible starting points are required. The concept of the approach is a minimum step size ϵ , a lower bound on the step sizes σ that prevents the evolutionary process from premature step size reduction. But it also prevents the optimization process from unlimited convergence to the optimum when reaching the range of ϵ . Consequently, a control mechanism is introduced with the task of reducing ϵ when approximating the optimum. Intuitively, the reduction process depends on the number of infeasible mutations produced when reaching the area of the optimum at the boundary of the feasible search space. Kramer and Schwefel introduced the DSES in [15].

	DSES	best	avg	dev
2.40	[100; 0.7]	-4999.9999999999	-4999.9999999995	7.99E-10
g01	[400; 0.5]	-14.9999999998	-14.9999999991	1.78E-09
g02	[15; 0.3]	-0.8036187549	-0.7658619287	2.9E-03
g04	[25; 0.7]	-30665.5386717833	-30665.5386717831	1.60E-10
g06	[10; 0.3]	-6961.8138755801	-6961.8138755801	1.90E-11
g07	[70; 0.7]	24.3062377392	24.3067230937	4.57E-04
g08	[2; 0.9]	-0.0958250414	-0.0958250414	9.06E-17
g09	[18; 0.7]	680.6301304921	680.6308434198	6.59E-04
g11	[10; 0.5]	0.7499000007	0.7499008144	1.02E-06
g12	[200; 0.5]	-0.9999999999	-0.9999999999	2.05E-12
g16	[100; 0.5]	-1.9051552585	-1.9051552584	9.12E-11
g24	[15; 0.3]	-5.5080132715	-5.5080132715	2.36E-11

Table 2: Summary of the experimental results of the DSES on the considered problems. The heuristic is able to approximate the optima of most of the problems.

The experimental results of the DSES on all considered problems are summarized in table 2. We consider a selection of the famous constrained g-problems. On Schwefel’s problem 2.40 the DSES is able to approximate the optimum with most of the tested settings. DP fails on these problems and suffers from premature mutation strength reduction. Various $[\varpi; \vartheta]$ -settings are successful on problem g01. On g02 the experiments show that also a slow reduction of ϵ can only improve the quality of the results slightly. It has to be paid with a high number of fitness and constraint function calls. No significant improvement can be achieved on problem g02, although we pay with inefficiency. For the DSES it is not difficult to obtain arbitrary convergence to the optimum of problem g04. On problem g06 all tested DSES variants achieved promising results. But the standard DP has to be recommended as the *ffc* values are lower in comparison. The behavior of the DSES on g07 has already been described. Like on g06 the DSES and DP are able to approximate the optimum of g08 very well. But here, the method DP exhibits no significant performance advantages. Only the slowest decrease of ϵ on problem g09 enables the DSES to approximate the optimum better than DP and better than *faster* DSES variants. Again, this improvement has to be paid with higher *ffc* and *ffc*. While problem g11 can be approximated sufficiently, the results of the DSES on problems g12 and g16 are exceeding good. As already stated, the method DP performs well on problem g24 while the DSES performs similarly on this problem. We can summarize, the DSES is based on the reduction of a minimum step size which prevents premature step size reduction. It is able to approximate most constrained problems of the *g-test suite*. The DSES is efficient concerning fitness function calls, but inefficient concerning constraint function calls.

2.4 Biologically Inspired Constraint-Handling

The two sexes evolution strategy (TSES) [15] works as follows . Every individual of the TSES is assigned to a new feature called its sex. Similar to nature, individuals with different sexes are selected according to different objectives. Individuals with sex *o* are selected by the objective function. Individuals with sex *c* are selected by the fulfillment of constraints. The intermediary recombination operator plays a key role. Recombination is only allowed between parents of different sex. The treatment of objective function and constraints as separate objectives sounds similar to the multiobjective optimization

approaches for constraint-handling. But instead of a multiobjective optimization method, a biologically inspired concept of pairing two sexes is introduced. Again, Kramer and Schwefel give a detailed introduction to the TSES in [15].

	TSES	κ	best	avg	dev
2.40	(8+8,13+87)	50	-5000.0000000000	-4999.9999999999	3.3E-07
g01	(40+40,50+400)	50	-14.9999999999	-14.9999999999	5.3E-15
g02	(40+40,50+400)	50	-0.7926079090	-0.6583469345	9.7E-03
g04	(8+8,13+87)	50	-30665.5386717833	-30665.5386717833	1.7E-11
g06	(8+8,10+90)	50	-6961.8138755801	-6961.8138755801	8.6E-13
g07	(40+40,50+400)	50	24.3171014481	24.4613633236	1.0E-02
g08	(8+8,15+85)	50	-0.0958250414	-0.0958250414	5.6E-18
g09	(8+8,13+87)	50	680.6303038768	680.6345393750	4.6E-04
g11	(8+8,10+200)	300	0.7499900000	0.7499900169	4.4E-09
g12	(8+8,10+90)	50	-1.0000000000	-0.9999999999	7.4E-16
g16	(40+40,50+400)	50	-1.9051552585	-1.9051552585	1.6E-15
g24	(8+8,10+90)	50	-5.5080132715	-5.5080132715	4.3E-13

Table 3: Comparison of experimental results of the TSES on our constrained test problems. In most cases the TSES shows satisfying approximation results.

Table 3 shows the experimental results of the TSES on all considered problems. While DP completely fails on Schwefel’s constrained problem 2.40 the (8+8,13+87)-TSES reaches the optimum in every run. On problem g01 increased population sizes achieve promising results. Despite an increase of the population sizes, the TSES could not achieve sufficient results on problem g02. On problem g04 a (8+8,15-85)-TSES failed, but the TSES with an offspring sex ratio of 13+87 and 10+90 found the optimum in every run. Again, the improvement in comparison to death penalty is significant. On problem g06, a highly constrained problem with a ratio of only 0.0066% of feasible search space, all algorithms, including death penalty were successful. Similar to problem g02 an increase of population sizes could not obtain promising approximation of the optimum of problem g07. The TSES could improve the results on problem g08 in comparison to death penalty, but demands a higher number of *ffc* and *cfc*. The results of the TSES on g09 were not satisfying, but slightly better than DP. Again, this improvement has to be paid with approximately 5 to 10 times higher number of *ffc* and *cfc*. On problem g11 various experiments around the sex ratio (8+8,13+87) like recommended by Kramer and Schwefel [15] could not achieve promising results. Tests around the ratio (8+8,10+200) were more successful. All experiments with the (8+8,10+200)-TSES and $\kappa = 200$ or $\kappa = 300$ showed sufficient results. A higher κ causes a loss of efficiency. We have to emphasize that DP shows comparable results. But the number of constraint function calls *cfc* is about ten times higher than the *cfc* of the TSES. So, we can observe an efficiency advantage of the TSES. The experiments on problem g12 have already been commented in this section. On problem g16 the (8+8,13+87)-TSES is not more successful, but less efficient than DP with an average accuracy of 3 decimal places. But the (40+40,50+400)-TSES reaches the optimum with arbitrary accuracy demanding 3-4 times more *ffc* and *cfc*. The (8+8,10+90)-TSES is the best compromise between quality of the results and efficiency on problem g24. But we recommend to use DP as it demands much less *ffc* and *cfc*.

3 The Biased Mutation Operator

Our experiments showed that self-adaptive biased mutation is also able to improve the results on constrained continuous problems. In this section we shortly introduce the Biased Mutation Operator (BMO) and two variants.

3.1 BMO Concept

Unlike directed mutation, the BMO does not change the skewness, but biases the mean of the Gaussian distribution to lead the search into a more beneficial direction. This is reflected in the success rate of reproducing superior offspring. For the BMO we introduce a *bias coefficient vector* $\vec{\xi}$ which indicates the level of bias relative to the standard deviation σ :

$$\vec{\xi} = (\xi_1, \dots, \xi_N) \text{ with } -1 \leq \xi_i \leq 1. \quad (14)$$

For every $i \in 1, \dots, N$ the bias vector $\vec{b} = (b_1, \dots, b_N)$ is defined by:

$$b_i = \xi_i \cdot \sigma_i \quad (15)$$

Since the absolute value of bias coefficient ξ_i is less than or equal to 1, the bias will be bound to the step sizes σ_i . This restriction prevents the search from being biased too far away from the parent. The BMO follows the standard way of mutation:

$$\vec{x}' := \vec{x} + \vec{z}. \quad (16)$$

The mutation of the BMO works as follows:

$$\vec{z} := (\sigma_1 \mathcal{N}_1(0, 1) + b_1, \dots, \sigma_N \mathcal{N}_N(0, 1) + b_N) \quad (17)$$

$$= (\sigma_1 \mathcal{N}_1(0, 1) + \xi_1 \sigma_1, \dots, \sigma_N \mathcal{N}_N(0, 1) + \xi_N \sigma_N) \quad (18)$$

$$= (\sigma_1 \mathcal{N}_1(\xi_1, 1), \dots, \sigma_N \mathcal{N}_N(\xi_N, 1)) \quad (19)$$

In terms of modifying the mutation strength, the log-normal rule is applied. Furthermore, in the BMO the bias coefficients are mutated in the following meta-EP way:

$$\xi_i' = \xi_i + \gamma \cdot \mathcal{N}(0, 1) \quad i = 1, \dots, N. \quad (20)$$

The parameter γ is a new parameter introduced for the BMO to determine the mutation strength on the bias. The BMO biases the mean of mutation and enables the ES to reproduce offspring outside the standard mutation ellipsoid. To direct the search, the BMO enables the mutation ellipsoid to move within the bounds of the regular step sizes $\vec{\sigma}$. The bias moves the center of the Gaussian distribution within the bounds of the step sizes. This is advantageous on ridge functions, on some multimodal functions and at the edge of feasibility. Without the BMO the success rate to reproduce better offspring is relatively low because many mutations lie beyond the feasible search space or have got a worse fitness, as described in section 2. The bias coefficient vector $\vec{\xi}$ improves the success rate situation as the success area increases. The BMO approach is as flexible as correlated and directed mutation, but is less computational expensive than both methods. Let N be the number of strategy variables. Uncorrelated Gaussian mutation with one step size requires only $O(1)$ operations, uncorrelated Gaussian mutation with N steps sizes $O(N)$ respectively. Correlated mutation needs $O(N^2)$ for the rotation of the mutation ellipsoid and is thus computationally expensive. The directed mutation [16] requires $O(2N)$ operations, but

the computation of the asymmetric probability density function is more complex. Also the BMO needs $O(2N)$ operations, but is based on the simpler Gaussian mutation. In comparison to correlated mutation the BMO saves $\frac{N \cdot (N-1)}{2} - N$ additional strategy parameters. In practice the implementation is less complex. In the following, we introduce various variants of the standard BMO.

3.2 Cube Biased Mutation Operator (cBMO)

Another variant of the BMO is the use of only one bias value b , which spans a hypercube with the edge length b , but N step sizes. The mutation \vec{z} becomes

$$\vec{z} := (\sigma_1 \mathcal{N}_1(b, 1), \dots, \sigma_N \mathcal{N}_N(b, 1)). \quad (21)$$

Mutations of the cBMO lie within the hyperellipsoid of the standard mutation, but the center of this hyperellipsoid is shifted by the vector (b, \dots, b) . Again, $\vec{\xi}$ is mutated analog to the biased mutation of the standard BMO.

3.3 The Descent Direction Mutation Operator (DMO)

The BMO enables to bias mutations into beneficial directions. Under the assumption that the optimum lies on the extension of the direction between the center of consecutive generations, the idea arises to adjust the bias along this direction. More precisely, the DMO adjusts the bias to the descent direction of the centers of two successive generations. Let $\vec{\chi}_t$ be the center of the population of generation t

$$\vec{\chi}_t = \sum_{i=1}^{\mu} \vec{x}_i \quad (22)$$

The normalized descent direction $\vec{\xi}$ of two successive population centers $\vec{\chi}_t$ and $\vec{\chi}_{t+1}$ is

$$\vec{\xi} = \frac{\vec{\chi}_{t+1} - \vec{\chi}_t}{|\vec{\chi}_{t+1} - \vec{\chi}_t|} \quad (23)$$

Similar to the BMO, the DMO now becomes

$$\vec{z} := (\sigma_1 \mathcal{N}_1(0, 1) + \xi_1 \sigma_1, \dots, \sigma_N \mathcal{N}_N(0, 1) + \xi_N \sigma_N) \quad (24)$$

$$= (\sigma_1 \mathcal{N}_1(\xi_1, 1), \dots, \sigma_N \mathcal{N}_N(\xi_N, 1)) \quad (25)$$

The DMO is reasonable as long as the assumption of locality is true: the estimated direction of the global optimum can be derived from the local information, i.e. the descent direction of two successive populations' centers.

3.4 Handling Constraints with Biased Mutation

For big step sizes the constrained search space cuts off a big area of success to reproduce better mutations than the parent. So the step sizes reduce self-adaptively before reaching the area of the optimum. Our experiments revealed the benefits of biased mutations for constraint handling. The feature of unbiasedness of mutation operators postulated by Beyer [5] is not applicable for constrained search domains. The experiments reveal

that the biased mutation, i.e. the BMO, the BMO variants and the DMO exhibit useful features, which prevent premature step size reduction while using the simple constraint handling technique *death penalty*². Intuitively speaking, the reason for this behavior is the following: the distribution function is shifted along the constraint boundary so that the success area is not cut off. At first, we compare the algorithms on the constrained function 2.40, see table 5. It turns out that the BMO and the DMO are both able to approximate the optimum, while the standard ES with death penalty fails. The cBMO is worse than BMO, DMO and even *almost* than ES. The measured standard deviation of the DMO on 2.40 is smaller than the accuracy of our data structure allows. Hence, the DMO allows the best approximation of the optimum.

	ES	BMO	cBMO	DMO
best	-4985.69	-5000.0	-4982.02	-5000.0
median	-4948.04	-5000.0	-4883.70	-5000.0
worst	-4676.76	-4999.99	-4683.89	-5000.0
mean	-4897.72	-5000.0	-4865.01	-5000.0
dev	96.69	0.0	91.99	0.0

Table 4: Experimental results of the BMO variants on the constrained function 2.40. BMO and DMO are able to find the optimum in every run while the ES with death penalty fails.

	BMO	cBMO	DMO
vs. ES	1.421E-09	0.09519	1.421E-09

Table 5: Wilcoxon Rank-Sum Test: The question for the significance of the experiments answers a Wilcoxon rank-sum test. It reveals that the BMO and the DMO are both better than the ES with DP ($1.4E-09 < 0.05$), but the cBMO is not ($0.09 > 0.05$). Obviously, a *cube bias* does not offer the necessary degree of freedom to shift the mutation ellipsoid appropriately.

A comparison of the BMO variants and ES with DP shows figure 3, where typical runs on problem g04 are drawn. BMO and DMO are able to approximate the optimum, while BMO is faster. This run of the ES fails. The results are similar to the behavior on problem 2.40. Similar results were obtained on other constrained functions.

4 Self-Adaptive Inversion Mutation

Inversion mutation is a mutation operator for combinatorial search domains. It is based on the random change of edges. A famous example for a combinatorial problem is the traveling salesman problem (TSP). Inversion mutation is a typical genetic operator for combinatorial problems like the TSP. It reverses the tour between two randomly chosen cities. Here, we propose a self-adaptive variant of inversion mutation. Self-adaptation is a successful control technique for the mutation strengths. Up to now, the concept of self-adaptation has not been applied to inversion mutation. As the number of successive inversion mutation operator applications can be seen as the mutation strength, we propose to control this parameter self-adaptively. In this section we introduce a self-adaptive variant of inversion mutation (SA-INV). We prove the convergence of inversion mutation and

²*Death penalty* discards infeasible solutions and produces new individuals until the whole offspring population is feasible.

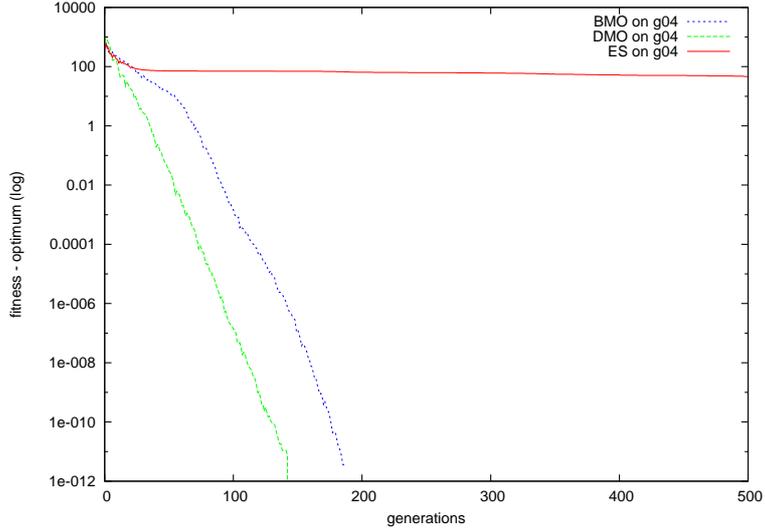


Figure 3: Fitness development on function g04. Again, we compare typical runs of BMO, DMO and ES on a logarithmic scale. The approximation speed of DMO is better than the speed of BMO. ES does not approximate the optimum.

its self-adaptive variant and show experimentally that SA-INV speeds up the evolutionary process, in particular at the beginning of the search. Later we encounter the *strategy bound problem* and introduce a heuristic to solve it.

4.1 Introduction of SA-INV

The *inversion mutation* is a mutation operator for combinatorial problems. It performs a mutual change of the connections of two entity pairs. Taking the TSP into account *inversion mutation* means that the part between the cities $p_1 - 1$ and $p_2 + 1$ is passed through reversely. Cities $p_1 - 1$ and p_2 as well as p_1 and $p_2 + 1$ are connected to each other after the procedure. INV is defined as follows.

Definition 4.1 (Inversion Mutation, INV) Let $p_1, p_2 \in \mathcal{N}_{>0}$ be two randomly chosen points with $1 \leq p_1 \leq p_2 \leq N$ and let π be a permutation of cities $\pi = (c_1, \dots, c_{p_1}, c_{p_1+1}, \dots, c_{p_2-1}, c_{p_2}, \dots, c_N)$. Inversion mutation inverts the part between p_1 and p_2 such that

$$\pi' = INV(\pi) = (c_1, \dots, c_{p_2}, c_{p_2-1}, \dots, c_{p_1+1}, c_{p_1}, \dots, c_N) \quad (26)$$

The number k of successive applications of the inversion mutation operator can be seen as the mutation strength. INV_k is the notation for k successive INV-operations. The idea of SA-INV is to evolve k self-adaptively. This leads to the following formalization of SA-INV.

Definition 4.2 (Self-Adaptive Inversion Mutation, SA-INV) Let k_{t-1} be the number of successive INV-operations of the last generation. Then k_t is mutated $k_t = mut(k_{t-1})$ and SA-INV is defined as

$$\pi' = SA-INV(\pi) = INV_{k_t}(\dots INV_1(\pi)). \quad (27)$$

Intuitively, as for nearly every optimization problem bigger mutations seem to be advantageous at the beginning of the search in order to explore the search domain. Later smaller steps make sense in order to optimize locally and to avoid the destruction of close-to-optimum solutions. Hence, it seems useful to adapt the number k of successive applications of inversion mutation. For the strategy variable mutation of k we use meta-EP mutation.

$$k' := k + \gamma \cdot \mathcal{N}(0, 1). \quad (28)$$

Figure 4 shows the pseudocode of SA-INV.

1	Mutate k
2	For $i=0$ to k
3	Randomly choose points p_1 and p_2
4	Inverse part between p_1 and p_2 of tour π
5	Next

Figure 4: Pseudocode of the SA-INV operator. The strategy variable k determines the number of edge swaps. An edge swap is the inversion of the part between the randomly chosen pairs, i.e. edges $(p_1 - 1, p_1)$ and $(p_2, p_2 + 1)$.

4.2 Convergence Properties

In the following we prove that an EA using the INV-Operator finds the global optimum in finite time. As stated by Rudolph [17] and Eiben [18] it is not necessary to build an quantitatively exact Markov model for each EA variant. Instead, we can use a qualitative model. Let \mathcal{X} be the search space and $(x_1, \dots, x_n) \in \mathcal{X}^n$ denote the population of parents. Parent selection is abbreviated with *mat*, recombination with *reco*, mutation with *mut* and selection with *sel*. Our convergence proof is based on theorem 4.1. So, we have to ensure the following assumptions (A₁) to (A₄) [17]:

- (A₁) $\forall x \in (x_1, \dots, x_n) : P\{x \in reco(mat(x_1, \dots, x_n))\} \geq \delta_r > 0$.
- (A₂) For every pair $x, y \in \mathcal{X}$ there exists a finite path x_1, x_2, \dots, x_l of pairwise distinct points with $x_1 = x$ and $x_l = y$ such that $P\{x_{i+1} = mut(x_i)\} \geq \delta_m > 0$ for all $i = 1, \dots, l - 1$.
- (A₂') For every pair $x, y \in \mathcal{X}$ holds $P\{y = mut(x)\} \geq \delta_m > 0$.
- (A₃) $\forall x \in (x_1, \dots, x_l) : P\{x \in sel(x_1, \dots, x_l)\} \geq \delta_s > 0$.
- (A₄) Let $v_l^*(x_1, \dots, x_l) = \max\{f(x_i) : i = 1, \dots, l\}$ denote the best fitness value within a population of l individuals ($l \geq n$). The selection method fulfills the condition

$$P\{v_n^*(sel(x_1, \dots, x_l)) = v_l^*(x_1, \dots, x_l)\} = 1. \quad (29)$$

Theorem 4.1 (Rudolph [17]). *If the assumptions (A₁), (A₂) and (A₃) are valid, then the EA visits the global optimum after a finite number of iterations with probability one, regardless of the initialization. If assumption (A₄) is valid additionally and the selection method chooses from parents as well as offspring, then the EA converges completely and in mean to the global optimum regardless of the initialization.*

We prove the global convergence of the following EA using INV as mutation operator:

1. recombination: no recombination is used, i.e. individuals for mutation are selected with uniform distribution,
2. mutation: INV/SA-INV see definitions 4.1 and 4.2,
3. selection: comma-selection with modification, i.e. comma-selection selects the $\mu - 1$ best parents from the λ descendants and one parent randomly with uniform distribution and probability $p^* \geq 1/\lambda$.

Theorem 4.2 *The EA with inversion mutation (INV) as defined above converges with probability 1 to the global optimum after a finite number of iterations.*

Proof 4.1 *We make use of theorem 4.1 and have to ensure that assumptions (A₁) to (A₃) are valid. Assumption (A₁) is valid, because $\text{mat}(x_1, \dots, x_n) = x_i, 0 \leq i \leq n$ with $P(X = i) = 1/n > 0$. Since no recombination is used $\text{reco}(x_i) = x_i$ with probability 1. So, $\forall x \in (x_1, \dots, x_n)$:*

$$P\{x \in \text{reco}(\text{mat}(x_1, \dots, x_n))\} = 1/n = \delta_r > 0 \quad (30)$$

To prove assumption (A₂) we have to show that for every tour π_x and tour π_y : \exists a finite sequence $\text{INV}_1, \dots, \text{INV}_r$ of INV-operations such that $\text{INV}_r(\dots \text{INV}_1(\pi_x)) = \pi_y$. We use the following lemma 4.1.

Lemma 4.1 *Let π be a tour with cities c_q with rank x and c_s with rank y , $x > y$. $\forall c_q, c_s \exists$ a finite sequence $\text{INV}_1, \dots, \text{INV}_r$ of INV-operations with $\pi' = \text{INV}_r(\dots \text{INV}_1(\pi))$ such that $\pi'(q) = y$, with probability $\delta_{m'(\text{INV})} > 0$.*

Proof 4.2 *It holds $x = \pi(q)$ and $y = \pi(s)$. For $x > y$, a sequence of $r = x - y$ successive $\text{INV}(\pi, i, j)$ -operations³ on neighbored cities moves city c_q from rank x to rank y , i.e.*

$$\text{INV}(\pi, x, x - 1), \text{INV}(\pi_2, x - 1, x - 2), \dots, \text{INV}(\pi_r, y + 1, y) \quad (31)$$

Since parameters i and j of the INV-operator are chosen randomly with uniform distribution, the probability to choose them is $P\{i = x, j = x - 1\} = 1/(N(N - 1))$. Hence, the probability of the successive INV-operations is

$$\delta_{m'(\text{INV})} = P\{i_1 = x, j_1 = x - 1, \dots, i_r = y + 1, j_r = y\} = \left(\frac{1}{N(N - 1)}\right)^r \quad (32)$$

with the difference of ranks $r = x - y$. The order of the other cities is not changed by the neighbored INV-operations.

If a sequence of mutations can move every city c_q with rank x to every rank y as proven with lemma 4.1, there must exist a sequence of mutations that transfers tour π_x into π_y with probability

$$\delta_{m(\text{INV})} = P\{\pi_x \rightarrow \pi_y\} \geq \left(\frac{1}{N(N - 1)}\right)^{r_{\max} \cdot (N - 1)} \quad (33)$$

³ $\text{INV}(\pi, i, j)$ is inversion mutation of tour π with inversion points $p_1 = i$ and $p_2 = j$.

with $r_{max} = \max\{r_i | x_i - y_i, 1 \leq i \leq N\} \leq N - 1$ and in the worst case $N - 1$ cities have to be moved by at most r_{max} . Hence, the probability for a finite path of pairwise distinct permutations $P\{\pi_{x_{i+1}} = mut(\pi_{x_i})\} \geq \delta_{m(INV)} > 0$ and assumption (A₂) is valid. Assumption (A₃) is valid with $\delta_s = 1/\lambda > 0$. We had to modify the comma selection in this kind of way to ensure convergence and are convinced that the modification will not influence our experimental results.⁴ Hence, the EA using the INV operator converges with probability 1 to the global optimum after a finite number of iterations. Assumption (A₄) would be valid for a 1-elitist strategy, i.e. the best individual is always kept in the population. In particular it is valid for plus selection and guarantees that the EA converges completely and in mean to the global optimum after a finite number of iterations. But this is not the case for comma selection.

Theorem 4.3 *The EA with self-adaptive inversion mutation (SA-INV) converges with probability 1 to the optimum after a finite number of iterations.*

Proof 4.3 *Assumptions (A₁) and (A₃) are valid analog to the proof of theorem 4.2. Let $P_{k=1}$ be the probability for $k = 1$. As SA-INV uses Gaussian mutation with rounding procedures to mutate strategy parameter k , it holds $P\{k = 1\} > 0$ and according to the argumentation of theorem 4.2's proof with $k = 1$,*

$$\delta_{m(SA-INV)} = P\{k = 1\}^{r_{max}(N-1)} \cdot \delta_{m(INV)} > 0 \quad (34)$$

Hence, it holds $P\{\pi_{x_{i+1}} = mut(\pi_{x_i})\} \geq \delta_{m(SA-INV)} > 0$ and (A₂) is valid.

4.3 Experimental Analysis

In this section we analyze the SA-INV operator empirically. The analysis contains a comparison with the standard inversion mutation (INV) with constant number of edge swaps in each generation. We conducted the experiments on various TSP instances of the TSPLib of Reinelt [19], e.g. *berlin52*, *bier127* and *gr666*. Furthermore, we analyzed the development of the strategy parameter k . In order to avoid side effects we do not apply crossover. We use the following notation: SA-INV _{ν - γ} is the self-adaptive inversion mutation with the initial setting $k^{(0)} = \nu$ and the strategy mutation strength γ and INV _{ν} is the common inversion mutation with constant $k = \nu$ edge swaps. Here we present the results of the experiments on Problem *bier127*, an *Euclidean distance* problem of 127 beer gardens in Augsburg. SA-INV₅₋₁ is the fastest variant, see figure 5, lower part. Between generations 70 to 80, INV₁ reaches the same approximation quality, we hold the *strategy bound problem*, see section 4.4, responsible for this. The *median* of the self-adaptive variants is approximately the same. The results of standard INV with $\nu = 10$ are poor. The number of ten edge exchanges seems to be too high in order to save the structure of the solutions, i.e. not to destroy them. This behavior is the basis for the success of SA-INV, which is able to reduce the mutation strength appropriately. Figure 5, upper part, shows the development of the strategy parameter k during the first 100 generations and the fitness development in the same time slot. For all SA-INV variants k increases within the first generations, very demonstrative in the case of SA-INV₅₋₁. This shows that the self-adaptation process is working properly: an increase of the mutation

⁴Assumption (A₃) is also valid for fitness proportional and tournament selection.

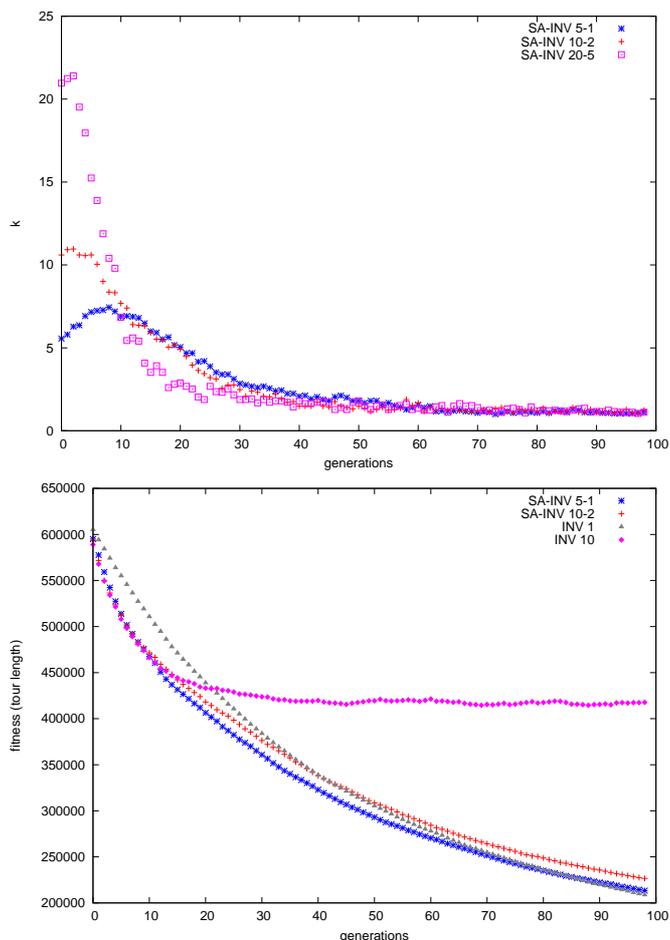


Figure 5: SA-INV on problem *bier127*. Upper part: Development of k averaged over 25 runs. For all SA-INV variants the *mutation strength* increases in the first generations. Lower part: Fitness development of SA-INV and INV averaged over 25 runs. SA-INV₅₋₁ is the fastest variant until generation $t \approx 80$. INV₁₀ is very fast at the beginning, but suffers from fitness stagnation after generation $t \approx 20$.

strength ($k > 5$) is desirable at the beginning of the search. Later the mutation strength developments of all SA-INV variants show similar properties. SA-INV₂₀₋₅ decreases its k faster between generation $t \approx 10$ and $t \approx 30$. We hold the higher strategy mutation parameter γ responsible for that. INV₁₀ suffers from fitness stagnation, here around $t \approx 20$. SA-INV₅₋₁ is better than INV₁ until generation $t \approx 80$.

4.4 The Strategy Bound Problem

The self-adaptation of k speeds up the starting phase of the search. At the beginning, more than one inversion mutation in each step leads to considerable improvements. Parameter k decreases constantly, as less and less edge swaps are necessary to improve the solution and not to destroy it. After a certain number of generations g , the strategy parameter k is almost constantly one. From now on more than one change of edges leads to a fitness deterioration. But the mutation of strategy variables in the context of the self-adaptive process still produces solutions with $k > 1$ and consequently offspring with potentially worse fitness. The progress of SA-INV slows down and soon INV₁ overtakes SA-INV concerning the approximation. We call this problem *strategy bound problem*:

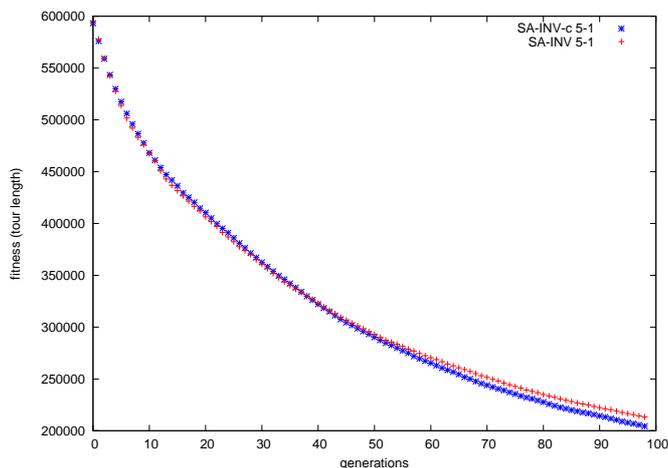


Figure 6: Comparison of the mutation strength k (upper part) and the fitness (lower part) of SA-INV- c_{5-1} and SA-INV $_{5-1}$ on problem *bier127* averaged over 25 runs. Once k reaches the region around one it is constantly set to $k = 1$. The fitness development shows the superiority of SA-INV- c in comparison to SA-INV, in particular at later generations.

Whenever the strategy variables reach a bound b , the self-adaptive process experiments with settings around b . But the bound itself may be an optimal setting. All other settings produce potentially worse solutions. Consequently, the success probability is smaller than p_s with the setting b . A bound of strategy variables is a phenomenon that occurs almost naturally for *discrete* strategy variables. In order to overcome this problem we propose the following modification SA-INV- c , which is based on SA-INV. Let u be the number of individuals with $k = 1$. If $u \geq \frac{\lambda}{2}$ we set $k = 1$ constantly and cancel the self-adaptation of k during the following generations. This modification makes sense under the assumption that after κ generations the optimal strategy setting is $k = 1$ and for $k > 1$ only worse solutions are produced. We assume that g is reached if the condition $u \geq \frac{\lambda}{2}$ is fulfilled. This bound is reasonable: if at least 50% of the strategy parameters *vote* for the setting $k = 1$, we approve this choice for all parameters. We tested the SA-INV- c_{5-1} algorithm experimentally on instance *bier127* and present the results in figure 6. It shows the fitness averaged over 25 runs and illustrates the superiority of SA-INV- c_{5-1} in comparison to SA-INV $_{5-1}$. SA-INV- c_{5-1} is faster, in particular in later phases of the search. Experiments on other TSP problems confirm these results while the Wilcoxon test reveals the statistical relevance.

5 Summary

Self-adaptation is an efficient way of improving evolutionary search. But a necessary condition for its success is a tight link between strategy parameters and fitness. This link is not existing for every type of parameter. Mutation strongly controls the protagonist exploration and antagonist exploitation and therefore offers a direct link between its strategy parameters and the corresponding fitness. The bias control of the BMO and the number of successive edge swaps of SA-INV are examples for the success of self-adaptation of mutation parameters. But self-adaptation can also be misleading: due to local optima in the strategy parameter space, it may result in premature fitness stagnation. Hence, it has to be used with care and sense for search domain characteristics. Furthermore, adaptive

heuristic extensions like the proposed constraint handling techniques offer useful means to overcome premature stagnation. To summarize all results, well-designed self-adaptive and adaptive heuristics are efficient techniques to improve evolutionary search and convergence properties.

References

- [1] Kramer, O.: *Self-Adaptive Heuristics for Evolutionary Computation*. Heidelberg: Springer. 2008.
- [2] Schwefel, H.-P.: Adaptive Mechanismen in der biologischen Evolution und ihr Einfluss auf die Evolutionsgeschwindigkeit. Interner Bericht der Arbeitsgruppe Bionik und Evolutionstechnik am Institut für Mess- und Regelungstechnik, TU Berlin. 1974.
- [3] Fogel, D. B.; Fogel, L. J.; Atma, J. W.: Meta-evolutionary programming. In: *Proceedings of 25th Asilomar Conference on Signals, Systems & Computers*, pp. 540–545. 1991.
- [4] Beyer, H.-G.; Schwefel, H.-P.: Evolution strategies - A Comprehensive Introduction. *Natural Computing* 1 (2002), pp. 3–52.
- [5] Beyer, H.-G.: *The Theory of Evolution Strategies*. Berlin: Springer. 2001.
- [6] Beyer, H.-G.; Meyer-Nieberg, S.: Self-Adaptation on the Ridge Function Class: First Results for the Sharp Ridge. In: *Proceedings of the 9th Conference on Parallel Problem Solving from Nature - PPSN IX*, pp. 72–81. 2006.
- [7] Auger, A.: Convergence results for $(1, \lambda)$ -SA-ES Using the Theory of ϕ -Irreducible Markov Chains. In: *Proceedings of the Evolutionary Algorithms workshop of the 30th International Colloquium on Automata, Languages and Programming*. 2003.
- [8] Semenov, M. A.; Terkel, D. A.: Analysis of Convergence of an Evolutionary Algorithm with Self-Adaptation Using a Stochastic Lyapunov Function. *Evol. Comput.* 11 (2003) 4, pp. 363–379.
- [9] Rudolph, G.: Self-Adaptive Mutations May Lead to Premature Convergence. *IEEE Transactions on Evolutionary Computation* 5 (2001) 4, pp. 410–414.
- [10] Rechenberg, I.: *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Frommann-Holzboog, Stuttgart. 1973.
- [11] Jägersküpper, J.: Rigorous Runtime Analysis of the $(1+1)$ ES: $1/5$ -Rule and Ellipsoidal Fitness Landscapes. In: *Proceedings of the FOGA*, pp. 260–281. 2005.
- [12] Jägersküpper, J.: Probabilistic Runtime Analysis of $(1 + \lambda)$ ES Using Isotropic Mutations. In: *Proceedings of the 8th conference on Genetic and evolutionary computation - GECCO*, pp. 461–468. New York: ACM. 2006.
- [13] Eiben, A.; Schut, M. C.; de Wilde, A.: Is Self-Adaptation of Selection Pressure and Population Size Possible? - A Case Study. In: *Proceedings of the 9th Conference on Parallel Problem Solving from Nature - PPSN IX*, pp. 900–909. 2006.
- [14] Joines, J.; Houck, C.: On the Use of Non-Stationary Penalty Functions to Solve Nonlinear Constrained Optimization Problems with GAs. In: *Proceedings of the 1st IEEE Conference on Evolutionary Computation* (Fogel, D. B., Hg.), pp. 579–584. Orlando, Florida: IEEE Press. 1994.
- [15] Kramer, O.; Schwefel, H.-P.: On Three New Approaches to Handle Constraints within Evolution Strategies. *Natural Computing* 5 (2006) 4, pp. 363–385.
- [16] Hildebrand, L.: *Asymmetrische Evolutionsstrategien*. Dissertation, University of Dortmund. 2002.
- [17] Rudolph, G.: Finite Markov Chain Results in Evolutionary Computation: A Tour d’Horizon. *Fundamenta Informaticae* 35 (1998) 1-4, pp. 67–89.
- [18] Eiben, A. E.; Aarts, E. H. L.; van Hee, K. M.: Global Convergence of Genetic Algorithms: A Markov Chain Analysis. In: *PPSN I: Proceedings of the 1st Workshop on Parallel Problem Solving from Nature*, pp. 4–12. Berlin: Springer. 1991.
- [19] Reinelt, G.: TSPLIB - A Traveling Salesman Problem Library. In: *ORSA J. Comput.* 3, pp. 376–384. 1991.

Benutzerpräferenzmodellierung zur interaktiven multiobjektiven evolutionären Optimierung

Johannes Krettek, Jan Braun, Frank Hoffmann, Torsten Bertram

Lehrstuhl für Regelungssystemtechnik, Technische Universität Dortmund

Otto-Hahn-Str. 4, 44221 Dortmund

Tel.: (0231) 755-2745

Fax: (0231) 755-2752

E-Mail: Johannes.Krettek@tu-dortmund.de

1 Einführung

Die Aspekte der Suche und Entscheidungsfindung sind im Rahmen der multikriteriellen Optimierung inhärent miteinander verknüpft. Das primäre Ziel der evolutionären Suche besteht in der Generierung einer Menge von Lösungen welche die tatsächliche Pareto-Front möglichst gut repräsentiert. Aus dieser Menge wählt der Entscheidungsfinder im Nachhinein die mit seinen Präferenzen oder seiner Nutzenfunktion übereinstimmende Kompromisslösung aus. Die Entscheidungsfindung im Anschluss an die evolutionäre Suche eignet sich für niedrigdimensionale Probleme mit etwa zwei bis vier Kriterien bei welcher die evolvierte Population untereinander nicht-dominiertes Lösungen die Pareto-Front hinreichend dicht abdeckt.

Bei der a priori Artikulation von Präferenzen spezifiziert der Entscheidungsfinder, der auch als Experte oder Benutzer bezeichnet wird, eine skalare Nutzenfunktion welche die Kompromisse zwischen den einzelnen Kriterien widerspiegelt im Vorhinein. Für die Mehrzahl praktischer Optimierungsprobleme bleibt diese Vorgehensweise insofern unbefriedigend als der Entscheidungsfinder, insbesondere ohne vorherige Kenntnis der letztendlich realisierbaren Kompromisslösungen, nicht in der Lage ist die einzelnen Kriterien quantitativ gegeneinander abzuwägen.

Die Rolle von Benutzerpräferenzen innerhalb der multikriteriellen evolutionären Optimierung und der damit verwandten Frage der multikriteriellen Entscheidungsfindung wurde von C. Coello in [1] diskutiert. Einen Übersichtsartikel jüngeren Datums zur Frage der Integration von Präferenzen und der Interaktion mit dem Benutzer bieten Rachmawati et al. [2].

Eine Alternative zu a priori und a posteriori Entscheidungen stellt die Benutzerbefragung während der laufenden Optimierung dar, bei welcher ein Experte seine Präferenzen innerhalb der aktuellen Population nicht-dominiertes Lösungen äußert. Seine Entscheidungen und die daraus abgeleiteten Präferenzen beeinflussen die Selektion der zukünftigen evolutionären Suche. Die Entscheidungsfindung fällt leichter, da der Experte lediglich alternative Lösungen qualitativ miteinander vergleicht ohne ein absolutes quantitatives Urteil zu fällen. Bei der rein interaktiven evolutionären Optimierung beruht die Selektion ausschließlich auf dem Experten welcher dadurch gezwungen ist alle Lösungen jeder Generation zu bewerten. Solch eine Vorgehensweise ist kaum praktikabel, da die Anzahl der zu treffenden Einzelbewertungen die Kapazität des Experten bei weitem überschreitet. Um den Experten so wenig wie möglich zu beanspruchen trifft seine Entscheidungen bezüglich einer repräsentativen Teilmenge gegenseitig nicht dominiertes Lösungen. Im Extremfall vergleicht der Experte lediglich zwei Lösungen miteinander. Aus der verglei-

chenden Bewertung der Repräsentanten wird auf die globalen Präferenzen des Benutzers hinsichtlich der ihm nicht gezeigten Lösungen geschlossen. Beispiele für diese Vorgehensweise beinhalten die so genannte probabilistische Entwicklung von Zielkonflikten (PROTRADE) [3], fortschreitende Artikulation von Präferenzen (STEP) [4] und sequentielle multikriterielle Problemlösung (SEMOPS) [5]. Ein interaktives Werkzeug um Lösungen der evolutionären Optimierung auszuwählen wird von Deb et al. in [6] vorgestellt.

Unser Ansatz zur fortlaufenden Präferenzbildung ist insofern neuartig und universell als dass ihm kein spezifisches Modell der Benutzerpräferenzen oder Ziele zu Grunde liegt. Die Expertenentscheidungen dienen als Trainingsbeispiele für ein instanzbasiertes Lernverfahren welches ein Präferenzmodell erzeugt. Die relative Präferenz für ein dem Experten nicht gezeigtes Lösungspaar wird aus dessen Ähnlichkeit mit expliziten Expertenentscheidungen hergeleitet. Aus der paarweisen Präferenz von Lösungen lässt sich die für den Experten maßgebliche Rangfolge der aktuellen Population herleiten. Unser Ansatz berücksichtigt zudem die Unvergleichbarkeit von Lösungen [7], ein Konzept welches die gegenläufige Bevorzugung zweier Lösungen ermöglicht indem sie die Vergleichbarkeit beider Lösungen zusätzlich zur Präferenzentscheidung erfasst. Lediglich miteinander vergleichbare Lösungen sind während der multikriteriellen Selektion dem gegenseitigen Wettbewerb untereinander ausgesetzt, wohingegen sich unvergleichbare Teilpopulationen unabhängig voneinander evolvieren.

Aus unserer Sicht spielt bei der interaktiven multikriteriellen Optimierung die Erkundung des Lösungsraumes eine gleichberechtigte Rolle zur Optimierung. Dabei ist der Experte nicht gezwungen seine Ziele ultimativ festzulegen, sondern ist vielmehr in der Lage unterschiedliche, nicht zusammenhängende Regionen des Suchraumes zu erkunden. Gerade im Zusammenhang mit evolutionärem Design von technischen Systemen bietet die iterative Verzahnung von Spezifikation und Optimierung deutliche Vorteile gegenüber einer sequentiellen Vorgehensweise. Die Eigenschaft nebenläufiger Präferenzen ist insbesondere in den frühen Phasen der Optimierung von Vorteil, in welchen die letztendlich realisierbaren Kompromisslösungen noch unbekannt sind. Unvergleichbare Lösungen sind dem wechselseitigen Wettbewerb entzogen, so dass sich die Population in voneinander unabhängige nicht miteinander wechselwirkende Inselformationen aufteilt. Der Selektionsdruck ist begrenzt auf Lösungen innerhalb der gleichen Teilpopulation. Auf diese Art und Weise verteilt der Experte die Optimierungsressourcen des evolutionären Algorithmus gegebenenfalls auf mehrere unzusammenhängende Regionen der Pareto-Front. In dem vorgeschlagenen Schema ist ein Paar von Lösungen charakterisiert durch den Grad ihrer gegenseitigen Präferenz und Vergleichbarkeit. Beide Eigenschaften sind im Sinne einer Fuzzy-Menge gradueller Natur, beinhalten jedoch die binären Fallunterscheidungen. Im Falle der vollständigen Vergleichbarkeit der Lösungen untereinander, konvergiert der evolutionäre Algorithmus auf eine einzelne, die Präferenzen des Experten zum höchsten Grade erfüllende Lösung. Dieser Fall ist äquivalent zu dem einer skalaren Optimierung. Im Falle der vollständigen Vergleichbarkeit bei nicht entscheidbarer Präferenz beruht der Wettbewerb und damit die Selektion alleinig auf dem Aspekt der Pareto-Dominanz. Dieses Szenario stellt den Fall der klassischen multikriteriellen evolutionären Optimierung dar, welche die gleichmäßige Verteilung von Lösungen entlang der Pareto-Front zum Ziel hat.

2 Interaktive Präferenzartikulation

Ohne ausreichendes Wissen um mögliche Alternativen, ist es für den Prozessexperten schwierig, unter Umständen auch unmöglich Präferenzen beziehungsweise einen Kompromiss für mehreren Kriterien konkret zu quantifizieren. Wesentlich einfacher hingegen fällt das Formulieren von Präferenzen für einen Vergleich von nur zwei möglichen Lösungen während des Verlaufs der Optimierung. Die interaktive Formulierung von Präferenzen wirft einige bereits bekannte [8, 9] Fragen auf:

1. Wann werden dem Benutzer welche Informationen oder Lösungen zur Bewertung vorgelegt?
2. Welche Art von Entscheidung muss der Benutzer treffen?
3. Wie beeinflusst die Entscheidung den Prozess der Optimierung?

2.1 Schema für paarweise Vergleiche von Lösungen

Für den Selektionsprozess formuliert der befragte Experte idealerweise seine Präferenzen für alle Lösungen der aktuellen Population. In diesem Falle würde die Selektion exakt die Präferenz des Experten abbilden und nicht nur annähern. Eine solche komplette Interaktion würde jedoch schon für Optimierungen kleineren Umfanges die menschlichen Fähigkeiten zur Informationsverarbeitung überschreiten. Die Menge der nicht-dominierten Lösungen zu einem beliebigen Zeitpunkt während der Optimierung ist üblicherweise größer als das eine Experte bereit wäre diese vollständig zu bewerten. Die Interaktive Entscheidung ist daher begrenzt auf eine Menge repräsentativer Lösungsprototypen die durch ein Clustering der gesamten Lösungsmenge gewonnen werden.

Für das vorgeschlagene Schema wählt ein hierarchischer Clustering-Algorithmus N Cluster um die gesamte Menge der n Pareto-optimalen Lösungen zu repräsentieren, für jeden Cluster wird die dem Clustermittelpunkt nächste Lösung als Lösungsprototyp gewählt. Der Experte bewertet nun diese Repräsentanten mittels eines paarweisen Vergleichs aller Prototypen hinsichtlich ihrer relativen Qualität sowie ihrer Vergleichbarkeit miteinander. Die Zahl von N Clustern erfordert vom Experten $N(N - 1)/2$ paarweise Vergleiche und Entscheidungen. Eine Lösung $S_i = \{\vec{x}_i, f(\vec{x}_i)\}$ wird repräsentiert durch den Parametervektor \vec{x}_i und den Kriterienvektor $f(\vec{x}_i)$. Der Experte formuliert im Hinblick auf ein Lösungspaar S_i, S_j seine Präferenz als $\sigma(S_i, S_j) \in [-1, 1]$.

Der Extremfall $\sigma(S_i, S_j) = 1$ bedeutet das S_i vollständig S_j vorgezogen wird, $\sigma(S_i, S_j) = -1$ bedeutet das genaue Gegenteil. Ein $\sigma(S_i, S_j) = 0$ bedeutet zwei gleichwertige Lösungen, Zwischenwerte werden als schwächere Präferenzen für die eine oder andere Lösung interpretiert. Zusätzlich klassifiziert der Benutzer ein Lösungspaar hinsichtlich seiner Vergleichbarkeit mittels eines Vergleichbarkeitsgrads $\rho \in [0, 1]$. Für $\rho = 1$ sind die beiden Lösungen vollständig Vergleichbar. Es besteht in fundamentaler Unterschied zwischen der Klassifikation von zwei Lösungen als "gleich" $\sigma = 0$ und "unvergleichbar" $\rho = 0$. Im erstgenannten Fall stehen beide Lösungen im Wettbewerb miteinander, können jedoch nicht unterschieden werden, im zweiten Fall findet kein Wettbewerb zwischen beiden Lösungen statt, die Selektion setzt ausschließlich vergleichbare Lösungen in Konkurrenz zueinander. Die Menge D_σ der verglichenen Paare aller Prototypen, $\{S_i, S_j\}$ werden in einer Datenbank abgelegt um für die Präferenzschätzung zur Verfügung zu stehen. Alle g

Generationen wird dieser Vorgang wiederholt und die ermittelten Vergleiche der Menge D_σ hinzugefügt.

2.2 Präferenzschätzung

Die Benutzerpräferenz über die gesamte Menge von Pareto-optimalen Lösungen in der aktuellen Population wird anhand der durch Benutzerbefragung ermittelten Datenbank vergleichener Lösungen D_σ vorhergesagt. Dem liegt die Annahme zu Grunde, dass der Benutzer Lösungspaare die sich ähnlich sind auch ähnlich bewerten würde.

Die Ähnlichkeit des zu bewertenden Paares $\{P_k, P_l\}$ mit jedem Paar $\{S_i, S_j\}$ aus der Menge D_σ wird bestimmt durch die Entfernung der Lösungen im normalisierten Kriterienraum nach folgender Gleichung:

$$\Delta(\{P_k, P_l\}, \{S_i, S_j\}) = \|f(\vec{x}_k) - f(\vec{x}_l)\|_2 + \|f(\vec{x}_i) - f(\vec{x}_j)\|_2 \quad (1)$$

Der gewichtete Einfluss eines einzelnen Trainingsdatenpaares ist bestimmt durch eine Gauß'sche Verteilung über dieses Abstandsmaß.

$$w(\{P_k, P_l\}, \{S_i, S_j\}) = \exp\left(-\frac{\Delta(\{P_k, P_l\}, \{S_i, S_j\})}{d_{min}}\right) \quad (2)$$

in der d_{min} dem mittleren kleinsten Abstand innerhalb der Lösungsmenge entspricht.

Die geschätzte Präferenzrelation für ein Paar wird dann als auf diese Art gewichtete Summe der Präferenzen aller Trainingspaare berechnet:

$$\hat{\sigma}(P_k, P_l) = \frac{\sum_{i,j} \sigma(S_i, S_j) w(\{P_k, P_l\}, \{S_i, S_j\})}{\sum_{i,j} w(\{P_k, P_l\}, \{S_i, S_j\})} \quad (3)$$

Die Vergleichbarkeit $\hat{\rho}$ jedes Paares wird auf die gleiche Art bestimmt:

$$\hat{\rho}(P_k, P_l) = \frac{\sum_{i,j} (\rho(S_i, S_j) - 1) w(\{P_k, P_l\}, \{S_i, S_j\})}{\sum_{i,j} w(\{P_k, P_l\}, \{S_i, S_j\})} \quad (4)$$

2.3 Integration von Benutzerpräferenzen in einen evolutionären Algorithmus

Der verwendete multikriterielle evolutionäre Algorithmus entspricht weitgehend den in der Literatur verbreiteten Methoden, der Ablauf ist in Abbildung 1 skizziert. Die Startpopulation wird zufällig initialisiert, ein diskreter Rekombinations-Operator sowie ein Mutations-Operator mit adaptiver Schrittweite finden Verwendung.

In der multiobjektiven evolutionären Optimierung beruht die Selektion hinsichtlich der ermittelten mehrkriteriellen Fitness der Individuen primär auf dem Konzept der Dominanz. Eine Lösung A dominiert eine Lösung B genau dann wenn sie in mindestens einem der Kriterien überlegen ist und sich in keinem der anderen Kriterien als schlechter erweist. Die Selektion bevorzugt diejenigen Lösungen, welche von keinem oder möglichst wenigen anderen Lösungen der Population dominiert werden. Der Rang einer Lösung ergibt sich zu

$$r_i = 1 + p_i \quad (5)$$

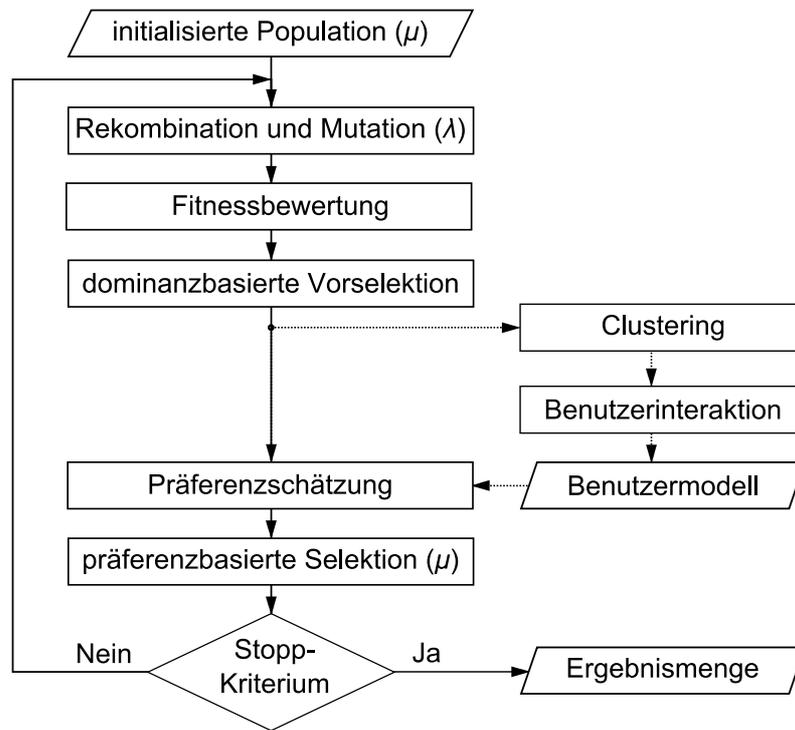


Bild 1: Evolutionärer Algorithmus mit Benutzerinteraktion

wobei p_i die Anzahl der gegenüber Lösung i dominierenden Lösungen darstellt. Alle nicht dominierten Lösungen erhalten den Rang eins und damit die höchste Fitness. Im Laufe der Evolution und damit verbundenen Annäherung an die Pareto-Front wächst der Anteil nicht dominierter Lösungen und übersteigt die Anzahl der zu selektierenden Eltern. Aus Sicht der multikriteriellen Optimierung sind alle nicht-dominierten Lösungen als gleichwertig zu betrachten, und das Ziel besteht darin die Pareto-Front durch homogen verteilte Lösungen abzudecken. Aus Sicht des Entscheidungsfinders unterscheiden sich jedoch die nicht-dominierten Lösungen hinsichtlich ihrer Güte. Im in Abbildung 1 gezeigten Ablauf folgt daher nun je nach Parametrierung die beschriebene interaktive Präferenzartikulation, in jedem Falle folgt jedoch ein auf dieser Präferenz basierender weiterer Selektionsschritt. Die Aufgabe der Präferenz-basierten Selektion besteht darin aus den bereits vorselektierten, nicht dominierten Lösungen die aus Sicht des Experten bevorzugten Lösungen als Eltern zu selektieren.

2.4 Präferenz-basierte Selektion

Im Anschluss an die Selektion aller n nicht-dominierter Lösungen aus der aktuellen Population wird ein weiterer Selektionsschritt anhand der Benutzerpräferenzen durchgeführt. Dazu trifft das instanzbasierte Präferenzmodell eine Vorhersage über die Präferenzrelation und Vergleichbarkeit aller Lösungspaare, gebildet aus der Kombination jeder der n Lösungen mit den jeweils anderen $n - 1$ Lösungen. Für jede Lösung wird aus allen Paaren an der diese beteiligt ist ein aggregierter Präferenz- und ein Vergleichbarkeitsindex nach Gleichung 6 beziehungsweise 7 ermittelt.

$$\gamma_{\sigma}(P_k) = \frac{\sum_{l=1}^n \hat{\sigma}(P_k, P_l) \hat{\rho}(P_k, P_l)}{\sum_{l=1}^n \hat{\rho}(P_k, P_l)} \quad (6)$$

Dieser Index bildet die mittlere relative Präferenz einer Lösung im Vergleich zu ihren unmittelbaren Mitbewerbern ab. Die Vergleichbarkeit $\hat{\rho}$ der Lösung beschränkt dabei den Wettbewerb auf mehr oder weniger disjunkte Untermengen ähnliche Individuen der gesamten Population. Der Vergleichbarkeitsindex $\hat{\rho}$ erfasst die relative Dichte der Lösungen im Kriterienraum.

$$\gamma_{\rho}(P_k) = \sum_{l=1}^n \hat{\rho}(P_k, P_l) \quad (7)$$

Um eine gleichmäßige Verteilung der Lösungen auf alle präferierten Regionen der Pareto-Front zu gewährleisten, wird dieser Index minimiert, ein Ansatz ähnlich zu verbreiteten Methoden zum Erhalt von Diversität in Pareto-Mengen, bei dem dicht benachbarte Lösungen für die Selektion abgewertet werden.

Als Ziel ergibt sich dadurch die simultane Maximierung von γ_{σ} und Minimierung von γ_{ρ} um ein ausgeglichenes Verhältnis zwischen Verbesserung und Erforschung zu erreichen. Die besten Lösungen werden gewählt entsprechend dem verbreiteten multikriteriellen Selektionsschema nach Fonseca [10], beim dem der Rang eines Individuums von der Anzahl der von ihm dominierten Lösungen abhängig ist. Die μ Individuen mit dem besten Rang hinsichtlich Präferenz und Vergleichbarkeit werden als Eltern der nächsten Generation selektiert.

2.5 Künstliches Benutzermodell

Bei der Entwicklung und Analyse interaktiver Mechanismen für evolutionäre Algorithmen erweist sich die Einbeziehung eines menschlichen Experten als problematisch. In erster Linie wäre die zeitliche Beanspruchung des Experten durch die hohe Anzahl von Versuchen und Entscheidungen kaum tragbar. Um das Verhalten verschiedener Varianten einer Optimierung objektiv miteinander zu vergleichen muss das Antwortverhalten des Experten reproduzierbar sein. Es ist jedoch fraglich, ob ein menschlicher Experte über einen längeren Zeitraum hinweg konsistente Entscheidungen trifft. Um eine objektive Analyse der Methoden zu gewährleisten, wird daher der Experte durch ein künstliches, dafür jedoch transparentes und konsistentes Expertenmodell, welches ein mögliches plausibles Entscheidungsverhalten nachbildet. Dieses künstliche Modell übernimmt anstelle des Experten die Beantwortung der ihm gestellten vergleichenden Bewertungen. Die evolutionäre Optimierung und dass von ihr erlernte Präferenzmodell verfügt über keinerlei Kenntnisse dieses künstlichen Modells. Das Modell erlaubt die reproduzierbare Analyse hinsichtlich Häufigkeit und Auswahl der Vergleiche unterschiedlicher Varianten der Benutzerinteraktion.

Das Antwortverhalten des künstlichen Benutzermodells zielt auf die Minimierung des Abstands von Lösungen zu entweder einem oder mehreren hypothetischen, optimalen Zielen im normalisierten Kriterienraum. Die das Benutzermodell definierenden hypothetischen Zielpunkte liegen dabei in der Nähe der tatsächlichen Pareto-Front jedoch im nicht realisierbaren Gebiet des Kriterienraums. Das Optimum aus Sicht des Benutzers ist die Paretolösung mit dem kleinsten Anstand zum hypothetischen Zielpunkt. Aus diesem Optimierungsziel leitet sich folgende abstands-basierte Präferenzrelation für ein Paar $\{S_i, S_j\}$

her

$$\sigma_M(S_i, S_j) = \frac{||f(\vec{x}_j) - R|| - ||f(\vec{x}_i) - R||}{||f(\vec{x}_i) - R|| + ||f(\vec{x}_j) - R||} \quad (8)$$

Die Präferenz ergibt sich der normierten Differenz der Abstände beider Lösungen zum Zielpunkt R wobei die nähere der beiden Lösungen bevorzugt wird. Im Falle mehrfacher konkurrierender Zielpunkte wird nur denjenigen Paaren eine Präferenz zugewiesen, bei denen beide Lösungen demselben Zielpunkt am nächsten liegen. Ist dies nicht der Fall, so wird das Paar als "nicht vergleichbar" ($\rho = 0$) bewertet.

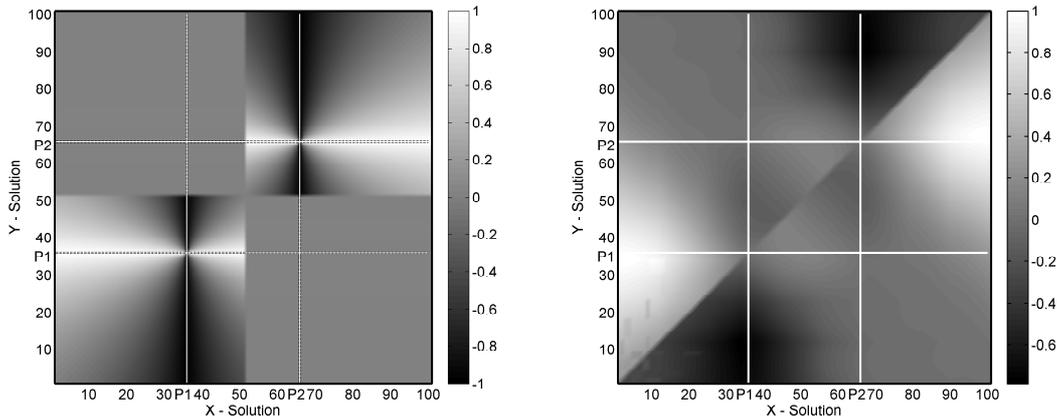


Bild 2: Präferenzrelation von Lösungspaaren (X,Y) des künstlichen Benutzermodells $\sigma_M(X, Y)$ (links) und des geschätzten Präferenzmodells $\hat{\sigma}(X, Y)$ basierend auf zehn Prototypen (rechts)

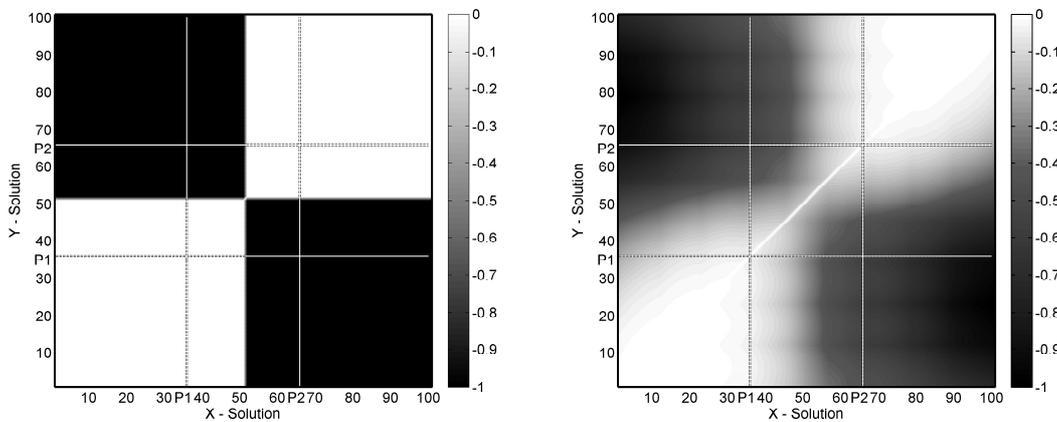


Bild 3: Vergleichbarkeitsindex von Lösungspaaren (X,Y) des künstliche Benutzermodells $\sigma_M(X, Y)$ (links) und des geschätzten Präferenzmodells $\hat{\sigma}(X, Y)$ basierend auf zehn Prototypen (rechts)

Die Abbildungen 2 und 3 vergleichen die nominelle künstliche Benutzerpräferenz und Vergleichbarkeit mit den geschätzten Modellen. Die geschätzten Modelle basieren auf 45 paarweisen Vergleichen der zehn gleichmäßig verteilten Prototypen. Der Kriterienraum ist für dieses Beispiel eindimensional gewählt, die dem Benutzermodell zugrunde liegenden Referenzziele sind $P_1 = 35$ und $P_2 = 65$. Die Abbildung 2 zeigt, dass die nominelle Präferenz $\sigma(X, Y)$ (links) und die geschätzte Präferenz $\hat{\sigma}(X, Y)$ (rechts) prinzipiell übereinstimmen, der in der Nähe der Zielpunkte nicht kontinuierliche Verlauf der nominellen

Tabelle 1: Rate der falschen Selektionen in Prozent

	$N = 100$	$N = 25$	$N = 10$	$N = 5$
$g = 1$	2.98	3.34	4.18	5.24
$g = 5$	4.09	4.91	5.39	6.68
$g = 10$	4.90	5.75	5.98	6.92

Präferenz wird jedoch durch die gewichtete Mittelwertbildung über die diskreten Prototypen deutlich geglättet.

Die Abbildung 3 vergleicht entsprechend die tatsächliche und geschätzte Vergleichbarkeit eines Lösungspaares (X, Y) . Für die nominelle Vergleichbarkeit sind die zwei Zielregionen $R_1 = \{X|X \in [0, 50]\}$ und $R_2 = \{X|X \in [50, 100]\}$ strikt disjunkt. Die Approximation dieser diskreten Partitionierung durch das geschätzte Modell weist wiederum in der Grenzregion einen geglätteten Verlauf auf. Dies führt dazu, dass Lösungen in der Grenzregion beiden Referenzpunkten gleichermaßen zugeordnet werden.

Die Approximationsgüte des erlernten Präferenzmodells hängt vor allem von der Art und Anzahl der zur Verfügung stehenden Lerndaten ab, und damit von der Anzahl der Cluster N beziehungsweise paarweisen Vergleichen und der Häufigkeit der Interaktionen ab. Im Gegensatz zu herkömmlichen Modellierungsaufgaben ist die absolute Approximationsgüte im Kontext der evolutionären Optimierung weniger relevant. Entscheidend ist vielmehr der Einfluss der geschätzten Benutzerpräferenzen auf die Selektionsentscheidung. Der Approximationsfehler des Benutzermodells bewirkt, dass die Selektion andere Lösungen auswählt als der Experte. Der Selektionsfehler ist definiert als die Anzahl der fehlerhaft selektierten besten Lösungen im Vergleich zur idealen Selektion anhand des nominellen Benutzermodells. Die Tabelle 1 zeigt den Einfluss der Anzahl der Clusterprototypen und der Häufigkeit der Interaktion auf den mittleren Selektionsfehler eines evolutionären Laufs für die Kursawe Testfunktion mit $\mu = 25$ Eltern und $\lambda = 100$ Nachkommen über 50 Generationen. Die angegebenen Fehlerraten sind das Ergebnis einer Mittelwertbildung über zehn unabhängige Läufe. Die Ergebnisse zeigen das eine deutliche Reduktion der Interaktionsrate sowohl in der Anzahl der Fragen als auch deren Häufigkeit lediglich zu einer moderaten und durchaus akzeptablen Verschlechterung der Selektionsqualität führt. In den meisten Optimierungsszenarien bestimmt sich der maßgebliche Aufwand für den interaktiven evolutionären Algorithmus durch die Interaktion selbst, weniger durch die Rechenzeit des Algorithmus. Ziel ist es daher, die Anzahl der Fragen, und damit die Inanspruchnahme des Experten weitestgehend zu minimieren. Dies geschieht durch gezielte Auswahl derjenigen paarweisen Vergleichen, die unabhängig von der zu erwartenden Antwort den größten Zuwachs an Information für das Präferenzmodell versprechen. Mit anderen Worten, gesucht sind diejenigen Vergleiche hinsichtlich derer die induzierte Selektionsentscheidung die höchste Sensitivität besitzt. In Anlehnung an das Konzept des "active learning" aus dem Gebiet des maschinellen Lernens, bei dem das Lernverfahren selber über die nächste Frage entscheidet, wird dem Benutzer dasjenige Lösungspaar zur Bewertung präsentiert, welches bei der zu erwartenden Entscheidung die größte Auswirkung auf das induzierte Ranking der Lösungen besitzt. In anderen Worten sind die informativsten Fragen die, die bei einer von der bisher geschätzten Präferenz abweichenden Bewertung im Vergleich zum Modell ohne diese zusätzliche Information die größte Veränderung in der Selektionsentscheidung bewirken.

2.6 Verbesserte Präferenzmodellierung durch gezielte Auswahl der Prototypen zur Interaktion

Nahezu unabhängig vom tatsächlich zu optimierenden Problem stellt zumindest aus Sicht des Experten die interaktive Bewertung von Prototypen den größten Teil des Gesamtaufwandes für die Optimierung dar. Tatsächlich kann angenommen werden, dass die Rechenzeit des Algorithmus für die evolutionäre Optimierung selbst gegenüber dem Zeitaufwand für die Interaktion zu vernachlässigen ist. Der Aufwand für die eigentliche Gütebewertung dagegen ist nicht in jedem Fall vernachlässigbar. Das Ziel einer interaktiv operierenden Optimierung ist es daher, mit möglichst wenigen Interaktionen eine schnelle Konvergenz der Suche zu erzielen, was beide für den Gesamtaufwand relevanten Aspekte nämlich Anzahl der Gütebewertungen und Anzahl der Interaktionen positiv beeinflusst.

Um die Anzahl der dem Experten gestellten Fragen zu reduzieren gilt es aus gestellten Fragen den größtmöglichen Informationsgewinn zu erzielen. Informationsgewinn bedeutet in diesem Zusammenhang eine Verbesserung der auf den modellierten Präferenzen basierenden Selektion. Daher ist die mit den möglichen Ergebnissen des Vergleichs einhergehende Änderung der Selektionsentscheidung das maßgebliche Kriterium zur Auswahl diskriminierender Fragen. Die zu erwartenden Änderung ergibt sich durch Vergleich des bisherigen Modells mit dem um das Ergebnis des Vergleichs als zusätzliches Trainingsdatum erweiterte Modell.

Für den untersuchten Vergleich $\{S_i, S_j\}$ wird zunächst die Präferenzentscheidung $\hat{\sigma}$ durch das aktuelle Modell geschätzt. Die tatsächliche Entscheidung wird der prognostizierten ähnlich sein aber dennoch von ihr abweichen. Diese Variabilität der Antwort wird durch zwei prototypische, der geschätzten Entscheidung ähnliche Antworten nachgebildet, nämlich $\hat{\sigma}\{S_i, S_j\} + \epsilon$ und $\hat{\sigma}\{S_i, S_j\} - \epsilon$. Für beide Varianten wird die induzierte Selektionsreihenfolge $(R_{+\epsilon}, R_{-\epsilon})$ hinsichtlich der geschätzten Präferenz ermittelt. Für die Variabilität der Antwort wird konkret $\epsilon = 0.2$ gewählt.

Die ersten μ potentiell selektierten Individuen beider induzierten Rangfolgen werden mit der Rangfolge der ohne dieses neue Trainingsbeispiel bisherigen Selektionsentscheidung R_0 verglichen. Als Sensitivitätsmaß dient die Summe der Verschiebungen der Rangfolge der μ besten Lösungen.

$$\Delta_{(+)} = \sum_{i=1}^{\mu} (R_{+\epsilon}(i) - R_0(i)) \quad \Delta_{(-)} = \sum_{i=1}^{\mu} (R_{-\epsilon}(i) - R_0(i)) \quad (9)$$

Die Abweichung Δ in der Selektionsreihenfolge stellt den Gradienten des möglichen Informationsgewinnes dar. Davon ausgehend, dass die größere der beiden Änderungen durch eine dem lokalen Trend gegenläufige Entscheidung hervorgerufen wird, dient die kleinere der beiden Änderung $\Delta = \min(\Delta_+, \Delta_-)$ als Kriterium. Bei N Prototypen ergeben sich $N * (N - 1)$ potentielle Vergleiche, von denen derjenige mit dem höchsten potentiellen Informationsgewinn ausgewählt wird.

Die Abbildung 4 vergleicht die Entwicklung des Selektionsfehlers über den Verlauf einer Optimierung für eine rein dominanzbasierte Selektion, eine interaktive Selektion bei Bewertung eines aus $N * (N - 1)$ möglichen Vergleichen zufällig gewählten Vergleichs in jeder Generation und eines anhand des Informationsgewinn gezielt ausgewählten Vergleichs in jeder Generation. Zur Bewertung der Effizienz dient der Selektionsfehler ausgedrückt als die Differenz des mittleren Abstandes zum Referenzpunkt der durch das approximierte Benutzermodell selektierten Lösungen im Vergleich zum selben Abstand der auf

dem tatsächlichen Benutzermodell beruhenden idealen Selektion. In anderen Worten, das gewählte Kriterium spiegelt die durch Approximation des Benutzermodells verursachte Verschleppung der Konvergenz des evolutionären Algorithmus wieder.

Wie zu erwarten, zeigt sich ein deutlich geringere Verlangsamung des evolutionären Fortschritts bei einer durch ein Präferenzmodell unterstützten Selektion gegenüber einer ausschließlich dominanzbasierten Selektion, die in Unkenntnis der Präferenzen des Experten agiert. Weiterhin wird deutlich, dass die gezielte Auswahl von Vergleichen gegenüber einer rein zufälligen Generierung von Trainingsbeispielen gerade in den späteren Generationen zu einer beschleunigten Konvergenz beiträgt.

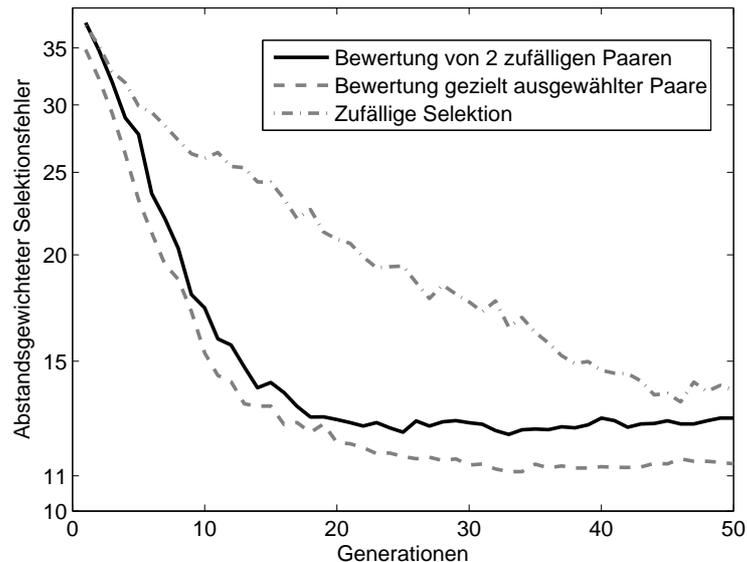


Bild 4: Vergleich der interaktiven Bewertung gezielt gewählter und zufällig gewählter Prototypen-Paare

Die Abbildung 5 illustriert den Einfluss der Interaktionsrate r , definiert als die Anzahl der Vergleiche pro Generation, auf die Entwicklung des durch das Präferenzmodell verursachten Selektionsfehlers. Die Bewertungen durch den modellierten Benutzer werden dabei gleichmäßig über die Generationen der Optimierung verteilt, r ersetzt damit die bisher verwendeten Parameter N und g durch den Zusammenhang $r = N(N - 1)/2g$. Die dargestellten Ergebnisse überlagern auf Grund der Gewichtung der Fehler mit deren Abstand zum Referenzpunkt den Fortschritt der Verbesserung des Benutzermodells mit dem allgemeinen Fortschritt der Optimierung. Dieser wird wiederum von der Qualität der präferenzbasierten Selektion beeinflusst. Deutlich wird die mit der Anzahl der Vergleiche zunehmende Güte des Benutzermodells. Eine zu niedrige Interaktionsrate hingegen führt zu einer suboptimalen Selektionsentscheidung die im Extremfall sogar schlechter als eine zufällig Auswahl abschneidet. Die Degradation bei sehr niedriger Interaktionsrate erklärt sich aus dem unterschiedlichen Schwerpunkt und Verteilung der aktuellen Population gegenüber den als Trainingsbeispiele fungierenden, mittlerweile veralteten Vergleichen aus früheren Generationen. Mit anderen Worten, die vom Experten gegebenen Antworten in lange zurück liegenden Generationen liefern keinerlei Anhaltspunkte für seine Präferenzen bezüglich der aktuellen alternativen Lösungen.

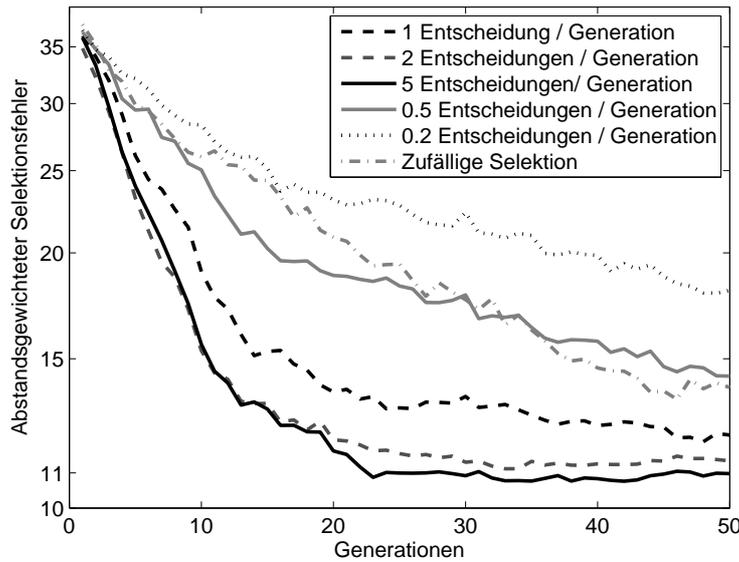


Bild 5: Vergleich des abstandsgewichteten Selektionsfehlers für verschiedene Interaktionsraten über den Verlauf der Generationen

3 Ergebnisse

Das Konvergenzverhalten des vorgestellten Algorithmus wird analysiert anhand der bekannten Kursawe Testfunktion

$$f_1(\vec{x}) = \sum_{i=1}^5 \left(-10 \exp \left(-0.2 \sqrt{x_i^2 + x_{i+1}^2} \right) \right) \quad (10)$$

$$f_2(\vec{x}) = \sum_{i=1}^6 (|x_i|^{0.8} + 5 \sin(x_i)^3) \quad (11)$$

mit zwei Kriterien f_1 und f_2 . Das hypothetische Benutzermodell basiert auf einem einzelnen Referenzpunkt $P_{Ref} = (-35; -21.5)$. Die Konvergenz der interaktiven Optimierung wird verglichen mit einem globalen multikriteriellen Algorithmus dessen Selektion nur auf dem Dominanz-Ranking und einem Niching-Mechanismus basiert. Ebenfalls verglichen wird ein skalarer evolutionärer Algorithmus, als dessen Fitness der euklidische Abstand der Lösungen zum Referenzpunkt P_{Ref} dient.

3.1 Vergleich der interaktiven Optimierung zu herkömmlichen Strategien

Es wird erwartet, dass sich die interaktive Methode als weniger effizient als die skalare Optimierung mit praktisch a priori bekannten Präferenzen verhält, jedoch die Konvergenz der ungerichteten multikriteriellen Optimierung zur Pareto-Front übertrifft. Die Konvergenz wird analysiert in Form des Abstandes zwischen dem Referenzpunkt und der am nächsten gelegenen Lösung in der aktuellen Population. Der skalare Algorithmus optimiert hierbei direkt den Abstand zum Referenzpunkt, der multikriterielle Algorithmus dagegen berücksichtigt diesen Abstand nicht, da alle nicht-dominierten Lösungen als gleichwertig

behandelt werden. Die interaktive Optimierung hat kein a priori Wissen über den Referenzpunkt, mit fortschreitender Optimierung bildet das erlernte Präferenzmodell jedoch den Abstand dazu ab. Zur Visualisierung der unterschiedlichen Konvergenzverhalten wird eine Population mit lediglich $\mu = 10$ Eltern und $\lambda = 50$ Nachkommen über 20 Generationen evolviert. Die Abbildung 6 zeigt die Entwicklung der Population gegen die Pareto-Front. Der Grauwert illustriert das Fortschreiten der Generationen, von hellgrau zu Beginn bis zu schwarz in der letzten Generation.

3.1.1 Skalare Optimierung:

Die skalare Optimierung des Abstandes zum Referenzpunkt, dargestellt in Abbildung 6 konvergiert schnell in Richtung Pareto-Front in der Umgebung des Referenzpunktes $P_{Ref} = (-35; -21.5)$. Eine lineare Gewichtung zu Grunde liegenden Kriterien konvergiert nicht unbedingt in ein einziges Minimum, die Verwendung des euklidischen Abstandes zum Referenzpunkt als Fitness-Bewertung führt zu einem eindeutigen Ergebnis.

3.1.2 Multikriterielle Optimierung:

Die multikriterielle Optimierung ohne die Berücksichtigung von Präferenzen entwickelt sich in Richtung der globalen Pareto-Front wie in Abbildung 6 gezeigt. Da der Referenzpunkt bei der Optimierung in keiner Weise berücksichtigt wird, verteilen sich die Lösungen gleichmäßig über die Pareto-Front. Die Konvergenz im Bezug auf den Referenzpunkt ist demnach erwartungsgemäß langsamer.

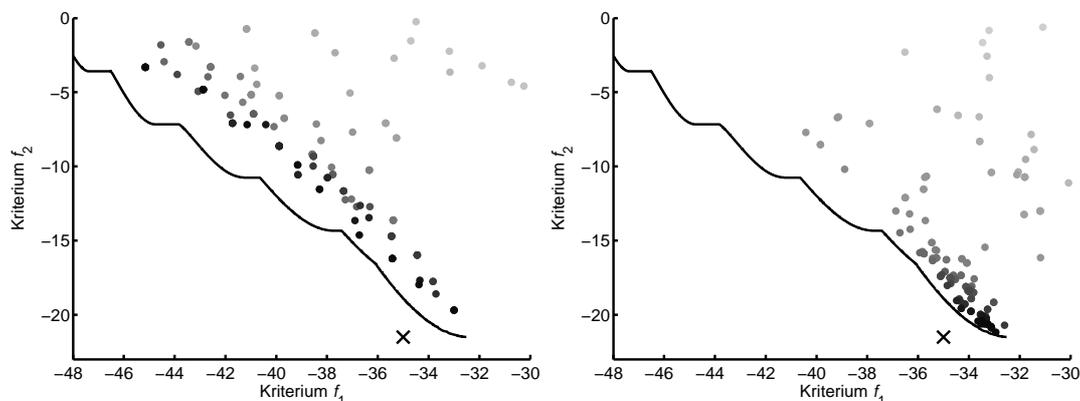


Bild 6: Vergleich der Algorithmen: Multikriterielle Optimierung (links) und Skalare Optimierung (rechts)

3.1.3 Multikriterielle Optimierung mit Präferenz-basierter Selektion

Die Integration des Präferenzmodells zur Selektion resultiert in einer schnelleren Konvergenz in Richtung des Referenzpunktes wie in Abbildung 7 dargestellt. Zu beachten ist, dass sich der Algorithmus auf Grundlage der paarweisen Vergleiche entwickelt. Die Population entwickelt sich in Richtung des Referenzpunktes, obgleich zu Beginn langsamer. Die Einfluss des Modells und auch das Lernen beginnen nicht mit der ersten Generation,

da nur für mehr als μ nicht-dominierte Lösungen die Präferenz in die Selektion einbezogen wird. Das Präferenzmodell wird mit dem fortschreiten der Optimierung genauer und nähert sich dem skalaren Optimum. Die Abbildung 7 (rechts) zeigt die Konvergenz der

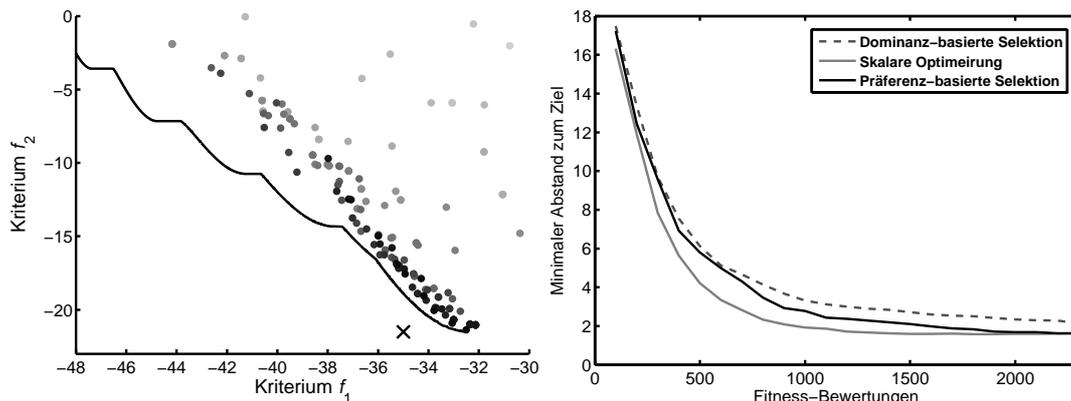


Bild 7: Präferenzbasierte Optimierung (links), Vergleich der Konvergenz der verschiedenen Verfahren (rechts)

aktuell besten Lösung in Richtung des Referenzpunktes über den Verlauf der Evolution. Die Darstellung ist das Ergebnis einer Mittelwertbildung über 25 unabhängige Optimierungen.

3.2 Interaktive Optimierung mit mehreren disjunkten Präferenzen

Das Konzept der Nichtvergleichbarkeit von Lösungspaaren ermöglicht es, die Optimierung auf mehrere disjunkte Regionen des Kriterienraums zu fokussieren. Dieser Abschnitt analysiert die Konvergenz der Optimierung in Richtung zweier separater Referenzpunkte für die 2-D Kursawe Testfunktion. Das verwendete Benutzermodell hat eine positive Präferenz für die dem Referenzpunkt näher gelegene Lösung sofern beide Lösungen nahe an einem der Referenzpunkte liegen. Liegen die Lösungen nahe zu verschiedenen Referenzpunkten so wird das Lösungspaar als nicht vergleichbar eingestuft. Die Konvergenz des Algorithmus, ausgedrückt durch die Summe der beiden kürzesten Abstände zu den Referenzpunkten, wird verglichen für die multikriterielle Optimierung, zwei parallele skalare Optimierungen sowie die Präferenz-basierte Optimierung mit jeweils $\mu = 20$ Eltern und $\lambda = 100$ Nachkommen. Abbildung 8 (links) zeigt die Entwicklung der Population über den Verlauf der Optimierung hin zu den beiden Referenzpunkten $P_{Ref1} = (-53; -21)$ und $P_{Ref2} = (-60; -13)$.

Die Abbildung 8 (rechts) vergleicht die Konvergenz der drei beschriebenen Algorithmen im Hinblick auf die Summe der kleinsten Abstände zu den Referenzpunkten. Die Ergebnisse basieren wieder auf der Mittelwertbildung über 25 unabhängige Optimierungen. Für den skalaren Algorithmus werden einzelne Läufe für beide Referenzpunkte zusammengefasst um einen fairen Vergleich zu ermöglichen, die Anzahl der Fitness-Bewertungen wird dabei auf beide Läufe verteilt. Der Algorithmus mit Präferenz-basierter Selektion übertrifft nicht nur den normalen multikriteriellen Algorithmus sondern ist im Mittel auch effizienter als die Kombination der skalaren Optimierungen der gewichteten Kriterien. Dies zeigt deutlich die Möglichkeiten zur Annäherung der Pareto-Front in Bereichen von Interesse für den Benutzer durch die interaktive Optimierung. Sogar im Fall eines nur

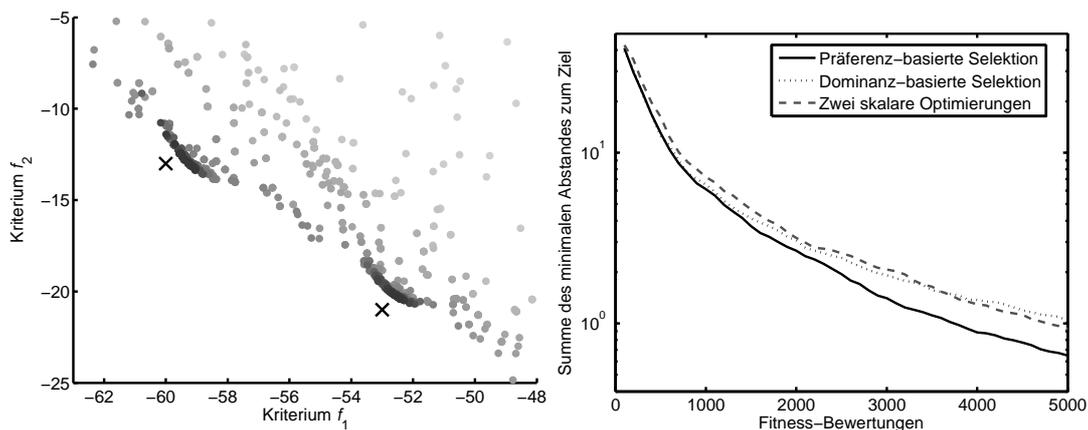


Bild 8: Entwicklung der Population (links), Konvergenzvergleich der verschiedenen Algorithmen (rechts)

zweidimensionalen Optimierungsproblems zeigt sich der Vorteil der interaktiven Integration von Präferenzen. Nachdem ein ausreichendes Präferenzmodell erlernt ist, ist die Leistung des interaktiven Algorithmus mit dem einer skalaren Optimierung mindestens vergleichbar. Anders als bei dieser ist jedoch keine a priori Festlegung der Präferenzen notwendig und der Algorithmus bleibt multikriteriell und ist nicht auf ein einzelnes Ziel beschränkt.

3.3 Konvergenzverhalten

Dieser Abschnitt analysiert das Konvergenzverhalten dreier Varianten multikriterieller Optimierung für eine unterschiedliche Anzahl an Kriterien. Als Konvergenzmaß dient der euklidische Abstand zum Referenzpunkt. Die skalare Optimierung verwendet als Gütemaß eben jenen euklidischen Abstand der Lösung zum Referenzpunkt. Dieses Maß ist in einigen Fällen inkompatibel mit der gemäß Gleichung 5 auf der Dominanz basierenden Rangfolge. Eine dominierte Lösung in der Nähe des Referenzpunktes wird durch das skalare Abstandsmaß gegenüber einer auf der Pareto-Front entfernt liegenden nicht dominierten Lösung bevorzugt. Die Antworten des Experten bei paarweisen Vergleichen von Lösungen basieren auf dem hypothetischen Präferenzmodell welches zum Referenzpunkt näher liegende Lösungen bevorzugt. Skalare und multikriterielle Optimierung mit Benutzermodell konvergieren langfristig gegen das gleiche Optimum. Die in Abbildung 9 dargestellten Verläufe des kleinsten euklidischen Abstands der Population zum Referenzpunkt basieren auf der Mittelung über zehn unabhängige Optimierungsläufe der zwei-, vier- und sechs-dimensionalen Kursawe-Funktion. Das vier- und sechs-dimensionale Optimierungsprobleme entsteht durch Replikation der Optimierungsparameter und Kriterien der eigentlich zwei-kriteriellen Kursawefunktion.

In allen Fällen ist die ungerichtete multikriterielle, globale Optimierung den beiden gezielten Varianten hinsichtlich der Konvergenz zum Referenzpunkt unterlegen. Dieses Verhalten ist zu erwarten, insofern als die Optimierung keinen Bezug zum Referenzpunkt besitzt. Die Population konvergiert gegen die Pareto-Front und verteilt sich auf ihr gleichförmig. Der letztendliche Abstand zum Referenzpunkt verhält sich umgekehrt zur Dichte der Lösungen entlang der Pareto-Front. Bei gleicher Anzahl an Lösungen verringert sich die Dichte exponentiell mit zunehmender Anzahl an Kriterien. Dementsprechend vergrößert sich der Abstand bei hochdimensionalen Problemen im stärkeren Maße als bei

den beiden zielgerichteten Varianten. Nach circa 20 Generationen hat sich die Population gleichmäßig über die Pareto-Front verteilt und der Abstand zum Referenzpunkt unterliegt nur noch einer zufälligen Drift verursacht durch die zufällige Umverteilung der Lösungen entlang der Pareto-Front. Bei den beiden zielgerichteten Optimierungsstrategien ist auch nach 50 Generationen keine vorzeitige Stagnation erkennbar. Mit zunehmender Anzahl an Kriterien erweist sich die direkte skalare Optimierung als zunehmend überlegen, allerdings erfordert diese wie bereits dargelegt eine genaue vorherige Kenntniss der erreichbaren Lösungen. Die Degradation des Konvergenzverhaltens der interaktiven multikriteriellen Optimierung mit zunehmender Anzahl an Kriterien ist deutlich robuster als im Fall der globalen Optimierung. Dennoch sinkt die Vorhersagekraft des Benutzermodells für hochdimensionale Kriterienräume durch dessen steigende Komplexität bei gleich bleibender Anzahl an Trainingsdaten. Im zweidimensionalen Fall, erweist sich die interaktive Optimierung der skalaren Variante als überlegen, da bei dieser auch dominierte Lösungen auf Kosten der Vielfalt nicht-dominiertes Lösungen selektiert werden. Zusammenfassend lässt sich feststellen, dass die interaktive multikriterielle Optimierung selbst bei einer großen Anzahl an Kriterien einen guten Kompromiss zwischen Erkundung der zunächst unbekanntenen Pareto-Menge einerseits und auf die Präferenzen des Experten zielender lokaler Optimierung ermöglicht.

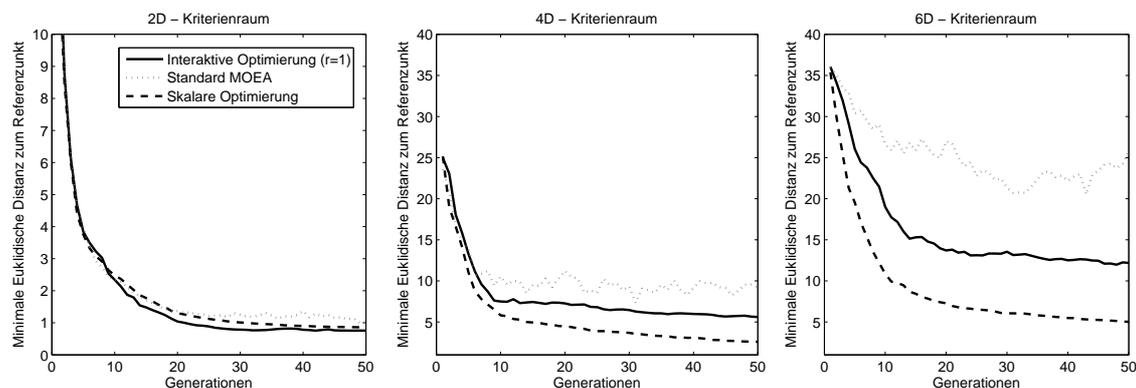


Bild 9: Konvergenz bei skalarer, multikriterieller und interaktiver Optimierung

4 Zusammenfassung

Dieser Beitrag präsentiert eine neue Methode zur Inkorporation von Benutzerpräferenzen in die multikriterielle evolutionäre Optimierung. Ein Präferenzmodell wird durch paarweisen Vergleich prototypischer Lösungen durch den Experten erlernt und dient innerhalb der Selektion der Unterscheidung und Auswahl nicht dominierter Lösungen. Das Modell beruht auf keinerlei Annahmen über die Struktur der Benutzerpräferenz und deckt damit sowohl den Fall einer einzelnen Ziellösung, mehrerer gleichrangiger Ziellösungen als auch globalen Approximation der Pareto-Front ab. Der Experte ist nicht gezwungen seine Ziele a priori in quantitativer Form zu definieren, sondern trifft lediglich qualitative Entscheidungen im Angesicht realisierbarer Kompromisslösungen. Die Beanspruchung des Experten durch die Befragung wird durch gezielte Auswahl der für die nachfolgende Optimierung besonders informationshaltigen Fragen begrenzt. Die Analyse des Konvergenzverhaltens verdeutlicht, dass der gewählte Ansatz sowohl dem Aspekt der breiten,

multikriteriellen Erkundung des Lösungsraumes als auch der mit den Expertenzielen in Einklang stehenden Fokussierung der Suche auf Teilregionen Rechnung trägt.

5 Danksagung

Die Arbeit wurde von der Deutschen Forschungsgemeinschaft (DFG) im SFB-Projekt 531 Design und Management komplexer technischer Prozesse und Systeme mit Methoden der Computational Intelligence gefördert.

Literatur

- [1] Coello, C. A. C.: Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. *Proceedings of the CEC2000* (2000), S. 30–37.
- [2] Rachmawati, L.; Srinivasan, D.: Preference Incorporation in Multi-objective Evolutionary Algorithms: A Survey. *Proceedings of the CEC2006* (2006), S. 962 – 968.
- [3] Goicoechea, A.; Duckstein, L.; Fogel, M.: Multiobjective Programming in Watershed Management: A Study of the Charleston Watershed. *Water Resources Research* (1976) 12, S. 1085–1092.
- [4] Cohon, J. L.: *Multiobjective Programming and Planning*. Dover Publications. 2004.
- [5] Monarchi, D.; Kisiel, C.; Duckstein, L.: Interactive multiobjective Programming in water resources: a case study. *Water Resources Research* (1973) 9, S. 837–850.
- [6] Deb, K.; Chaudhuri, S.: I-EMO: An interactive evolutionary multi-objective optimization tool. *First International Conference on Pattern Recognition and Machine Intelligence* (2005), S. 690–695.
- [7] Hüllermeier, E.; Brinker, K.: Fuzzy-Relational Classification: Combining Pairwise Decomposition Techniques with Fuzzy-Preference Modeling. *Proc. of the 5th International Conference of the European Society for Fuzzy Logic and Technology* (2007).
- [8] Parmee, I. C.; Cvetkovic, D.; Watson, A.; Bonham, C.: Multiobjective Satisfaction within an Interactive Evolutionary Design Environment. *Evolutionary Computation* 8 (2000) 2, S. 197–222.
- [9] Takagi, H.: Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE* 89 (2001) 9, S. 1275–1296.
- [10] Fonseca, C.; Fleming, P.: Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. *Proceedings of the 5th International Conference on Genetic Algorithms* (1993), S. 416–423.

Robuster Beobachterentwurf für Takagi-Sugeno Fuzzy Systeme zur Überwachung hydrostatischer Fahrtriebe

Horst Schulte

Bosch Rexroth AG, Hydraulics
E-Mail: horst.schulte@boschrexroth.de

In mobilen Arbeitsmaschinen werden häufig hydrostatische Fahrtriebe eingesetzt. Um den wachsenden Anforderungen bezüglich Kraftstoffverbrauch und Emissionsverhalten technologisch begegnen zu können, werden zunehmend elektronische Steuerungen mit automatisierten Fahrstrategien eingesetzt, bei denen der Fahrer lediglich als übergeordnete Kontrollorgan fungiert. Dieser Trend stellt neue Anforderungen an die Betriebssicherheit, da bewährte mechanische durch mechatronische Systeme ersetzt werden und dabei in vielen Fällen ebenso die mechanische bzw. hydraulische Rückfallebene entfällt. Um jedoch innerhalb sicherheitskritischer Funktionen Fehler rechtzeitig zu erkennen und darauf geeignet reagieren zu können, müssen zusätzlich Überwachungsfunktionen mit sicherheitsgerechten Systemarchitekturen in die elektronischen Steuerungen integriert werden.

In dieser Arbeit wird ein modellbasierter Ansatz verfolgt, mit dem exemplarisch die Schwenkwinkeldynamik von Hydraulikpumpen und der Druckaufbau in hydrostatischen Antrieben mit elektronischen Steuerungen überwacht werden kann. Ausgehend von einem generalisierten Modell hydrostatischer Getriebe in Takagi-Sugeno Fuzzy Form wird eine robuste LMI-basierte Beobachtersynthese vorgestellt. Im Einzelnen werden hierzu systematisch zwei reduzierte Zustandsbeobachter zum Schätzen der Druckänderung und der Pumpenschwenkwinkel abgeleitet. Es werden die Vorteile aber auch Grenzen dieses Modellansatzes zur Überwachung hydrostatischer Antriebe in mobilen Arbeitsmaschinen aufgezeigt.

1 Einleitung

Immer mehr Arbeitsprozesse mobiler Arbeitsmaschinen zeichnen sich durch einen steigenden Automatisierungsgrad aus. Dabei werden automatisierte Vorgänge sowohl bei Funktionen im Fahrtrieb als auch bei der Arbeitshydraulik eingesetzt. Zum Beispiel wird durch die automatisierte Steuerung hydrostatischer Fahrtriebe der Bediener entlastet während die prinzipbedingte Entkopplung der Dieseldrehzahl von der Fahrzeuggeschwindigkeit eine energieoptimierte Steuerung des gesamten Antriebsstrangs ermöglicht. Da in Systemen mit einem hohen Automatisierungsgrad die mechanischen bzw. hydraulischen Rückfallebenen durch X-by-wire-Technologien ersetzt werden, müssen zusätzlich Überwachungsfunktionen mit sicherheitsgerechten Systemarchitekturen in die elektronischen Steuerungen integriert werden [6]. Aufgabe der Überwachungsfunktionen ist es sicherheitskritische Fehlfunktionen zu erkennen um geeignete Reaktionen rechtzeitig ausführen zu können. Unter einer geeigneten Reaktion versteht man in diesem Zusammenhang eine Aktion, die den sicheren Zustand des Gesamtsystem wieder herstellt oder aufrecht erhält. Das kann entweder heißen, dass das System unmittelbar nach Erkennen einer Fehlfunktion abgeschaltet wird (Fail-Safe), dass seine Funktion eingeschränkt (Fail-Soft) oder sogar voll aufrecht erhalten wird (Fail-Operational). Systeme mit Fail-Operational-Verhalten müssen fehlertolerant ausgelegt sein, d.h. die Funktionen bleiben

ausfallfrei, auch wenn einzelne Komponentenfehler im Rahmen der Fehlervorgabe auftreten [15]. Hierzu zählen z.B. mehrkanalig ausgeführte X-by-wire Systeme, die auch nach Ausfall eines Kanals durch Einspringen eines zweiten bzw. n-ten Kanals redundant verfügbar bleiben.

Die in dieser Arbeit vorgestellte modellbasierte Überwachungsfunktion ist Teil einer sicherheitsgerechten Systemarchitektur für hydrostatische Fahrtriebe. Im Einzelnen ist die Überwachungsfunktion entweder Bestandteil einer Selbstdiagnose oder -prüfung einer Signalquelle bzw. eines Kanals oder ermöglicht durch analytische Redundanz mehrkanalige Anordnungen mit reduziertem Aufwand an Hardware. Beim erstgenannten Fall wird durch Selbstdiagnose der sicherheitskritische Fehler erkannt und die Sicherheitsfunktion ausgeführt. Das heißt, das System wird unter Fehlermeldung in einen sicheren Zustand überführt (Fail-Safe). Beim zweitgenannten Fall nutzt man die Redundanz zur sicheren Beherrschung von Einzelfehlern. Ab 3-kanaligen Systemen wird mittels Entscheidungslogiken, sogenannten Votern, festgestellt welcher Kanal fehlerhaft ist. Dies hat den entscheidenden Vorteil, dass die Funktionalität des Gesamtsystems dabei sichergestellt bleibt (Fail-Operational).

Die modellbasierte Überwachungsfunktion wird im Kern aus zwei reduzierten Zustandsbeobachtern gebildet. Mittels analytischer Redundanz wird die gemessene Schwenkwinkeldynamik von Hydraulikpumpen und der gemessene Druckaufbau im geschlossenen Hauptkreislauf kontinuierlich überwacht. Aus der Abweichung zwischen den errechneten und gemessenen Prozessgrößen lassen sich Merkmale zur Fehlererkennung gewinnen.

Auf die Merkmalsgewinnung wird in dieser Arbeit nicht eingegangen. Schwerpunkt ist die robuste Beobachtersynthese und die experimentelle Validierung der reduzierten Beobachterstrukturen unter Zuhilfenahme der Modellgleichung hydrostatischer Fahrtriebe in Form von Takagi-Sugeno Fuzzy Systemen [12]. Die eigentliche Beobachtersynthese besteht aus der Anwendung und Erweiterung von zwei allgemein gültigen Sätzen zur globalen asymptotischen Stabilität der Fehlerdifferentialgleichung aus [7] und [13] für Takagi-Sugeno Fuzzy Systeme.

Zum Entwurf nichtlinearer Beobachter wurden in den letzten 20 Jahren eine Vielzahl von Ansätzen verfolgt (siehe z.B. die Übersicht [9] oder die Arbeiten in [8]). Die wesentliche Schwierigkeit bei der nichtlinearen Beobachtersynthese liegt im Bestimmen einer nichtlinearen Beobachter-Normalform. In den Koordinaten der nichtlinearen Beobachter-Normalform besitzen solche Beobachter eine exakt lineare Fehlerdynamik. Der eigentliche Entwurf erfolgt dann z.B. mit der Polvorgabe in diesen Koordinaten. Ein Nachteil dieser Methode ist, dass die nichtlineare Beobachter-Normalform nur unter sehr restriktiven Bedingungen, die an das zu beobachtende System gestellt werden, eingesetzt werden kann. Folglich kann diese Methodik der Beobachtersynthese nur in wenigen Fällen angewandt werden [1].

Der in dieser Arbeit vorgestellte Ansatz hat gegenüber Beobachtern mit nichtlinearer Beobachter-Normalform zwei wesentliche Vorteile:

1. Durch Verwendung von Stabilitätssätzen im Sinne von Lyapunov ist die Existenzbedingung von Takagi-Sugeno Beobachtern mit global asymptotisch stabiler Fehlerdynamik (Beobachterkonvergenz) wesentlich schwächer als bei Normalform-Beobachtern mit exakt linearer Fehlerdynamik.
2. Nebenbedingungen wie das Einschwingverhalten unter Stellgrößenbeschränkungen und Modellunsicherheiten der zu beobachtenden Strecke sind mathematisch formulierbar und Bestandteil der Beobachtersynthese.

Wesentliches Ziel dieser Arbeit ist es eine Beobachterstruktur mit Hilfe von wohldefinierten mathematischen Sätzen zu entwerfen, die auf generalisierten physikalischen Modellen hydrostatischer Fahrtriebe beruht. Damit soll zum einen ermöglicht werden, technische Anforderungen an die Prozessüberwachung durch mathematisch formulierbare Nebenbedingungen in den Entwurfprozess zu integrieren und zum anderen soll ermöglicht werden, die Ergebnisse auf neue Antriebsvarianten einfach übertragen zu können. Ein weiteres wichtiges Kriterium ist der Rechenaufwand bei der Online-Überwachung. Dieser ist möglichst gering zu halten, um auch bei umfangreichen Steuerungsaufgaben (die Überwachungsfunktion ist immer nur eine Unterfunktion in einem Steuerprogramm der elektronischen Fahrtriebsregelung) die Implementierung auf elektronischen Seriensteuergeräten gewährleisten zu können. Letzte Anforderung an die Online-Berechenbarkeit schließt zudem die Anwendung von Zustandsbeobachtern basierend auf Extended Kalman-Filter (EKF) Ansätzen zur Berechnung der zu überwachenden Prozessgrößen aus.

Im folgenden zweiten Abschnitt dieser Arbeit werden zunächst kurz die in [10] und [11] abgeleiteten reduzierten physikalischen Modellgleichungen und das daraus gebildete nichtlineare Zustandsraummodell zur Beschreibung und Simulation hydrostatischer Fahrtriebe vorgestellt.

Das Ziel besteht darin, handhabbare mathematische Modelle anzugeben. Diese sollen die dominanten Zeitkonstanten, die wesentlichen Nichtlinearitäten und Modellunsicherheiten enthalten, um eine systematische Beobachtersynthese zu ermöglichen. Danach werden mit der Einführung geeigneter Sektornichtlinearitäten die Modellgleichungen auf zwei äquivalente Takagi-Sugeno Fuzzy Systeme in Standardform transformiert [13]. Diese werden anschließend experimentell an einem Versuchsfahrzeug validiert. Im dritten Abschnitt werden, basierend auf den zuvor abgeleiteten TS-Fuzzy Systemen, zwei reduzierte Beobachteransätze zum Schätzen der Druckänderung und der Pumpenschwenkwinkel vorgestellt. Damit können die Beobachtermatrizen mittels linearer Matrixungleichungen unter Verwendung der Matlab Toolbox *LMI Control* [3] berechnet werden.

2 Modellbildung und Ableitung geeigneter TS-Fuzzy Systeme

2.1 Physikalische Modellgleichungen

Der prinzipielle Aufbau von hydrostatischen Fahrtrieben ist in Bild 1 dargestellt. Ein Dieselmotor treibt eine hydraulische Verstellpumpe an, die durch Rohr- oder Schlauchleitungen in einem geschlossenen Kreis mit einem Hydromotor verbunden ist. Der Hydromotor ist abtriebsseitig über feste Getriebestufen (meist zwischen Motorwelle und Kardanwelle sowie zwischen Kardanwelle und den Radachsen) mit den Antriebsrädern verbunden. Die häufigste Bauart von Pumpen und Motoren in mobilen Anwendungen ist die Axialkolbenmaschine, wobei die Pumpe mit einer Schrägscheibenverstellung und der Motor mit einer Schrägachsenverstellung versehen ist. Bei der Axialkolbenpumpen wird, falls die Schrägscheibe nicht senkrecht gegenüber der Triebwerksachse steht (Pumpenschwenkwinkel $\alpha_P \neq 0$), bei Drehung von Triebwelle und Zylinderblock an den Kolben ein Hub erzeugt. Bei Axialkolbenmotoren wird an den Kolben eine Hubbewegung ermöglicht, falls die Zylinderblockachse gegenüber der Triebwellenachse schräg steht (Motorschwenkwinkel $\alpha_M \neq 0$). Dies bewirkt eine Drehung von Triebwelle und Zylinderblock. Aufgrund der hohen Leistungsdichte und Zuverlässigkeit ist die Kombination aus Pumpe mit Schrägscheibenverstellung und Motor mit Schrägachsenverstellung besonders für

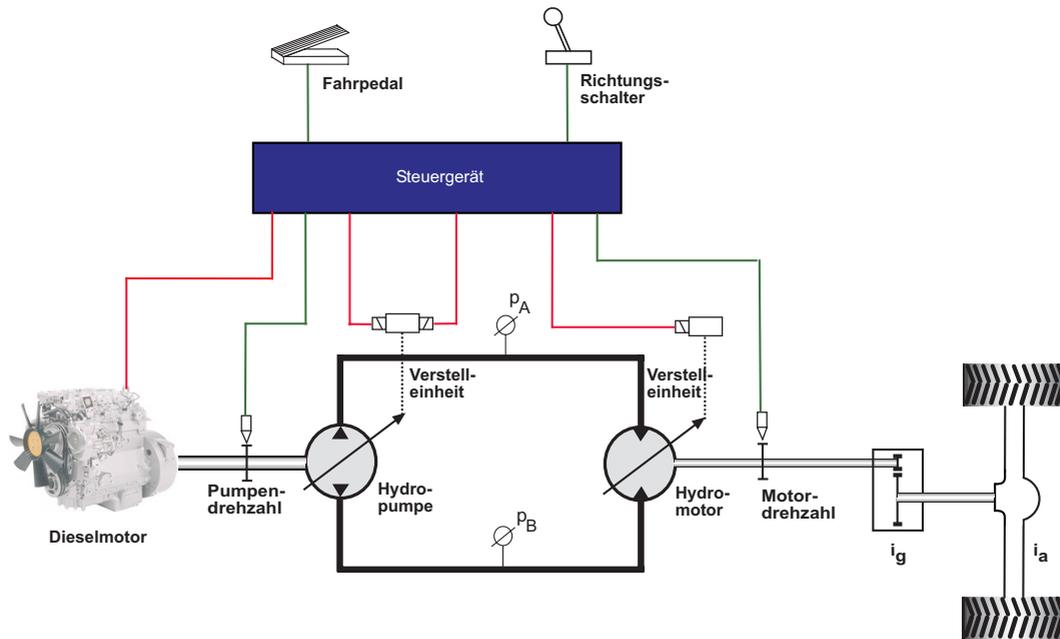


Bild 1: Aufbau und Komponenten elektronisch geregelter hydrostatischer Fahrtriebe

stufenlose hydrostatische Getriebe geeignet.

Bei Fahrtrieben mit elektronischer Steuerung und elektro-proportionaler Verstellung wird über ein elektronisches Fahrpedal und einem Richtungsschalter die gewünschte Getriebeübersetzung eingestellt. Dabei wird mittels elektro-hydraulischer Verstellung das Fördervolumen der Pumpe q_P sowie mit der elektro-hydraulischen Ansteuerung vom Motor der aufgenommene Volumenstrom q_M in Menge und Richtung derart variiert, dass sich das gewünschte Übersetzungsverhältnis einstellt. Hierzu werden die Hydropumpendrehzahl n_P , die Hydromotordrehzahl n_M sowie die Drücke p_A und p_B im geschlossenen Kreis erfasst. Die grundlegenden physikalischen Zusammenhänge werden im Folgenden kurz vorgestellt. Eine ausführliche Beschreibung der Modellbildung und -reduktion erfolgt in [11].

Der von der Pumpe geförderte Volumenstrom q_P ergibt sich aus dem verstellbaren Verdrängungsvolumen $V_P(\alpha_P)$ und der Pumpendrehzahl n_P bzw. ω_P

$$q_P = V_P(\alpha_P) n_P = \tilde{V}_P(\alpha_P) \omega_P \quad \text{mit} \quad \tilde{V}_P = \frac{V_P}{2\pi}. \quad (1)$$

Um das gewünschte Verdrängungsvolumen zu erreichen, wird die Schwenkscheibe mit einem hydraulischen Zylinder verstellt. Die Dynamik zwischen der elektromagnetischen Ansteuerung der Ventile des Stellzylinders und dem Schwenkwinkel kann in erster Näherung durch ein Verzögerungsglied erster Ordnung beschrieben werden:

$$T_{u_P} \dot{\tilde{\alpha}}_P + \tilde{\alpha}_P = k_P u_P \quad \text{mit} \quad \tilde{\alpha}_P = \frac{\alpha_P}{\alpha_{max_P}}, \quad (2)$$

dabei bezieht sich der Schwenkwinkel auf den maximalen Wert α_{max_P} . Da die Pumpe in beide Richtungen ausschwenkbar ist, gilt für den Schwenkwinkel $\tilde{\alpha}_P \in \{-1, 1\}$.

Der den Hydromotor antreibenden Volumenstrom ergibt sich aus

$$q_M = V_M(\alpha_M) n_M = \tilde{V}_M(\alpha_M) \omega_M \quad \text{mit} \quad \tilde{V}_M = \frac{V_M}{2\pi}. \quad (3)$$

Wie bei der Pumpe wird die Verstellodynamik vom Motor durch ein Verzögerungsglied erster Ordnung beschrieben:

$$T_{u_M} \dot{\tilde{\alpha}}_M + \tilde{\alpha}_M = k_M u_M \quad \text{mit} \quad \tilde{\alpha}_M = \frac{\alpha_M}{\alpha_{max_M}} \quad (4)$$

Im Gegensatz zur Hydropumpe kann der Motor nur in eine Richtung verstellt werden, das heißt, das Verdrängungsvolumen des Motors ist stets positiv, somit ist $\tilde{\alpha}_M \in \{0, 1\}$. Berücksichtigt man im weiteren, dass der nominelle Arbeitsbereich der beiden Schwenkwinkel $|\alpha_P| \leq 18^\circ$ und $\alpha_M \leq 20^\circ$ ist, so gilt für die Annahme kleiner Winkel

$$q_P = \tilde{V}_P \tilde{\alpha}_P \omega_P, \quad q_M = \tilde{V}_M \tilde{\alpha}_M \omega_M. \quad (5)$$

Die Dynamik im geschlossenen Kreis wird weitestgehend durch den Kompressibilitätsvolumenstrom q_c zwischen Pumpe und Motor bestimmt. Dieser wird für den unteren und oberen Hydraulikzweig mit Hilfe der konzentrierten hydraulischen Kapazitäten C_A und C_B berechnet, welche die Kapazität des Öls in den Leitungen und die Kapazität der Leitungen berücksichtigen:

$$q_{C_A} = C_A \dot{p}_A, \quad q_{C_B} = C_B \dot{p}_B \quad (6)$$

Mit der Einbeziehung des Kompressibilitätsvolumenstroms im unteren und oberen Leitungszweig sowie der druckabhängigen Leckflüsse

$$q_{leak_P} = k_{leak_P} (p_A - p_B), \quad q_{leak_M} = k_{leak_M} (p_A - p_B) \quad (7)$$

mit $k_{leak_{P,M}}$ als Leckflusskoeffizienten ergeben sich die Volumenstrombilanzen zu

$$q_P - q_M - q_{leak_P} = q_{C_A}, \quad -q_P + q_M + q_{leak_M} = q_{C_B}. \quad (8a)$$

Mit dem Einsetzen von (5), (6) und (7) in (8) erhält man nach kurzer Umformung die Druckaufbaugleichungen für beide Zweige im geschlossenen Kreis:

$$\begin{aligned} \dot{p}_A &= \frac{1}{C_A} \left(\tilde{V}_P \tilde{\alpha}_P \omega_P - \tilde{V}_M \tilde{\alpha}_M \omega_M - k_{leak_P} (p_A - p_B) \right), \\ \dot{p}_B &= -\frac{1}{C_B} \left(\tilde{V}_P \tilde{\alpha}_P \omega_P - \tilde{V}_M \tilde{\alpha}_M \omega_M - k_{leak_M} (p_A - p_B) \right). \end{aligned} \quad (9)$$

Durch die Einführung einer neuen Zustandsgröße $\Delta p = p_A - p_B$ wird nun die mathematische Beschreibung der Hochdrücke um eine Systemordnung reduziert. Unter der Voraussetzung der Symmetrieannahmen

$$C_A = C_B =: C_H, \quad k_{leak_A} = k_{leak_B} =: k_{leak} \quad (10)$$

gilt

$$\Delta \dot{p} = \frac{1}{C_H} \left(\tilde{V}_P \tilde{\alpha}_P \omega_P - \tilde{V}_M \tilde{\alpha}_M \omega_M - k_{leak} \Delta p \right). \quad (11)$$

Dies ist zulässig, da sich der Öldruck einer Kreislaufseite schon bei kleinen Lastwechseln auf den nahezu konstanten Speisedruck einstellt. Die reduzierte Bewegungsgleichung für den gesamten Fahrtrieb wird angegeben durch

$$\dot{\omega}_M J_v + \tilde{d}_{vc} i_a^2 \omega_M = i_g^2 i_a^2 \eta_g M_M \tilde{\alpha}_M - M_{L_w} i_g i_a \quad (12)$$

mit den Parametern i_g als Geriebeübersetzung zwischen Hydromotor- und Kardanwelle, i_a als Achsübersetzung und η_{ga} als mechanischer Getriebewirkungsgrad $\eta_{ga} = \eta_g \eta_a$. Die Massenträgheitsmomente der Antriebsräder und der Einfluss der Fahrzeugmasse werden durch einen konzentrierten Parameter J_v zusammengefasst. Der Dämpfungsfaktor \tilde{d}_{vc} bezieht sich auf die Wirkung vom Verlustmoment bezogen auf die Kardanwelle. Das Antriebsmoment vom Hydromotor M_M auf der rechten Seite ergibt sich zu

$$M_M = \Delta p \tilde{V}_M \tilde{\alpha}_M \eta_{mh} \quad (13)$$

mit dem in erster Näherung konstant angenommenen hydromechanischen Wirkungsgrad η_{mh} des Hydromotors. Die aufgeprägten äußeren Momente an der Radachse werden in dem Lastmoment M_{L_w} zusammengefasst. Insgesamt erhält man für den hydrostatischen Fahrtrieb ein Modell 4. Ordnung. Mit den definierten Zustandsgrößen

$$\mathbf{x} = [x_1, x_2, x_3, x_4]^T := [\tilde{\alpha}_P, \tilde{\alpha}_M, \Delta p, \omega_M]^T, \quad (14)$$

den Eingangsgrößen

$$\mathbf{u} = [u_1, u_2]^T := [u_P, u_M]^T \quad (15)$$

und den zuvor eingeführten Modellgleichungen (2), (4), (11) und (12) ergibt sich das nichtlineare Differentialgleichungssystem zu

$$\begin{aligned} \dot{x}_1 &= -\frac{1}{T_{u_P}} x_1 + \frac{k_P}{T_{u_P}} u_1, \\ \dot{x}_2 &= -\frac{1}{T_{u_M}} x_1 + \frac{k_M}{T_{u_M}} u_2, \\ \dot{x}_3 &= \frac{10}{C_H} \tilde{V}_{max_P} x_1 \omega_P - \frac{10}{C_H} \tilde{V}_{max_M} x_2 x_4 - \frac{10}{C_H} k_{leak} x_3, \\ \dot{x}_4 &= \frac{1}{J_v} i_g^2 i_a^2 \eta_g \eta_{mh} \tilde{V}_{max_P} 10^{-4} x_2 x_3 - \frac{1}{J_v} \tilde{d}_{vc} i_a^2 x_4 - \frac{1}{J_v} M_{L_w} i_g i_a. \end{aligned} \quad (16)$$

Anmerkung: Die in (16) vorkommenden Zustandsgrößen und Parameter haben die Einheiten $C_H [\frac{\text{mm}^5}{\text{N}}]$, $T_{P,M} [\text{s}]$, $V_{max_{P,M}} [\text{mm}^3]$, $k_{leak} [\frac{\text{mm}^3}{\text{s bar}}]$, $x_3 : [\text{bar}]$, $x_4 [\frac{1}{\text{s}}]$ wobei $u_{1,2}$ und $x_{1,2}$ einheitenlos sind.

2.2 Überführung in Takagi-Sugeno Fuzzy Systeme

Die exakte Überführung von nichtlinearen Differentialgleichungssystemen in ein Takagi-Sugeno Fuzzy System der Standardform

$$\dot{\mathbf{x}} = \sum_{i=1}^{N_r} h_i(\mathbf{z}) \mathbf{A}_i \mathbf{x} + \sum_{i=1}^{N_r} h_i(\mathbf{z}) \mathbf{B}_i \mathbf{u} \quad (17)$$

$$\mathbf{y} = \sum_{i=1}^{N_r} h_i(\mathbf{z}) \mathbf{C}_i \mathbf{x} \quad (18)$$

mit $\mathbf{z} \in \mathbb{R}^l$ ist gegeben, falls Nichtlinearitäten durch Sektornichtlinearitäten ersetzt werden können. Die Transformation ist jedoch nicht ein-eindeutig, d.h. aus einem gegebenem

nichtlinearen Differentialgleichungssystem können verschiedene Takagi-Sugeno Fuzzy Systeme exakt abgeleitet werden. Dieser Freiheitsgrad wird in dieser Arbeit ausgenutzt, um eine wesentliche Forderung, die an die Streckenmodelle zur Beobachtersynthese gestellt werden, erfüllen zu können. In [14] wurde gezeigt, dass das Separationsprinzip für T-S Fuzzy Systeme nur dann gilt, falls die einzelnen Variablen in z , vgl. (17), messbar sind. Ist dies nicht der Fall, ist ein separater Beobachterentwurf mit LMI-basierten Methoden nicht ausführbar. Mit der notwendigen Forderung der vollständigen Messbarkeit von z werden nun in zwei Schritten ausgehend von (16) T-S Fuzzy Systeme mit dem Ziel der Überwachung vom

1. Pumpenschwenkwinkel ($\Rightarrow x_1 \neq z_j$ für $j = 1, \dots, l$)
2. Differenzdruck ($\Rightarrow x_3 \neq z_j$ für $j = 1, \dots, l$)

bestimmt.

1. Schritt: Da die Eingänge u linear in (16) eingehen können die Nichtlinearitäten allein durch

$$\mathbf{A}(x_3, x_4, \omega_P) = \begin{bmatrix} -\frac{1}{T_{uP}} & 0 & 0 & 0 \\ 0 & -\frac{1}{T_{uM}} & 0 & 0 \\ \frac{10}{C_H} \tilde{V}_{maxP} \omega_P & -\frac{10}{C_H} \tilde{V}_{maxM} x_4 & -\frac{10}{C_H} k_{leak} & 0 \\ 0 & \gamma x_3 & 0 & -\frac{\tilde{d}_{vc}}{J_v} i_a^2 \end{bmatrix} \quad (19)$$

mit

$$\gamma = \frac{1}{J_v} i_g^2 i_a^2 \eta_g \eta_{mh} \tilde{V}_{maxP} 10^{-4} \quad (20)$$

oder mit einer zweiten Realisierung

$$\tilde{\mathbf{A}}(x_2, \omega_P) = \begin{bmatrix} -\frac{1}{T_{uP}} & 0 & 0 & 0 \\ 0 & -\frac{1}{T_{uM}} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{10}{C_H} \tilde{V}_{maxP} \omega_P & 0 & -\frac{10}{C_H} k_{leak} & -\frac{10}{C_H} \tilde{V}_{maxM} x_2 \\ 0 & 0 & \gamma x_2 & -\frac{\tilde{d}_{vc}}{J_v} i_a^2 \end{bmatrix} \quad (21)$$

erfasst werden. Aufgrund der natürlichen unteren und oberen Beschränktheit der Variablen ω_P, x_3, x_4 in (19) und ω_P, x_2 in (21) können diese jeweils einzeln durch die Linearkombination der Sektorfunktionen w_{j1} und w_{j2} ersetzt werden:

$$z_j = f_j(z_j) = \underbrace{\underline{f}_j \frac{\bar{f}_j - z_j}{\bar{f}_j - \underline{f}_j}}_{=w_{j1}(z_j)} + \underbrace{\bar{f}_j \frac{z_j - \underline{f}_j}{\bar{f}_j - \underline{f}_j}}_{=w_{j2}(z_j)}, \quad \underline{f}_j = \min[z], \quad \bar{f}_j = \max[z] \quad (22)$$

Damit ergeben sich die Systemmatrizen zu

$$\mathbf{A}(z_1, z_2, z_3) = \begin{bmatrix} -\frac{1}{T_{uP}} & 0 & 0 & 0 \\ 0 & -\frac{1}{T_{uM}} & 0 & 0 \\ \frac{10}{C_H} \tilde{V}_{maxP} f_3(z_3) & -\frac{10}{C_H} \tilde{V}_{maxM} f_2(z_2) & -\frac{10}{C_H} k_{leak} & 0 \\ 0 & \gamma f_1(z_1) & 0 & -\frac{\tilde{d}_{vc}}{J_v} i_a^2 \end{bmatrix}, \quad (23)$$

$$\tilde{\mathbf{A}}(z_1, z_2) = \begin{bmatrix} -\frac{1}{T_{uP}} & 0 & 0 & 0 \\ 0 & -\frac{1}{T_{uM}} & 0 & 0 \\ \frac{10}{C_H} \tilde{V}_{maxP} f_2(z_2) & 0 & -\frac{10}{C_H} k_{leak} & -\frac{10}{C_H} \tilde{V}_{maxM} f_1(z_1) \\ 0 & 0 & \gamma f_1(z_1) & -\frac{\tilde{d}_{vc}}{J_v} i_a^2 \end{bmatrix}. \quad (24)$$

Im *zweiten Schritt* werden nun die Variablen der Systemmatrizen (23) und (24) herausgelöst und konzentriert in den Zugehörigkeitsfunktionen $h_i(\mathbf{z})$ mit $i = 1, \dots, N_r$. Diese ergeben sich für (23) aus der Kombination von $l = 3$ Linearkombinationen der Sektor-nichtlinearitäten mit $N_r = 2^l = 8$.

$$\{w_{11}, w_{12}\} \times \{w_{21}, w_{22}\} \times \{w_{31}, w_{32}\}$$

Die Zugehörigkeitsfunktionen werden aus dem Produkt der Sektorfunktionen $w_{jk}(z_j)$ für $k = 1, 2$ ermittelt (vgl. [10])

$$\begin{aligned} h_1(z_1, z_2, z_3) &= w_{11}(z_1) \cdot w_{21}(z_2) \cdot w_{31}(z_3), \dots, \\ h_8(z_1, z_2, z_3) &= w_{12}(z_1) \cdot w_{22}(z_2) \cdot w_{32}(z_3). \end{aligned}$$

Alle zeitinvarianten Systemmatrizen lassen sich kompakt angeben

$$\mathbf{A}_i = \begin{bmatrix} -\frac{1}{T_{uP}} & 0 & 0 & 0 \\ 0 & -\frac{1}{T_{uM}} & 0 & 0 \\ \frac{10}{C_H} \tilde{V}_{maxP} *_3 & -\frac{10}{C_H} \tilde{V}_{maxM} *_2 & -\frac{10}{C_H} k_{leak} & 0 \\ 0 & \gamma *_1 & 0 & -\frac{\tilde{d}_{vc}}{J_v} i_a^2 \end{bmatrix} \quad (25)$$

mit $*_1 \in [\underline{x}_3, \bar{x}_3]$, $*_2 \in [\underline{x}_4, \bar{x}_4]$ und $*_3 \in [\underline{\omega}_P, \bar{\omega}_P]$. Analog werden die Zugehörigkeitsfunktionen für (24) mit der Kombination

$$\{w_{11}, w_{12}\} \times \{w_{21}, w_{22}\}$$

berechnet. Damit erhält man $N_r = 2^2 = 4$ zeitinvariante Systemmatrizen:

$$\tilde{\mathbf{A}}_i = \begin{bmatrix} -\frac{1}{T_{uP}} & 0 & 0 & 0 \\ 0 & -\frac{1}{T_{uM}} & 0 & 0 \\ \frac{10}{C_H} \tilde{V}_{maxP} *_2 & 0 & -\frac{10}{C_H} k_{leak} & -\frac{10}{C_H} \tilde{V}_{maxM} *_1 \\ 0 & 0 & \gamma *_1 & -\frac{\tilde{d}_{vc}}{J_v} i_a^2 \end{bmatrix} \quad (26)$$

mit $*_1 \in [\underline{x}_2, \bar{x}_2]$ und $*_2 \in [\underline{\omega}_P, \bar{\omega}_P]$.

Zusammengefasst ergeben sich für den anschließenden Beobachterentwurf bei Vernachlässigung externer Lasten M_{L_w} zwei T-S Fuzzy Systeme:

$$\dot{\mathbf{x}} = \sum_{i=1}^8 h_i(z_1, z_2, z_3) \mathbf{A}_i \mathbf{x} + \mathbf{B} \mathbf{u} \quad (27)$$

mit $z_1 := x_3, z_2 := x_4, z_3 := \omega_P$ und

$$\dot{\mathbf{x}} = \sum_{i=1}^4 h_i(z_1, z_2) \tilde{\mathbf{A}}_i \mathbf{x} + \mathbf{B} \mathbf{u} \quad (28)$$

mit $z_1 := x_2, z_2 := \omega_P$ bei identischer Eingangsmatrix

$$B = \begin{bmatrix} \frac{k_P}{T_{u_P}} & 0 \\ 0 & \frac{k_M}{T_{u_M}} \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (29)$$

und identischer Ausgangsgleichung

$$y = C x \quad . \quad (30)$$

Anmerkungen:

1. Beide T-S Fuzzy Systeme (27) und (28) erfüllen die Forderung, dass die Variablen in z messbar sind. Das erste System (27) ist damit geeignet zur beobachterbasierten Überwachung der Schwenkwinkel x_1, x_2 . Das zweite System (28) wird verwendet zur Überwachung des Differenzdrucks x_3 .
2. Durch Vernachlässigung der externen Last in (27) und (28) ist z.Z. die Drucküberwachung in hydrostatischen Antrieben eingeschränkt auf Fahrsituationen in der Ebene und mit geringer Steigung. Um externe Lasten (die ein Vielfaches der bereits berücksichtigten Last einnehmen kann) modellbasiert im Beobachter berücksichtigen zu können, muss über eine Fahrsituationserkennung [4] die Beobachterstruktur entsprechend angepasst werden.

2.3 Experimentelle Modellvalidierung

Die im vorhergehenden Abschnitt abgeleiteten T-S Fuzzy Systeme werden nun anhand von realen Messungen validiert. Hierzu wurden Messungen an einer mobilen Arbeitsmaschine bei einer Straßenfahrt (in der Ebene) aufgezeichnet. Im einzelnen wurden die Drücke im geschlossenen Kreis, die Pumpen-, Hydromotordrehzahl und die Stellsignale u_P, u_M gemessen. Anschliessend wurden die Stellsignale und die Pumpendrehzahl als Eingangsgrößen in die Simulation von (27) eingespeist. In Bild 2 sind typische Stellsignalverläufe für mehrere Beschleunigungs-Verzögerungszyklen (0 bis ca. 11 km/h) einer mobilen Arbeitsmaschine dargestellt. Man sieht deutlich wie die Schwenkwinkel von Pumpe und Hydromotor nacheinander verstellt werden (sogenannte Primär-/Sekundärverstellung), wodurch die Getriebeübersetzung $r := \frac{n_M}{n_P}$ kontinuierlich von Null bis zur maximalen hydrostatischen Übersetzung eingestellt wird. Gleichzeitig wird zur Beschleunigung des Fahrzeugs die Dieseldrehzahl elektronisch von der Leerlauf- bis zur Nenn-drehzahl verstellt, siehe zum Beispiel das Messintervall von $t = 2$ s bis 20 s sowie von $t = 38$ s bis 42 s. Der Vergleich zwischen den Messungen und der Simulation erfolgt im Bild 3. Auf den ersten Blick erkennt man eine gute Übereinstimmung zwischen dem simulierten und gemessenen Differenzdruck sowie der simulierten und gemessenen Achsdrehzahl. Im Detail sieht man in den Simulationsdaten Druckspitzen, die in den Messungen nicht enthalten sind. Dies ist zurückzuführen auf eine nicht modellierte Dämpfung zwischen Rad und Fahrbahn. Diese Dynamik kann mit dem reduzierten Modell nicht erfasst werden.

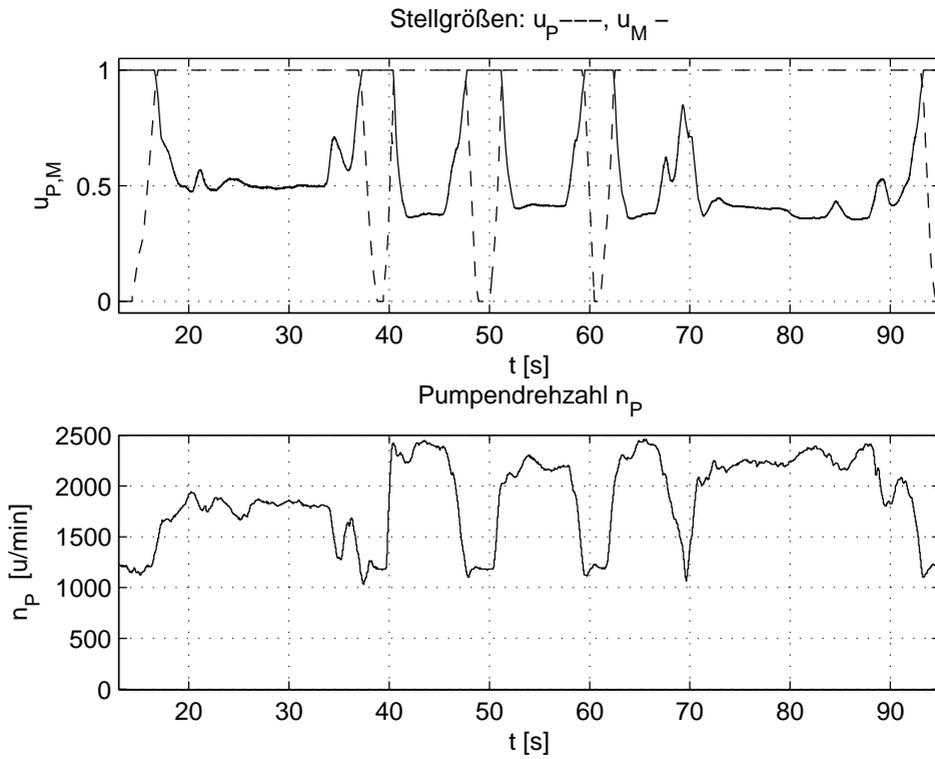


Bild 2: Stellsignale und Pumpendrehzahl bei Beschleunigungs-Verzögerungszyklen

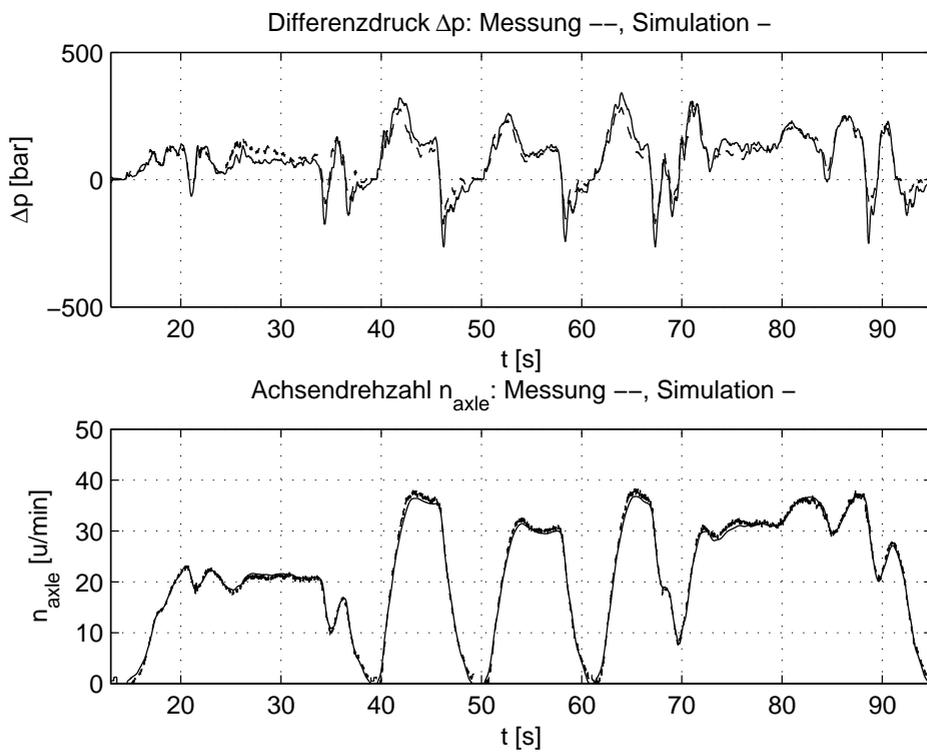


Bild 3: Vergleich Experiment/Simulation mit Gl.(27): Differenzdruck und Achsdrehzahl

2.4 Berücksichtigung von Modellunsicherheiten

Zur Integration von parametrischen Unsicherheiten in T-S Fuzzy Systemen wird in der Literatur [5, 13] häufig eine Form mit additiven Unsicherheiten angegeben

$$\dot{\mathbf{x}} = \sum_{i=1}^{N_r} h_i(\mathbf{z}) (\mathbf{A}_i + \Delta\mathbf{A}_i) \mathbf{x} + \sum_{i=1}^{N_r} h_i(\mathbf{z}) (\mathbf{B}_i + \Delta\mathbf{B}_i) \mathbf{u} \quad (31)$$

$$\mathbf{y} = \sum_{i=1}^{N_r} h_i(\mathbf{z}) (\mathbf{C}_i + \Delta\mathbf{C}_i) \mathbf{x} \quad (32)$$

Bei den untersuchten hydrostatischen Antrieben beschränken wir uns in dieser Arbeit auf eine wesentliche Unsicherheit, die durch Vereinfachungen im Leckflussmodell hervorgerufen wird. In erster Näherung wird der Leckfluss durch die Beziehungen (7) quantifiziert, wobei die unbekannt Parameter $k_{leak_{P,M}}$ häufig aus Messungen geschätzt werden. Tatsächlich hängt der Leckfluss nicht nur vom Differenzdruck, sondern ebenfalls von den aktuellen Schwenkwinkeln, der Pumpen- und der Hydromotordrehzahl ab. Ebenso sind die geschätzten Parameter temperaturabhängig aufgrund der Viskositätsänderung des Öls. In diesem Fall werden die nicht modellierten Dynamiken und die physikalischen Parameterunsicherheiten in einer Funktion $\Delta a(t)$ zusammengefasst. Somit ergibt sich für das nominelle Modell (27) eine Erweiterung

$$\dot{\mathbf{x}} = \sum_{i=1}^8 h_i(z_1, z_2, z_3) (\mathbf{A}_i + \Delta\mathbf{A}_i) \mathbf{x} + \mathbf{B} \mathbf{u} \quad (33)$$

mit

$$\Delta\mathbf{A}_i = \mathbf{H} \Delta a(t) \mathbf{E} \quad (34)$$

wobei \mathbf{H} und \mathbf{E} konstante Matrizen sind und so gewählt werden, dass $a^2(t) \leq 1$ gilt.

Anmerkung: Auf gleichem Wege kann auch das nominelle Modell (28) erweitert werden.

3 Beobachtersynthese zur Überwachung hydrostatischer Antriebe

3.1 Reduzierter Beobachteransatz für Takagi-Sugeno Fuzzy Systeme

Zunächst wird ein reduzierter Beobachteransatz aus [2] für lineare zeitinvariante Systeme in den Grundzügen vorgestellt. Dieser wird dann übertragen auf die bisher abgeleiteten T-S Fuzzy Systeme.

Sei $\mathbf{x} \in \mathbb{R}^n$ ein Zustandsvektor der in einen messbaren $\mathbf{x}_a \in \mathbb{R}^{n-p}$ und in einen beobachtbaren Zustandsvektor $\mathbf{x}_b \in \mathbb{R}^p$ aufgeteilt wird. Daraus folgt mit der Bedingung das $\mathbf{y} = \mathbf{x}_a$ ist direkt die Beziehung

$$\begin{bmatrix} \dot{\mathbf{x}}_a \\ \dot{\mathbf{x}}_b \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{aa} & \mathbf{A}_{ab} \\ \mathbf{A}_{ba} & \mathbf{A}_{bb} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_a \\ \dot{\mathbf{x}}_b \end{bmatrix} + \begin{bmatrix} \mathbf{B}_a \\ \mathbf{B}_b \end{bmatrix} \mathbf{u} \quad (35)$$

$$\mathbf{y} = \begin{bmatrix} \mathbf{E}^{(n-p) \times (n-p)} & \mathbf{0}^{(n-p) \times p} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_a \\ \dot{\mathbf{x}}_b \end{bmatrix} . \quad (36)$$

Die Differentialgleichung der nichtmessbaren Zustandsvariablen kann dann wie folgt angegeben werden

$$\dot{\mathbf{x}}_b = \mathbf{A}_{bb} \mathbf{x}_b + \underbrace{\mathbf{A}_{ba} \mathbf{x}_a + \mathbf{B}_b \mathbf{u}}_{\text{bekannte Größen}} . \quad (37)$$

Aus Analogiebetrachtungen zwischen dem vollständigen Entwurf und der reduzierten Form folgt der Beobachter (vgl. [2])

$$\dot{\hat{\mathbf{x}}}_b = \mathbf{A}_{bb} \hat{\mathbf{x}}_b + \mathbf{A}_{ba} \mathbf{x}_a + \mathbf{B}_b \mathbf{u} + \mathbf{L} (\dot{\mathbf{y}} - \mathbf{A}_{aa} \mathbf{x}_a - \mathbf{B}_a \mathbf{u} - \mathbf{A}_{ab} \hat{\mathbf{x}}_b) . \quad (38)$$

Die Gleichung (38) enthält die Ableitung des Ausgangs \mathbf{y} . Um den Beobachter auch realisieren zu können wird ein neuer Zustandsvektor

$$\mathbf{x}_c = \dot{\hat{\mathbf{x}}}_b - \mathbf{L} \mathbf{y} \quad (39)$$

eingeführt. Mit der Erweiterung vom einzelnen linearen System auf die gewichtete Kombination von N_r Systemen, erhält man nach kurzer Umformung den reduzierten Beobachter für T-S Fuzzy Systeme

$$\begin{aligned} \dot{\hat{\mathbf{x}}}_c &= \sum_{i=1}^{N_r} h_i(\mathbf{z}) [(\mathbf{A}_{i_{bb}} - \mathbf{L}_i \mathbf{A}_{i_{ab}}) \hat{\mathbf{x}}_b + (\mathbf{A}_{i_{ba}} - \mathbf{L}_i \mathbf{A}_{i_{aa}}) \mathbf{y} + (\mathbf{B}_{i_b} - \mathbf{L}_i \mathbf{B}_{i_a}) \mathbf{u}] , \\ \hat{\mathbf{x}}_b &= \hat{\mathbf{x}}_c + \sum_{i=1}^{N_r} h_i(\mathbf{z}) \mathbf{L}_i \mathbf{y} . \end{aligned} \quad (40)$$

Angewandt auf den Beobachterentwurf für hydrostatische Antriebe heißt das, dass zur Überwachung vom Pumpenschwenkwinkel für (26) mit $\mathbf{z} = [x_3, x_4, \omega_P]^T$ entweder

$$\mathbf{x}_a = [x_2, x_3, x_4]^T , \quad \mathbf{x}_b = x_1$$

oder

$$\mathbf{x}_a = [x_3, x_4]^T , \quad \mathbf{x}_b = [x_1, x_2]^T$$

gesetzt wird. Im ersten Fall wird nur der Schwenkwinkel der Pumpe und im zweiten Fall auch der Schwenkwinkel vom Hydromotor beobachtet. Dies ist bei den praktischen Anwendungen sinnvoll, wo eine zusätzliche Messung nicht zuverlässig und/oder kostengünstig erfolgen kann. Zur Überwachung des Differenzdrucks Δp im geschlossenen Kreis ist die Messung des Hydromotorschwenkwinkels jedoch zwingend erforderlich, da die Variablen in $\mathbf{z} = [x_2, \omega_P]^T$ mit $x_2 = \tilde{\alpha}_M$ in (28) messbar sein müssen. Andernfalls können die nun folgenden Entwurfsverfahren nicht angewandt werden.

3.2 LMI-basierte Synthese

Der Schätzfehler ist definiert mit $\mathbf{e}_c := \mathbf{x}_c - \hat{\mathbf{x}}_c$. Die Matrizen \mathbf{L}_i für $i = 1, \dots, N_r$ sind dabei so zu wählen, dass der Fehler \mathbf{e}_c für beliebige Anfangszustände \mathbf{x}_0 und $\hat{\mathbf{x}}$ für

$t \rightarrow +\infty$ gegen Null strebt. Drückt man den Schätzfehler e_c mit Hilfe der Gleichungen (38), (39) und (40) aus, so erhält man nach kurzer Umformung bei Berücksichtigung der Normierungsbedingung $\sum_{i=1}^{N_r} h_i(\mathbf{z}) = 1$ für T-S Fuzzy Systeme das Fehlersystem

$$\dot{e}_c(t) = \sum_{i=1}^{N_r} \sum_{j=1}^{N_r} h_i(\mathbf{z}(t)) h_j(\mathbf{z}(t)) (\mathbf{A}_{i_{bb}} - \mathbf{L}_j \mathbf{A}_{i_{ab}}) e_c(t). \quad (41)$$

Zu diesem Fehlersystem kann, abgeleitet aus dem Satz für vollständige Beobachter [13], der folgende Satz angegeben werden

Theorem 1. *Gegeben seien die Matrizen \mathbf{L}_j für $j = 1, \dots, N_r$, das Streckenmodell (27) oder (28) und das System zur Rekonstruktion der Zustände (40). Dann ist für beliebige Anfangszustände $\mathbf{x}_0 \in \mathbb{R}^n$ und $\hat{\mathbf{x}}_0 \in \mathbb{R}^n$ das Fehlersystem (41) global asymptotisch stabil, wenn eine gemeinsame, symmetrisch positiv definite Matrix \mathbf{P} für alle $\mathbf{A}_{i_{bb}}$, $\mathbf{A}_{i_{ab}}$, \mathbf{L}_i und \mathbf{N}_i mit $i = 1, \dots, N_r$ und eine symmetrisch positiv definite Matrix \mathbf{Q} existiert, so dass die Matrixungleichungen*

$$\begin{aligned} \mathbf{A}_{i_{bb}}^T \mathbf{P} - \mathbf{A}_{j_{ab}}^T \mathbf{N}_i^T + \mathbf{P} \mathbf{A}_{i_{bb}} - \mathbf{N}_i \mathbf{A}_{j_{ab}} - 2\mathbf{Q} + 4\alpha \mathbf{P} \\ + \mathbf{A}_{j_{bb}}^T \mathbf{P} - \mathbf{A}_{i_{ab}}^T \mathbf{N}_j^T + \mathbf{P} \mathbf{A}_{j_{bb}} - \mathbf{N}_j \mathbf{A}_{i_{ab}} < 0 \end{aligned} \quad (42)$$

für $i, j = 1, \dots, N_r$ erfüllt sind.

Die Beobachtermatrizen werden aus $\mathbf{L}_i = \mathbf{P}^{-1} \mathbf{N}_i$ bestimmt. Mit dem Designparameter α wird das Abklingverhalten vom Fehlersystem und damit die Beobachterdynamik eingestellt. Die Matrixungleichungen enthalten nur Variablen, die linear in (42) eingehen. Eine Lösung, falls (42) eine Lösung besitzt, kann effizient u.a. mit der Matlab LMI-Toolbox von Gahinet [3] ermittelt werden.

4 Zusammenfassung und Ausblick

Vorgestellt wurde ein Verfahren zur beobachterbasierten Überwachung von Prozessgrößen (Druck und Schwenkwinkel) in hydrostatischen Antrieben. Hierzu wurden systematisch physikalische Modelle abgeleitet und überführt in zwei dynamisch äquivalente T-S Fuzzy Systeme. Ein Freiheitsgrade beim Beobachterentwurf wurde ausgenutzt um notwendige Bedingungen für LMI-basierte Berechnungsverfahren nicht verletzen zu müssen.

Die bisher vorgestellte Modellstruktur zur Überwachung hydrostatischer Antriebe ist nur geeignet für Fahrsituationen in der Ebene und bei Fahrten mit leichter Steigung. Durch die Vernachlässigung zusätzlicher externer Lasten im aktuellen Modell ist die Drucküberwachung in hydrostatischen Antrieben entsprechend eingeschränkt. Um bisher nicht modellierte Zusammenhänge geeignet zur Rekonstruktion der Zustände im Beobachter berücksichtigen zu können, wird das Modell um eine z-Variable zur Gewichtung lastabhängiger lokaler Modelle erweitert. Die neue Variable kann dann extern, fahrsituationsabhängig basierend auf der Methode in [4] bestimmt werden.

Literatur

- [1] Deutscher, J.: Entwurf nichtlinearer Beobachter für autonome Systeme basierend auf dem Hilfssatz von Lyapunov, *at - Automatisierungstechnik*, vol. 55, no 9, pp. 468–476, 2007.
- [2] Franklin, G. F., Powell, J. D. und Emami-Naeini, A., *Feedback Control of Dynamic Systems*, Addison-Wesley Publishing Company, 1991.
- [3] Gahinet, P., Nemirovski, A., Laub, A. J. und Chilali, M. : *LMI Control Toolbox*, The Mathworks Inc., 1995.
- [4] Gerland, P., Schulte, H., Kroll, A.: Wahrscheinlichkeitsbasierte globale Situationserkennung bei komplexen Systemen, In: *Proc. 18. Workshop Computational Intelligence*, Universitätsverlag Karlsruhe, 2008.
- [5] Lee, K. R., Jeung, E. T., Park, H. B., Robust fuzzy H-infinity control for uncertain nonlinear systems via state feedback: An LMI approach, *Fuzzy Sets and Systems*, vol. 120, pp. 123–134, 2001.
- [6] Martinus, M. A.: *Funktionale Sicherheit von mechatronischen Systemen bei mobilen Arbeitsmaschinen*, Dissertation, Technische Universität München, Institut für Maschinentechnik - Lehrstuhl für Fahrzeugtechnik, 2004.
- [7] Messoussi, W. E., Pagès, O. und Hajjaji, A. E.: Observer-Based Robust Control of Uncertain Fuzzy Dynamic Systems with Pole Placement Constraints: An LMI Approach, In: *Proc. of the 2006 American Control Conference*, 14–16 June 2006, Minneapolis, Minnesota, USA, pp. 2203–2208.
- [8] Nijmeijer, H. und Fossen, T.I.: New directions in nonlinear observer design, In: *Lecture notes in control and information science*, vol. 244, Springer-Verlag, New York, 1999.
- [9] Schaffner, J. und Zeitz, M.: Entwurf nichtlinearer Beobachter, In: S. Engell (Editor), *Entwurf nichtlinearer Regelungen*, pp. 53–76 , Oldenbourg, 1995.
- [10] Schulte, H.: Modellierung und Simulation hydrostatischer Getriebe in mobilen Arbeitsmaschinen mittels Takagi-Sugeno-Fuzzy-Systemen, In: *Proc. 16. Workshop Computational Intelligence*, Universitätsverlag Karlsruhe, 2006, pp. 62-70.
- [11] Schulte, H.: Control-Oriented Modeling of Hydrostatic Transmissions using Takagi-Sugeno Fuzzy Systems, In: *IEEE International Conference on Fuzzy Systems*, 23-26 July 2007, London, UK, pp. 2030–2035.
- [12] Takagi, T. und Sugeno, M.: Fuzzy identification of systems and its application to modelling in control, *IEEE Transactions on System, Man and Cybernetics*, vol. 15, no. 1, pp. 116–132, 1985.
- [13] Tanaka, K. und Wang, H. O.: *Fuzzy Control Systems Design and Analysis: A linear matrix inequality approach*, John Wiley & Sons, New York, Chichester, 2001.

- [14] Yoneyama, J., Nishikawa, M., Katayama, H., Ichikawa A.: Output stabilization of takagi-sugeno fuzzy systems, *Fuzzy Sets and Systems*, vol. 111, no. 2, pp. 253–266, 2000.
- [15] Norm DIN EN 61508: *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbar elektronischer Systeme*. Norm DIN EN 61508. Berlin Beuth Verlag, 2001.

Disrupted Learning — Lernen bei harten Zustands- oder Strukturwechseln

Nils Rosemann, Jens Hülsmann und Werner Brockmann

Universität Osnabrück

Albrechtstraße 28

49069 Osnabrück

Tel.: (0541) 969 2598

E-Mail: Werner.Brockmann@uni-osnabrueck.de

Zusammenfassung

Bei Regelungssystemen muss vielfach ein Kompromiss gefunden werden zwischen dem Entwurfsaufwand und der Güte einer Regelung. Bei manchen Prozessen müssen dabei verschiedene Prozessphasen unterschieden und für jede Phase ein eigener Regler entworfen werden. Vielfach reicht aber die Aufteilung in Phasen noch nicht aus, um den Entwurf der Regler wirklich handhabbar zu gestalten. Bestimmte Prozessparameter können zur Entwurfszeit noch unbekannt sein oder sich während des Betriebs ändern. Das System kann zu starke Nichtlinearitäten aufweisen oder nicht vorhersehbaren Störungen, Fehlern oder Alterung unterworfen sein. In diesen Fällen bietet es sich an, die einzelnen Regler selbst-optimierend zu gestalten, so dass sie sich durch inkrementelles Online-Lernen laufend auf Parametervariationen und Störungen einstellen. Je nach Prozessdynamik und Eigenschaften dieser Regler beeinflusst das Umschalten sowohl die Robustheit als auch die Konvergenz des Lernverhaltens der Regler und somit des Gesamtsystems. Im *worst case* konvergiert das Lernen wegen des Umschaltens nicht bzw. die einzelnen Regler sind nur auf einem kleinen Teil des Eingangsraums robust. Diese Probleme ergeben sich insbesondere dadurch, dass die lernfähigen Systeme in ihrem jeweiligen Lernvorgang unterbrochen werden (müssen), sobald auf eine andere Prozessphase umgeschaltet wird. Wir sprechen daher von *disrupted* bzw. *interrupted learning*. Um diesen Problemen zu begegnen, wird in der vorliegenden Arbeit der ELISE-Ansatz vorgestellt und an einem Anwendungsszenario untersucht.

1 Einleitung

1.1 Motivation

Bei modernen Automatisierungssystemen ist die praktische Handhabbarkeit des Entwurfs oft ein großes Problem. Dies trifft vor allem auf die Regelungssysteme für nichtlineare Systeme mit einer komplexen Dynamik zu. Bei diesen muss in der Praxis oft ein Kompromiss aus Entwurfsaufwand und Regelungsgüte gefunden werden.

Es gibt viele Einflüsse, die das Verhältnis von erreichter Regelgüte und Entwurfsaufwand ungünstig beeinflussen können. Die wichtigsten sind die Eigenschaften des zu regelnden Systems, wie Nichtlinearitäten, ein ungenaues bzw. fehlendes Systemmodell, unbekannte Systemparameter und vor allem Zeitvarianz. Zeitvariant können Systemparameter sein, aber auch strukturelle Zusammenhänge und Umgebungsbedingungen oder wechselnde Betriebsmodi.

Beim Reglerentwurf gilt es, unter Berücksichtigung von Randbedingungen (Entwurfskosten, minimale Regelungsgüte, ...) diejenige Entwurfsmethode zu wählen, die das beste Verhältnis von Regelgüte zum Entwurfsaufwand hat. Es gibt zwei Kriterien, nach denen sich die meisten Methoden einordnen lassen. Das erste dieser Kriterien betrifft den *Modellbezug* der Entwurfsmethode. Die beiden Eckpfeiler sind Methoden, die ein exaktes Modell voraussetzen und ausnutzen [1], und Methoden, die gar kein Modell benötigen, indem sie z. B. Methoden der *Computational Intelligence* nutzen [2]. Das zweite der beiden Kriterien betrifft den Umgang mit zur Entwurfszeit *unbekannten Systemeigenschaften* (z. B. Zeitvarianz). Dabei sind die Eckpfeiler zum einen die Methoden, die Abschätzungen der unbekannt Einflüsse zur Entwurfszeit einfließen lassen (statische Robustheit), und zum anderen Methoden, die sich im Betrieb an vorher unbekannte oder sich ändernde Einflüsse anpassen, also adaptiv sind [3].

Grundlage dieser Arbeit ist eine Entwurfsmethodik, die kein formales Modell benötigt und sich durch lernfähige Algorithmen im Betrieb adaptieren kann. Speziell geht es um Regelstrecken, bei denen sich unterschiedliche Betriebsmodi bzw. -phasen unterscheiden lassen. Es kann also wechselnde Strukturen geben, wechselnde Regelziele oder variierende Parametersätze. Eine solche Betriebsphase (nachfolgend einfach nur *Phase* genannt) ist also ein Bereich des Zustandsraumes, für den eine bestimmte Regelstrategie anzuwenden ist. Diese Strategie kann sich in manchen Anwendungen aus der physikalischen Struktur ergeben, oftmals ist sie aber durch unterschiedliche Anforderungen an das Regelverhalten, durch unterschiedliche Zielgrößen oder andere funktionale Anforderungen bedingt. Diese Phasen folgen also nicht aus einer formalen Beschreibung, sondern aus der gewünschten Funktionalität des Systems. Aufgrund der Aufteilung in Phasen ist es bei solchen Regelstrecken besser oder sogar notwendig, für jede Phase einen einzelnen Regler bzw. eine Reglerstruktur zu entwerfen, anstatt einen monolithischen Regler für das Gesamtsystem.

Bezüglich des zweiten Kriteriums, also dem Umgang mit unbekannt Systemeigenschaften, setzt die Entwurfsmethodik auf Online-Adaption. Insgesamt ist das Vorgehen derart, dass für die Phasen unabhängige, (bei Bedarf) adaptive Einzelregler entworfen werden. Dazu gibt es zwei Möglichkeiten. Bei der *indirekten* Adaption werden im Betrieb lokale Modelle für die einzelnen Phasen gelernt. Aus diesen lokalen Modellen werden dann lokal gültige Regler abgeleitet. Der Lernerfolg hängt bei diesem Vorgehen stark von der Genauigkeit der Modelle ab [4, 5, 6]. Im Gegensatz dazu wird bei der *direkten* oder auch verhaltensbasierten Adaption aus dem beobachtbaren Systemverhalten eine unmittelbare Reglerveränderung abgeleitet [7]. Falls man für die einzelnen Phasen lineare Regler nutzt, so spricht man von *gain scheduling* [8, 9]. Das gleiche Prinzip findet sich generalisiert in Fuzzy-Reglern, die auf Takagi-Sugeno-Systemen basieren [10].

Für beide Adaptionarten stellt sich die Frage, ob im Übergangsbereich gleichzeitig mehrere Regler aktiv sein können, indem zwischen den Einzelreglern interpoliert wird, oder ob man hart umschaltet. In der Literatur findet man häufiger den interpolierenden Ansatz [5, 11], da man dadurch ein glatteres Verhalten erreicht und Unstetigkeiten vermeidet. Wie oben erwähnt ist es aber bei unterschiedlichen Regelstrategien üblich, dass Reglerstrukturen mit verschiedenen Teilreglern eingesetzt werden, um je nach Phase ein gewünschtes Ziel an Regelgüte und Stabilität zu erreichen. Um diese reglerspezifischen Eigenschaften nicht zu verlieren, konzentriert sich diese Arbeit daher auf Fälle, in denen *hart* zwischen den einzelnen Reglerstrukturen umgeschaltet wird, und auf Methoden, um die dadurch auftretenden Probleme zu vermeiden. Ein übergeordneter Entscheider bestimmt dabei, wann und wohin umgeschaltet wird. Dadurch lässt sich Hysterese oder

anderes komplexeres Umschaltverhalten erzielen.

Ein solches hartes Umschalten ist sowohl für direkte als auch für indirekte Adaption problematisch. Denn zum einen ist das Verhalten in den Phasengrenzgebieten in der Regel transient, sie werden also schnell wieder verlassen. Entsprechend können nur wenige Lernvorgaben generiert werden. Die Konvergenz des Lernens ist sehr langsam oder überhaupt nicht gegeben. Darüber hinaus werden bei zu kurzem Verweilen in einer Phase keine oder nur wenige sinnvollen Lernvorgaben generiert. Zum anderen bedeutet das Umschalten eine neue, zusätzliche Dynamik, die mit drastischen Änderungen der Systemeigenschaften einhergehen kann. Es wechselt also auch das Lernproblem in signifikanter Weise. So kann es zu unerwünschten bzw. erfolglosen Phasenwechseln kommen. Diese Arbeit geht nur auf direkte Adaption ein, so dass keine Annahmen über ein formales Modell oder funktionale Eigenschaften eines formalen Modelles nötig sind.

Das harte Umschalten hat daher Auswirkungen auf das Lernen bzw. Adaptieren der Einzelregler. Durch dieses unterbrochene Lernen (*disrupted* bzw. *interrupted learning*) dauert der Adaptionsvorgang sehr lange bzw. führt zu suboptimalen Resultaten. Damit die Stabilität und der Optimierungserfolg nicht gefährdet werden, müssen Lernvorgänge an den Phasengrenzen möglichst effektiv ausgenutzt werden. Genau das ist das Ziel des im nächsten Kapitel vorgestellten ELISE¹-Ansatzes.

1.2 Randbedingungen für einzelne adaptive Regler

Wie oben beschrieben zeichnen sich die einzelnen Regler dadurch aus, dass sie *lernen* können, potentiell nichtlineare, zeitvariante Systeme ohne ein formales Modell in einem gewissen Bereich des Eingangsraums erfolgreich zu regeln. Sie müssen also beliebige nichtlineare Funktionen darstellen können. Damit der Entwurf solcher Regler aber beherrschbar bleibt, ist es vorteilhaft, wenn zum einen vorhandenes Expertenwissen zum Betriebsbeginn eingesetzt und zum anderen das im Betrieb optimierte bzw. gelernte Wissen jederzeit *interpretiert* werden kann.

Formal handelt es sich bei dem Lernvorgang um eine Funktionsapproximation. Es können also beliebige Funktionsapproximatoren eingesetzt werden, die inkrementell lernen können und für die es sowohl für jeden Lernvorgang als auch für jede Auswertung eine obere Zeitschranke gibt (Echtzeitfähigkeit). Für den Einsatz in eingebetteten Systemen muss der Bedarf an Rechenleistung und Speicher gering sein. Vorteilhaft für die Beherrschbarkeit des Entwurfs ist zudem, wenn es sich um einen lokalen Funktionsapproximator handelt, so dass jeder Lernvorgang nur für die Situationen wirkt, für die die neuen Lerndaten vorhanden sind.

Zusätzlich zu diesen technischen Randbedingungen ist natürlich gefordert, dass die eigentliche Adaption so schnell und genau wie möglich geschieht. Denn je länger sie dauert, desto länger wird das System suboptimal und evtl. kritisch betrieben. Gleichzeitig muss die Adaption aber auch so robust wie möglich sein. Verrauschte Daten, Ausfälle und Fehler dürfen den Lernvorgang und die Stabilität nicht gefährden. Diese Anforderungen sind besonders dadurch schwer zu erfüllen, dass die Adaption bzw. das Lernen einen zusätzlichen Wirkungskreis ins System einführt. Denn das, was in einer Situation gelernt wird, verändert das zukünftige Systemverhalten, und damit auch das, was in der Zukunft

¹Exploiting Learning stimuli in Interrupted SElfoptimization

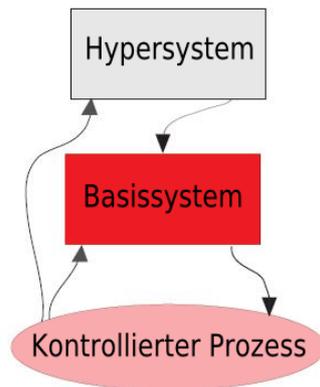


Bild 1: Aufbau einer adaptiven Regelung mit direkter Adaption nach dem DSL-Schema

gelernt werden wird. Durch diesen Wirkungskreis ist das System potentiell chaotisch, es können also kleine Änderungen mit der Zeit beliebig groß werden. Entsprechend kann sich selbst bei sehr ähnlichen Startbedingungen ein völlig unterschiedliches Regelverhalten ausprägen.

Direkte Adaption durch dirigiertes Selbstlernen (DSL) bedeutet nun, dass der aktuelle Zustand des Systems als Kriterium für die Bewertung und eventuelle Korrekturen vergangener Stellaktionen genutzt wird. Die eigentliche Korrektur wird dann aber für einen vergangenen Systemzustand vorgenommen [12]. Dabei wird immer eine feste Zeit zurückgegriffen (Rückgriffszeit). Dadurch wird berücksichtigt, dass es eine gewisse Zeit dauert, bis sich Stellaktionen signifikant auf den Prozess auswirken. Abbildung 1 zeigt die Struktur der direkten Adaption mit DSL in schematischer Form. Das gezeigte Basissystem kapselt die Systemfunktionalität und wird von dem Hypersystem durch Lernvorgaben optimiert.

2 Der ELISE-Ansatz für direkte Adaption

Bei komplexeren Regelungen aus mehreren Einzelreglern R_i , von denen einer oder mehrere adaptiv sind, treten an Phasengrenzen und -übergängen die oben beschriebenen Probleme auf. Das erste Problem besteht darin, dass Lernvorgänge an den Phasengrenzen selten, aber kritisch sind. Dies liegt einerseits daran, dass das System nur selten an den Phasengrenzen betrieben wird. Andererseits liegt es an der Verzögerungszeit von Stellaktionen. Denn wenn auf einen adaptiven Einzelregler umgeschaltet wurde, können so lange keine Korrekturen an ihm vorgenommen werden, bis sich seine Stellaktionen auswirken und diese bewertet werden können. Ähnliches gilt für den Einzelregler, der zuvor aktiv war. Die Verzögerungszeiten im System erzeugen also einen zeitlichen Bereich um die Phasengrenze, in dem Lernvorgänge problematisch sind. Das zweite Problem besteht darin, dass die Phasenübergänge eine zusätzliche Dynamik erzeugen können. Dies liegt daran, dass sich die Regelstrategie und damit das Verhalten abrupt ändern. Selbst wenn dies insgesamt die Stabilität und Robustheit des Gesamtsystems nicht gefährdet, wirkt es sich in der Regel nachteilig auf das Konvergenzverhalten des Lernens aus, und zwar hinsichtlich Konvergenzgeschwindigkeit und -güte.

Beide Probleme geht der ELISE-Ansatz dadurch an, dass er die bisher aufgezeigte Systemstruktur an zwei Stellen modifiziert. Aus dem ersten Problem ergibt sich die Anforderung, gerade auch in der Nähe der Phasengrenzen, also in der Nähe der Umschaltgrenzen, Lernstimuli möglichst effektiv auszunutzen. Der erste Baustein des ELISE-Ansatzes ist eine Verlängerung des Lernvorgangs über den Umschaltzeitpunkt hinaus mit einem *graduellen* Übergang des Lernens von dem einen adaptiven Einzelregler zum anderen. Obwohl also die eigentlichen Regler hart umgeschaltet werden, gilt dies nicht mehr für das Lernen. Wenn von einem Regler R_i auf einen Regler R_j umgeschaltet wird, dann lernt R_i nach dem Umschalten für einen Zeitraum noch weiter. Dieser Zeitraum muss auf die relevanten Systemeigenschaften zugeschnitten sein, also vor allem auf die Latenzzeiten des Prozesses und die Rückgriffszeiten der DSL-Adaptionen der einzelnen Regler. Für diese zusätzlichen Lernvorgänge wird allerdings die *Lernrate* von R_i mit einem Faktor multipliziert, der von Eins nach und nach auf Null reduziert wird (z. B. linear). Dadurch wird berücksichtigt, dass die letzten Stellaktionen von R_i *graduell* weniger Einfluss haben, je länger R_j aktiv ist. In umgekehrter Weise wird das Lernen von R_j nicht erst aktiviert, wenn die zugehörige DSL-Rückgriffszeit vorbei ist. Statt dessen wird direkt nach dem Umschalten gelernt, allerdings wird die Lernrate mit einem Faktor multipliziert, der sich (von Null aus) *graduell* erhöht. Insgesamt wird der alte Lernvorgang also aus- und der neue eingeblendet.

Dieser Mechanismus basiert auf der Idee, dass die kurz vor dem Umschalten ausgeführten Stellaktionen von R_i erst nach dem Umschalten eine signifikante Wirkung entfalten. Sie können also auch erst dann bewertet und eventuell korrigiert werden. Dennoch beeinflussen sie das Systemverhalten, obwohl schon R_j aktiv ist. Auf diese Weise beeinflussen sich die beiden Einzelregler, zwischen denen umgeschaltet wird. Und je nach Prozesseigenschaften und je nachdem, wie weit fortgeschritten die bisherigen Adaption der Regler sind, beeinflussen sich die Einzelregler unterschiedlich stark. Um dies zu berücksichtigen, gibt es für das Überblenden des Lernens unterschiedliche Varianten. Zum einen kann man unterschiedliche Zeiträume für das Ein- und Ausblenden verwenden, um den Eigenschaften der beteiligten Phasen Rechnung zu tragen. Desweiterern kann man die beteiligten Lernvorgänge linear, exponentiell oder stufenförmig ein- bzw. ausblenden. Und schließlich ist zu klären, ob für den jeweiligen Prozess sich die Faktoren für die Ein- und die Ausblendung jeder Zeit zu Eins addieren sollten oder eben nicht.

Um zusätzliche, durch das Lernen induzierte Phasenübergänge zu erkennen, ist der zweite Baustein des ELISE-Ansatzes, dass der zeitliche Verlauf der Phasenübergänge in einem *Phasen-FIFO-Speicher* abgelegt wird. Dabei können auch problemspezifische Zusatzinformationen abgelegt werden, beispielsweise über die Kriterien, die zum Umschalten führten. In einem solchen FIFO-Speicher wird die Umschaltdynamik des Systems beobachtet, beispielsweise, dass das System zwischen zwei Phasen sehr schnell oszilliert, das System sich also sehr oft in dem Umschaltbereich aufhält. Dieser Bereich sollte eigentlich transient sein. Eine Oszillation deutet demnach auf ein ungewünschtes Verhalten und suboptimales Lernen hin. Auf solche Beobachtungen kann dann mit vorher festgelegtem Expertenwissen reagiert werden, das beispielsweise zusätzliche Lernvorgaben für die beteiligten Einzelregler generiert.

Ein wichtiger, charakteristischer Fall ist gegeben, wenn ein Einzelregler R_{vor} den Prozess in gewisser Weise vorbereitet, damit dieser nach dem Umschalten dauerhaft von einem zweiten Einzelregler R_{nach} geregelt wird (z. B. Verlassen der Haftreibung). Wenn sich nun im Phasen-FIFO zeigt, dass von R_{vor} auf R_{nach} und praktisch direkt wieder zurück

geschaltet wird, dann ist die Vorbereitung nicht erfolgreich gewesen. Es ist dann sinnvoll, aus diesem erfolglosen Phasenwechsel gezielt zusätzliche Lernvorgaben zu generieren. Eine solche Heuristik muss natürlich an den Prozess angepasst werden, ist aber dennoch ein Beispiel für eine allgemeine Herangehensweise für bestimmte Phasenkonstellationen. Im folgenden Kapitel wird eine exemplarische Umsetzung des ELISE-Ansatzes beschrieben. Sie dient vor allem dazu, die Anwendung und die Praxistauglichkeit an einem realen, nichtlinearen Prozess zu zeigen. Zusätzlich ist es aber auch eine Demonstration, wie das Expertenwissen aussehen kann, das im ELISE-Ansatz eine Rolle spielt, insbesondere auch für die Heuristik bei zu häufigen Phasenwechseln.

3 Untersuchungen

3.1 Aufbau

Bei dem Anwendungsbeispiel handelt es sich um einen Wagen mit aufmontiertem Pendel (Pendelwagen). Dieser kann sich vor und zurück bewegen (entlang der Pendelebene) und damit das Pendel in Schwingung versetzen. Das Ziel dabei ist es zu lernen, das Pendel so weit aufzuschwingen, dass es über Kopf steht, und es dann in dieser Stellung zu balancieren. Dabei muss der Wagen selbst möglichst nah bei einer vorgegebenen Zielposition bleiben.

Der real eingesetzte Pendelwagen (Abbildung 2) ist schwer zu modellieren, da Dreckeffekte wie starkes Getriebeispiel, Reibung, Wegrutschen der Räder und Rauschen auftreten. Es sind wesentliche Prozesseigenschaften zur Entwurfszeit unbekannt, wie z. B. schwer bestimmbare, aber im Prinzip konstante Effekte (z. B. Auswirkung des Getriebeispiels), aber auch für situations- bzw. umgebungsabhängige Effekte wie Haftung der Räder auf dem Untergrund. Bei diesem Pendelwagen misst ein Sensor den Winkel des Pendels, während der andere zur Positionsbestimmung dient. Die zu entwickelnde Regelung hat den Pendelwinkel und die Position als Messgrößen und die Motorspannung als Stellgröße. Im hier beschriebenen Fall soll die komplette Regelung gelernt werden. Dabei sind zwei Betriebsphasen zu unterscheiden, und zwar das Aufschwingen des Pendels und das Balancieren, zwischen denen ein Entscheider hart umschaltet. Die beiden Betriebsphasen werden in diesem Beispiel durch zwei Teilregler mit völlig unterschiedlichen Regelzielen betrieben. Entsprechend unterscheiden sich auch die Lernstrategien. Der Balancierregler setzt kontinuierliches, lokales Lernen ein, während der Aufschwingregler mit unregelmäßigen, globalen Lernvorgaben trainiert wird.

In Abb. 3 ist das Konzept der Pendelwagenregelung schematisch dargestellt. Die Erweiterung durch den ELISE-Ansatz wird weiter unten erläutert. Ein Entscheider wählt jeweils einen der beiden adaptiven Regler anhand der momentanen Situation des Pendelwagens aus. Die Umschaltkriterien des Entscheiders legen den aktiven Regler auf der Basis von Winkel und Winkelgeschwindigkeit des Pendels mit einer Hysterese fest. Ist der Winkel φ und die Winkelgeschwindigkeit $\dot{\varphi}$ kleiner als $(\varphi_{ein}, \dot{\varphi}_{ein})$, wird die Kontrolle an den Balancierregler übergeben. Überschreitet der Winkel den Schwellwert φ_{aus} , übernimmt wieder der Aufschwingregler. Die Parameter haben Größenordnungen von einigen Grad bzw. einigen zehn Grad pro Sekunde.

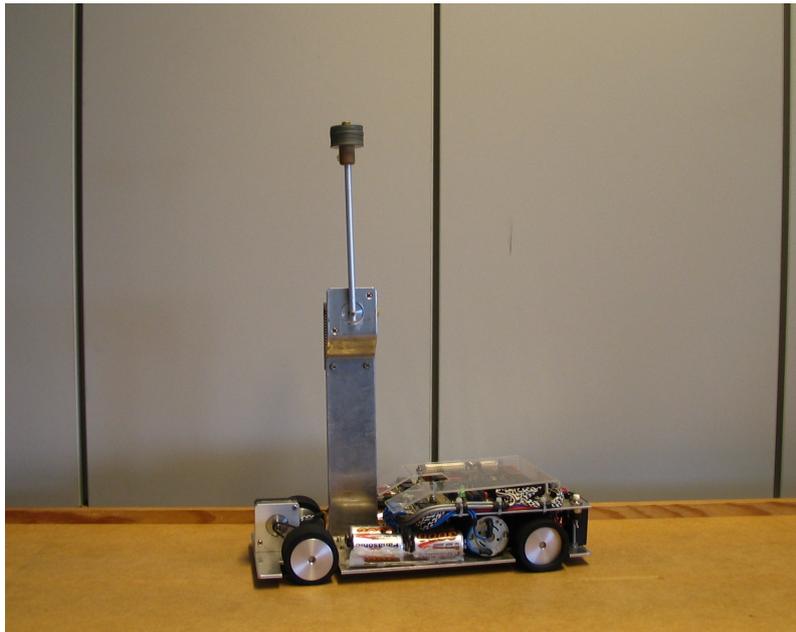


Bild 2: Verwendeter Pendelwagen mit balanciertem Pendel

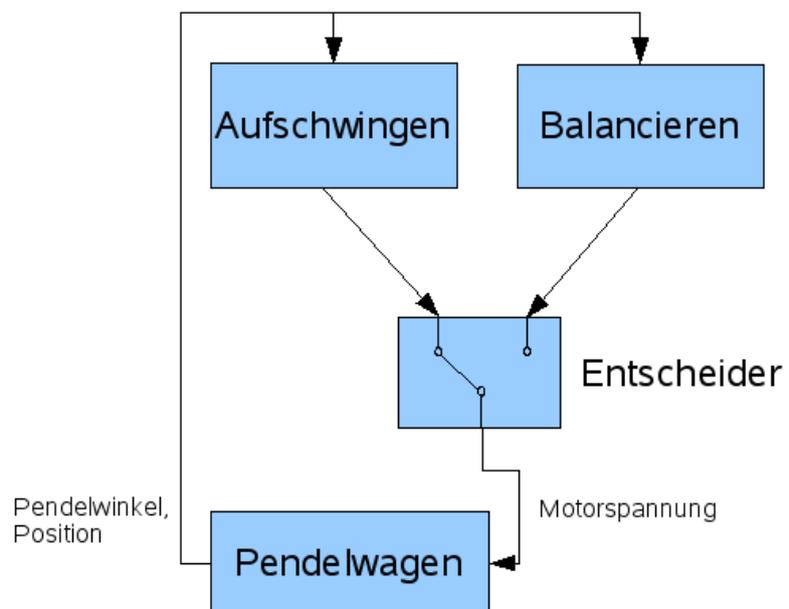


Bild 3: Funktionaler Aufbau der Pendelwagenregelung

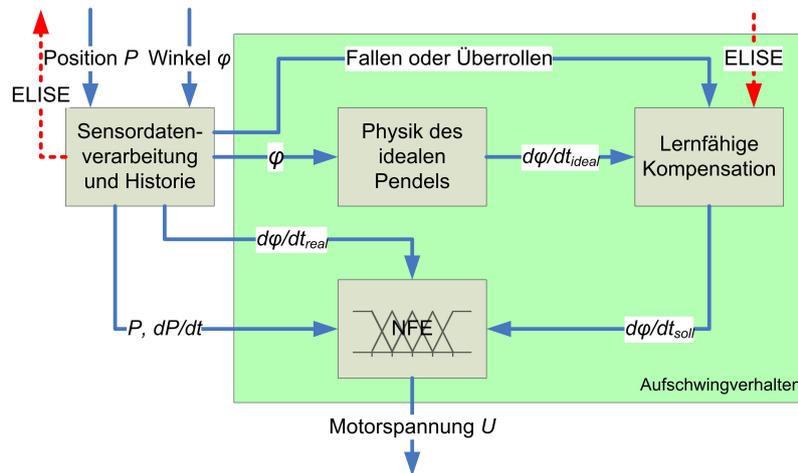


Bild 4: Aufbau des Aufschwingreglers

3.2 Aufschwingen des Pendels

In Abb. 4 ist die Funktionsweise des Aufschwingreglers dargestellt. Dieser Regler basiert auf der Physik des idealen Pendels. Er regelt die Geschwindigkeit des Pendels. Um festzustellen, ob mehr oder weniger Schwung auf das Pendel gegeben werden muss, wird die aktuelle Winkelgeschwindigkeit mit der Geschwindigkeit verglichen, die benötigt würde, damit das Pendel im oberen Punkt stehen bleibt. Hierzu verwendet man zunächst die Phasenraumrelation des idealen starren Pendels:

$$\dot{\varphi}_{ideal} = \frac{360^\circ \cdot \sqrt{g \cdot l \cdot \frac{\cos(\varphi)+1}{2}}}{\pi \cdot l}$$

Mit der Erdbeschleunigung $g = 9,81 \text{ m s}^{-2}$ und l als Länge des Pendels.

Da hierbei Reibungseffekte und andere Nichtlinearitäten nicht berücksichtigt werden, wird die Sollgeschwindigkeit $\dot{\varphi}_{ideal}$ mit einem veränderlichen Faktor f multipliziert. Dieser Faktor wird wie folgt durch den Lernalgorithmus angepasst: Erreicht das Pendel den oberen Winkelbereich von $\pm \varphi_{ein}$ nicht, wird er erhöht; rollt das Pendel über den oberen Winkelbereich mit einer Geschwindigkeit größer als $\dot{\varphi}_{ein}$, wird er verringert.

Nun werden in dem Fuzzy-System NFE die reale Winkelgeschwindigkeit $\dot{\varphi}_{real}$ und die Sollwinkelgeschwindigkeit $\dot{\varphi}_{soll} = f \cdot \dot{\varphi}_{ideal}$ verglichen. Ist die reale Winkelgeschwindigkeit zu niedrig, wird der Pendelwagen beschleunigt. Damit der Pendelwagen eine vorgegebene Fahrstrecke nicht verlässt, wird im Fuzzy-System zusätzlich die Positionsabweichung des Wagens berücksichtigt. Das Hin- und Herfahren des Wagens geschieht somit nur in der Nähe der Zielposition. Bis auf diese Einschränkung versucht der Regler also, die kinetische Energie im Pendel auf den Wert zu regeln, mit dem es mit einer Geschwindigkeit von null Grad pro Sekunde im oberen Punkt ankommt. Diese Energie wird durch den Lernalgorithmus über den Faktor f eingestellt.

3.3 Balancieren des Pendels

In Abb. 5 ist die Funktionsweise des Balancierreglers dargestellt (vgl. dazu [13]). Bei diesem Regler handelt es sich um eine Kaskadenstruktur aus einem Positions- und einem Winkelregler. Der Positionsregler ist als PD-Regler realisiert und generiert einen

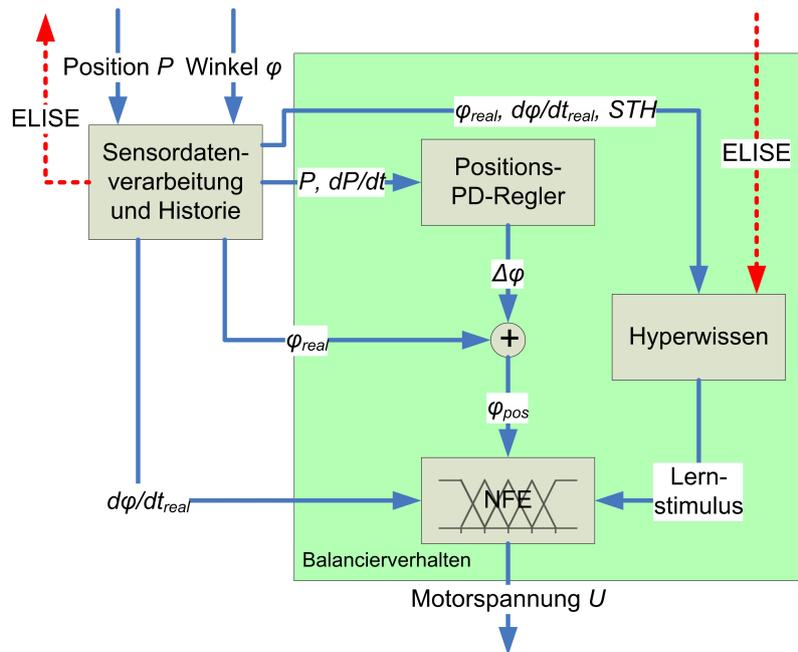


Bild 5: Aufbau des Balancierreglers

Sollwinkel für den nachgeschalteten Winkelregler. Dieser ist als lernfähiger adaptiver Fuzzy-Regler realisiert. Das Regelverhalten wird also durch inkrementelle, lokale Funktionsapproximation gelernt. Das zugehörige Takagi-Sugeno-Fuzzy-System lernt ohne Initialwissen den Zusammenhang zwischen dem Tupel $(\varphi, \dot{\varphi})$ und der anzulegenden Motorspannung U . Hierzu nutzt man entsprechend der DSL-Lernstruktur Hyperwissen aus, das die Reaktion des Systems auf Stellaktionen in der Vergangenheit beurteilt. Die Strategie dahinter ist, dass auf große positive Winkelfehler und positive Winkelableitungen mit hoher negativer Beschleunigung reagiert wird, während auf große positive Winkelfehler und negative Winkelableitung nur wenig reagiert werden muss (Symmetrie entsprechend für negative Winkelfehler). Wenn nun eine Stellaktion in der Vergangenheit den Winkelfehler und die Winkelgeschwindigkeit nicht auf Null regeln konnte, dann muss beim nächsten Mal entsprechend der Strategie stärker oder weniger stark reagiert werden. Die Rückgriffszeit τ für diesen Lernalgorithmus beträgt hier 15 ms.

3.4 Umsetzung des ELISE-Ansatzes

Die beiden Bausteine des ELISE-Ansatzes werden in diesem Szenario wie folgt eingesetzt. Das Überblenden der Lernvorgänge geschieht nur für das Umschalten vom Balancierregler in den Aufschwingregler. Diese Vereinfachung ist hier sinnvoll, da nur der Balancierregler lokal lernt. Bei einem solchen Umschaltvorgang wird das Lernen des Balancierreglers also nicht abgeschaltet. Statt dessen lernt er für 15 ms weiter. Dabei wird aber die Lernrate linear auf Null reduziert. Auf diese Weise kann der Balancierregler in einem Bereich sinnvolles Verhalten ausprägen, in dem dies sonst nur sehr langsam oder gar nicht möglich wäre.

Der zweite Baustein ist ein Phasen-FIFO mit einer entsprechenden Heuristik. Dieser FIFO speichert hier, welche Phase bzw. welcher Regler in den letzten Abtastungen aktiv war, und von welcher Seite des Pendelwagens aus das Umschalten erfolgte. Da der Auf-

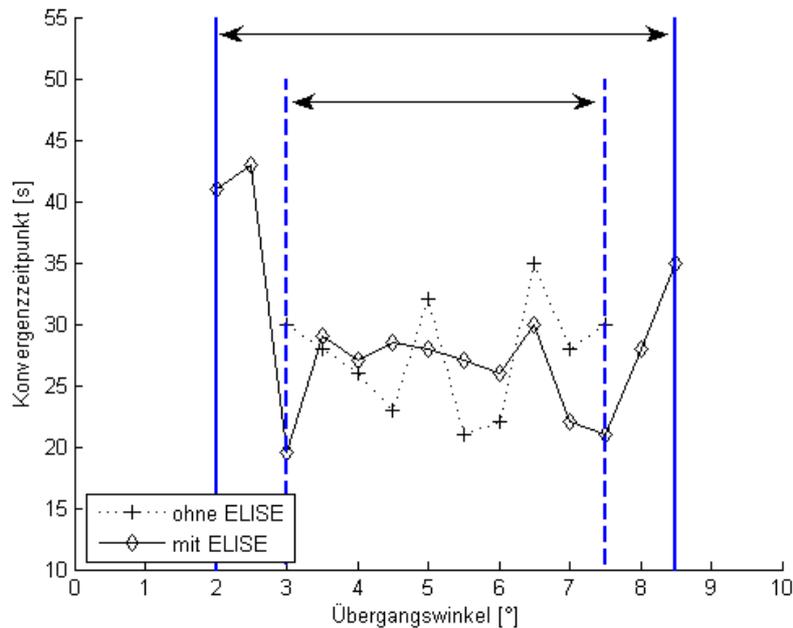


Bild 6: Wenn man den Winkel variiert, bei dem vom Aufschwing- zum Balancierregler umgeschaltet wird (hier Übergangswinkel genannt), dann bestimmt das den Zeitpunkt, ab dem das Pendel zum ersten Mal stabil balanciert werden kann (hier Konvergenzzeitpunkt genannt). Man erkennt, dass es ein Intervall gibt, in dem das System innerhalb der Messzeit konvergiert. Dieses Intervall vergrößert sich durch den ELISE-Ansatz.

schwingregler dazu dient, den Prozess auf den Betrieb des Balancierreglers vorzubereiten, findet die oben erwähnte Heuristik für eine solche Phasenkonstellation Anwendung. Wird also erkannt, dass der Balancierregler aktiviert und direkt wieder deaktiviert wird, so war die Vorbereitung durch den Aufschwingregler nicht gut genug. Es werden daher zusätzliche Lernvorgaben generiert. Dabei wird berücksichtigt, ob das Pendel auf die Seite kippt, von der es aufgeschwungen wurde, oder ob es überschwingt und auf die andere Seite kippt. Ersteres sorgt für den Lernstimulus, beim nächsten Mal für mehr Schwung zu sorgen, letzteres für den Stimulus, beim nächsten Mal für weniger Schwung zu sorgen.

3.5 Untersuchungsszenario

Um die Tauglichkeit der beschriebenen Erweiterungen zu überprüfen wurden zwei Situationen am folgenden Szenario untersucht. Hierzu fährt der Pendelwagen auf einer Strecke von ca. einem Meter hin und her und versucht das Pendel zu balancieren. Die maximale Zeit bis zum stabilen Balancieren des Pendels wurde auf 50 Sekunden festgelegt, da erfahrungsgemäß auch mit längeren Fahrten keine Konvergenz der Regler mehr erreicht werden kann.

Bei der ersten Untersuchung wird ermittelt, wie lange die beschriebene Reglerstruktur braucht, um das Pendel stabil zu balancieren. Dazu wird ein fester Wert für das Übergangskriterium φ_{ein} eingestellt, festgelegt, ob der ELISE-Ansatz in dieser Messung verwendet wird oder nicht und das Lernen gestartet. Wenn das Pendel innerhalb von 50 Sekunden erfolgreich balanciert wird, dann zählt das als erfolgreiches Lernen, und der Zeitpunkt wird festgehalten. Das gelernte Wissen wird dann zurückgesetzt, ein neuer Wert für

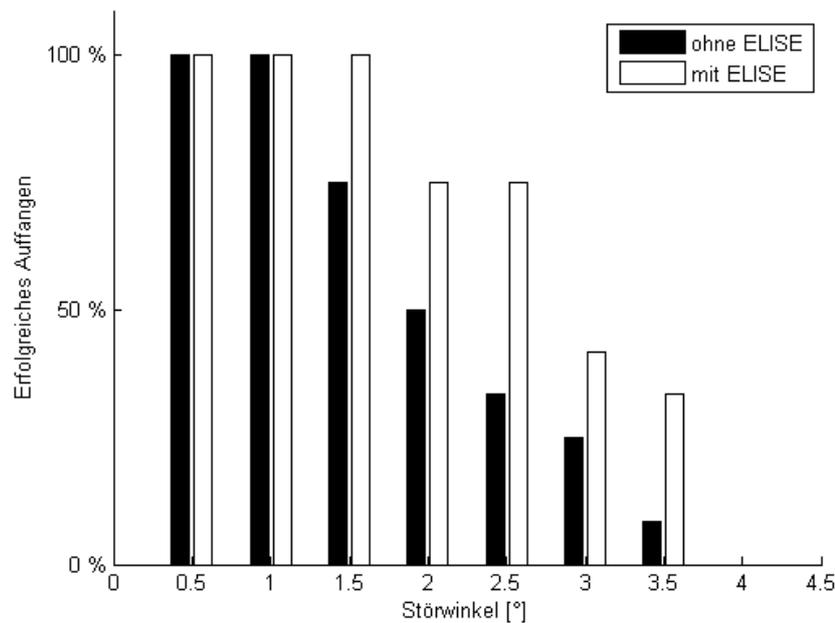


Bild 7: Messung der Erfolgsrate des Balancierreglers bei der Aufgabe, das Pendel nach einem künstlich induzierten Kippen um den Störwinkel wieder aufzufangen

das Übergangskriterium wird eingestellt, und das Vorgehen wiederholt sich. Abbildung 6 zeigt die Ergebnisse, also die Abhängigkeit des Konvergenzzeitpunktes (erstes stabiles Balancieren) vom Übergangswinkel (dem gewählten Kriterium). Dabei handelt es sich um Mittelwerte, da für jeden Wert von φ_{ein} die Messung fünf Mal durchgeführt wurde. Die Ergebnisse zeigen, dass für einen weiten Bereich von Übergangswinkeln die Konvergenzzeit nahezu konstant ist. Sie beträgt ca. 27 Sekunden (verrauscht). Unterhalb von 3 Grad und überhalb von 7.5 Grad konvergiert das Lernen ohne den ELISE-Ansatz allerdings nicht (innerhalb der Messzeit). Dieses Konvergenzintervall vergrößert sich um 1 Grad zu jeder Seite, wenn man den ELISE-Ansatz verwendet. In diesem Vergrößerungsbereich lernt der Aufschwingregler also ausreichend schnell, das Pendel erfolgreich aufzuschwingen, wenn der ELISE-Ansatz die erfolglosen Phasenwechsel mit dem Phasen-FIFO erkennt und mithilfe der verwendeten Heuristik zusätzliche Lernvorgaben generiert. Ohne den ELISE-Ansatz schafft er es nicht.

Wenn der Balancierregler das Pendel erfolgreich und stabil balanciert, dann ist direkt am Balancierpunkt erfolgreich gelernt worden. Das Ziel der zweiten Untersuchung ist es daher, das gelernte Wissen des Balancierreglers in der Nähe des Übergangswinkels zu überprüfen, also in einer gewissen Entfernung zum Balancierpunkt. Ist dieses Wissen in der Lage, das Pendel wieder zum Balancierpunkt zu bewegen, dann ist der Balancierregler insgesamt also robust.

Für diese Untersuchung wird das Pendel mit funktionierenden Übergabeparametern von $\varphi_{ein} = 4,5^\circ$, $\dot{\varphi}_{ein} = 50^\circ s^{-1}$ und $\varphi_{aus} = 5,0^\circ$ betrieben. Nach den normalen 50 Sekunden, in denen das Wissen in den Reglern ausreichend konvergiert ist, um das Pendel zu balancieren, werden Störungen simuliert. Hierzu wird der Motor des Pendelwagens abgeschaltet, bis ein gewisser Fehlerwinkel erreicht ist. Der Balancierregler muss nun das Pendel wieder auffangen. Diese Untersuchung wird mehrmals ohne und mit dem ELISE-Ansatz durchgeführt. In Abbildung 7 ist die Häufigkeit des Auffangens in Abhängigkeit

vom Störwinkel in einem Histogramm dargestellt. Ohne den ELISE-Ansatz kann das Pendel schon bei kleineren Winkeln nicht mehr gefangen werden. Dies liegt daran, dass erst mit dem ELISE-Ansatz die nötigen Lernvorgaben in den Randbereichen möglich sind.

4 Diskussion

In der ersten Untersuchung wurden die Auswirkungen der Phasengrenze auf den Erfolg der Adaption mit und ohne den ELISE-Ansatz bestimmt. Ist die Phasengrenze zu eng, so wird der Balancierregler viel zu selten aktiviert. Ist sie dagegen zu groß, ist der Balancierregler bei der Aktivierung noch nicht in der Lage, das Pendel zu balancieren. Dazwischen gibt es also ein Intervall von Umschaltwinkeln, in dem die Adaption erfolgreich sein kann. Wichtiger ist, dass dieses Intervall durch die Verwendung des ELISE-Ansatzes größer wird. Dies liegt daran, dass am Rand des Intervalls erfolglose Phasenwechsel passieren. Durch den ELISE-Ansatz werden diese Fälle von ungewollter Umschaltdynamik erkannt und durch zusätzliche Lernvorgaben an den Aufschwingregler kompensiert. Die benutzte Heuristik kann also mit Fällen umgehen, in denen ein Einzelregler erfolgreich lernen würde, wenn nicht auf einen anderen umgeschaltet würde. Durch die Heuristik kann dieses Lernen trotzdem stattfinden.

In der zweiten Untersuchung wurden die Fähigkeit des Balancierreglers mit und ohne den ELISE-Ansatz getestet, das Pendel nach einer Störung zu kontrollieren. Dabei hat sich gezeigt, dass kleine Störungen problemlos sind. Größere Störungen können allerdings nur durch den ELISE-Ansatz kompensiert werden. Das liegt daran, dass die größeren Störungen erfordern, dass der Balancierregler in der Nähe der Phasengrenze erfolgreich gelernt hat. Und an diesen kritischen Stellen konnten aufgrund des ELISE-Ansatzes die wenigen Lernvorgaben viel effektiver genutzt werden. Hohe Konvergenzgeschwindigkeit und Robustheit kann somit auch an den Phasengrenzen erreicht werden.

Die Untersuchungen konnten somit die Relevanz der Probleme des *interrupted learning* aufzeigen. Diese sind umso stärker, je mehr Dynamik ein Phasenwechsel ins System bringen kann, je mehr Lernvorgaben nur aus dem Zusammenspiel der Einzelregler gewonnen werden können und je größer die Verzögerungszeiten sind. Denn größere Verzögerungszeiten führen zu einem größeren Einfluss der Phasengrenzgebiete, in denen ohne den ELISE-Ansatz schlecht gelernt werden kann. Wenn solche Probleme gegeben sind, können mit dem ELISE-Ansatz durch das Umschalten entstehende Lernprobleme vermindert werden, zudem kann das Lernen an den kritischen Phasengrenzen beschleunigt werden.

Diese ersten Arbeiten lassen natürlich eine Reihe von Fragen offen. Es wird sicherlich weitere sinnvolle Heuristiken geben, mit denen sich unerwünschte dynamische Effekte verhindern lassen. Diese können sogar verallgemeinert werden, wenn man die Prozessphasen sinnvoll kategorisieren und die Heuristiken auf diese Kategorien anwenden kann. Dies wurde für eine Heuristik gezeigt, mit der sich Oszillationen zwischen einer vorbereitenden und einer stationären Phase vermeiden lassen. Außerdem gilt es, das Überblenden der Lernvorgänge an die Eigenschaften und die Anzahl der Einzelregler anzupassen. Und schließlich ist die Frage offen, welche Probleme des *disrupted learning* sich auch bei indirekter Adaption stellen.

Literatur

- [1] Khalil, H.: *Nonlinear systems*. Prentice Hall Upper Saddle River, NJ. 2002.
- [2] Michels, K.; Kruse, R.; Andreas, N.; Klawonn, F.: *Fuzzy Control: Fundamentals, Stability and Design of Fuzzy Controllers*. Springer, Heidelberg. 2006.
- [3] Farrell, J.; Polycarpou, M.: *Adaptive Approximation Based Control*. Wiley–Interscience. 2006.
- [4] Foss, B.; Johansen, T.: On local and fuzzy modelling. *Industrial Fuzzy Control and Intelligent Systems, IFIS'93, Third International Conference on* (1993), S. 80–87.
- [5] Jakubek, S.; Keuth, N.: Optimierte Neuro-Fuzzy-Modelle für Auslegungsprozesse und Simulation im Automotive-Bereich. *at-Automatisierungstechnik* 53 (2005) 9, S. 425–433.
- [6] Fink, A.; Töpfer, S.; Isermann, R.: Nonlinear model–based control with local linear neuro–fuzzy models. *Archive of Applied Mechanics* 72 (2003) 11–12, S. 911–922.
- [7] Brockmann, W.; Huwendiek, O.: Adaptive Control of Friction Processes with Neuro-Fuzzy Elements. In: *Proc. 2nd European Congress on Intelligent Techniques and Soft Computing - EUFIT*, S. 835–839. Verlag der Augustinus Buchhandlung, Aachen. 1994.
- [8] Shamma, J.; Athans, M.: Analysis of gain scheduled control for nonlinear plants. *Automatic Control, IEEE Transactions on* 35 (1990) 8, S. 898–907.
- [9] Oosterom, M.; Babuska, R.: Design of a gain-scheduling mechanism for flight control laws by fuzzy clustering. *Control Engineering Practice* 14 (2006) 7, S. 769–781.
- [10] Takagi, T.; Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics* 15 (1985) 1, S. 116–132.
- [11] Angelov, P.; Filev, D.: An Approach to Online Identification of Takagi-Sugeno Fuzzy Models. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 34 (2004) 1, S. 484–498.
- [12] Shao, S.: Fuzzy self-organizing controller and its application for dynamic processes. *Fuzzy Sets and Systems* 26 (1988) 2, S. 151–164.
- [13] Rosemann, N.; Brockmann, W.: Concept for Controlled Self-optimization in Online Learning Neuro-fuzzy Systems. In: *Lecture Notes in Computer Science*, Bd. 4667, S. 498–502. Springer, Heidelberg. 2007.

Geometrische Einflüsse auf das Training von Künstlichen Neuronalen Netzen bei nichtlinearem Systemverhalten

**M. Schneider¹, E. Kraft¹, U. Lehmann¹, J. Krone¹, P. Langbein¹,
Ch. Ament², P. Otto²**

¹Institut für Computer Science, Vision und Computational Intelligence,
Fachhochschule Südwestfalen, Frauenstuhlweg 31, 58644 Iserlohn
Tel. (02371) 566-303 Fax (02371) 566-209

E-Mail: {MSchneider, Lehmann, Krone, Langbein}@fh-swf.de

²Institut für Automatisierungs- und Systemtechnik
Technische Universität Ilmenau, Postfach 10 55 65, 98684 Ilmenau
Tel. (03677) 69-2815 Fax (03677) 69-1434

E-Mail: {christoph.ament, peter.otto}@tu-ilmenau.de

Zusammenfassung

Mit Künstlichen Neuronalen Netzen (KNN) soll ein Modell zur Berechnung der Kräfte an den Auflagerpunkten eines großflächig gekrümmten Bauteils einer Flugzeugrumpfssektion erstellt werden.

1 Motivation und Zielsetzung

Es ist bei Montageunternehmen wettbewerbsentscheidend, wie genau und in welcher Geschwindigkeit einzelne Bauteile in einer Montageeinrichtung aneinandergesetzt werden. Dabei ist es von großer Bedeutung, dass die optimale Montageposition des Bauteils unter Berücksichtigung verschiedener Randbedingungen in einem Schritt erreicht wird. Die nichtlinear wirkenden Kräfte könnten eine solche Randbedingung sein. Bei diesem Anwendungsbeispiel können diese an den Auflagerpunkten eines großflächig gekrümmten Bauteils aus der Flugzeugrumpfmontage gemessen werden. Diese Kräfte dürfen sich nur in festgelegten Grenzen bewegen. Es soll nun versucht werden, mit Hilfe Künstlicher Neuronaler Netze (KNN) ein stabiles Kräftemodell zu entwickeln.

2 Modellbildung

Künstliche Neuronale Netze besitzen die Fähigkeit aus Beispielen zu lernen, ohne dass sie programmiert werden müssen. Sie können sich durch spezielle Lernverfahren selbst während des Trainings adaptieren und verfügen auch in einem gewissen Maße über eine Generalisierungsleistung, so dass sie auch für unbekannte Eingangswerte sinnvolle Ergebnisse liefern. Dazu werden wichtige Zusammenhänge in den Trainingsdaten in den Verbindungsgewichten des Künstlichen Neuronalen Netzes abgebildet. Künstliche Neuronale Netze sind aufgrund ihrer zahlreichen Verbindungen zwischen den Neuronen fehlertolerant, so dass sie auch unvollständige Eingangsdaten sinnvoll verarbeiten können.

Da Künstliche Neuronale Netze aus vorhandenen Daten lernen, können sie auch nur den Ausschnitt der Realität wiedergeben, der in den Trainingsdaten enthalten ist. Deshalb bekommt bei der experimentellen Prozessanalyse die Versuchsplanung eine große Bedeutung, weil diese für den Erfolg beim „realen“ Test der Künstlichen Neuronalen Netze mitverantwortlich sein kann, gerade wenn nicht der gesamte Arbeitsbereich durch die Trainingsdaten abgedeckt ist. Das Grundlagenwissen zu Künstlich Neuronalen Netzen wird in [3] und [4] sehr anschaulich beschrieben.

Mit Hilfe der Korrelationskoeffizienten R soll die Stärke des linearen Zusammenhangs zwischen zwei Messgrößen bestimmt werden. Hier soll nun auch der Zusammenhang zwischen Eingangs- und Ausgangsgröße des Künstlichen Neuronalen Netzes ermittelt werden. Dazu wird die Korrelation jeder einzelnen Eingangsgröße in das Künstliche Neuronale Netz zur Ausgangsgröße ermittelt. Für das erfolgreiche Training eines Künstlichen Neuronalen Netzes ist es wichtig, dass die Ausgangsdaten des Künstlichen Neuronalen Netzes korrekt durch die Eingangsdaten beschrieben werden. Deshalb wird im Folgenden die Korrelation für einige mögliche Kombinationen der Daten untersucht. Die Korrelationskoeffizienten befinden sich im Intervall $[-1, +1]$. Eine Korrelation nahe -1 bzw. $+1$ bedeutet, dass die Messgrößen linear abhängig sind, dagegen sind Messgrößen unabhängig voneinander, wenn der Korrelationskoeffizient gleich 0 ist. Die Ausgangsdaten lassen sich dann nur schwer durch die Eingangsdaten beschreiben.

$$R_{x,y} = \frac{\frac{1}{n} \sum_{i=1}^n [(x_i - \frac{1}{n} \sum_{i=1}^n x_i) \cdot (y_i - \frac{1}{n} \sum_{i=1}^n y_i)]}{\sqrt{[\frac{1}{n} \sum_{i=1}^n (x_i - \frac{1}{n} \sum_{i=1}^n x_i)^2] \cdot \frac{1}{n} \sum_{i=1}^n (y_i - \frac{1}{n} \sum_{i=1}^n y_i)^2}} \quad (1)$$

Dabei bedeutet:

$R_{x,y}$ *Korrelationskoeffizient von x_i und y_i*

Korrelation zwischen Positioniererbewegung bezogen auf die Eintaktposition ΔP_i und der Kraftänderung bezogen auf die Eintaktposition ΔF_i

Beim Training des Neuro-Reglers [2] wurde erfolgreich der Zusammenhang zwischen relativer Targetabweichung und relativem Positioniererverfahrweg gelernt. Deshalb wurde in Tabelle 1 die Korrelation zwischen dem Positioniererverfahrweg bezogen auf die Eintaktposition als Eingang in das Künstliche Neuronale Netz und die Kraftänderung bezogen auf die Eintaktposition als Ausgang des Künstlichen Neuronalen Netzes berechnet. Keine der Korrelationen weichen nach [1] signifikant von 0 ab, weshalb nur ein geringer Zusammenhang zwischen Eingangs- und Ausgangsgröße des Künstlichen Neuronalen Netzes besteht. Ein Korrelationskoeffizient signifikant ungleich 0 befindet sich vom Betrag her im Intervall $[0,6; 1]$. Zur Modellierung der Kräfte mit einem Künstlichen Neuronalen Netz ist dieser Zusammenhang aufgrund der zu geringen linearen Abhängigkeit eher ungeeignet.

	ΔF_{POVy}	ΔF_{POVz}	ΔF_{POMy}	ΔF_{POMz}	ΔF_{POHy}	
$\Delta PUVx$	-0,42	-0,14	-0,11	-0,34	-0,19	
$\Delta PUVy$	-0,09	-0,40	0,03	0,04	0,13	
$\Delta PUVz$	-0,03	-0,24	0,04	0,02	0,18	
$\Delta PUMy$	-0,26	0,13	-0,10	-0,28	-0,16	
$\Delta PUMz$	-0,25	-0,13	-0,05	-0,20	-0,24	
$\Delta PUHy$	-0,36	-0,06	-0,08	-0,32	-0,35	
$\Delta PUHz$	0,07	-0,48	0,23	0,05	-0,03	...

Abbildung 2.1: Auszug aus der Korrelationsanalyse $R(\Delta P_i, \Delta F_i)$

Korrelation R zwischen den Differenzen von Targetabweichungen ΔT_i und der Kraft F_i

Bei der Berechnung der Targetdifferenzen werden jeweils zwei Targetkomponenten voneinander abgezogen und diese Targetdifferenz für die weiteren Schritte verwendet. So ist z.B. TOVUVy die Differenz der y-Komponente des Targets oben vorne (TOVy) und der y-Komponente des Targets unten mitte (TUVy). In Abb. 2.2 und 2.3 ist die Differenzenbildung der Targets zu, besseren Verständnis graphisch dargestellt.

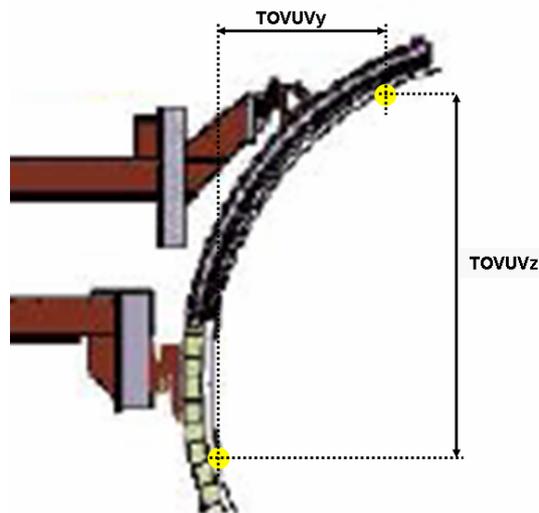


Abbildung 2.2: Seitenansicht eines Bauteils einer Flugzeugrumpfsktion mit den Bezeichnungen der Targetdifferenzen [7]

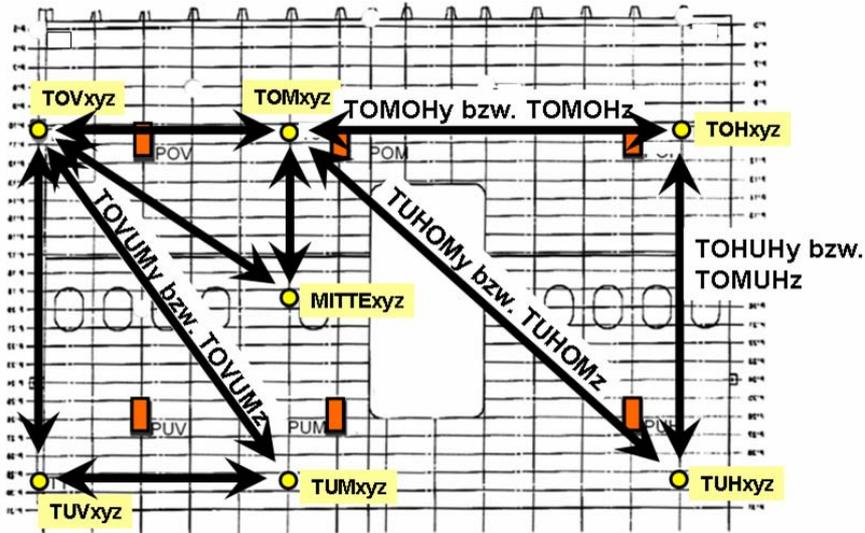


Abbildung 2.3: Ansicht eines Bauteils einer Flugzeuggruppensektion mit den Bezeichnungen der Targetdifferenzen (Schema zur Berechnung der Targetdifferenzen) [7]

	F_POVy	F_POVz	F_POMy	F_POMz	F_POHy
TOVOMy	0,69	-0,08	0,54	0,67	0,74
TOVOMz	-0,36	0,27	-0,39	-0,45	-0,43
TOMOHy	-0,72	0,16	-0,66	-0,66	-0,57
TOMOHx	0,81	-0,06	0,67	0,71	0,70
TUVUMy	-0,84	-0,33	-0,66	-0,43	-0,60
TUVUMz	0,04	0,64	-0,05	-0,44	-0,20
TUMUHy	0,68	-0,35	0,59	0,82	0,71
TUMUHz	0,68	0,53	0,47	0,18	0,41

Abbildung 2.4: Auszug aus der Korrelationsanalyse $R(\Delta T_i, F_i)$

Die Korrelationskoeffizienten R , welche signifikant ungleich Null sind, sind in Abb. 2.4 hellgrün markiert. Es scheint, dass sich diese Zusammenhänge aufgrund der größeren Korrelation besser mit einem Künstlichen Neuronalen Netz trainieren lassen.

3 Ergebnis

Es konnte die Korrelation zwischen den Eingaben und den Ausgaben der Künstlichen Neuronalen Netzes durch die paarweise Differenzenbildung der Targetkomponenten erhöht werden (Abb. 2.4). Dieser Vorverarbeitungsschritt lieferte nach dem Training des Künstlichen Neuronalen Netzes eine bessere Generalisierungsleistung (Abb. 3.2) als die Variante mit den Positioniererbewegungen als Input (Abb. 3.1).

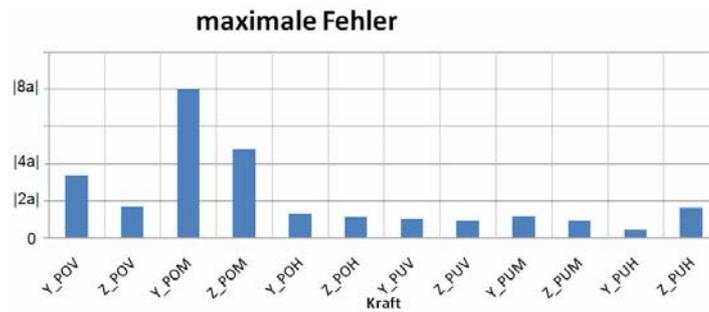


Abbildung 3.1: Trainingsergebnis des KNN mit den Positionierbewegungen als Input und den Kraftänderungen als Output

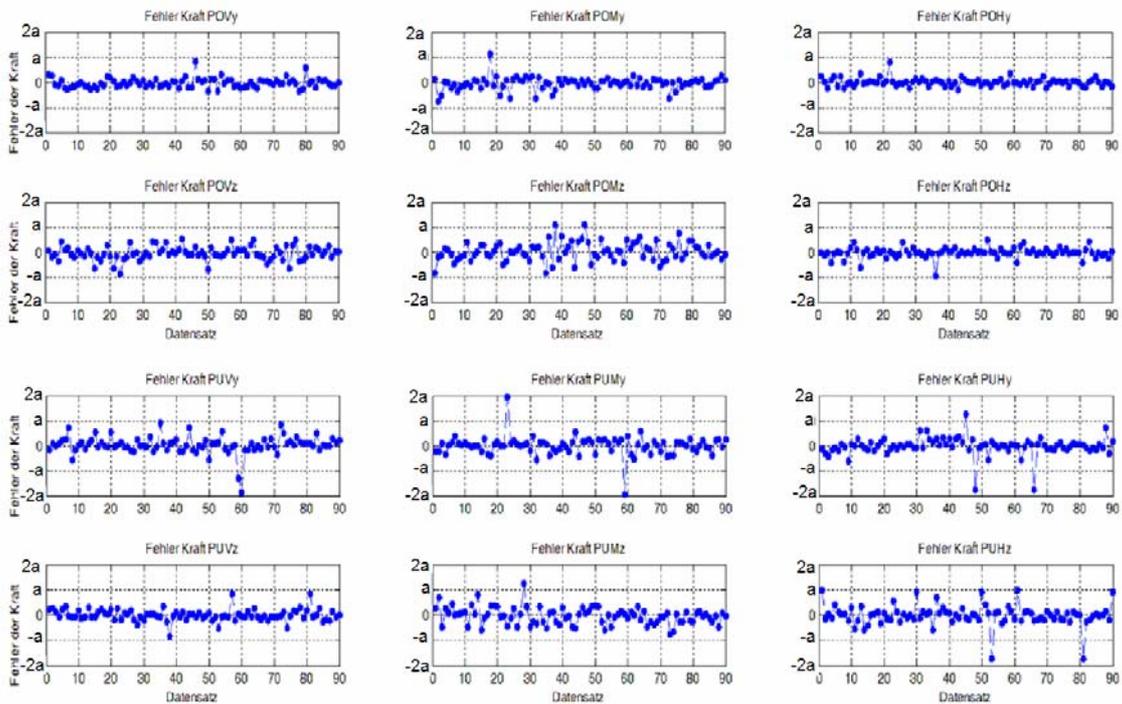


Abbildung 3.2: Trainingsergebnis des KNN mit den Targetdifferenzen als Input und den Kräften als Output

Die Geometrie des großflächig gekrümmten Bauteils aus der Flugzeugrumpfmontage hat somit einen großen Einfluss auf den Trainingserfolg des Künstlichen Neuronalen Netzes zur Berechnung der an den Auflagerpunkten wirkenden Kräfte während der Montage.

4 Ausblick

Dieses Teilmodell wird als nächstes mit den beiden anderen Komponenten Materialmodell und Neuro-Regler in ein Gesamtmodell zusammengefügt. Die Aufgabe des Gesamtmodells ist es ein großflächig gekrümmtes Bauteil aus einer beliebigen Position in die Sollposition zu regeln, ohne dabei die zulässigen Kraftgrenzen zu überschreiten

5 Danksagung

Das Institut für Computer Science, Vision und Computational Intelligence ist aus der Kompetenzplattform Computer Vision und Computational Intelligence (KOPF CV&CI) heraus gegründet worden. Die KOPF CV&CI wird vom Ministerium für Innovation, Wissenschaft, Forschung und Technologie des Landes Nordrhein-Westfalen gefördert. Die Autoren wollen ihren aufrichtigen Dank an Herrn Dipl.-Ing. Wolfgang Brandt von der Airbus Deutschland GmbH für die Unterstützung dieser Forschungsarbeit ausdrücken.

Literatur/Veröffentlichungen

- [1] D. Stoyan: Stochastik für Ingenieure und Naturwissenschaftler. Akademie Verlag, 1993
- [2] M. Schneider, U. Lehmann, P. Langbein, J. Krone, Ch. Ament, P. Otto, U. Stark: Künstliches Neuronales Netz zur Regelung von Montageprozessen im Flugzeugbau. Automatisierungstechnik at, Themenheft "Computational Intelligence", Oldenburg Verlag, 07/2008
- [3] A. Zell: Simulation neuronaler Netze. Oldenburg Verlag, 1994
- [4] R. Rojas: Theorie der neuronalen Netze. Springer-Verlag, 1996
- [5] Airbus Deutschland GmbH, 2006

Lösung komplexer Probleme mittels heuristischer Verfahren

L. Schreiber, H. Sommer, A. Kroll

Universität Kassel, FB Maschinenbau, FG Mess- und Regelungstechnik
E-Mail: lsch@ifm.maschinenbau.uni-kassel.de

Zusammenfassung

Komplexe Probleme sind nicht mit den üblichen mathematischen Lösungsmethoden bearbeitbar. Ausgehend von einer strukturellen Interpretation des Komplexitätsbegriffs werden Vorgehensweisen vorgeschlagen, um direkt aus der Problembeschreibung heraus Lösungsheuristiken zu entwickeln. Die qualitative und nicht die quantitative Bedeutung der Komplexität ist die leitende Idee dieses Ansatzes.

1 Einleitung

Das Problem der Komplexität ist mit den Computerwissenschaften zu einem der wichtigsten Probleme für die Technik geworden. Aus dem Scheitern vieler Projekte zum Einsatz von Computern zur Anlagensteuerung musste man um 1980 erkennen, dass das ursprüngliche Kriterium der 'Schwierigkeit einer Aufgabe' nicht ausreicht, um deren Machbarkeit abschätzen zu können. Auch als 'einfach' einzuschätzende Aufgaben wie die Verwaltung der Daten des Telefonverkehrs (Verbindungsaufbau, Gebührenerfassung und Dokumentation der Teilnehmergegespräche) scheiterten damals an der Nichtabschätzbarkeit ihrer Komplexität (SIEMENS EWS). Aus dieser Problemsituation heraus entwickelte sich die Wissenschaft der Softwaretechnologie mit der Bereitstellung der Methoden der Strukturierten Programmierung und der Eingebetteten Systeme. Schon um 1980 war jedoch abzusehen, dass das Komplexitätsproblem nicht auf die Computerwissenschaften beschränkt ist, sondern sich auf die gesamte Technik ausbreiten würde, und so gab es bereits 1982 ein DFG Schwerpunktprojekt in der Regelungstechnik zu diesem Thema [1].

Aber was macht eigentlich das Thema der Komplexität so speziell? Als Wissenschaftler sind wir es gewohnt, ein Problem zunächst klar zu erfassen, seine Charakteristika zu erkennen und dann die zu diesen Charakteristika passenden spezifischen Lösungsmethoden zu entwickeln. Beim Komplexitätsproblem führt jedoch schon der erste Schritt zur Wahrnehmung, dass der Komplexitätsbegriff nicht eine, sondern viele sehr unterschiedliche Definitionen ermöglicht:

- Kolmogoroffsche Komplexität [2],
- Grzegorzky-Hierarchie [2],
- NP-Schwierigkeit [3],
- Hohe Anzahl von Multiplikationen in einem Algorithmus [4],
- Inseparabilität eines Systems in seriell und parallel geschaltete Komponenten [5],
- ein System wird komplex genannt, wenn es kein simulierbares Modell hat [6], und viele andere mehr.

Auch die vorgeschlagenen Lösungsmethoden können nicht im Sinne des klassischen Lösungsbegriffs als wirkliche Lösungen verstanden werden. Sie bestehen darin, Komplexität einzubetten (indem man Kompliziertes nach außen unsichtbar macht), oder indem man die Unlösbarkeit eines komplexen Problems in Kauf nimmt und, anstatt es zu lösen, lernt, mit den nun unvermeidbar 'emergierenden' Fehlern mittels fehlertoleranter Systeme zu überleben.

Gerade dieser zweite Weg sprengt aber ureigenste technische Zielsetzungen und das bisherige technische Denken und ist daher für unser Technikverständnis prinzipiell problematisch.

Wir wollen hier folgende Thesen begründen:

- I Das Problem der Komplexität wird immer stärker bei der Fortentwicklung technischer Systeme auftreten.
- II Die Komplexität der Probleme kann nicht allgemein mit Komplexitätsmaßen gemessen werden, um diese algorithmisch zu lösen.
- III Komplexität ist nicht als quantitatives Problem, sondern nur als ein strukturelles Problem bearbeitbar. Bei komplexen Aufgaben muss ein Lösungsweg aus der Problemstruktur heraus entwickelt werden. Die Strukturen komplexer Probleme gilt es zu kennzeichnen und die 'Verhaltensregeln' zu finden, die beim Auftreten des jeweiligen Problemtyps erfolgreich sind.
- IV Die Lösungsmethoden, die letztlich zu komplexen Problemen gefunden werden, können keine Algorithmen sein, sondern es kann sich nur um Heuristiken handeln.

2 Strukturelle Charakterisierung komplexer Systeme

Umgangssprachlich nennen wir ein Problem komplex, wenn es uns unmöglich ist, alle Bestimmungsgrößen und Charakteristika dieses Problems vollständig zu erfassen. Während also bei einem schwierigen Problem nur der Weg zur Lösung lang ist oder unkonventionelle Ideen erfordert, das Problem selbst aber in allen seinen Charakterisierungen, die zur Lösungsfindung relevant sind, exakt fixiert ist, sind diese Charakterisierungen, die zur Ableitung der Lösung benötigt werden, bei komplexen Problemen selbst unklar [7].

Ein besseres Verständnis für diese Problematik versuchen Busch et al [8] aus einer Betrachtung der Navier-Stokes'schen Gleichung (NSG) zu gewinnen. Die Frage nach der Existenz und Eindeutigkeit klassischer Lösungen der NSG ist bis heute ungelöst und sie wird als so wichtig angesehen, dass diese Frage in die Millenniumsprobleme aufgenommen wurde, diejenigen Probleme, die vom zweiten Jahrtausend dem dritten Jahrtausend gestellt werden. Charles L. Feferman schreibt bei der Vorstellung des Problems [9]:

'Standard methods (from partial differential equations) appear inadequate to settle the problem. Instead, we probably need some deep, new ideas.'

Natürlich sind auch Busch et al. weit entfernt davon, dieses Problem zu lösen. Streng genommen wird in ihrer Arbeit das Problem der Lösungsfindung zu einer mathematisch exakt gestellten Differential-Anfangsrandwertaufgabe überhaupt nicht behandelt. Anstatt sich auf die Lösungssuche zu diesem wirklich schwierigen Problem zu begeben, ändern

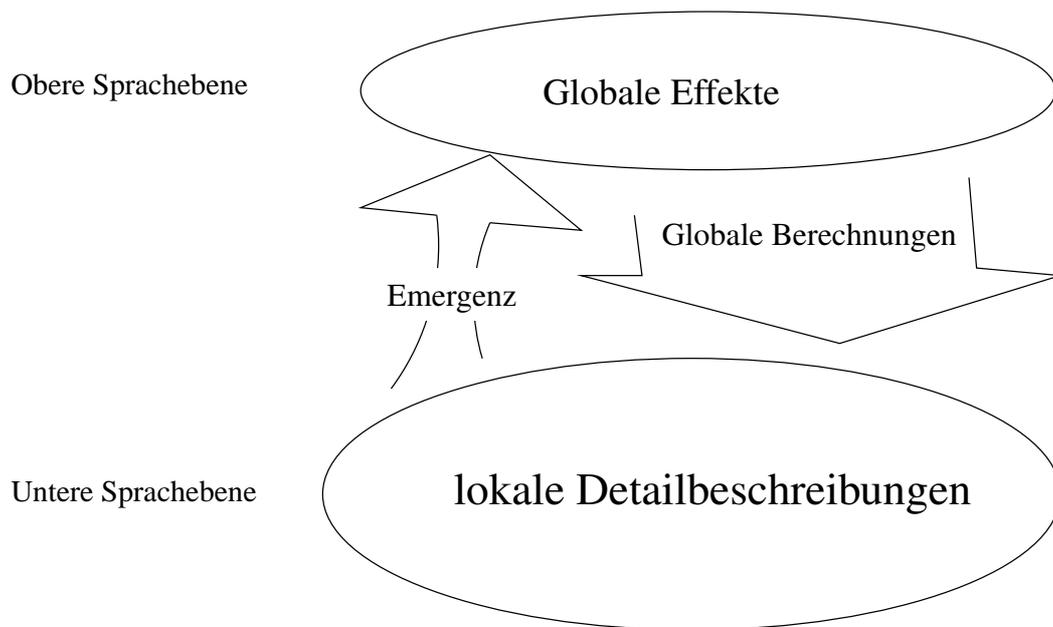


Bild 1: Das Zweisprachenmodell zur Darstellung komplexer Systeme.

sie die Betrachtungsweise. Sie machen aus einem schwierigen ein komplexes Problem und erhalten damit die Freiheit, durch geeignete Festlegung der Grundcharakteristika, dieses neue Problem lösbar zu machen. Aus der Fragestellung: 'Warum ist das Lösungsproblem der NSGI eigentlich so schwierig?' heraus entwickeln sie eine Problemcharakterisierung auf zwei Sprachebenen (Bild 1).

Einer unteren Sprachebene der Detailbeschreibung und einer oberen Sprachebene der globalen Effekte. Das Wechselspiel zwischen beiden Beschreibungsebenen ist nicht in einer der beiden Beschreibungssprachen allein darstellbar. Emergierende Ereignisse, die aus der unteren Ebene in die obere einfallen, sind von dieser aus nicht prognostizierbar, und umgekehrt sind die globalen Zwangsbedingungen der oberen Ebene in der lokalen Sprache der unteren Ebene nicht formulierbar.

Bei turbulenten Strömungen können Wirbel als emergente Ereignisse aufgefasst werden, und durch Zähigkeitseffekte entstehen globale Abhängigkeiten.

Das Zwei-Sprachenmodell der NSGI zeigt also all die Phänomene, die wir von einem komplexen System erwarten:

- Nichtprognostizierbarkeit des Verhaltens,
- Unbestimmtheit relevanter Eigenschaften (z.B. der Orte, an denen eine Strömung abreißt), etc.

Da ein solches Mehr-Sprachenmodell in der Lage ist, die Phänomene darzustellen, die komplexe Systeme kennzeichnen, machen wir den Vorschlag, ein System dann als 'strukturell komplex' zu bezeichnen, wenn es nur mittels eines Mehr-Sprachenmodells effektiv (d.h. in lösbarer Form) dargestellt werden kann.

Die vorgeschlagene Charakterisierung ist nicht objektiv, sondern auch abhängig von den Komplexitätsbeherrschungsfähigkeiten der Problembearbeiter. Was sich für die eine Arbeitsgruppe schon als zu komplex herausstellt, könnte für eine andere durchaus noch beherrschbar sein. Aber wie die Erfahrung lehrt, liegen die Grenzen in der Regel sehr eng zusammen.

3 Zur Unvermeidbarkeit des Komplexitätsproblems

Der im vorherigen Abschnitt angegebene Vorschlag zur strukturellen Charakterisierung von Komplexität erscheint zunächst nur theoretisch begründet und nicht von technischen Anwendungen her motiviert zu sein. Um eine praktische Motivation für diesen Vorschlag zu geben, erinnern wir an die sogenannten Baraban-Albert-Modelle, die zur Beschreibung gewachsener technischer Systeme verwendet werden [10].

Technische Systeme werden heute in ihrer Mehrheit nicht vollständig neu konzipiert, sondern sie bestehen aus bereits entwickelten bewährten Komponenten, die miteinander vernetzt werden, bzw. die wir in bereits bestehende Systeme neu einfügen. Dabei wird davon ausgegangen, dass eine neu hinzugefügte Komponente um so wahrscheinlicher mit einer anderen Komponente des Systems vernetzt wird, je wichtiger diese Komponente ist, d.h. je vernetzter sie bereits ist.

Für die Wahrscheinlichkeit $p(K_i)$ der Verknüpfung der neuen Komponente mit der i -ten Komponente des Systems K_i gilt:

$$p(K_i) = \frac{\text{Anzahl der Verknüpfungen von } K_i}{\text{Anzahl aller Verknüpfungen}}$$

Die so entstehenden Netzwerke haben folgende Eigenschaften [10]:

- I Die Grade der Knoten, d.h. die Anzahlen ihrer Verknüpfungen sind nach einem Potenzgesetz verteilt.
- II Es existieren keine Korrelationen zwischen den Graden von Nachbarknoten.
- III Clusterbildung, d.h. Strukturentstehung auf dem Netz ist wesentlich höher als dies bei einem reinen Zufallsgraphen der Fall wäre.
- IV Es entsteht der sogenannte 'small-world'-Effekt. D.h. die kürzesten Verbindungen zwischen zwei Knoten sind im Mittel wesentlich kürzer als dies auf einem Zufallsgraphen mit vergleichbarem durchschnittlichen Knotengrad der Fall wäre.

Die Eigenschaften [I] bis [IV] zeigen, dass gewachsene technische Systeme nur schlecht mittels reiner Wahrscheinlichkeitstheorie beschreibbar sind. [III] und [IV] bedeuten, dass, im Gegensatz zu den Erwartungen, die man auf Grund der Wahrscheinlichkeitstheorie hätte, in der Regel auf einem gewachsenen System eine Strukturierung und damit eine Beschreibungssprache für diese existiert. Diese Beschreibungssprache ist dann die höhere Sprache, die wir zur vollständigen Charakterisierung eines gewachsenen technischen Systems benötigen. Mit dem Wachstum technischer Systeme scheint daher das Komplexitätsproblem immer wichtiger zu werden.

Mathematische Problemlösungsmethode

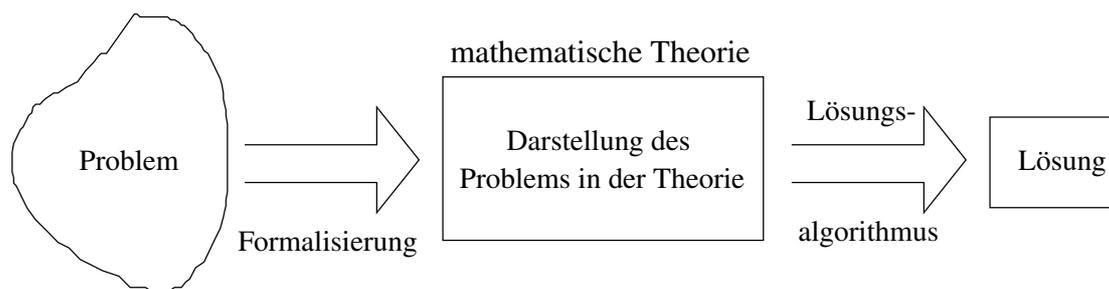


Bild 2: Umformung eines Problems in eine mathematische Aufgabe und deren Lösung.

4 Probleme mit mathematischen Lösungsmethoden zu komplexen Problemen

Die Frage, die sich an dieser Stelle unserer Überlegungen aufdrängt, lautet natürlich: 'Hilft uns die angegebene Charakterisierung bei der Bearbeitung (eventuell auch Lösung) komplexer Probleme?'

Um diese Frage zu beantworten, müssen wir zunächst die übliche Herangehensweise an Probleme betrachten und uns deren prinzipielle Schwierigkeiten vergegenwärtigen.

Ausgehend von einem Problem müssen wir dieses beim üblichen Problemlösungsverfahren zunächst in einer vollständig festgelegten mathematischen Struktur (Theorie) darstellen und dann einen Lösungsalgorithmus finden, der die Problemlösung findet (Bild 2).

Die Vorteile dieses Vorgehens sind allgemein bekannt:

Wir haben mit ihm die Lösungsmöglichkeiten vollständig eingegrenzt, wir können auch nach der Eindeutigkeit und Optimalität der Lösung fragen und wir haben klar fixierte mathematische Grundstrukturen, aus denen eventuell die Unlösbarkeit des Problems erkennbar sein müsste.

Die Nachteile des mathematischen Verfahrens liegen in der strikten Separation, die es zwischen den Schritten bedingt. Die formale Darstellung des Problems erfolgt unabhängig von der Suche und den Eigenschaften des Problemlösungsalgorithmus. Sie erlaubt keine Anpassung, mit der dessen Auffinden vereinfacht oder anhand der die Eigenschaften des Lösungsalgorithmus im Voraus gesichert werden könnten. In der Praxis führt dies häufig bei komplexen Problemen dazu, dass die Lösungsalgorithmen nicht in polynomialer Zeit ablaufen und damit praktisch unbrauchbar werden. Trotz ihrer prinzipiellen Durchführbarkeit ist dann die mathematische Lösungsmethode sinnlos.

5 Heuristiken als Lösungsverfahren für komplexe Probleme

Heuristiken sind sehr generell anwendbare, aber auch unspezifische Lösungsverfahren, die im Allgemeinen:

- nicht mit Sicherheit eine Problemlösung ergeben,
- keine Optimalität der Lösung garantieren,
- meist keinen Hinweis zum prinzipiellen Problemverständnis liefern.

Durch die größeren Freiheiten, die wir mittels des Zwei-Sprachenmodells zur Darstellung komplexer Probleme haben, können wir dann aber versuchen, unsere Problemdarstellung nicht nur an die Problemstellung, sondern auch an die Lösungsfindung anzupassen.

Ein einfaches Beispiel für ein solches Vorgehen wird von Fan und Mumford [11] zur Problemstellung der optimalen Gestaltung von Bus-Fahrplänen gegeben. Hierbei gilt es, mehrere Bedingungen einzuhalten:

- a: Die Fahrtwünsche aller Kunden sollen befriedigt werden,
- b: die Wartezeiten der Kunden sollen möglichst kurz sein,
- c: die Fahrtzeiten der Kunden sollen möglichst kurz sein, d.h die Verbindungen sollen direkt sein,
- d: die Personenbeförderungskapazitäten der Busse müssen eingehalten werden.
- e: die Kosten des Busverkehrs sollen möglichst gering sein,
- f: die maximalen Fahrzeiten, Ruhezeiten und Fahrzeitreserven der Busfahrer müssen eingehalten werden.

Eine vollständige mathematische Darstellung dieses Problems wäre sicherlich möglich, aber sehr kompliziert und sie würde auf ein NP-schwieriges mathematisches Problem führen, d.h. zu einem mathematisch nicht effektiv lösbaren mathematischen Problem. Fan und Mumford gehen daher so vor, dass sie sich im ersten Schritt auf eine lokale Lösungssuche festlegen. Als Lösungssprache verwenden sie die Fahrplanlisten, die auch im praktischen Busverkehr üblich sind und die Fahrplanänderungen, die auch die Praktiker vornehmen. Diese Änderungen definieren sie als lokale Änderungsmöglichkeiten. A_0 sei die Liste aller Änderungsmöglichkeiten.

Mit diesen Definitionen ist dann ein heuristisches Suchverfahren nach einer günstigen Fahrplangestaltung realisierbar:

Heuristisches Verfahren zur Fahrplangestaltung:

Schlage einen Fahrplan vor.

$A = A_0$

DO UNTIL ($A = \emptyset$)

Wähle Änderung aus A und setze $A := A - \{\text{Änderung}\}$.

Wende Änderung auf Fahrplan an und erzeuge dadurch einen geänderten Fahrplan.

IF NOT (geänderter Fahrplan widerspricht den Bedingungen [a],[d] bzw. [f])
AND (geänderter Fahrplan besser als Fahrplan bezüglich der Bedingungen [b],[c]
bzw. [e])
THEN Fahrplan = geänderter Fahrplan, $A = A_0$
END DO
Gib Fahrplan aus.

Führt man dieses einfache heuristische Verfahren für mehrere Anfangs-Fahrpläne durch, so kommt man, wie Fan und Mumford nachweisen, zu einem akzeptablen Ergebnis.

Wie wir aus dem Beispiel sehen, sind heuristische Verfahren nicht theorie- sondern anwendungsorientiert. Dies führt aber dazu, dass es keine theoriegestützten Entwicklungs- und Auswahlverfahren für Heuristiken gibt. In den Zeitschriften, die sich mit Heuristiken beschäftigen (Artificial Intelligence, Artificial Intelligence Reviews, Journal of Heuristics, Advanced Engineering Informatics, Mind and Machines, etc.) wird daher die Forderung nach einer übergreifenden Theorie zu diesen Verfahren immer lauter ([12], [13], [14]). Sie ist bekannt unter dem Schlagwort:

'Hooker's challenge for a more structured approach to heuristics.'

6 Ein Zwei-Ebenenmodell zur strukturierten Lösung komplexer Probleme mittels heuristischer Verfahren

Im Abschnitt 4 ergab die Diskussion mathematischer Lösungsmethoden, dass deren Problem bei der praktischen Anwendung darin besteht, dass sie nach der Formalisierung des Problems oft nicht aus unserem Problemwissen heraus direkt abgeleitet werden können. Mathematische Lösungsverfahren können auch unkonventionelle ,d.h. aus der Problemstellung nicht direkt entdeckbare Ideen enthalten und sie sind daher nicht immer einfach zu finden [7]. Der große Vorteil der heuristischen Verfahren besteht dagegen darin, dass sie in sehr flexiblen allgemeinen Begriffen definiert sind. Wir können daher versuchen, unsere Beschreibungssprache und die Problembeschreibung an ein geplantes Verfahren anzupassen. Dieser Vorschlag ist im dargestellten Lösungsstrukturplan verdeutlicht (Bild 3):

Der Lösungsstrukturplan macht deutlich, an welchen Punkten weitere theoriegestützte Methoden hilfreich wären. Zunächst einmal muss das Zusammenpassen von Problem und Heuristik überprüfbar sein. In der Literatur findet man diese Forderung unter den Slogan: 'Problem structures have to be tied to solution methods.'

Der zweite wichtige Punkt betrifft die Auswahl, bzw. Anpassung der Beschreibungssprache.

Zum ersten Punkt könnte eine Charakterisierung grundlegender heuristischer Verfahren sinnvoll sein, um ein 'Katalogbasiertes Auswahlverfahren' zu ermöglichen. Die nachfolgend dargestellte Tabelle ist ein erster Schritt in dieser Richtung. Selbstverständlich erhebt diese Tabelle keinen Anspruch auf Vollständigkeit.

Lösungsstrukturplan

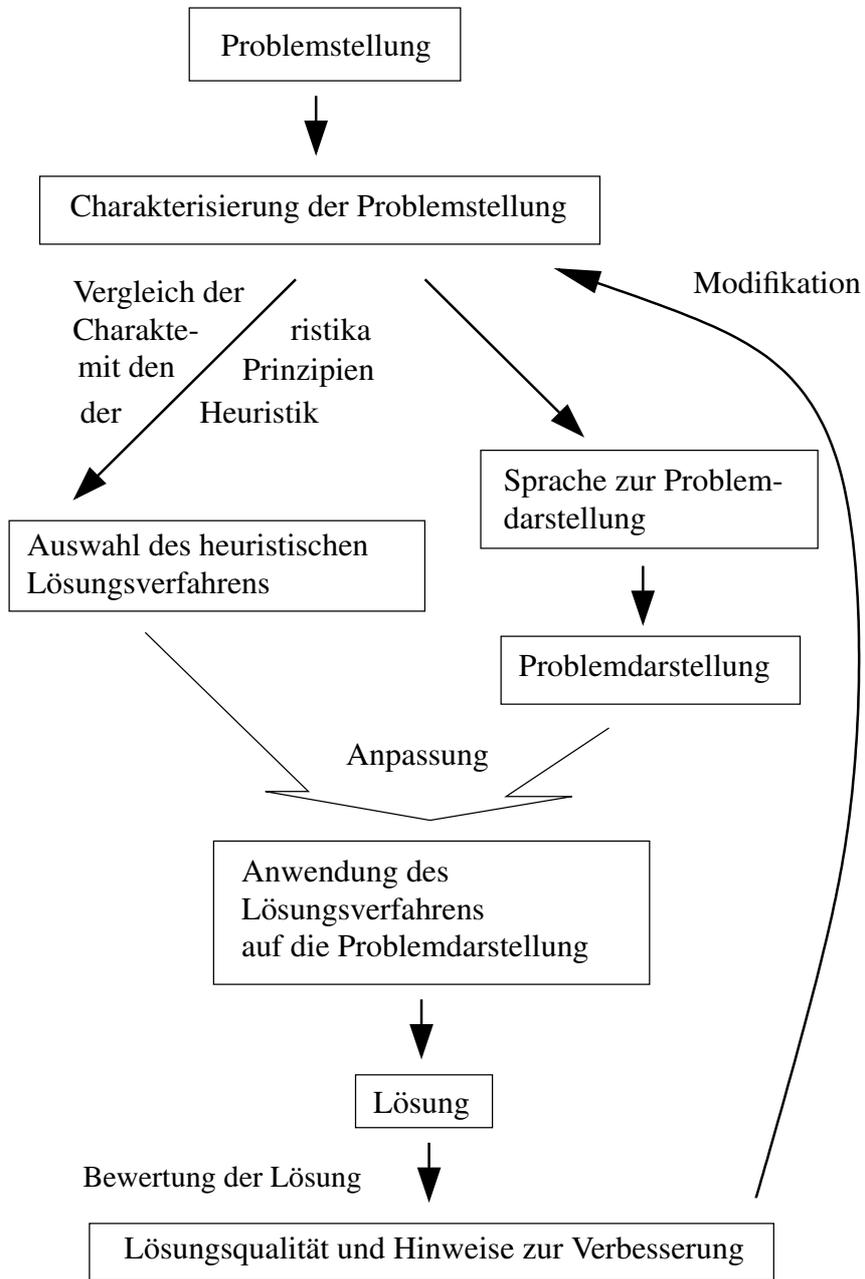


Bild 3: Heuristisches Lösungsverfahren.

Übersichtstabelle heuristischer Verfahren:

Verfahren	Idee des Verfahrens
Neuronales Netz	Lasse viele versuchen, eine Lösung zu finden, und stärke die Besten.
Gradientenverfahren oder lokale Suchverfahren	Suche das Bessere nahe dem Guten.
Genetische Algorithmen / Evolutionsstrategie	Bilde das Bessere aus Teilen der guten Lösungen.
Simulierte Abkühlung	Lege durch eine grobe Betrachtung zuerst den Bereich fest, in dem dann eine feinere Suche erfolgen soll.
Tabu-Search	Kennzeichne nutzlose Versuche und wiederhole sie nicht.
Schwarm-Algorithmen	Orientierung an den Nachbarn und zusätzlich einem global gesuchten Vorbild dessen Methoden übernommen werden.
Cross-Entropie-Methode	Vereinfache zunächst die Aufgabe und ermittle dann bezüglich der vereinfachten Aufgabe eine Verteilung der Lernbeispiele bezüglich der das gesuchte Ergebnis in einem Suchraum mit geringerer Streuung entsteht.

Die gegebene Idee der Verfahren gibt manchen Hinweis zur Charakterisierung derselben und zu deren Anwendbarkeit. Wer z.B. die Idee der Genetische Algorithmen berücksichtigt, wird diese nicht einsetzen, wenn die Güte des Suchziels vor allem von der Koordination der Teile desselben abhängt. In diesem Falle würde ja die individuelle Güte jedes Teils nur wenig zur Gesamtgüte des Suchziels beitragen. Auch bei der Verkoppelung von Heuristiken ist es wichtig auf die Stimmigkeit ihrer Grundcharakteristiken zu achten. Nicht stimmige Verfahren, die z.B. lokale und globale Suchziele vermengen, (also z.B. Schwarmalgorithmen und Genetische Verfahren,) sollten nicht verkoppelt werden.

Jedes Verfahren, auch jede Heuristik ist strukturiert, aber durch die Stärke der Ausprägung derselben wird ihre Anwendbarkeit begrenzt. Gute Heuristiken müssen daher einfach sein und sie sollten in flexibel anpassbaren Begriffen formuliert sein. Eine ideale Heuristik ist aus mathematischer Sicht strukturlos. D.h. das Vollendete aus einer Perspektive erscheint unzulänglich aus der anderen. Es gilt hier Laotses Leitspruch:

Große Vollendung muss wie unzulänglich erscheinen,
Große Fülle muss wie leer erscheinen,
Große Geradheit muss wie krumm erscheinen,
Große Begabung muss wie dumm erscheinen,
Große Beredsamkeit muss wie stumm erscheinen.

7 Konzeption zur Bearbeitung komplexer Probleme

Komplexe Probleme sind gemäß unseren Ausführungen diejenigen, deren Beschreibungsstruktur, in der sie bearbeitbar sind, nicht eindeutig vom Problem selbst her bestimmbar ist. Sie sollten daher mittels Heuristiken nach dem folgenden Bearbeitungsschema gelöst werden:

Bearbeitungsschema für komplexe Probleme:

- I Bestimmung der Charakteristika des Problems und Auswahl einer zu diesen passenden Heuristik.
- II Zielorientierte Bestimmung einer formalen Problembeschreibung:
Alle für die Heuristik notwendigen Begriffe müssen mittels der Probleminformation spezifiziert werden.
- III Anwendung und Bewertung der Heuristik.
- IV Ergibt die Heuristik keine geeignete Lösung, so sind die Gründe für dieses Scheitern festzustellen. Diese Gründe sind dann bei der Verbesserung oder neuen Wahl desselben in [I] zu berücksichtigen.

Die vorgestellte Konzeption wurde schon in mehreren Projekten getestet. Eine Erkennungssprache für das Versagen von Bohrern beim Gewindeschneiden wurde von Dürrbaum et al. mittels von Glaubenskalkülen nach den angegebenen Prinzipien entwickelt [15], und Gerland et al. stellen die Konzipierung einer Beschreibungssprache vor, die problemorientiert mittels Klassifikatoren zu einer komplexen Regelungsaufgabe erfolgt [16].

Zum Problem der Korrespondenz von Problem und Lösungsheuristik liegen zur Zeit erst wenige Ergebnisse vor. Dieses Problem ist aber keineswegs einfach und vielleicht auch nicht vollständig lösbar. Zum analogen Problem der Lösungsübertragung von einer Aufgabe auf andere Aufgaben, findet man in der Literatur die Feststellung, dass es kein 'unified framework for a definition of task relatedness' gibt ([17],[18]). Methoden zur direkten Ableitung von Heuristiken aus Problemstellungen sind in Vorbereitung [19].

8 Fazit

Fortschritte bei der Bearbeitung komplexer Probleme können nicht darin bestehen, neue Methoden oder sogar Algorithmen zu finden, die diese lösbar machen. Würde zu einem Problem ein solcher Algorithmus gefunden, so wäre das Problem nicht mehr als komplex einzuschätzen. Komplexe Probleme sind auch nicht umgehbar, da wir mit dem Wachsen der technischen Systeme ihre Zunahme erwarten müssen. Die Bearbeitung komplexer Probleme erfordert daher eine ganz neue Herangehensweise. Es geht nicht mehr darum, nur ein Lösungsverfahren zu finden, sondern darum, einen gesamten Lösungsprozess zu konzipieren, dem auch die Problembeschreibungssprache angepasst wird. Dieser gesamte Lösungsprozess soll zielgerichtet verlaufen und daher ohne orientierungslose Suchverfahren (exhaustive search method) auskommen, die bei großen Problemen nicht effizient durchführbar wären. Zur Entwicklung einer entsprechenden Methodologie existieren

heute, verglichen mit der Zeit des DFG-Schwerpunktprojekts von 1982, wesentlich bessere Voraussetzungen. Es gab in der Zwischenzeit Weiterentwicklungen im Bereich des Software-Engineering, es wurden neue Paradigmen in der Computational Intelligence entwickelt und eine Theorie komplexer Netze ist vollständig neu entstanden [20]. In der Physik hat sich die Computational Physics als fester Bestandteil durchgesetzt. R.H. Landau schreibt dazu [21]:

'The power of computation is leading us to think about problems differently than we might have before and even to think about different problems than we might have before.'

Es geht darum, dieses neue Denken weiter zu entwickeln und zu systematisieren, um komplexe Fragestellungen bearbeitbar zu machen. Dieses Ziel ist einer der Arbeitsschwerpunkte des neu gebildeten Fachgebiets Mess- und Regelungstechnik an der Universität Kassel.

Literatur

- [1] Schmidt, G.: *Was sind und wie entstehen komplexe Systeme, und welche spezifischen Aufgaben stellen sie für die Regelungstechnik*. Regelungstechnik 30 Heft 10, 331-339. 1982.
- [2] Schnorr C.P.: *Rekursive Funktionen und ihre Komplexität*. Teubner: Stuttgart. 1974.
- [3] Garey M.R. Johnson D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman: New York. 1979. 1974.
- [4] Blahut R.E.: *Fast Algorithms for Digital Signal Processing*. Addison-Wesley Publishing Company: Massachusetts. 1984.
- [5] Caserta M. Nodar M.C.: *A Cross Entropy Based Algorithm for Reliability Problems*. J. Heuristics, DOI 10.1007/s10732-008-9074-2. 2008.
- [6] Rosen R.: *Essays on Life Itself*. Columbia University Press: New York. 2000.
- [7] Arana A.: *On Formally Measuring and Eliminating Extraneous Notions in Proofs*. Philosophia Mathematica(III), 1-19. 2008.
- [8] Busch M. Schweizer B. Sommer H. Sommer V.: *Ein einfaches Berechnungsverfahren zu den Navier-Stokes'schen Gleichungen*. Proc., 18 Workshop Computational-Intelligence Universitätsverlag: Karlsruhe. 2008.
- [9] Feferman C.L.: *Existence and Smoothness of the Navier-Stokes Equation*. at:<http://www.claymath.org/millennium/>. 2001.
- [10] Caldarelli G. Garlaschelli D.: *Self-Organized Criticality*. arXiv:0806.1655v1[cond-mat.dis-nn]10 Jun. 2008.
- [11] Fan L. Mumford C.L.: *A Metaheuristic Approach to the Urban Transit Routing Problem*. J. Heuristics DOI 10.1007/s10732-008-9089-8. 2008.
- [12] Bench-Capon T.J.M. Dune P.E.: *Argumentation in Artificial Intelligence*. Artificial Intelligence 171, 619-641. 2007.

- [13] Atkinson K. Bench-Capon T.: *Practical Reasoning as Presumptive Argumentation Using Action Based Alternating Transition Systems*. Artificial Intelligence 171, 855-874. 2007.
- [14] Verdegay J.L. Yager R.R. Bonissone P.P.: *On Heuristics as a Fundamental Constituent of Soft Computing*. Fuzzy Sets and Systems, doi:10.1016/j.fss.2007.08.014. 2007.
- [15] Dürrbaum A. Scherm W. Sommer H.: *Systemidentifikation mittels Glaubenskalkül*. at 7, 350-349. 2008.
- [16] Gerland P. Schulte H. Kroll A.: *Wahrscheinlichkeitsbasierte globale Situationserkennung bei komplexen Systemen*. Proc., 18 Workshop Computational-Intelligence Universitätsverlag: Karlsruhe. 2008.
- [17] Mehta N. Natarajan S. Tadepalli P.: *Transfer in Variable-Reward Hierarchical reinforcement*. Mach Learn DOI 10.1007/s10994-008-5061-y. 2008.
- [18] Kotsiantis S.B. Zaharakis I.D. Pintelas P.E.: *Machine Learning: a Review of Classification and Combining Techniques*. Artif Intell Rev DOI 10.1007/s10462-007-9052-3. 2007.
- [19] Schreiber L. Sommer H.: *Heuristiken, die Algorithmen einer natrlichen Logik*. Bericht des Instituts für Mechanik, Universität Kassel: Kassel. 2008.
- [20] Northover M. Kourie D.G. Boake A. Gruner S. Northover A.: *Towards a Philosophy of Software Development: 40 Years after the Birth of Software Engineering*. J. Gen. Philos. Sci. DOI 10.1007/s108838-008-9068-7. 2008.
- [21] Landau R.H.: *Resource Letter CP-2: Computational Physics*. Am.J.Phys.76(4,5),296-306. 2008.
Letters to the Editor, Landaus Reply,Am.J.Phys.76(9) (2008) 797.

Parameterselektion für komplexe Modellierungsaufgaben der Wasserwirtschaft

T. Bartz-Beielstein, T. Zimmer und W. Konen

Fakultät für Informatik und Ingenieurwissenschaften

Fachhochschule Köln

Tel.: +49 2261 8196-0

Fax: +49 2261 8196-15

E-Mail: {thomas.bartz-beielstein | wolfgang.konen}@fh-koeln.de

Zusammenfassung

Die Simulation komplexer technischer Vorgänge kann mit unterschiedlichen Modellen durchgeführt werden. Meistens erfolgt die Modellauswahl und anschließende Parametrisierung des Modells nach subjektiven Kriterien. Wir demonstrieren, wie eine systematische Vorgehensweise zur Modellselektion mittels SPO (sequentieller Parameteroptimierung) diesen Vorgang objektivieren kann. Um die einzelnen Schritte nachvollziehbar darzustellen, basiert unsere Darstellung auf einem Beispiel aus der Praxis: der Modellierung von Füllständen in Regenüberlaufbecken.

1 Einleitung

Verfahren der *Computational Intelligence* (CI) haben in den letzten Jahrzehnten einen festen Platz im Repertoire der Simulations- und Optimierungsverfahren erobert. *Neuronale Netze* (NN), *genetische Algorithmen* (GA) oder *genetisches Programmieren* (GP) – um nur einige CI-Verfahren zu nennen – werden tagtäglich in der industriellen Praxis eingesetzt. Die Entscheidung für den Einsatz dieser Verfahren wird dabei häufig von subjektiven Kriterien beeinflusst. Als Kriterien können hier z.B. die Vertrautheit mit einem Verfahren, das persönliche Interesse an beispielsweise bioanalogen Verfahren oder Vorträge auf Fachkonferenzen genannt werden. Nach unserer Erfahrung spielt bei dieser Auswahl der Zufall eine große Rolle. Nur in seltenen Fällen erfolgt die Auswahl allein nach objektiven Maßstäben. Außerdem ist es nicht ausreichend, das beste Verfahren auszuwählen, da der Auswahlprozess selbst gewisse Eigenschaften besitzen sollte¹. Es ist also nicht alleine ausreichend, das beste Verfahren zu finden; der Findungsprozess selbst muss optimal gestaltet werden. Dieser Artikel beschäftigt sich zentral mit diesem Findungsprozess und stellt ein Verfahren (SPO) vor, das diesen Prozess verbessern kann.

¹So beschreiben Beyth-Marom et al. [1] diese Situation wie folgt:

... According to the most general normative model, a person facing a decision should (a) list relevant action alternatives, (b) identify possible consequences of those actions, (c) assess the probability of each consequence occurring (if each action were undertaken), (d) establish the relative importance (value or utility) of each consequence, and (e) integrate these values and probabilities to identify the most attractive course of action, following by a defensive decision rule. People who follow these steps are said to behave in a *rational* way. People who do so effectively (e.g., they have accurate probability estimates, they get good courses of action into their list of possibilities) are said to behave *optimally*. Thus, if one does not execute these steps optimally, one can be rational without being very effective at getting what one wants.

Uns ist bewusst, dass es einen 100% objektiven Auswahlprozess allein schon aus Zeitgründen nicht geben kann. Stehen aber bereits einzelne Verfahren zur Verfügung und ist ein effizienter Vergleich gewünscht, so kann das in diesem Artikel beschriebene Verfahren hilfreich sein. In diesem Artikel werden drei Zielsetzungen behandelt. Wir betrachten die Modellauswahl und die Bestimmung guter Parameter für ein einzelnes Modell. Diese Zielsetzungen lassen sich mittels eines Meta-Modells in einem gemeinsamen statistischen Framework behandeln. Eine mögliche und bereits in vielen Bereichen erfolgreich angewandte Technik ist die *sequentielle Parameteroptimierung* (SPO) [2, 3, 4]. Als dritte Zielsetzung untersuchen wir verschiedene Einstellung des SPO Meta-Modells.

Zunächst stellen wir in Abschnitt 2 ein Problem aus der Wasserwirtschaft dar. In Abschnitt 3 wird die SPO als Meta-Modellierungsansatz vorgestellt. Abschnitt 4 stellt die verschiedenen Prognoseansätze vor und beschreibt die zugehörigen Schritte zur Datenvor- und -nachbearbeitung. Ebenfalls in diesem Abschnitt wird dargelegt, wie die Parameter des von uns favorisierten Modells eingestellt wurden. Abschnitt 5 beschreibt die Experimente, die in Abschnitt 6 diskutiert werden. Im letzten Abschnitt geben wir eine kurze Zusammenfassung.

2 Füllstandsprognose in Regenüberlaufbecken

In dieser Arbeit vergleichen wir systematisch verschiedene Modellansätze für eine Anwendung aus der Wasserwirtschaft. Dabei steht nicht nur der Vergleich verschiedener Modellierungsansätze im Vordergrund, sondern vor allem die Beschreibung der Verbesserungspotentiale, die sich mit einer gezielten Vorgehensweise erreichen lassen².

Der Vergleich von Modellen zur Prognose der Füllstände in *Regenüberlaufbecken* (RÜB) aufgrund von Regeneintrag und Bodenbeschaffenheit war Gegenstand unserer Untersuchungen. Im Forschungsprojekt KANNST (KANalNetz-Steuerung), aus das von uns benutzte Datenmaterial stammt, wird die Modellierung und Prognose von Füllstandshöhen in RÜB auf Basis einzelner Regenmessungen untersucht [6]. Kanalnetze werden nach dem heutigen Stand der Technik meistens als unregelmäßige Systeme betrieben, bei denen z.B. am Ablauf von Regenüberlaufbecken feste Drosselmengen eingestellt werden [7, 8]. Im Vordergrund einer Weiterentwicklung der Kanalnetzbewirtschaftung steht der Gewässerschutz. In jüngster Zeit werden die hierfür erforderlichen vernetzten Regelungssysteme im Rahmen des Projektes KANNST [6] unter Nutzung moderner Messtechnik und neuer CI-Verfahren entwickelt. Eine vorausschauende Prognose der Füllstände aufgrund von Regeneintrag und Bodenbeschaffenheit ist für diese Aufgaben unerlässlich. Die in [9] dargestellten Ergebnisse basierten auf den mit dem Simulator SWMM [10] erzeugten Regendaten und Füllständen. Hingegen beruhen die in dieser Arbeit präsentierten Ergebnisse auf empirisch erhobenen Daten. Die Anforderungen an die zugrunde liegenden Modellierungsansätze werden dadurch drastisch erhöht, denn die Daten weisen ein intermittierendes und burstartiges Verhalten auf, s. Bild 1. Dieses stellt rückgekoppelte Modelle vor starke Probleme. Wir untersuchen, wie verschiedene Prognosemodelle mit solchen Inputdaten arbeiten können. Unterschiedliche Ansätze wie *neuronale Netze* (NN), FIR-Modelle, *Nonlinear AutoRegressive models with eXogenous inputs* (NARX) [11], *Echo*

²Während in [5] die Prognosemodelle im Vordergrund der Analyse standen, untersuchen wir in diesem Artikel zusätzlich die SPO.

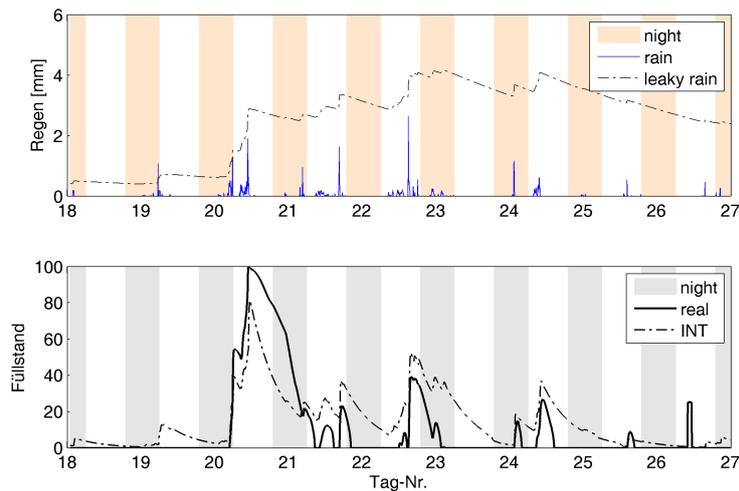


Bild 1: Ausschnitt aus den Kanalnetzdaten. Oben: Die Regendaten haben einen stark burstartigen Charakter. Der 'leaky rain' ist eine gleitende Integration der Regenaktivität. Unten: Die zugehörigen Füllstände im RÜB. Tag Nr. 18 entspricht dem Datum 16.05.2006.

State Networks (ESN) [12], Differentialgleichungen (ODE) oder die Modellierung mit Integralgleichungen (INT) werden dabei von uns eingesetzt.

3 Statistische Meta-Modellierung

Wir unterscheiden exogene Faktoren von endogenen Parametern. Endogene Parameter ändern sich während einer Untersuchung (z.B. Gewichte bei NN während des Lernvorgangs), exogene Faktoren bleiben hingegen konstant (z.B. die Anzahl der Neuronen). Exogene Faktoren werden im weiteren Verlauf unserer Untersuchung betrachtet, endogene Parameter sind nicht Gegenstand unserer Untersuchung. Die exogenen Faktoren werden im Folgenden als *Modellfaktoren* bezeichnet. Zusätzliche Parameter, wie z.B. der Beobachtungszeitraum, sind problemspezifisch. Diese werden im Folgenden als *Problemfaktoren* bezeichnet. Ein Modelldesign beschreibt die Gesamtheit aller modellspezifischen Faktoren, ein Problemdesign beschreibt die problemspezifischen Faktoren. Ein wichtiges Ziel in der Simulation ist die Bestimmung eines optimalen Modelldesigns für ein (oder sogar mehrere) gegebene Problemdesigns. Wir betrachten daher eine typische Situation der Meta-Modellierung, siehe Bild 2. Um eine Vergleichbarkeit zu gewährleisten und zu erreichen, dass jeder Prognoseansatz nach Möglichkeit mit seinem besten Modelldesign operiert, nutzen wir für alle Verfahren die SPO [3, 4], um die bestmögliche Faktoreinstellungen systematisch zu bestimmen. Neu ist in dieser Arbeit auch, dass SPO erstmalig für die Optimierung von ODE-Modellen und Integralgleichungsmodellen (INT) eingesetzt wird. Diese Modelle bieten das Potential, komplexe Anwendungszusammenhänge zu modellieren, die von rein datengetriebenen Verfahren in der Regel nicht erfasst werden. Traditionell haben sie aber das Problem, dass eine größere Anzahl von Parametern einzustellen sind, um quantitativ gute Übereinstimmung zu erzielen. Hier bietet SPO einen systematischen Ansatz. Ziel unserer Untersuchungen ist festzustellen, ob SPO auch im Fall von ODE und INT gute Optimierungsergebnisse liefert.

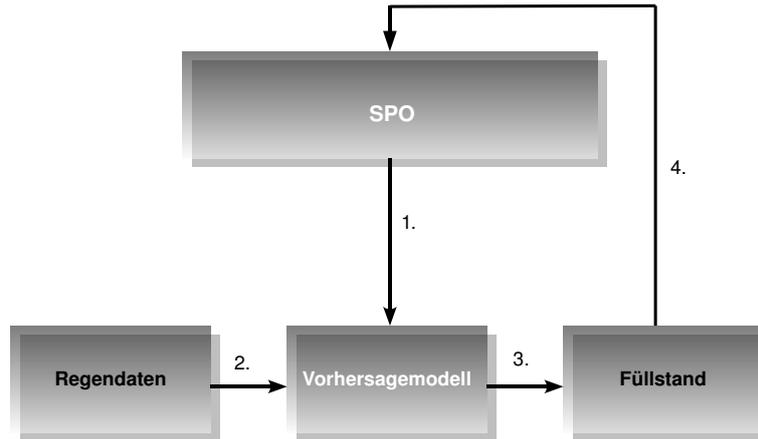


Bild 2: Datenfluss in der SPO Meta-Modellierung. 1. SPO generiert Einstellungen für das Vorhersagemodell (z.B. ein NN). Mit den Regendaten (2.) berechnet das Vorhersagemodell die zugehörigen Füllstände im RÜB (3.) und den Vorhersagefehler. Mit dem Vorhersagefehler (4.) berechnet SPO ein Meta-Modell und generiert weitere Einstellungen für das Vorhersagemodell.

4 Material und Methoden

4.1 Daten und Zielfunktion

Die Regendaten wurden minütlich ab dem 28.04.2006 über einen Zeitraum von 108 Tagen erhoben, so dass insgesamt $n = 155.521$ Datensätze zur Verfügung standen. Basierend auf den gemessenen Regendaten r_t ($t = 1, 2, \dots, n$), in den Abbildungen mit „rain“ bezeichnet, sollen die realen Füllstandshöhen in den Becken vorhergesagt werden. Die realen Werte y_t werden mit „real“ und die von unseren Modellen vorhergesagten Werte \hat{y}_t werden mit dem Modellkürzel (z.B. INT) bezeichnet. Die minütlichen Regenmengen wurden zunächst in 5minütigen Intervallen zusammengefasst, die Füllstände in 5minütigen Intervallen gemittelt. Bekannt sind somit r_t und y_t , zu bestimmen ist ein Modell zur Berechnung von \hat{y}_t , so dass der Fehler zwischen y_t und \hat{y}_t möglichst gering ist^{3 4}. Als Gütemaß der Prognosen betrachten wir den sog. *Mean Square Error* (MSE): $\frac{1}{n} \sum_t^n (y_t - \hat{y}_t)^2$ und den *Root Mean Square Error* (RMSE): $\sqrt{\frac{1}{n} \sum_t^n (y_t - \hat{y}_t)^2}$.

³Es ist zu betonen, dass die hier vorgestellten Modelle (bis auf NARX) alle r_t als einzigen zeitveränderlichen Input benutzen.

⁴Auf den ersten Blick mag es erscheinen, dass die Prognose $r_t \rightarrow \hat{y}_t$ gar keine Prognose darstellt, da der Prognosehorizont 0 ist. Jedoch ist die wahre Aufgabe eines Modells, aus den Regeneinträgen bis zum aktuellen Zeitpunkt t eine Prognose über die Bodenbeschaffenheit und deren Auswirkung auf den Füllstand abzugeben. Des Weiteren führt der bis zum Zeitpunkt t gemessene Regeneintrag erst 30-60 min später zu einer Füllstandsänderung, ein Zeitfenster, welches sinnvoll zu Steuerungsmaßnahmen in Kanalnetzen genutzt werden kann.

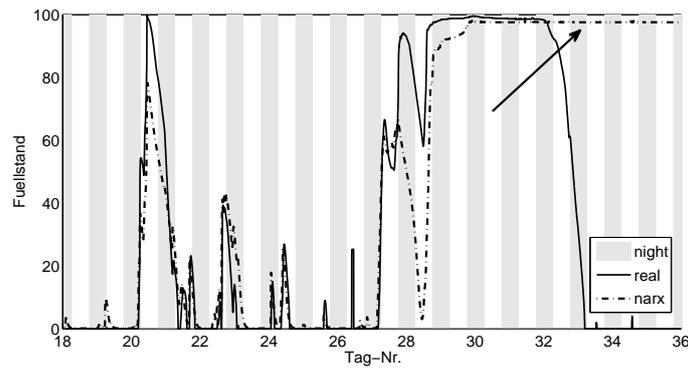


Bild 3: NARX Modellierungsfehler. Der Pfeil kennzeichnet den Bereich, in dem das NARX-Modell nicht „zurück schwingt“, sondern einen konstant hohen Füllstand (größer als 90%) vorhersagt. Dieser Fehler trat systematisch in jeder NARX Simulation auf.

4.2 Prognoseverfahren

Im Rahmen unserer Analysen haben wir sechs verschiedene Prognoseverfahren betrachtet: NN, ESN, NARX, FIR, ODE und INT. Neuronale Netze finden häufig Anwendung, um komplexe Beziehungen zwischen Ein- und Ausgabegrößen zu modellieren [9]. Dabei ist die zugrunde liegende Abbildungsvorschrift nicht genau bekannt, sondern wird vom NN gelernt. Echo State Networks sind rekurrente neuronale Netze, die ein sog. Reservoir rückgekoppelter Neuronen nutzen, um komplexe dynamische Signale zu generieren [12]. NN und ESN wurden zwar untersucht, aber werden hier nicht weiter dargestellt, da sie im Vergleich mit den anderen Verfahren deutlich schlechtere Ergebnisse lieferten. Der in Bild 3 dargestellte Fehler trat bei NN und ESN bereits nach kurzer Zeit auf und ließ sich nicht durch eine Änderung der Einstellungen der Modellfaktoren ändern. Filter mit endlicher Impulsantwort, engl.: *finite impulse response filter* (FIR-Filter), werden häufig in der Signalverarbeitung eingesetzt. NARX Modelle werden in der Zeitreihenmodellierung und -analyse eingesetzt. Der aktuelle Wert der Zeitreihe wird durch Werte aus der Vergangenheit und zusätzlich aus aktuellen und vergangenen Werten der exogenen Reihe bestimmt [11]. Wir betrachten zudem ein gewöhnliche Differentialgleichungssystem, engl. *ordinary differential equations* (ODE), um die wichtigsten Kausalzusammenhänge des dynamischen Systems Regen-Böden-RÜB zu modellieren. Nachteilig an diesem einfachen ODE-Modell ist, dass es an exponentielles Abklingverhalten des Füllstandes gebunden ist. Dies entspricht nicht den realen Verläufen, weil die Laufzeiten des Wassers durch verschiedene Bodenschichten natürlich unterschiedliche Verzögerungen haben. Deshalb formen wir das ODE-Modell in ein vergleichbares Integralgleichungsmodell (INT) um, bei dem das Abklingverhalten durch Modifikation geeigneter Faltungskerne anpassbar ist.

4.3 SPO-basierte Meta-Modellierung

Die FIR-, NARX-, ODE- und INT-Modelle besitzen zwischen zwei und dreizehn Parametern. In vielen Situationen wird versucht, manuell, d.h. durch Probieren, günstige Parametereinstellungen für das Modell zu finden. Die sog. *one-factor-at-a-time* (OAT) Vorgehensweise findet dabei häufig Anwendung. Allerdings wächst die Anzahl der mög-

lichen Einstellungen mit der Anzahl der Faktoren exponentiell⁵. Daher besitzt die OAT-Vorgehensweise in vielerlei Hinsicht große Nachteile im Vergleich zu einer systematischen Vorgehensweise. Im Bereich der Simulation wurde dies von Kleijnen ausführlich dargestellt [14]. Wir setzten die SPO [4] ein. Diese kombiniert Ansätze aus der klassischen Regressions- und Varianzanalyse mit modernen statistischen Verfahren wie Kriging [15, 16]. SPO ist eine umfassende, auf einem rein experimentellen Ansatz beruhende Methode zur Analyse und Verbesserung deterministischer und stochastischer Simulations- und Optimierungsverfahren [2]. Ein eingangs kleiner Stichprobenumfang mit wenigen Wiederholungen wird im Laufe der SPO vergrößert. Zur Bestimmung neuer Stichproben und der Anpassung der Anzahl der Wiederholungen fließen die bisher gewonnenen Informationen ein, so dass das Modell sequenziell verbessert und die Aussagen immer sicherer werden. Die sequenzielle Vorgehensweise ermöglicht ein Lernen der zugrunde liegenden Abhängigkeiten. SPO kann zudem automatisiert ablaufen und ist für die numerische Simulation und Optimierung praktisch konkurrenzlos. Uns ist zur Zeit kein vergleichbares Verfahren bekannt, das einem Anwender mit geringem Aufwand die Anpassung, den Vergleich und die Analyse verschiedener Modelle ermöglicht. SPO kann direkt zur Optimierung komplexer Probleme eingesetzt werden und wird von den Anwendern, aber auch in der empirisch orientierten Grundlagenforschung an stochastischen Suchverfahren stark nachgefragt⁶. Zudem wird eine frei verfügbare *SPO Toolbox* (SPOT) momentan entwickelt. Mittels SPOT gelang in der hier vorliegenden Arbeit z.B. die Analyse, welcher der Faktoren den größten Einfluss auf die Vorhersagegüte hat, so dass Faktoren mit nur geringem Einfluss im Laufe der SPO-Analyse nicht weiter betrachtet werden müssen.

5 Experimente und Resultate

Es wurden drei ausführliche Studien durchgeführt. Zunächst wurde die Vorhersagegüte auf dem gesamten Datensatz untersucht. Im zweiten Schritt wurde der Datensatz partitioniert, um Training- und Testdaten zur Verfügung zu haben. Im dritten Schritt wurde die Spezifikation des SPO Meta-Modells untersucht.

Für das FIR-Modell mussten fünf Faktoren angepasst werden. Der erste Faktor (s) modelliert die Verdunstung des Niederschlags. Die Gewichte, die nach einer gewissen Verzögerung mit wachsendem t exponentiell abnehmen, benötigen die Einstellung der Verzögerung (d), der Länge des exponentiellen Abklingens (l) und des Abklingfaktors A .

⁵Dieses grundlegende Problem ist in der Literatur auch als „Fluch der Dimensionen“ (engl. *curse of dimensionality* [13]) bekannt: Im \mathbb{R}^1 reichen 100 Punkte, um das Einheitsintervall im Abstand 0.01 zu belegen. Im \mathbb{R}^{10} sind dafür 10^{20} Punkte erforderlich. Ein Einheitswürfel im \mathbb{R}^{10} ist in diesem Sinne um das 10^{18} -fache größer als im \mathbb{R}^1 .

⁶SPO wurde beispielsweise in den folgenden Anwendungsfeldern eingesetzt: Maschinenbau: Temperierbohrungen; Luft- und Raumfahrt: Tragflächenoptimierung; Simulation und Optimierung: Fahrstuhlsteuerung; Algorithm Engineering: Graphenalgorithmen; Computational Intelligence: Algorithmische Chemie; Verfahrenstechnik: Entwurf von Destillationsanlagen; Wirtschaftswissenschaft: Modellierung eines Bodenmarktes; Statistik: Selektionsverfahren für Partikelschwarm Verfahren; Informatik: Threshold Selektion und Schrittweitensteuerung für Evolutionsstrategien, Analyse und Anwendung von Partikelschwarm Verfahren; Numerische Mathematik: Vergleich/Analyse klassischer und moderner Optimierungsalgorithmen; Logistik: Touroptimierungs- und Torzuordnungsprobleme; Bioinformatik: Optimierung von Hidden-Markov-Modellen.

Eine Übersicht mit Literaturstellen und eine kurze Einführung in SPO werden in [17] gegeben.

Schließlich muss die Skalierung s_2 angepasst werden. Im NARX-Modell sind zwei Faktoren anzupassen: Die Anzahl der Verzögerungen (delay-states, d) und die Anzahl der Neuronen n . Das INT-Modell besitzt 13 zu optimierenden Faktoren und für das ODE-Modell waren sechs Faktoren zu bestimmen. Diese bilden eine Teilmenge der Faktoren des INT-Modells. Um die Übersichtlichkeit zu wahren, haben wir die einzelnen Faktoren in Tab. 1 zusammenfassend dargestellt.

5.1 Erste Untersuchung: Vergleich der Verfahren auf dem gesamten Datensatz

Um einen ersten Vergleich der Prognosegüte der verschiedenen Verfahren zu erhalten, haben wir in diesen Experimenten den gesamten Datensatz (108 Tage) zugrunde gelegt. In diesem Abschnitt beschreiben wir exemplarisch, wie SPO-basiert verbesserte Modellfaktoreinstellungen bestimmt werden können. Diese Vorgehensweise wurde für alle betrachteten Modelle benutzt und wird anhand des INT-Modells (da dies die meisten Faktoren besitzt) im Folgenden skizziert⁷.

I. Vorstudien zur Bestimmung der ROI. Ausgangspunkt waren in Vorstudien [18] ermittelte gute Einstellungen. Diese ersten Einstellungen basierten auf Plausibilitätsüberlegungen. Es wurde eine geringe Anzahl von Testläufen (ohne Designüberlegungen zu berücksichtigen) durchgeführt. Anschließend wurde systematisch ein experimentelles Design entworfen, in dem Intervallgrenzen großzügig gesetzt wurden. Diese Intervallgrenzen werden in der Literatur als *region of interest* (ROI) bezeichnet. Bereits nach drei SPO Runden wurde eine Verbesserung des erwarteten RMSE von 39.25 auf 10.95 erreicht, Überschreitungen des zulässigen Bereichs wurden dabei bewusst in Kauf genommen, so dass SPO einen möglichst großen Bereich explorieren konnte. Nicht zulässige Lösungen erhielten einen Funktionswert, der signifikant über dem in den Vorstudien gefundenen schlechtesten Wert lag⁸.

II. Screening. Im diesem Schritt, der in der Literatur auch als *screening* bezeichnet wird, bestimmten wir die wichtigsten Parameter. Dazu wurde die Laufzeit auf 500 Berechnungen des INT Modells beschränkt, da bereits nach einer kurzen Anfangszeit die wichtigsten Parameter feststehen sollten. Die SPO Analyse zeigte, dass der Parameter α_L den größten Einfluss auf die Vorhersagegüte hat (Bild 5). In diesem Stadium können Ausreißer, die eine valide Modellbildung nicht zulassen, erkannt werden, so dass eine Veränderung der ROI durchgeführt werden kann.

Die baumbasierte Analyse in Bild 4 bestätigt diese Ergebnisse. Sie zeigt, dass der beste Wert wird für $\alpha_L < 0.00357935$ und $3.85311 < \Delta$ erzielt wird.

Die Analysen legen nahe, dass eine Anpassung der ROI für den Faktor α_L bessere Ergebnisse produziert, da dieser mit Abstand den größten Einfluss auf die Vorhersagegüte hat. Im nächsten Schritt beschreiben wir, wie eine Anpassung der ROI durchgeführt werden kann: Für $\alpha_L = 0.0009618$ wurde der beste Funktionswert gefunden. Die Betrachtung des Regressionbaums (Abb. 4) legt nahe, den Wert für α_L kleiner als 0.00357935 und den Wert für Δ größer als 3.85311 zu wählen. Mit der so veränderten ROI werden nun weitere SPO Läufe durchgeführt, wobei die

⁷Eine detaillierte Darstellung dieser Modelle findet sich in [5], generelle Erläuterungen zur SPO-Methodik mit weiteren Beispielen sind in [4] dargelegt.

⁸Es wird ein Minimierungsproblem betrachtet.

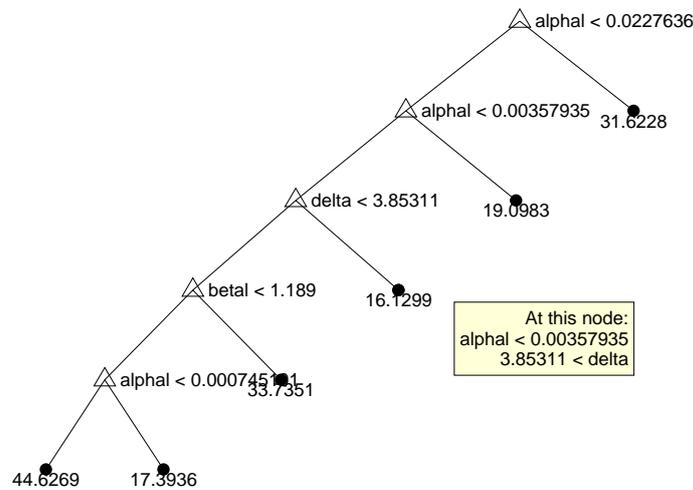


Bild 4: Regressionsbaum. Die Bedingung an jedem Knoten gilt für den linken Teilbaum. Der Faktor mit dem größten Effekt steht an der Wurzel. Der Faktor α_L hat anscheinend den größten Effekt.

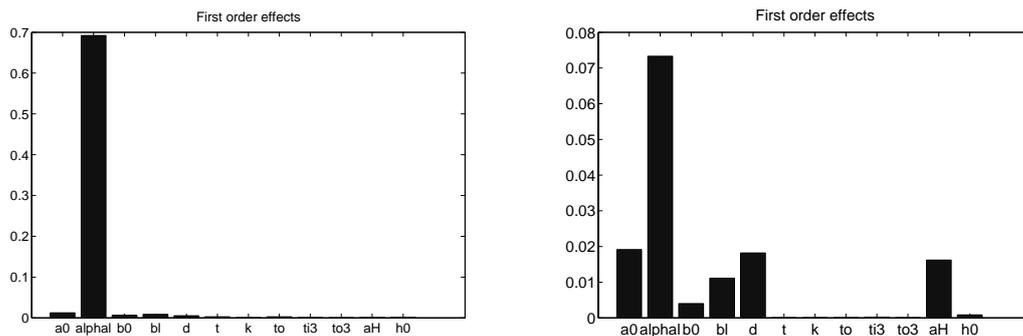


Bild 5: Links: Effekte erster Ordnung (Haupteffekte ohne Interaktionen) nach 500 Auswertungen des INT Modells. Der Faktor α_L hat anscheinend den größten Effekt. Rechts: Haupteffekte im zweiten Schritt der Modellierung. Dadurch, dass passenden Werte für α_L gefunden werden konnten, kommt es zu weniger Ausreißern und die anderen Faktoren besitzen einen größeren Einfluss.

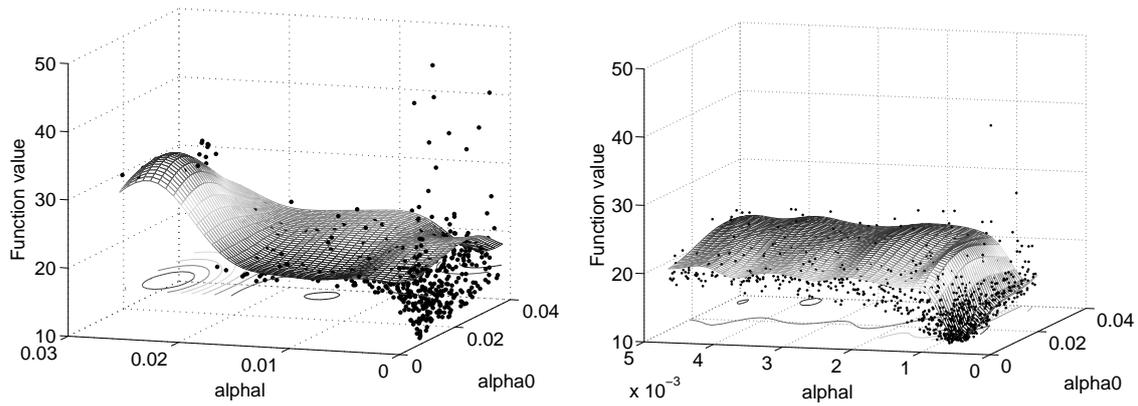


Bild 6: Links: Interaktion der Faktoren α_L und α nach 500 Auswertungen des INT Modells. Die Punkte stellen die Auswertungen dar, sie zeigen starke Fluktuationen im Zielwert (aufgrund der Variation der anderen, hier nicht gezeigten Parameter). Die Fläche ist die SPO-Approximation an die Punkte, die die Basis für weitere Auswertungen ist. Der Faktor α_L hat anscheinend den größten Effekt, kleinere α_L Werte führen zur Verbesserung, aber auch zu höherer Varianz. Rechts: In einer späteren Phase der SPO-Optimierung kann durch Einschränkung der ROI eine wesentliche Reduktion der Varianz erzielt werden, die schwarzen Punkte liegen dichter beieinander.

Anzahl n der Simulatoreufrufe (bzw. Aufrufe des INT-Modells) auf 1000 vergrößert wird. Aus unseren bisherigen Experimenten tragen Werte, die sehr viel größer als 1000 sind, nicht zu einer Verbesserung der Modellfaktoren bei. Im Gegenteil, es können sogar gegenteilige Effekte bei einer Kriging-basierten Modellierung auftreten [19]. Zusätzlich wird durch Bild 6 offensichtlich, dass die Varianz auf diesem Intervall für α_L deutlich zunimmt. Eine ähnliche Betrachtung für Δ zeigte, dass größere Δ Werte vermutlich zu besseren Ergebnissen führen werden, allerdings im Gegensatz zu α_L ohne Veränderung der Varianz.

III. Modellierung und Optimierung. Nachdem in Schritt II von den ursprünglich zwölf Faktoren sechs als signifikant erkannt wurden, nämlich α_L , α , β , β_L , Δ und α_H , beschränkten wir die Suche im weiteren Verlauf auf diese Parameter. Die restlichen Parameter beließen wir auf den bislang besten gefundenen Werten. Mit den so eingestellten Parametern erzielten wir nach 1000 Simulationen den Bestwert RMSE=9.49, siehe Tab. 2.

Der Rechenaufwand bei der unter I.-III. dargestellten Vorgehensweise liegt im Stundenbereich für die FIR-, ODE- und INT-Modelle. Für die NARX-Modellierung war der Aufwand um den Faktor 10 größer, da in jedem Lauf das NARX-Netz trainiert werden musste.

Interessant ist in diesem Zusammenhang auch die Fragestellung, wie weit die von SPO gefundenen Faktoreinstellungen von den manuell bestimmten Werten abweichen, d.h. ist der Anwender auch in der Lage, ohne SPO günstige Einstellungen zu finden und welche Faktoren werden manuell völlig falsch eingestellt? Deshalb stellen wir in Tab. 2 drei verschiedene Messungen vor: Erstens haben wir in einem vom Anwender vorgegebenen Bereich (jeweils kleinster und größter zulässiger Wert) 100 zufällige Einstellungen der Modellfaktoren generiert und die zu jeder Einstellung gehörenden Vorhersagefehler berechnet. Zweitens wurde eine vom Anwender nach manuellem Tuning der Modellfaktoren gefundene Einstellung simuliert. Drittens wurde mit einem Budget von $n = 1000$ Modellierungsläufen eine Optimierung der einzelnen Verfahren mittels SPO durchgeführt.

Tabelle 1: Die 13 Parameter des INT-Vorhersagemodells mit den vom Anwender vorgegebenen zulässigen Bereichen. Das ODE-Modell benutzt daraus nur die 6 Parameter α , β , τ_{rain} , Δ , α_L , β_L , die in der Tabelle in hellerem Grau unterlegt sind.

Parameter	Symbol	manuell	Best SPO	Bereich SPO
Abklingkonstante Füllstand (Filter g)	α	0.0054	0.00845722	[0, 0.02]
Abklingkonstante Filter h	α_H	0.0135	0.309797	{0 ... 1}
Abklingkonstante 'leaky rain'	α_L	0.0015	0.000883692	{0 ... 0.0022}
Einkopplung Regen in Füllstand	β	7.0	6.33486	{0 ... 10}
Einkopplung Regen in 'leaky rain'	β_L	0.375	0.638762	{0 ... 2}
Einkopplung K -Term in Füllstand	h_0	0.5	6.87478	{0 ... 10}
Schwelle für 'leaky rain'	Δ	2.2	7.46989	{0 ... 10}
Flankensteilheit aller Filter	κ	1	1.17136	{0 ... 200}
Zeitverzögerung Füllstand zu Regen	τ_{rain}	12	3.82426	{0 ... 20}
Startzeitpunkt Filter h	τ_{in3}	0	0.618184	{0 ... 5}
Endzeitpunkt Filter h	τ_{out3}	80	54.0925	{0 ... 500}
Endzeitpunkt Filter g	τ_{out}	80	323.975	{0 ... 500}
RMSE		12.723	9.48588	

Tabelle 2: Überblick über die Prognosegüte der einzelnen Verfahren mit und ohne SPO auf dem gesamten Datensatz (108 Tage). Das beste Verfahren in jeder Spalte ist weiß unterlegt.

Verfahren	$\langle RMSE \rangle$ zufällige Parametersets	RMSE manuelles Parameterset	RMSE optimiertes Parameterset
FIR	25.42	25.57	20.10
NARX	85.22	75.80	38.15
ODE	39.25	13.60	9.99
INT	31.75	12.72	9.49

Diese Werte sind in Tab. 1 exemplarisch für das INT Modell dargestellt. Die von SPO nach bestmöglicher Einstellung der Modellparameter erzielten RMSE-Werte sind in der vierten Spalte von Tab. 2 dargestellt.

Für die SPO Läufe wurden 5 Wiederholungen berechnet, da NARX ein stochastisches Verfahren ist. Die oben dargestellten Ergebnisse zeigen für das NARX Modell ein eher ernüchterndes Bild: Im Mittel wurde ein RMSE von 35.50 erzielt. Die mit dem NARX Modell erzielten Werte weisen eine hohe Varianz auf. Aber: Das Minimum liegt bei 12.35 und ist durchaus mit den anderen Verfahren vergleichbar. Allerdings ist dies ein zufälliger Wert, der von der Initialisierung des Zufallszahlengenerators (Seed) abhängt.

5.2 Zweite Untersuchung: Generalisierbarkeit der Ergebnisse auf neuen Daten

Die zweite Serie von Experimenten wurde durchgeführt, um festzustellen, ob die gefundenen Einstellungen sich auf neue Daten übertragen lassen. Dazu wurden die Daten in Trainings- und Testdaten unterteilt. Die erste Hälfte, also 54 Tage, wurde zum Lernen benutzt. Der Fehler wurde dann auf den restlichen Daten bestimmt. Es wurden wiederum die in Abschnitt 5.1 beschriebenen Modellfaktoren optimiert. Tabelle 3 stellt die Ergebnisse zusammen. Es mag zunächst verwundern, dass beim ODE- und INT-Modell die Fehler auf den Trainingsdaten fast immer höher sind als auf den Testdaten. Dies liegt daran, dass der Testzeitraum (Mitte Juni - Mitte August) weniger Niederschläge enthielt als der Trainingszeitraum (Mitte April - Mitte Juni) und damit an trockenen Tagen leichter vorhersagbar war. Korrigiert man aber den Testzeitraum auf gleiche Regenintensität wie

Tabelle 3: Vergleich der Generalisierbarkeit der einzelnen Verfahren mit und ohne SPO-Optimierung. Die ersten beiden Spalten zeigen den RMSE auf dem Trainingszeitraum (Tage 0...53), die letzten beiden Spalten den RMSE auf dem Testzeitraum (Tage 54...107). Die Zahlen in geschweiften Klammern sind die Testfehler hochgerechnet auf gleiche Regenintensität wie im Trainingszeitraum (siehe Abschnitt 5.2).

Verfahren	RMSE Train (manuell)	RMSE Train (mit SPO)	RMSE Test (manuell)	RMSE Test (mit SPO)
FIR	33.45	20.05	13.73 {19.61}	13.91 {19.86}
NARX	50.86	17.25	20.38 {29.10}	9.91 {14.15}
ODE	13.15	11.03	12.18 {17.40}	8.14 {11.62}
INT	14.99	10.82	9.47 {13.53}	7.71 {11.01}

den Trainingszeitraum (Faktor 1.428 = 30/21 Regentage), so zeigen die Zahlen in geschweiften Klammern, dass Trainings- und Testfehler jeweils gleichauf liegen. Es findet also durch die Modelle ODE und INT kein „Auswendiglernen“ statt.

5.3 Dritte Untersuchung: Wahl der initialen Designgröße

Zu Beginn einer SPO wird das zu optimierende Modell $M \in \mathcal{M}$ mehrfach, z.B. $k = 10$ mal, ausgewertet. Anschließend wird ein SPO-Metamodell $M^* \in \mathcal{M}^*$, z.B. ein stochastisches Prozessmodell oder ein Regressionsbaum, bestimmt, um weitere, verbesserte Einstellungen für M vorherzusagen. Um Zeit und Kosten zu sparen, ist es wünschenswert, k möglichst klein zu wählen. Eine untere Schranke ist durch das Metamodell M^* und durch die Anzahl der zu optimierenden Faktoren n im Modell M gegeben. So benötigt ein stochastisches Prozessmodell mit linearem Regressionsterm (also ein Polynom erster Ordnung, siehe [15, 20]) $k = n + 1$ initiale Experimente. Ein quadratisches Metamodell benötigt bereits $k = \frac{1}{2}(n + 1)(n + 2)$ initiale Experimente. Sollen, wie im INT-Modell $n = 13$ verschiedene Faktoren bestimmt werden, dann müssen $k = 91$ initiale Faktoreinstellungen ausgewertet werden. Dies ist in einigen industriellen Anwendungen nicht möglich.

Doch dies ist nicht das einzige Problem. Wir betrachten zudem ein „Henne-Ei Problem“, da die Wahl des optimalen Metamodells (z.B. linear oder quadratisch) von modellinhärenten Strukturen beeinflusst wird[21]: Gibt es keinen linearen Zusammenhang in den Daten (oder dem den Daten zugrunde liegenden Prozess), dann ist auch ein lineares Modell ungeeignet. Zu Beginn eines SPO Laufs liegen allerdings keine Daten vor – diese sollen ja gerade durch SPO möglichst optimal erzeugt werden. Letztendlich ist die Wahl einer passenden initialen Designgröße n nur durch Erfahrungen des Anwenders zu bestimmen. Liegen keine Erfahrungen vor, dann können vorexperimentelle Testläufe erste Hinweise geben.

Eine formale Beschreibung der SPO ist in Alg. 1 dargestellt. Um Floor- oder Ceiling-Effekte zu vermeiden, muss eine geeignete Problem Instanz ausgewählt werden. Es muss sichergestellt werden, dass diese nicht zu leicht, aber auch nicht zu schwierig ist. In diesen beiden Fällen sind die Ergebnisse der Vergleiche der Vorhersagemodelle bedeutungslos. Wir benutzen, abhängig vom verwendeten Meta-Modell ($M^* \in \mathcal{M}^*$), unterschiedliche Initialdesigns wie *Latin Hypercube Designs* (LHD) oder fraktionelle faktorielle Designs. Nachdem das Vorhersagemodell ($M \in \mathcal{M}$) mit den k initialen Faktoreinstellungen gelaufen ist, wird das Meta-Modell M^* berechnet, um weitere, interessante Designpunkte zu

Algorithm 1 Sequentielle Parameter Optimierung

```
1: procedure SPO( $\mathcal{A}_d, \mathcal{P}_d$ ) ▷ Algorithmus und Problem Design
2:   Select  $p \in \mathcal{P}_d$  and set  $t = 0$  ▷ Probleminstanz
3:    $X_A(t) = \{x^{(1)}, x^{(2)}, \dots, x^{(k)}\}$  ▷ Sample  $k$  initiale Punkte, z.B. mit LHS
4:   repeat
5:      $y_j^{(i)} = Y_j(x^{(i)}, p) \forall x^{(i)} \in X_A(t)$  and  $j = 1, \dots, r(t)$  ▷ Fitnessauswertung
6:      $\bar{Y}^{(i)}(t) = \sum_{j=1}^{r(t)} y_j^{(i)}(t) / r(t)$  ▷ Statistik (Mittelwert) für den  $i$ -ten Punkt
7:      $x_b$  with  $b = \arg \min_i(\bar{y}^{(i)})$  ▷ Bestimmung des besten Punktes
8:      $Y(\cdot) = \mathcal{F}(\beta, \cdot) + \mathcal{Z}(\cdot)$  ▷ Meta-Modell (hier: Kriging)
9:      $X_S = \{x^{(k+1)}, \dots, x^{(k+s)}\}$  ▷ Erzeuge  $s$  neue Samples,  $s \gg k$ 
10:     $y(x^{(i)}), i = 1, \dots, k + s$  ▷ Vorhersage der Fitness mittels Meta-Modell
11:     $I(x^{(i)})$  for  $i = 1, \dots, s + k$  ▷ Bestimme erwartete Verbesserung, cf. [16]
12:     $X_A(t + 1) = X_A(t) \cup \{x^{(k+i)}\}_{i=1}^m$  ▷ Füge  $m$  Punkte mit dem größten  $I(\cdot)$  dem
    Design hinzu
13:    if  $x_b(t) = x_b(t + 1)$  then
14:       $r(t + 1) = 2r(t)$  ▷ Anpassung der Wiederholungen
15:    end if
16:     $t = t + 1; k = k + m$  ▷ Zähler hochsetzen
17:  until Budget exhausted
18: end procedure
```

bestimmen. Weitere m Punkte mit der größten erwarteten Verbesserung werden dem Design hinzugefügt. Hierbei sollte m im Vergleich zu s relativ klein sein. Die Update-Regel für die Anpassung der Anzahl der Wiederholungen ist nur erforderlich bei stochastischen Verfahren wie z.B. NN. Bei den deterministischen Verfahren wird nur eine Auswertung benötigt ($r(t) = r = 1$). Letztendlich gibt es je nach Optimierungsziel unterschiedliche Terminierungsbedingungen. In der dritten Untersuchung habe wir die Größe des Initialdesigns bei konstantem Budget (500 Berechnung des Vorhersagemodells M) systematisch variiert. Diese Versuchsserie wurde zudem für unterschiedliche Meta-Modelle M^* durchgeführt. Im Einzelnen wurden ein Polynom nullter Ordnung (in der Literatur auch als ordinary Kriging bezeichnet), ein Polynom erster und ein Polynom zweiter Ordnung benutzt. Zusätzlich wurde ein baumbasiertes Meta-Modell untersucht. Die Modelle wurden mit `regpoly0`, `regpoly1`, `regpoly2` und `tree` bezeichnet. Die Ergebnisse sind graphisch in Bild 7 dargestellt.

6 Diskussion

6.1 Vergleich der Prognosemodelle (M)

Es wurde deutlich, dass Ansätze wie NN, ESN, FIR oder NARX mit den Daten gar nicht oder nur teilweise zurecht kommen. Im Vergleich zu den NN-Modellen lassen sich die FIR-Modelle sehr schnell und einfach implementieren [9]. Die Kernroutine besteht aus ca. 10 Zeilen Programmcode. Die Berechnung der Vorhersagewerte ist entsprechend schnell, FIR-Modelle sind zudem sehr stabil. Interessant ist, dass die FIR-Modelle bessere Ergebnisse erzielen als alle anderen Modelle, wenn die Faktoreinstellungen zufällig (allerdings aus einer vom Anwender als sinnvoll beschriebenen ROI) gewählt werden, siehe Tab. 2. Stellt man jedoch die Faktoren ein, z.B. indem man Erfahrungen aus Test-

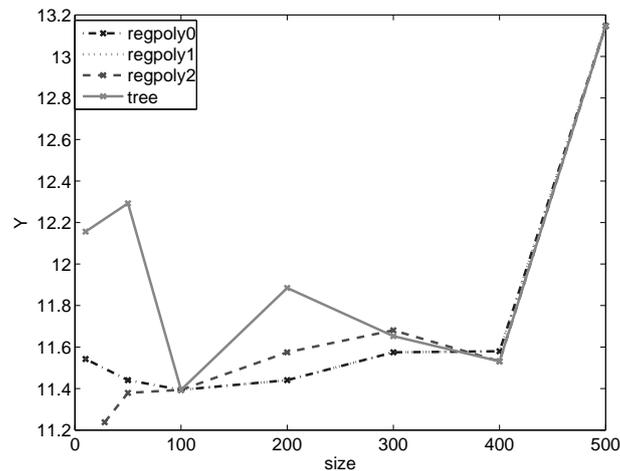


Bild 7: Auswirkungen unterschiedlicher initialer Designgrößen auf das SPO Ergebnis.

läufen heranzieht oder SPO anwendet, dann verlieren FIR-Modelle im Vergleich zu ODE und INT eindeutig. Damit liefern die FIR-Modelle im Portfolio eine gute Basis für einen Vergleich mit komplexeren Modellen (NN, NARX, ODE und INT). Modelle, die nicht in der Lage sind, die FIR Resultate zu übertreffen, dürfen ohne Weiteres als ungeeignet bezeichnet werden.

In der Anfangsphase der Simulation, z.B. während der ersten 25 Tage (siehe Bild 3), erzielten NARX-Modelle einen guten Vorhersagefehler. Das Modell ging aber dann in einen nahezu gesättigten Zustand (ca. 90% Füllstand) über, der konstant bis zum Ende der Simulation prognostiziert wurde. Ein ähnliches Verhalten zeigte auch ESN. Dieser Fehler trat systematisch auf, so dass wir NARX Modelle für diese Art von Daten als nicht geeignet halten. Zudem erfordern NARX-Modelle durch den Trainingsaufwand eine signifikant längere Laufzeit als alle anderen von uns betrachteten Modelle. Sicherlich gibt es effektivere Verfahren für Verbesserung der Faktoren des NARX Modells. Die von uns gewählte SPO Modellierung ist für zwei Faktoren überdimensioniert. SPO findet normalerweise Anwendung, falls drei oder mehr Faktoren zu optimieren sind. Eine sinnvolle Obergrenze liegt bei 20 Faktoren. Aber auch Modelle mit mehreren hundert oder gar tausend Faktoren können nach einem vorher durchgeführten Screening, z.B. *sequential bifurcation* [19], mit SPO optimiert werden. Um eine Vergleichbarkeit mit den anderen Techniken zu gewährleisten, haben wir uns auch für das NARX-Modell für die Standard-SPO Vorgehensweise entschieden. Jedes der von uns betrachteten Modelle (FIR, NARX, ODE und INT) wurde mit der gleichen Systematik optimiert. Die modellbasierten Ansätze ODE und INT zeigen hier deutliche Vorteile. Bereits mit den manuell, vor dem SPO-Einsatz gefundenen Parameter erreicht man mit RMSE 13.60 bzw. 12.72 eine starke Verbesserung (61% bzw. 64%) gegenüber FIR und NARX. Der Einsatz von SPO ist wichtig, denn diese bereits guten Werte konnten durch SPO nochmals um 26% gesteigert werden (s. Tab. 2).

6.2 Vergleich der Meta-Modelle (M^*)

Der Modellentwurfsprozess wurde durch systematische Experimente mittels SPO zielgerichtet gesteuert. Da SPO selbst verschiedene Modelle und Designs zur Verfügung

stellt, ist eine Analyse der Abhängigkeit der Ergebnisse der Meta-Modellierung von dem verwendeten Meta-Modell und Design von großem Interesse. Die Meta-Modelle aus der Klasse der Kriging-Modelle (also `regpoly0`, `regpoly1`, `regpoly2`) zeigten ein ähnliches Verhalten. Die besten Ergebnisse wurden mit kleinen Initialdesigns erzielt, siehe Bild 7. Für das baumbasierte Modell ist die Situation nicht so eindeutig: Ein Trend ist hier nicht zu erkennen. Die folgende Aussage trifft wiederum auf alle Meta-Modelltypen (Kriging und Regressionebäume) zu: Die von SPO verwendete sequentielle Vorgehensweise ist von Vorteil für die Optimierung. Für eine initiale Designgröße von $k = 500$ ⁹ wurde das schlechteste Ergebnis erzielt.

7 Zusammenfassung

Zeitreihen mit intermittierender Aktivität treten in zahlreichen Modellierungs- und Prognoseanwendungen der Automatisierungstechnik auf und sind schwierig zu prognostizieren. Wir betrachteten exemplarisch einen Anwendungsfall aus der Wasserwirtschaft. Es liegt nicht an ungünstig gewählten Parametern, dass Modelle wie FIR oder NARX eine schlechtere Performance zeigen. Mit hoher Wahrscheinlichkeit sind FIR und NARX für diese Modellierungsaufgabe nicht gut geeignet. Die bereits recht guten manuellen Ergebnisse für ODE und INT (Faktor 2 bis 6 besserer RMSE als FIR bzw. NARX) konnten durch SPO nochmals um 30-40% gesteigert werden (s. Tabelle 2). Durch die SPO erschließen sich oftmals überraschende Einsichten über die Modellierung sowie Ansätze, wie sich möglicherweise ein verbessertes Modell bauen läßt.

Die SPO selbst zeigte sich als robustes Optimierungsverfahren, das mit einer kleinen Samplezahl gestartet werden sollte. Anschließend sollten weitere Punkte hinzugenommen werden. Das Meta-Modell ermöglicht eine signifikante Verbesserung der Optimierung. Dabei ist die Wahl des Meta-Modells als nicht sehr kritisch zu bewerten. Das baumbasierte Modell zeigte größere Schwankungen in der Optimierungsqualität, ist aber ca. doppelt so schnell zu berechnen wie die Kriging Modelle. Es kann zusätzlich mit kategorischen Daten umgehen – dies ist für viele Modelle ein entscheidender Vorteil. Insgesamt kann der SPO-basierte Findungsprozess als erfolgreich bezeichnet werden. SPO konnte die manuell gefundenen Modelldesigns signifikant verbessern. Aus den Tab. 2 und 2 ist aber auch ersichtlich, dass das vom Entscheider (Experten, Anwender) bevorzugte Modell (INT) mit dem von SPO gefundenen Modell überstimmt. Lediglich wurde von SPO eine bessere Einstellung der Modellfaktoren gefunden.

Danksagung:Für die Bereitstellung von Daten und Fotos aus dem KANNST-Projekt danken wir Prof. Dr. Michael Bongards und Dipl.-Ing. Tanja Hilmer. – Teile dieser Arbeit wurden durch die FH Köln im Rahmen des Forschungsschwerpunktes COSA gefördert.

Literatur

- [1] Beyth-Marom, R.; Fischhoff, B.; Quadrel, M.; Furby, L.: Teaching decision making to adolescents: A critical review. In: *Teaching Desision Making to Adolescents* (Baron, J.; Brown, R., Hg.), S. 19–60. Mahwah, NY: Erlbaum. 1991.

⁹Zur Erinnerung: Insgesamt stand ein Budget von 500 Modellberechnungen zur Verfügung. Dieses war somit bereits ohne Meta-Modellbildung ausgeschöpft.

- [2] Bartz-Beielstein, T.; Parsopoulos, K. E.; Vrahatis, M. N.: Design and analysis of optimization algorithms using computational statistics. *Applied Numerical Analysis and Computational Mathematics (ANACM)* 1 (2004) 2, S. 413–433.
- [3] Bartz-Beielstein, T.; Lasarczyk, C.; Preuß, M.: Sequential Parameter Optimization. In: *Proceedings 2005 Congress on Evolutionary Computation (CEC'05), Edinburgh, Scotland* (McKay, B.; et al., Hg.), Bd. 1, S. 773–780. Piscataway NJ: IEEE Press. 2005.
- [4] Bartz-Beielstein, T.: *Experimental Research in Evolutionary Computation—The New Experimentalism*. Natural Computing Series. Berlin, Heidelberg, New York: Springer. 2006.
- [5] Konen, W.; Zimmer, T.; Bartz-Beielstein, T.: Optimierte Modellierung von Füllständen in Regenüberlaufbecken mittels CI-basierter Parameterselektion. *at – Automatisierungstechnik* (2008). Eingereicht.
- [6] Bongards, M.: Online-Konzentrationsmessung in Kanalnetzen – Technik und Betriebsergebnisse. Techn. Ber., Cologne University of Applied Sciences. 2007.
- [7] Grüning, H.: Abflusssteuerung - quo vadis? *KA Korrespondenz Abwasser, Abfall* 55 (2008), S. 358–364.
- [8] Hilmer, T.; Bongards, M.: Integrierte Steuer- und Regelstrategien für Kanalnetz und Kläranlage. Techn. Ber., Cologne University of Applied Sciences. 2008.
- [9] Bartz-Beielstein, T.; Bongards, M.; Claes, C.; Konen, W.; Westenberger, H.: Datenanalyse und Prozessoptimierung für Kanalnetze und Kläranlagen mit CI-Methoden. In: *Proc. 17th Workshop Computational Intelligence* (Mikut, R.; Reischl, M., Hg.), S. 132–138. Universitätsverlag, Karlsruhe. 2007.
- [10] U.S. Environmental Protection Agency: Storm Water Management Model. www.epa.gov/ednrmrl/models/swmm/index.htm, Online; Stand 16.08.08. 2008.
- [11] Siegelmann, H. T.; Horne, B. G.; Giles, C. L.: Computational capabilities of recurrent NARX neural networks. Techn. Ber. UMIACS-TR-95-12 and CS-TR-3408, University of Maryland. URL citeseer.ist.psu.edu/siegelmann97computational.html. 1995.
- [12] Jaeger, H.: The echo state approach to analysing and training recurrent neural networks. Techn. Ber. 148, Fraunhofer Institute for Autonomous Intelligent Systems (AIS), Sankt Augustin. 2001.
- [13] Bellman, R. E.: *Adaptive Control Processes*. Princeton University Press. 1961.
- [14] Kleijnen, J. P. C.: *Statistical Tools for Simulation Practitioners*. New York NY: Marcel Dekker. 1987.
- [15] Sacks, J.; Welch, W. J.; Mitchell, T. J.; Wynn, H. P.: Design and analysis of computer experiments. *Statistical Science* 4 (1989) 4, S. 409–435.
- [16] Santner, T. J.; Williams, B. J.; Notz, W. I.: *The Design and Analysis of Computer Experiments*. Berlin, Heidelberg, New York: Springer. 2003.
- [17] Bartz-Beielstein, T.; Preuss, M.: Moderne Methoden zur experimentellen Analyse evolutionärer Verfahren. In: *Proc. 16th Workshop Computational Intelligence* (Mikut, R.; Reischl, M., Hg.), S. 25–32. Universitätsverlag, Karlsruhe. 2006.
- [18] Konen, W.; Bartz-Beielstein, T.; Westenberger, H.: Computational Intelligence und Data Mining – Datenanalyse und Prozessoptimierung am Beispiel Kläranlagen. Techn. Ber., Cologne University of Applied Sciences. 2007.
- [19] Kleijnen, J. P. C.: *Design and analysis of simulation experiments*. New York NY: Springer. 2008.
- [20] Lophaven, S.; Nielsen, H.; Søndergaard, J.: Aspects of the Matlab Toolbox DACE. Techn. Ber. IMM-REP-2002-13, Informatics and Mathematical Modelling, Technical University of Denmark, Copenhagen, Denmark. 2002.
- [21] Bartz-Beielstein, T.: Review: Design and Analysis of Simulation Experiments by Jack P. C. Kleijnen. *ICS News* (2008). <http://computing.society.informs.org/newsletter.php>.

Ein einfaches Berechnungsverfahren zu den Navier-Stokesschen Gleichungen – Die Struktur berechenbarer Modelle für komplexe Systeme

M. Busch, B. Schweizer, H. Sommer, V. Sommer

Universität Kassel
Mönchebergstraße 7, 34109 Kassel
hans.sommer@mrt.uni-kassel.de
busch@mks.uni-kassel.de
schweizer@mks.uni-kassel.de

Zusammenfassung

Jeder Naturwissenschaftler wird aus seiner Studienzeit den Sprung erinnern, der in allen Einführungen in die Partiiellen Differentialgleichungen zwischen der linearen und der nichtlinearen Theorie besteht. Während in der linearen Theorie Klassen von Gleichungen untersucht werden können, muss man sich im nichtlinearen Fall auf einzelne Gleichungen beschränken und erhält auf viele Fragen keine Antwort. Obwohl sich die Problemstellungen in beiden Theorien im ersten Augenschein oft nur ganz wenig unterscheiden, offenbart die nichtlineare Theorie, bei genauerem Hinsehen, ihre prinzipielle, nicht nur qualitativ sondern auch quantitativ größere Komplexität.

Da die Probleme der nichtlinearen Theorie sich jedoch in den praktischen Anwendungen immer dringlicher stellen, wurden sie zum Gegenstand eines Seminars des Fachgebietes Mehrkörpersysteme, das über mehrere Semester an der Universität Kassel stattfand. Diskutiert wurden die Fragen:

- Warum sind manche nichtlineare Differentialgleichungen so ungleich schwieriger lösbar als die entsprechenden linearen Gleichungen?
- Lässt sich der Begriff der Komplexität nicht nur quantitativ sondern auch als qualitatives Phänomen verstehen?
- Ist das Phänomen der Komplexität theorieübergreifend oder immer nur jeweils theoriespezifisch verstehbar?
- Welche Hilfsmittel ergeben sich aus den Methoden der Computational Intelligence?

Unsere Ergebnisse, die im vorgelegten Beitrag vorgestellt werden sollen, sind:

- I Komplexität entsteht durch die Koppelung von Schwierigkeiten. Im konkreten Falle der Navier-Stokesschen Gleichungen dadurch, dass die Lösungsschwierigkeiten, die bei linearen Differentialgleichungen zu getrennten Klassen gehören (elliptisch/hyperpolisch), sich verkoppeln und damit die Lösungsmethoden zum jeweils reinen Problemtyp verhindern. Gelingt es, mit einem Trick die Problemschwierigkeiten zu separieren, so können die Navier Stokesschen Gleichungen unter der Verwendung von Aggregationsoperatoren mit Gitterverfahren einfach gelöst werden. Diese Lösungsmethode und die physikalische Rechtfertigung zum verwendeten Trick sollen vorgestellt werden.
- II Das am Beispiel der Navier-Stokesschen Gleichungen gewonnene Verständnis für Komplexität wird dann als allgemeiner Begriffsrahmen gefasst. Zunächst völlig unterschiedliche Problemgebiete, wie die Untersuchung von Schwarmverhalten, die Theorie der Co-Simulation und der Simulator-Kopplung und die Quantenmechanik erscheinen dann, auf Grund der (jedem der Problemfelder eigenen) qualitativen Komplexität, einheitlich in einer hierarchischen Modellbeschreibung.

1 Einleitung

Die Verwendung von Gittermethoden zur Lösung von Differentialgleichungen ermöglicht oft nicht nur ein einfaches Lösungsverfahren, sondern auch eine Veranschaulichung der in den Gleichungen kodierten Gesetzmäßigkeiten. Durch diese Veranschaulichung werden komplizierte Sachverhalte selbst mit Schülern diskutierbar und was noch wesentlich wichtiger ist, es können dadurch die Ursachen dieser Kompliziertheiten ins Blickfeld gebracht werden. Am Beispiel der Navier-Stokesschen Gleichungen wird ausgeführt, wie die Veranschaulichung ihrer Terme zu einem Verständnis führt, das es selbst Schülern (15 Jahre 9. Gymnasialklasse) ermöglicht, eine mathematische Darstellung dieser Terme und schließlich deren Realisierung in einem im C-Code geschriebenen Programm durchzuführen.

Dabei muss jedoch an einer Stelle (mittels der Verwendung von Aggregationsoperatoren) 'gemogelt' werden. Eine genaue Untersuchung dieser 'Mogelei' soll uns einen Zugang zum Verstehen der Komplexität des vorliegenden Problems geben.

Die Ergebnisse, die wir mit dieser Vorgehensweise erhalten, können wie folgt in Worten zusammengefasst werden:

Komplexität ist ein Begriff, der vom Beobachter abhängt: Ein Problem ist komplex, wenn es uns in einer Sprache vorgelegt ist, in der wir alle seine Wirkungen nicht vollständig überblicken können.

Um ein komplexes System numerisch zu erfassen, muss es daher mittels eines Zwei-Sprachen-Modells dargestellt werden, wobei diese unterschiedliche, inkommensurable Begriffswelten darstellen.

Die Struktur, die wir aus dem untersuchten Beispiel abgeleitet haben, können wir als gemeinsame Grundstruktur der verschiedensten, als komplex bewerteten Systeme nachweisen. Typische Phänomene, wie Emergenz, Unvorhersehbarkeit und Unbestimmtheit folgen in direkter Weise aus der Grundstruktur.

Mit dem vorgestellten Beispiel wollen wir aufzeigen, wie mittels Methoden der Computational Intelligence (Aggregationsoperatoren) ein numerisch beherrschbarer Zugang zu sonst unzugänglichen Problemklassen ermöglicht werden kann.

2 Die Navier-Stokesschen Gleichungen

Die Navier-Stokesschen Gleichungen (NS-Gleichungen) lauten für ein kompressibles Fluid mit Strömungsgeschwindigkeit $u(x, t)$, der Dichte $\rho(x, t)$ und den Zähigkeitskoeffizienten η und ζ [WG78]:

$$\frac{\partial u(x, t)}{\partial t} = \mathcal{O}(u(x, t)) = -u(x, t) \frac{\partial u(x, t)}{\partial x} - \frac{1}{\rho(x, t)} \frac{\partial p(x, t)}{\partial x} + \frac{\eta}{\rho(x, t)} \frac{\partial^2 u(x, t)}{\partial x^2} + \frac{(\zeta + \frac{\eta}{3})}{\rho(x, t)} \frac{\partial(\nabla u(x, t))}{\partial x}$$

Eine genaue Spezifikation der Koeffizienten wird hier nicht benötigt, da wir uns auf prinzipielle Aussagen zur Lösungsberechnung beschränken. Um eine einfache Beziehung zur Berechnung des Drucks zu haben, nehmen wir an, dass dieser proportional zur Dichte anwächst.

Bei der Lösung der NS-Gleichungen mittels Operator Splitting wird der Operator $\mathcal{O}(u(x, t))$ in drei Bestandteile zerlegt:

I. Änderung durch Transport:

$$\frac{\partial u(x, t)}{\partial t} = \mathcal{O}_{\mathcal{T}}(u(x, t)) = -u(x, t) \frac{\partial u(x, t)}{\partial x}$$

II. Änderung durch Druck:

$$\frac{\partial u(x, t)}{\partial t} = \mathcal{O}_{\mathcal{P}}(u(x, t)) = -\frac{1}{\rho(x, t)} \frac{\partial p(x, t)}{\partial x}$$

III. Änderung durch Diffusion:

$$\frac{\partial u(x,t)}{\partial t} = \mathcal{O}_{\mathcal{D}}(u(x,t)) = \frac{\eta}{\rho(x,t)} \frac{\partial^2 u(x,t)}{\partial x^2} + \frac{(\zeta + \frac{\eta}{3})}{\rho(x,t)} \frac{\partial(\nabla u(x,t))}{\partial x}$$

In jedem Zeitintervall dt wird dann angenommen, dass im ersten Drittel $t \in [t_0, t_0 + \frac{dt}{3})$ nur die Transportänderung aktiv ist, allerdings in dreifacher Stärke, im zweiten Drittel $t \in [t_0 + \frac{dt}{3}, t_0 + \frac{2dt}{3})$ wirkt die Druckänderung in dreifacher Stärke und im letzten Drittel $t \in [t_0 + \frac{2dt}{3}, t_0 + dt)$ die dreifach verstärkte Diffusionsänderung.

Das Problem bei der Lösung der NS-Gleichungen besteht nun darin, dass die Wirkungen der Operatoren $\mathcal{O}_{\mathcal{T}}$ und $\mathcal{O}_{\mathcal{P}}$ vom anderen Typ sind, als diejenigen des Operators $\mathcal{O}_{\mathcal{D}}$. Die Operatoren $\mathcal{O}_{\mathcal{T}}$ und $\mathcal{O}_{\mathcal{P}}$ haben eine lokale Wirkung, die sich mit einer endlichen Geschwindigkeit durch das ganze Fluid ausbreitet. Ein solches Verhalten ist von linearen hyperbolischen Differentialgleichungen her gut bekannt. Der Operator $\mathcal{O}_{\mathcal{D}}$ repräsentiert dagegen einen Vorgang bei dem die Strömung in ihrer Gesamtheit erfasst wird. Alle Strömungsteile 'kleben' sozusagen zusammen und versuchen ihre Nachbarn in den gleichen Zustand zu überführen. Dieses Verhalten ist von Diffusionsvorgängen, z.B. der Ausbreitung eines Tintenklecks in einem Glas mit Wasser gut bekannt. An den Lösungen elliptischer Differentialgleichungen kann dies studiert werden. Wir finden nur halb so viele Lösungen wie zur entsprechenden hyperbolischen Differentialgleichung, dafür aber sind nun die Lösungen im gesamten Lösungsraum analytisch. Durch die Analytizität werden alle Lösungsgebiete instantan verkoppelt. Die kleinste Änderung an einer Stelle wirkt sofort auf alle anderen Gebiete.

Eine von W. Greiner und H. Stock beschriebene Beobachtung des Verhaltens realer Strömungen, sowie numerische Untersuchungen, die eine sensible Abhängigkeit der Lösungen von den Anfangswerten nachweisen, scheinen jedoch aufzuzeigen, dass das 'Zusammenhalten' der Lösungsteile im Zusammenwirken der Operatoren aufgebrochen wird. Insgesamt entsteht also ein Verhalten, das der Wirkung des Operators $\mathcal{O}_{\mathcal{D}}$ zuwiderläuft und daher dessen Realisierung im Kontext einer Gesamtlösung der NS-Gleichungen, mit einem endlichen Berechnungsverfahren, in Frage stellt.

- **Empirische Beobachtung durch W.Greiner, H.Stock [WG78]:** Hinter dem Punkt, an dem die Grenzschicht abreißt, bildet sich das sogenannte Totwassergebiet mit vielen Wirbeln. Es ist weitgehend **unabhängig von der Hauptströmung**.
- **Numerische Untersuchungen durch Hoaran, Braza, Yertibos, Tzabras von Strömungen hinter Tragflächen ergaben [YH03]:** By performing a very detailed three-dimensional study, it has been found that **the flow forgets its initial conditions**.

Obwohl mit sehr aufwändigen Verfahren Lösungen der NS-Gleichungen gefunden werden können, ist deren Eindeutigkeit wegen der sensiblen Abhängigkeit von den Anfangswerten oder den Zwischenergebnissen nicht gewährleistet. Die NS-Gleichungen können daher nicht als berechenbare Darstellung ihrer Lösungen aufgefasst werden [RA97, Som98, HK02, PC08].

Der in den nächsten Abschnitten vorgeschlagene Lösungsansatz diskutiert eine Möglichkeit zur Auflösung des dargelegten Widerspruchs in der Wirkung der Operatoren $\mathcal{O}_{\mathcal{T}}$, $\mathcal{O}_{\mathcal{P}}$ und $\mathcal{O}_{\mathcal{D}}$.

3 Wirkungsbeschreibung der Operatoren $\mathcal{O}_{\mathcal{T}}$, $\mathcal{O}_{\mathcal{P}}$ und $\mathcal{O}_{\mathcal{D}}$ mittels eines Gitterverfahrens

Die Darstellung der Operatoren erfolgt mit der Strategie, die bei den Gittermethoden üblich ist [FM07, MS08, AS07, XN08, YG08, Kad86, NG99]. Wir stellen uns das Strömungsgebiet vor als einen großen Platz, auf dem sich viele Menschen bewegen. Die Gitterpunkte entsprechen Messstellen, in deren Umgebung

- (a) die Anzahl der Personen bestimmt wird, die sich nahe der Messstelle aufhalten, und
- (b) die Durchschnittsgeschwindigkeit dieser Personen berechnet wird.

Das Gesetz der Massenerhaltung besagt, dass die Gesamtanzahl der Personen nur durch neu das Gebiet betretende Personen oder dieses verlassende geändert werden kann. Die Beachtung des Impulserhaltungssatzes bedingt, dass die Durchschnittsgeschwindigkeit aller Personen ebenfalls nur durch äußere Einflüsse änderbar ist.

Mittels diesen Vorbemerkungen ist nun die Beschreibung der Wirkung der Operatoren \mathcal{O}_T , \mathcal{O}_P und \mathcal{O}_D möglich:

Beschreibung von \mathcal{O}_T :

Bewegen sich Personen von einer Messstelle zu einer Nachbar-Messstelle, so nehmen sie sowohl ihren Anteil an der Dichte als auch ihre mit diesem Dichteanteil gewichtete Geschwindigkeit ins neue Messgebiet mit. Anschaulich ist dieser Vorgang in Bild 1 dargestellt.

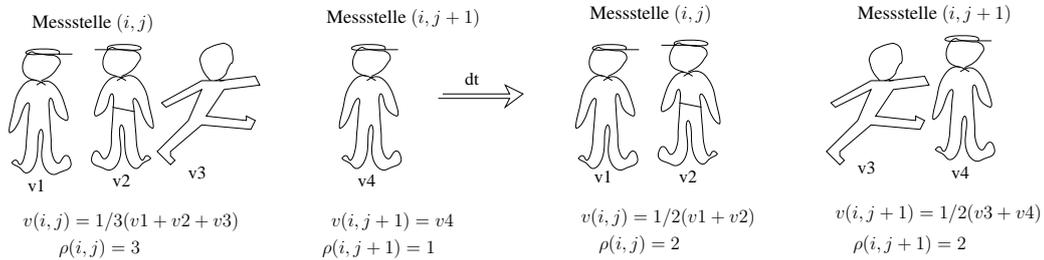


Bild 1: Wirkung durch Transport

Beschreibung von \mathcal{O}_P :

Aus dichten Gebieten streben die Personen in dünner besetzte Gebiete. Der durch diesen Effekt bewirkte Personenfluss ist proportional zur Dichtedifferenz zwischen den Gebieten.

Beschreibung von \mathcal{O}_D :

In jedem Gitterpunkt passen die Personen ihre Geschwindigkeit der Geschwindigkeit der Personen in den Nachbarpunkten an, wobei der Anpassungsdruck, der von einem Gitterpunkt an seine Nachbarpunkte ausgeht, proportional zu seiner Personendichte ist.

Diese Idee kann jedoch nicht, wie dies für die Operatoren \mathcal{O}_T und \mathcal{O}_P möglich war, in lokaler Weise realisiert werden. Der Diffusionsoperator beschreibt einen globalen Zusammenhang und würde zunächst ein Abreißen der Strömung unmöglich machen. Wir verwenden daher ein übergeordnetes Prinzip:

Falls Abreißgebiete oder Vergessenzonen existieren, so wird eine Lösung aus der globalen Gesamtstruktur und nicht mittels lokaler Bedingungen konstruiert.

Im nächsten Abschnitt wird gezeigt, wie dieses Prinzip mit Aggregationsoperatoren in berechenbarer Weise dargestellt werden kann.

Die Operatoren \mathcal{O}_T , \mathcal{O}_P und \mathcal{O}_D könnten natürlich den wahren Bedingungen, die in einem speziellen Fluid herrschen, noch genauer angepasst werden. Wichtig ist jedoch, dass in jedem Fall der Gegensatz zwischen der lokalen Wirkungsweise der Operatoren \mathcal{O}_T und \mathcal{O}_P und dem, das gesamte Fluid erfassenden Operator \mathcal{O}_D auftritt.

4 Aggregationsoperatoren

Die Aggregation der lokalen Entscheidungen zur Bestimmung einer globalen Lösung kann man sich vorstellen als den Prozess, mittels dessen aus den individuellen Bewertungen einer Gruppe von Entscheidern eine Gesamtentscheidung gebildet wird. Operatoren, die diesen Prozess formal darstellen, werden Aggregationsoperatoren genannt.

Um Aggregationsoperatoren zur Berechnung von Lösungen physikalischer Probleme verwenden zu können, muss sichergestellt sein, dass durch sie keine neuen, nicht von der Physik bestimmten Abhängigkeiten eingeführt werden. Sie müssen dem **Objektivitätsprinzip** genügen:

Der mathematische Ansatz zur Beschreibung eines physikalischen Phänomens muss alle Symmetrien erfüllen, die das physikalische Phänomen erfüllt.

Die allgemeinste Definition von Aggregationsoperatoren die R.R. Yager eingeführt hat, erfüllt diese Bedingung. Sie basiert auf den im folgenden gegebenen Bezeichnungen und Definitionen [Yag01, Som95].

Bezeichnungen: $U^{[0,1]} := \{ \langle u_1, u_2, \dots, u_n \rangle \text{ ungeordnete } n\text{-Tupel} \}$
mit Wiederholung von Elementen $u_i \in [0, 1]$
 $\langle u_1, u_2, \dots, u_n \rangle \oplus \langle v_1, v_2, \dots, v_m \rangle := \langle u_1, \dots, u_n, v_1, \dots, v_m \rangle$

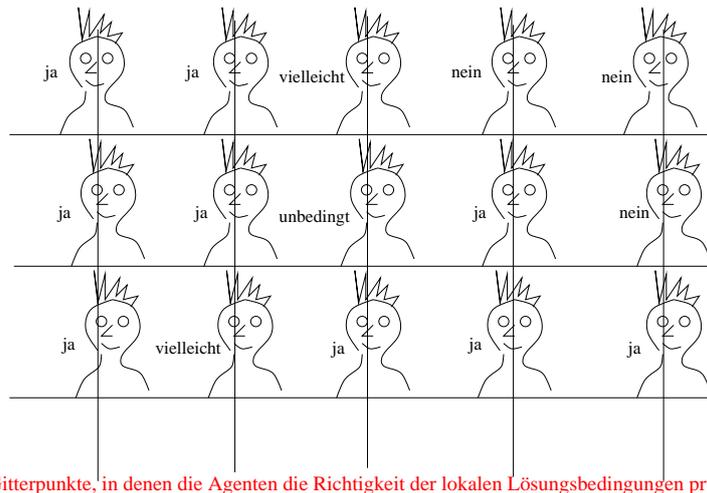


Bild 2: Ausgaben der Agenten

$\langle u_1, u_2, \dots, u_n \rangle \leq \langle v_1, v_2, \dots, v_m \rangle \Leftrightarrow n = m$ und es gibt eine Umordnung

$\Pi : \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ so dass für $i = 1, \dots, n$ gilt: $u_i \leq v_{\Pi(i)}$.

Definition: $A : U^{[0,1]} \rightarrow [0, 1]$ heißt **Aggregationsoperator**: \Leftrightarrow

Für alle $\langle u_1, u_2, \dots, u_n \rangle, \langle v_1, v_2, \dots, v_m \rangle \in U^{[0,1]}$ gilt:

- $\langle u_1, u_2, \dots, u_n \rangle \leq \langle v_1, v_2, \dots, v_m \rangle \Rightarrow A(\langle u_1, u_2, \dots, u_n \rangle) \leq A(\langle v_1, v_2, \dots, v_m \rangle)$
- Es gibt ein $u_0 \in [0, 1]$ (die Stimmenthaltung) mit:
 $A(\langle u_1, u_2, \dots, u_n \rangle \oplus \langle u_0 \rangle) = A(\langle u_1, u_2, \dots, u_n \rangle)$

Definition: Ein Aggregationsoperator A heißt

(a) **und-Operator**, (b) **oder-Operator**, bzw. (c) **assoziativ** falls

zusätzlich für alle $\langle u_1, \dots, u_n \rangle, \langle v_1, \dots, v_m \rangle, \langle w_1, \dots, w_m \rangle \in U^{[0,1]}$ gilt:

- $A(\langle u_1, \dots, u_n \rangle) \geq A(\langle u_1, \dots, u_n \rangle \oplus \langle v_1, \dots, v_m \rangle)$ bzw.
- $A(\langle u_1, \dots, u_n \rangle) \leq A(\langle u_1, \dots, u_n \rangle \oplus \langle v_1, \dots, v_m \rangle)$
- $A(\langle A(\langle u_1, \dots, u_n \rangle \oplus \langle v_1, \dots, v_m \rangle) \oplus \langle w_1, \dots, w_m \rangle) =$
 $A(\langle u_1, \dots, u_n \rangle \oplus \langle A(\langle v_1, \dots, v_m \rangle \oplus \langle w_1, \dots, w_m \rangle) \rangle) =$
 $A(\langle u_1, \dots, u_n \rangle \oplus \langle v_1, \dots, v_m \rangle \oplus \langle w_1, \dots, w_m \rangle)$

SATZ (a): Ein **assoziativer und-Aggregationsoperator** kann nach entsprechender Umskalierung des Bewertungsintervalls $[0, 1]$ mit einer bijektiven, bistetigen, monoton steigenden Skalierungsfunktion $\Psi_{\wedge} : [0, 1] \rightarrow [0, 1]$ dargestellt werden mittels dem **Fuzzy-und-Operator** \wedge der für $a, b \in [0, 1]$ definiert ist durch: $a \wedge b := a \cdot b$ (bzw. $a \wedge b := \min(a, b)$):

$$A(\langle u_1, u_2, \dots, u_n \rangle) := u_1 \wedge u_2 \wedge \dots \wedge u_n$$

(b): Ein **assoziativer oder-Aggregationsoperator** kann nach entsprechender Umskalierung des Bewertungsintervalls $[0, 1]$ mit einer bijektiven, bistetigen, monoton steigenden Skalierungsfunktion $\Psi_{\vee} : [0, 1] \rightarrow [0, 1]$ dargestellt werden mittels dem **Fuzzy-oder-Operator** \vee der für $a, b \in [0, 1]$ definiert ist durch: $a \vee b := 1 - (1 - a) \cdot (1 - b)$ (bzw. $a \vee b := \max(a, b)$):

$$A(\langle u_1, u_2, \dots, u_n \rangle) := u_1 \vee u_2 \vee \dots \vee u_n$$

Definition: Ein Aggregationsoperator heißt **verstärkend**, falls eine Bestätigung mit $u > u_0$ (bzw. eine Verneinung mit $v < u_0$) durch mehrere Agenten stärker wirkt als nur diejenige eines einzelnen Agenten:

$$A(\langle u, u \rangle) > A(\langle u \rangle) > A(\langle u_0 \rangle) \quad (\text{bzw. } A(\langle v, v \rangle) < A(\langle v \rangle))$$

SATZ: Ein **assoziativer, verstärkender Aggregationsoperator** kann (nach entsprechender Umskalierung des Bewertungsintervalls $[0, 1]$ mit bijektiven, bistetigen, monoton steigenden Skalierungsfunktionen $\Psi_{\wedge} :$

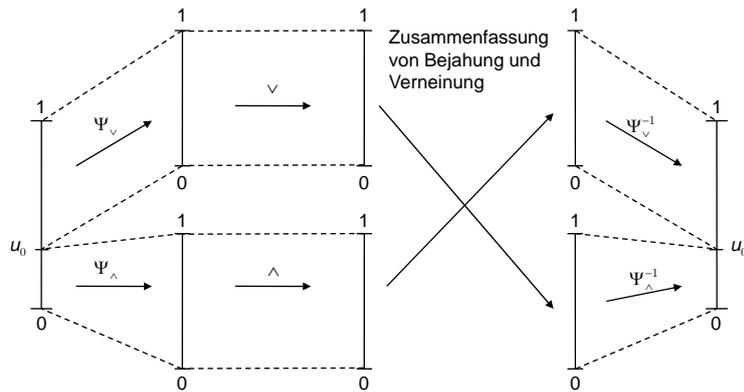


Bild 3: Aggregation

$[0, u_0] \rightarrow [0, 1]$ und $\Psi_\vee : [u_0, 1] \rightarrow [0, 1]$ dargestellt werden) mit den **Fuzzy-Operatoren** \wedge und \vee , die für $a, b \in [0, 1]$ definiert sind durch: $a \wedge b := a \cdot b$, bzw. $a \vee b := 1 - (1 - a) \cdot (1 - b)$.

Möglichkeiten der Zusammenfassung von Bejahung und Verneinung:

- **Starkes Veto:** Wird eine Verneinung ausgesprochen, so werden alle Bejahungen vernachlässigt.
- **Starkes Ja:** Wird eine Bejahung ausgesprochen, so werden alle Verneinungen vernachlässigt.
- **Ausgewogene Entscheidung:** Seien $u_i, i = 1, \dots, n$ Bejahungen oder Verneinungen mit:

$$m_i := \begin{cases} \ln(u_i) & \text{für eine Verneinung} \\ -\ln(1 - u_i) & \text{für eine Bejahung} \end{cases}$$

$$m_{\text{aggregiert}} := \sum_{i=1}^n m_i$$

$$u_{\text{aggregiert}} := \begin{cases} \exp(m_{\text{aggregiert}}) & \text{für } m_{\text{aggregiert}} < 0 \\ 1 - \exp(-m_{\text{aggregiert}}) & \text{für } m_{\text{aggregiert}} \geq 0 \end{cases}$$

Die dargelegte Formalisierung von Aggregationsoperatoren zeigt, dass deren Festlegung nicht willkürlich getroffen werden kann, sondern dass die allgemeinen Prinzipien, die diese erfüllen müssen, sie weitgehend festlegen.

Im Algorithmus, der die Wirkung des Operators $\mathcal{O}_{\mathcal{D}}$ im Programm zur Berechnung der NS-Gleichungen bestimmt, konnte aus Gründen der Rechenzeit keine wirklich globale Aggregation realisiert werden. Es werden Gebiete um die Gitterpunkte erfasst und die Geschwindigkeit in jedem Gitterpunkt so angepasst, dass dieser mit einem möglichst großen Teil dieses Gebiets eine zusammenhängende Einheit bildet.

5 Programmerstellung und Ergebnisse

Anhand der vorgestellten Überlegungen konnten die Prinzipien, die in den NS-Gleichungen kodiert sind, soweit veranschaulicht werden, dass sie von einem Schüler der Gymnasialklasse 9 (Vincent Sommer) innerhalb eines dreiwöchigen Praktikums in Gleichungen für die Strömungsgeschwindigkeit $u(x, t)$ und die Dichte $\rho(x, t)$ umgeschrieben und in ein C-Programm umgesetzt werden konnten. Betreut wurden diese Arbeiten vom ersten Autor, der die Programmierung anleitete und die Strukturierung des Programmaufbaus überwachte.

Abbildung 4 zeigt das Geschwindigkeitsfeld, welches durch das vorgestellte Verfahren berechnet wurde. Dabei strömt das Fluid an einer starren Platte vorbei, wodurch sich hinter der Platte zum einen die Stömungsgeschwindigkeit in der laminaren Hauptströmung erhöht und zum anderen ein Totwassergebiet (Kavität) mit Wirbeln ausbildet. Die Strömungen im Totwassergebiet sind deutlich von der Hauptstömung abgekoppelt.

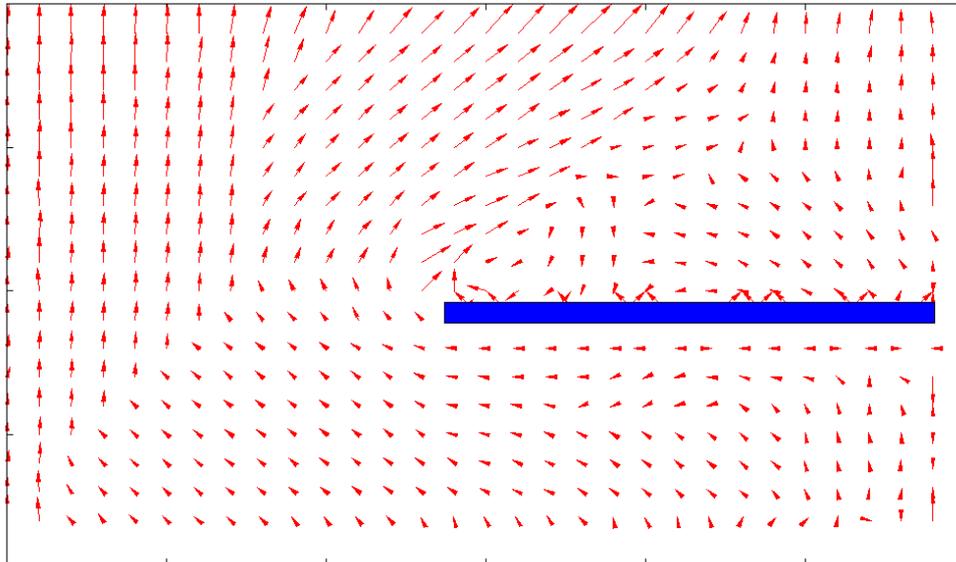


Bild 4: Geschwindigkeitsfeld der Navier-Stokes-Gleichungen

6 Das Sprachebenenmodell zur Beschreibung komplexer Systeme

Unser Vorgehen, mit dem wir zu einem allgemein als sehr komplex eingestuften Problem eine numerisch realisierbare Lösung erhielten, soll nun in größerer Allgemeinheit untersucht werden.

Die wesentlichen Charakteristika desselben sind:

- Zu erkennen, dass in der lokalen Sprache, dem Infinitesimalkalkül, keine Umsetzungsmöglichkeit der Problemstellung in ein Berechnungsverfahren zur Lösungsermittlung möglich ist.
- Daher ist es notwendig, Informationen auch auf einer höheren Sprachebene zu sichern.
- Diese Information der höheren Sprache wird dann später mit einem Aggregationsoperator in das Berechnungsverfahren eingebracht, das sonst im Wesentlichen auf der lokalen Sprachebene abläuft.

Mit einem solchen Zwei-Sprachen-Modell ließ sich der Widerspruch zwischen dem Wirken der Diffusion zur Herstellung globaler Gleichförmigkeit und dem Phänomen des Abreißens der Strömung beseitigen, indem die Gleichförmigkeit nicht mehr in jeder infinitesimalen Umgebung, sondern nur noch bezüglich einem Wahrheitsbegriff gefordert wurde, der mit einem Aggregationsoperator global definiert ist.

Dabei ist zu beachten, dass es zwischen der lokalen Sprache, mit der wir die Wirkung der Operatoren $\mathcal{O}_{\mathcal{T}}$ und $\mathcal{O}_{\mathcal{P}}$ in Algorithmen umsetzen, und der Darstellungssprache von $\mathcal{O}_{\mathcal{D}}$ keine Übersetzung gibt. In der lokalen Sprache war z.B. die Geometrie des Gebiets, das vom Fluid erfüllt wird, nicht present und in der Sprache für $\mathcal{O}_{\mathcal{D}}$ gibt es keine Aussagen zur Kennzeichnung der Stellen, an denen die Strömung abreißt.

Unser Beispiel legt die folgende allgemeine Begriffsbildung nahe:

Definition (Komplexität): Wir bezeichnen ein Problem oder auch ein System als komplex, wenn es von der Sprache aus, in der es uns gegeben ist, von uns nicht vollständig erfasst oder beherrscht werden kann.

Die vorgelegte Definition erfasst den Begriff der Komplexität in subjektiver Weise. Das 'Erfassbare' hängt selbstverständlich vom Erfassenden ab, eine Tatsache die jedem älteren Schachspieler nur zu gut bekannt ist. Beim Spiel gegen jüngere Gegner muss dieser fast immer anerkennen, dass deren Vermögen zur Vor-ausberechnung einige Züge weiter reicht, als die eigenen Fähigkeiten. Um dennoch gegen diese, eigentlich überlegenen Gegner bestehen zu können, muss der ältere Spieler sein besseres Spielverständnis und Strategiewissen einsetzen, die sein Wissen über das Schachspiel auf einer höheren Ebene darstellen. Dass dann auch der ältere Spieler nicht notwendigerweise chancenlos ist, hat schon Botwinnik in seinem zweiten Wettkampf um die Weltmeisterschaft gegen Tal nachgewiesen.

Fordern wir von einem Modell eine Darstellung der Realität, die diese nicht nur in abstrakter Weise erfasst (z.B. mit den Sprachen der nichtkonstruktiven Mathematik), sondern die Fragen, die wir stellen, auch

konstruktiv beantwortbar macht, so führt uns dies zu einer in folgender Weise strukturierten **Modellbildung komplexer Systeme**: 'Das Wissen über ein komplexes System ist auf verschiedenen Sprachebenen zu kodieren, zwischen denen keine generelle Übersetzungsmöglichkeit besteht.'

In diesen Zwei-Sprachen Modellen hat der Vorgang der Emergenz eine exakt definierte Bedeutung: **Emergenz** tritt auf, wenn Effekte, die auf der unteren Sprachebene ablaufen, sich zu einem Begriff konstituieren, der auf der oberen Sprachebene Bedeutung hat.

Das Abreißen der Strömung stellt ein solches Emergenzphänomen dar. Es ist ein Vorgang, der zwar von der unteren Sprachebene aus bewirkt wird, der aber im Beschreibungsmodell erst auf der oberen Sprachebene, im Bezug zur Gesamterscheinung der Strömung, beschreibbar wird. Auf der unteren Sprachebene bleibt dieser Vorgang dagegen unbestimmt.

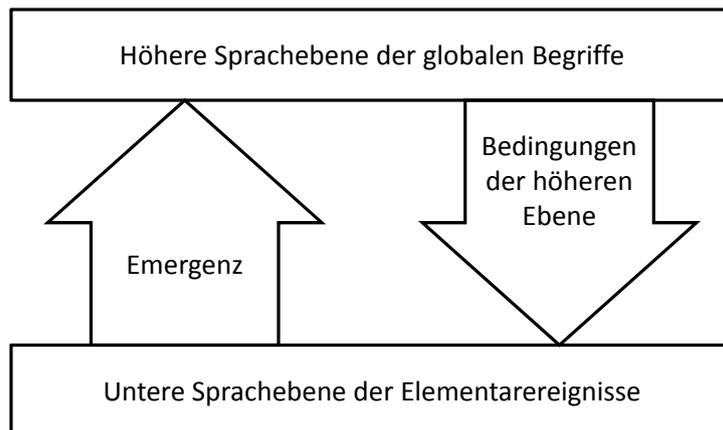


Bild 5: Idee der Sprachebenen

Anhand der folgenden Beispiele wollen wir die Allgemeingültigkeit des dargelegten Schemas zur Modellbildung nachweisen.

6.1 Schwarmverhalten und Schwarmalgorithmen

Große Fisch- und Vogelschwärme bilden ein faszinierendes Naturphänomen. Schwarmalgorithmen orientieren sich an diesem Vorbild. Jedes Tier sucht nach dem Lösungsziel (z.B. einem Gebiet mit hoher Futterkonzentration) und orientiert sich dabei sowohl an seinen Nachbarn (lokale Information), als auch am Verhalten des gesamten Schwarms oder am entfernt fliegenden Leittier (globale Information). Lange glaubte man, dass die Koordinierung des Schwarms unmöglich nur lokal, d.h. durch die Orientierung jedes Tiers an seinen Nachbarn erzeugt sein könnte. Bis neuere Simulationsexperimente diese Sicherheit ins Wanken brachten. Sie zeigten, dass fast alle Phänomene, die beim Verhalten der Schwärme beobachtet werden, rein lokal, ohne zusätzliche globale Orientierung, erklärbar sind. Es stellt sich dann natürlich die Frage: Wie müssten denn Phänomene aussehen, die nur durch eine zusätzliche globale Orientierung hervorgebracht sein können?

Dies ist eine offene Frage und sie ist natürlich für die ganze Theorie der Schwarmalgorithmen von größter Bedeutung [QH07, PV05, LL06, RM07]. Denn wenn es gar kein solches Phänomen gäbe, dann könnten wir keinen Grund für die Effizienz der Schwarmalgorithmen angeben.

Unsere Ausführungen zu den NS-Gleichungen zeigen nun aber, dass es Phänomene gibt, die nicht ausschließlich durch lokale Koordinationen hervorgerufen sein können. Schwarmalgorithmen sind daher nicht allgemein durch lokale Verfahren ersetzbar.

6.2 W. Zureks Modell zum Verständnis der Quantenmechanik

Mit der Nanotechnologie und dem Vordringen der Technik in den atomaren Bereich wird ein Verstehen der Grenze zwischen klassischer Mechanik und Quantenmechanik immer wichtiger. Dabei stehen wir vor dem

Problem, dass die Quantenmechanik, als Theorie der fundamentalen Erscheinungen, nicht unabhängig ist von der klassischen Physik. Zur Definition der Pointer-Zustände, d.h. sowohl zur Fixierung der Ausgangssituation eines Experiments, als auch der Experimentergebnisse benötigt sie die Begriffswelt der klassischen Mechanik (Messproblem). Die Vorstellung, einer Fundierung aller Erscheinungen auf der Kenntnis des Verhaltens der Grundbausteine der Natur, ist damit in Frage gestellt.

Zureks Vorschlag, den er selbst als Quanten-Darwinismus beschreibt, besteht nun darin, alle Erscheinungen in einem Zwei-Sprachen-Modell zu erfassen [Zur07, Zur03, Som07]. Die grundlegenden Erscheinungen sind diejenigen der Quantenwelt, die in deren Operatoren-Sprache beschrieben werden. Uns, als menschlichen Beobachtern, sind jedoch nur diejenigen Ereignisse zugänglich, die in der Lage sind, in vielen Kontexten zu überleben ('die fittesten Zustände'). Diese Zustände konstituieren unser Wissen, das den Gesetzen der klassischen Logik gehorcht. Zureks Modell der Welt ist also mit Zwei-Sprachen gebildet. Unser Zugang zu dieser Welt besteht nur über die obere Sprache (diejenige der klassischen Physik). Die Ereignisse der Quantenwelt haben aber die Möglichkeit zu Ereignissen der oberen Sprache zu emergieren. Dies geschieht beim Messprozess, wenn das Messinstrument in einen Pointer-Zustand übergeht und beim Vorgang der Dekohärenz, der verhindert dass makroskopische Gebilde (wie Katzen) von uns in gemischten Zuständen (gleichzeitig tot und lebendig) vorgefunden werden.

6.3 Co-Simulation / Simulator Coupling

Zur Simulation komplexer Systeme ist häufig eine multidisziplinäre Modellierung notwendig (z.B. Fahrzeugsystemdynamik). Da für verschiedene physikalische Teilgebiete (MKS, Hydraulik, Regelungstechnik, ...) verschiedene Programme mit optimierten Solvoren angeboten werden, können komplexe Gesamtsysteme durch eine Kopplung von Simulationsumgebungen (Subsystemen) abgebildet werden.

Üblicherweise werden die einzelnen Subsysteme durch Systemeingänge u und Systemausgänge y gekoppelt, die im Subsystem (lokal) nicht mehr interpretierbar sind. Der „globale Koppler“ kann jedoch auf höherer Ebene erkennen, wie die Ein- und Ausgänge verbunden sind und kann diese z.B. als Kraft-Weg-Kopplung interpretieren (siehe Abb. 6).

Man erhofft sich nun, die Effizienz der einzelnen Programme (Subsysteme) für die Simulation des Gesamtsystems auszunutzen. In früheren Arbeiten [RK00, MA01] konnte jedoch gezeigt werden, dass selbst bei stabiler Simulation der einzelnen Subsysteme trotzdem die Simulation des Gesamtsystems instabil werden kann. Solche Instabilitäten können nur noch auf höherer (globaler) Ebene erklärt werden, z.B. wenn das Massenverhältnis beider Subsysteme direkt in den Spektralradius und damit in die Stabilität eingeht.

Diese Erkenntnis ist entscheidend für die effiziente Kopplung von Simulationsumgebungen, wenn Stabilisierungstechniken, adaptive Kommunikationsgitter und globale Ordnungssteuerungen einbezogen werden sollen. Weiter kann hieraus abgeleitet werden, dass der häufig für die adaptive Co-Simulation verwendete Solver-Verbund [Bus07] (Steuerung der Kommunikation aller Solver über die variable Schrittweite eines bestimmten Solvers) wegen seiner „Lokalität“ für theoretische Stabilitäts- und Fehleraussagen ungeeignet ist, wenngleich dieser Ansatz für viele Modelle praktikabel und effizient ist.

Vielversprechendere Kopplungsansätze sind in der Erweiterung der äquidistanten aber global gesteuerten „klassischen“ Co-Simulation zu finden, wenn die Kommunikationsschrittweite global adaptiv berechnet und für die Subsysteme vorgegeben wird.

7 Diskussion und Schlussbemerkungen

Vom Standpunkt der klassischen Mechanik aus, ist die eingeführte Verwendung von Aggregationsoperatoren 'unphysikalisch', da ihre Wirkung nicht aus den NS-Gleichungen abgeleitet wurde. Aus dieser Sichtweise haben wir daher 'gemogelt'. Da wir jedoch nur auf Gitterpunkten über Wissen verfügen und uns alle Vorkommnisse, die zwischen diesen stattfinden, unbekannt sind, ist klar, dass das zu lösende Problem von unserer Darstellung nicht vollständig erfasst wird. Wegen der sensiblen Abhängigkeit der Ergebnisse von den Anfangswerten ist es auch nicht möglich, die Gitterpunkte entsprechend dicht zu wählen, um die wahre Lösung zu approximieren.

Schon 1906 hat Ludwig Boltzmann dieses prinzipielle Problem erkannt. Er schreibt (zitiert nach [Wil08]): 'It is the core of my definition that matter must consist of a discrete number of material points. We must

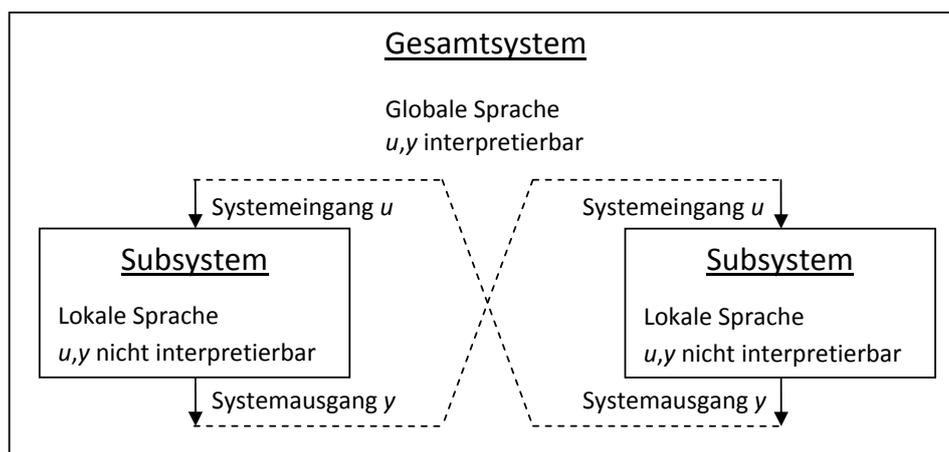


Bild 6: Sprachebenen bei der Systemzerlegung

conceive of it as composed of a finite number of discrete points **if we are to be capable of drawing secure conclusions**. If we allow an absolute continuum, this does not hold anymore.'

Auch unser Vorgehen kann als ein Übergang zu einem berechenbaren Modell interpretiert werden. Um die Kennzeichnung von $u(x, t)$ mittels der NS-Gleichungen numerisch auswerten zu können, gehen wir unter Informationsverlust zunächst zu einer numerisch beherrschbaren Darstellung über (der Gitterdarstellung). Den Informationsverlust dieses Schritts kompensieren wir durch die Forderung: ' $u(x, t)$ soll von einer globalen Sichtweise aus eine sinnvolle (d.h. einfache oder gleichförmige) Struktur haben.'

Diese Forderung entspricht einer in der Literatur häufig dargelegten Interpretation des Prinzips der kleinsten Aktion:

'Durch die Gesetze der Physik wird die empirische Erfahrung so geordnet, dass Prognosen bestmöglichst möglich werden, d.h. so, dass wir eine möglichst einfache Sicht der Realität erhalten'

Die zusätzlich zu den NS-Gleichungen eingeführte Information hat also ebenfalls ihre physikalische Begründung, die jedoch auf einer anderen Argumentationsebene liegt.

Die Darstellung mit einem Zwei-Sprachen-Modell ermöglicht es, physikalische Prinzipien in allgemeinerer Weise einzubringen, als dies über nur eine Sprachebene möglich wäre.

Der Übergang zu höheren Ebenen ist auch bei der häufig verwendeten Co-Simulation zur Kopplung von Simulationsumgebungen notwendig. Dort erzeugen Kopplungsalgorithmen zusätzliche Informationen, die bei der Formulierung als Gesamtsystem nicht auftreten [EL82] (u.a. auch Instabilitäten) und die nur global interpretierbar sind. Hier kann die vorgestellte Aggregation bei Stabilisierungstechniken und Methoden zur globalen Schrittweitensteuerung für Effizienzsteigerungen von Nutzen sein.

Weiter konnte anhand unserer Untersuchung der Navier-Stokes-Gleichungen gezeigt werden, dass es Systeme gibt, die sich einer Modellierung als „Ganzes“ im klassischen Sinne vollständig entziehen. Hier ist eine Systemzerlegung nicht nur wegen der begrenzten Fähigkeit einer Simulationsumgebung oder aus numerischen Gründen notwendig, sondern wegen der Unvereinbarkeit verschiedener mathematischer Betrachtungsweisen. Dabei kann die Aggregation als Kopplung verschiedener Modellbildungen verwendet werden, um überhaupt erst eine Beschreibung des Gesamtsystems zu erhalten.

Zusammenfügen von Information aus verschiedenen Argumentationsebenen erscheint uns als einer der vielversprechendsten Ansätze zur Lösung komplexer Probleme. Mittels der Einführung der verschiedensten Sprachen (Fuzzy-Logik, Glaubenskalküle, Neuronale Netze, Schwarmalgorithmen und der Untersuchung des Zusammenwirkens verschiedener Sprachebenen [RM05]) hat die Computational Intelligence auf diesem Weg bereits wichtige Vorarbeiten geleistet.

Literatur

[AS07] A. STERN, Y. TONG, M. DESBRUN UND J.E. MARSDEN: *Computational electromagnetism*

with variational integrators and discrete differential forms. 2007. arXiv:0707.4470v2.

- [Bus07] BUSCH, M.: *Entwicklung einer SIMPACK-Modelica/Dymola Schnittstelle.* DLR Deutsches Zentrum für Luft- und Raumfahrt e.V., Institut für Robotik/Mechatronik, 2007. Oberpfaffenhofen.
- [EL82] E. LELARASMEE, A. RUEHLI, A. SANGIOVANNI-VINCENTELLI: *The waveform relaxation method for time domain analysis of large scale integrated circuits.* IEEE Trans. on CAD of IC and Syst., Vol. 1, 1982.
- [FM07] F. MILICCHIO, A. DICARLO, A. PAOLUZZI UND V. SHAPIRO: *A codimension-zero approach to discretizing and solving field problems.* Advanced Engineering Informatics, 2007.
- [HK02] H. KELLAY, W.I. GOLDBURG: *Two-dimensional turbulence: a review of some recent experiments.* Reports on Progress in Physics, Vol.65, S.845-894, 2002.
- [Kad86] KADANOFF, L.P.: *On Two Levels.* Physics Today, Vol.39, S.7-9, 1986.
- [LL06] L. LI, Y. YANG, H. PENG: *Parameters identification of chaotic systems via chaotic ant swarm.* Chaos Solitons Fractals, Vol.28, S.1204-1211, 2006.
- [MA01] M. ARNOLD, M. GÜNTHER: *Preconditioned Dynamic Iteration for Coupled Differential-Algebraic Systems.* BIT Numerical Mathematics, 2001.
- [MS08] M. SIPOS, B.G. THOMPSON: *on a grid: The finite-difference time-domain method applied to optics and cloakin.* American Journal of Physics, Vol.76, S.464-459, 2008.
- [NG99] N. GOLDENFELD, L.P. KADANOFF: *Simple Lessons form Complexity.* Science, Vol.184, S.87-89, 1999.
- [PC08] P. CORTET, P. DIRIBARNE, R. MONCHAUX: *Characterization of turbulence in inhomogeneous anisotropic flows.* Physics of Fluids, 2008. arXiv:0805.3092v1.
- [PV05] P. VANDERSCHRAAF, J. MCKENZIE ALEXANDER: *Follow the Leader: Local Interactions with Influence Neighborhoods.* Philosophy of Science, Vol.72, S.86-113, 2005.
- [QH07] Q. HE, L. WANG, B. LIU: *Parameter estimation for chaotic systems by particle swarm optimization.* Chaos, Solitons and Fractals, Vol.34, S.654-661, 2007.
- [RA97] R. ANSORGE, TH. SONAR: *Informationsverlust, abstrakte Entropie und die mathematische Beschreibung des zweiten Hauptsatzes der Thermodynamik.* ZAMM Z.angew.Math,Mech., Vol.77,11, S:803-821., 1997.
- [RK00] R. KÜBLER, W. SCHIEHLEN: *Two methods of simulator coupling.* Mathematical and Computer Modelling of Dynamical Systems, Reihe 6(Nr. 2), 2000.
- [RM05] R. MIKUT, J. JÄCKEL, L. GRÖLL: *Interpretability issues in data-based learning of fuzzy systems.* Fuzzy Sets and Systems, Vol.150, S.179-197, 2005.
- [RM07] R. MACH, F. SCHWEITZER: *Modeling Vortex Swarming In Daphnia.* Bulletin of Mathematical Biology, Vol.69, S.539-562, 2007.
- [Som95] SOMMER, H.: *The Logic in Knowledge Processing Formalisms.* Proceedings EUFIT, Vol.1, S.274-278, 1995. Aachen.
- [Som98] SOMMER, H.: *Eigenbehaviour in Deterministic Systems.* Int. J. Computing Anticipatory Systems, Vol 4, S.235-250, 1998. Inst.Math.Liege.
- [Som07] SOMMER, H.: *Emergence of classical reality from a quantum mechanical background.* Chaos Solitons and Fractals, Vol.75, S.715, 2007. doi: 10.1016/j.chaos.2007.04.002.
- [WG78] W. GREINER, H. STOCK: *Theoretische Physik Bd.2A: Hydrodynamik.* Verlag Harri Deutsch, 1978. Thun Frankfurt a.M.
- [Wil08] WILHOLT, T.: *When realism made a difference: The constitution of matter and its conceptual enigmas in the late 19th century physics.* Studies in History and Philosophy of Modern Physics, Vol.39, S.1-16, 2008.
- [XN08] X.B. NIE, X. SHAN, H. CHEN: *Galilean invariance of lattice Boltzmann models.* 2008. arXiv:0801.2924v1.

- [Yag01] YAGER, R.R.: *Uninorms in Fuzzy Systems Modeling*. Fuzzy Sets and Systems, Vol.122, S.167-175., 2001.
- [YG08] Y. GAN, A. XU, G. ZHANG UND X. YU UND Y. LI: *Two-Dimensional Lattice Boltzmann Model For Compressible Flows With High Mach Number*. Physica A, Vol.387, S.1721-1732, 2008. arXiv:0801.4169.
- [YH03] Y. HOARAN, M. BRAZA, Y. VENTIKOS UND D. FAGHANI UND G. TZABIRAS: *Organized Modes and the Three-Dimensional Transition to Turbulence in the Incompressible Flow Around a NACA0012 Wing*. J.Fluid.Mech., Vol.496, S.63-72., 2003.
- [Zur03] ZUREK, W.H.: *Decoherence, einselection, and the quantum origins of the classical*. Reviews of modern Physics, Vol.75, S.715, 2003. arXiv:quant-ph/0105127v3.
- [Zur07] ZUREK, W.H.: *Relative States and the Environment: Einselection, Envariance, Quantum Darwinism, and the Existential Interpretation*. Quantum Physics, 2007. arXiv:0707.2832v1.

Fuzzy-basiertes System zur Optimierung eines Reglersystems für die Niederhalterkraft beim Tiefziehen

Ulrich Priber, Thomas Päßler

Fraunhofer-Institut für Werkzeugmaschinen und Umformtechnik

09126 Chemnitz, Reichenhainer Str. 88

Tel.: (0371) 5397 1435

Fax: (0371) 5397 1488

E-Mail: ulrich.priber@iwu.fraunhofer.de

Zusammenfassung

Die kraftgeregelten hydraulischen Achsen, mit denen beim Tiefziehen der Blecheinzug am Niederhalter gezielt beeinflusst wird, sind in ihren Einstellparametern (PID und weitere) von den verschiedenen Prozeßbedingungen (Blechdicke, Werkstoff u.a.) und von schwankenden Parametern des Hydrauliksystems (Öltemperatur, Ventileigenschaften) abhängig. Diese schwankenden Einflüsse sind nicht ohne weiteres extern bestimmbar und sollen durch ein selbstlernendes System abgefangen werden.

Übliche Kriterien an das Folgeverhalten der Niederhalterkraft bewerten das Ergebnis eines Hubes. Sie sind die Informationsquelle für das zu entwerfende Fuzzy-System. Es soll durch Parameteradaption schrittweise (von Hub zu Hub) die qualitätsbestimmenden Zielgrößen des Ziehvorganges verbessern bzw. erhalten.

Für die Voruntersuchungen des Regelkreises wird ein Simulink-Modell benutzt, das das Verhalten des Kissenregles in einer Presse weitgehend abbildet. Somit können die Auswirkungen von Parameteränderungen auf das System erprobt werden. Damit besteht über die Einbeziehung von erfahrenen Experten hinaus ein Zugang zum erforderlichen Regelwissen über das System. Die Wirkungsweise des Fuzzy-Systems kann ebenfalls mit der Simulation getestet werden.

Nach den Simulationsuntersuchungen wird das Prinzip in einer Versuchspresse implementiert. Das Ziel des Projektes¹ ist die Verkürzung und Objektivierung des derzeit zeitaufwendigen Einstellens der Ziehkissenregler, das derzeit i.a. nur durch hochqualifizierte Fachkräfte möglich ist.

Das Projekt befindet sich in der Anfangsphase, weshalb zum Workshop das Gesamtergebnis noch nicht vorliegen wird.

1 Problembeschreibung

Gegenstand der Untersuchungen ist das Tiefziehen von Blechen mittels einer Kurbelpresse (Bild 1), bei der das Blech beim Ziehvorgang durch einen Niederhalter geführt wird,

¹„Entwicklung einer Methode für einen selbstoptimierenden Regler für Ziehkissensysteme“, gefördert durch die AiF (15286BR1) über die Europäische Forschungsgesellschaft für Blechverarbeitung e.V. (EFB)

der (i.a. mehrfach) hydraulisch abgestützt wird. Für einen erfolgreichen Ziehvorgang (keine Falten, keine Risse) ist die Einhaltung einer definierten Niederhalterkraft während des Ziehvorganges entscheidend. Die zunächst lagegeregelten Hydraulikachsen werden mit Auftreffen des Stößels kraftgeregelt betrieben. Die optimale Einstellung des Reglers (PID) und des Umschaltregimes ist von einer Reihe von verschiedenen Prozeßbedingungen (Blechdicke, Werkstoff u.a.) und von schwankenden Parametern des Hydrauliksystems (Öltemperatur, Ventileigenschaften) abhängig. Diese schwankenden Einflüsse sind nicht ohne weiteres extern bestimmbar und sollen durch ein selbstlernendes System abgefangen werden (Bild 2). Für dieses Aufgabenfeld haben sich Fuzzy-Ansätze vorteilhaft herausgestellt [2].

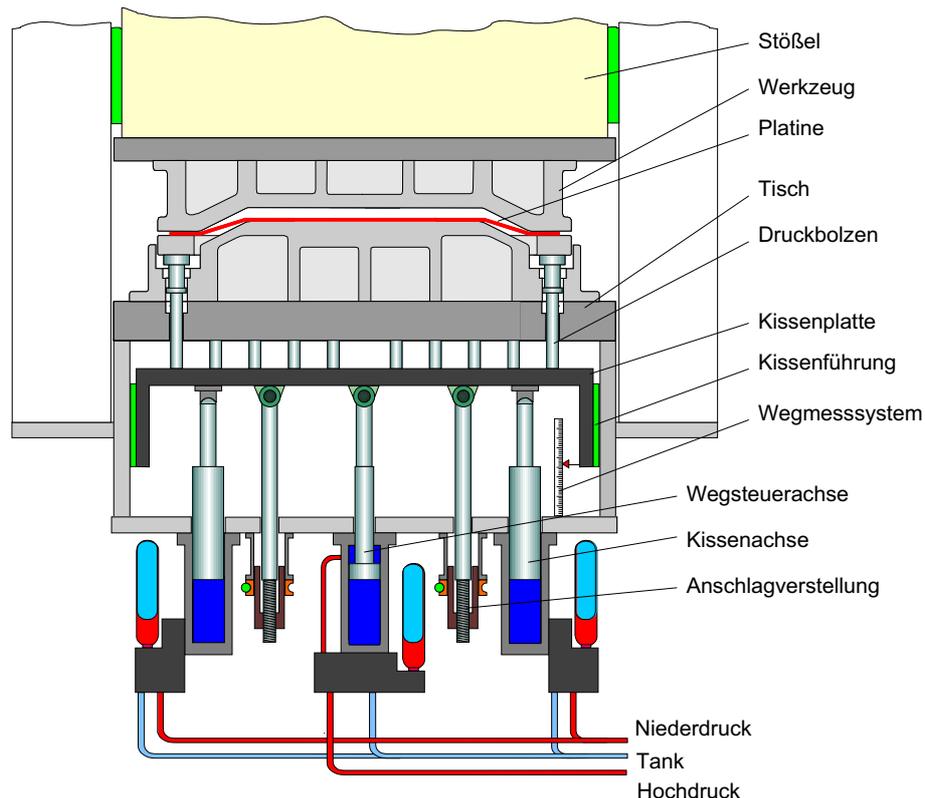


Bild 1: Schema einer Tiefziehpresse

Die Aufgabenstellung geht dabei weit über das Einstellen eines PID-Reglers hinaus (Einstellregeln), da

- neben den PID-Parametern (K_P , T_N , T_V) weitere Übergangseigenschaften bzw. -parameter festgelegt werden müssen,
- die Anregung durch das Auftreffen des Stößels mit einem einfachen Sprung nicht beschreibbar ist, sowie
- das nichtlineare Verhalten der (hydraulischen) Regelstrecke unvorhersehbaren Schwankungen unterliegt.

Die Problemstellung ist in gewisser Weise ähnlich zu der in [1] beschriebenen Problematik. Die größere Anzahl von Einflußparametern macht es schwieriger, jedoch sind (vereinfachend) keine online-Änderungen vorgesehen. Die Tatsache, dass erfahrene Bediener

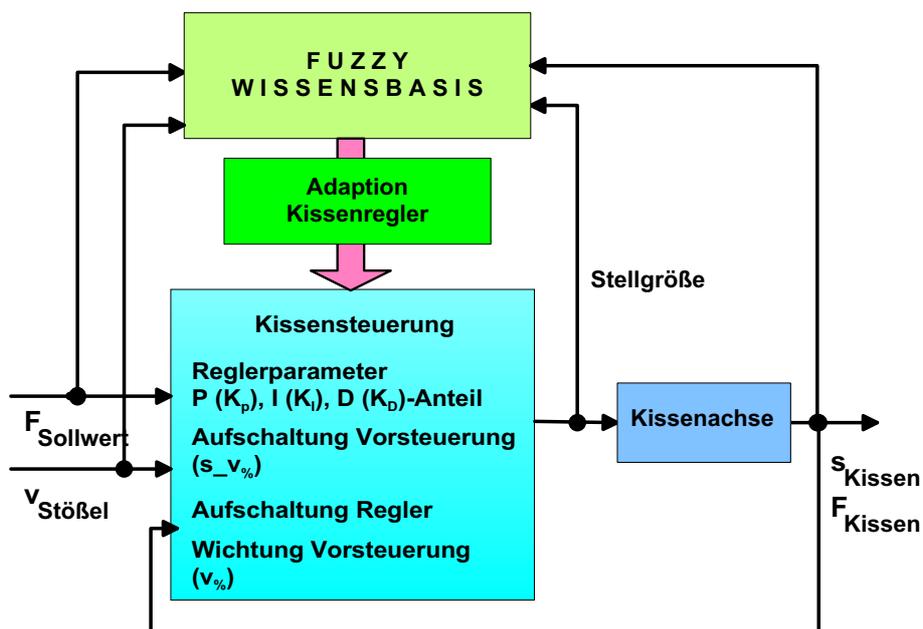
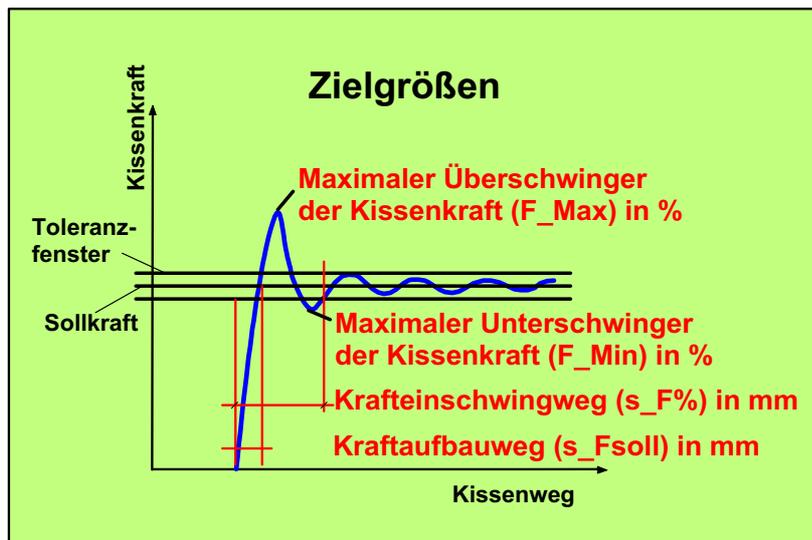


Bild 2: Grundschaema des Kissenreglers

in der Lage sind, das gewünschte Verhalten nach etlichen Probehüben den Anforderungen entsprechend zu erreichen, motiviert den Ansatz, diese Aufgabe einem Fuzzy-System zu übertragen, das ohne Spezialist zielführend diese Optimierung durchführen soll und außerdem prozessbegleitend auf mögliche Veränderungen (Blechqualität, Veränderungen im Hydrauliksystem) reagieren kann. Die Wirkung des Fuzzy-Systems erfolgt zwischen 2 Hüben, d.h. während des Ziehvorgangs gibt es keine Parameteränderungen.

2 Modellierung

Das dynamische Verhalten und die Arbeitsweise des Hydrauliksystems einschließlich PID-Regler mit den Abläufen in der Presse (Kissenachse) kann mittels Matlab-Simulink nachgebildet werden. Dieses Modell ist der Ausgangspunkt der Untersuchungen, die später auf einer realen Presse fortgesetzt werden sollen.

2.1 Bewertung des Verhaltens

In der Simulation können markante Auswirkungen ungünstiger Einstellungen wie Risse oder Falten nicht nachgebildet werden. Es werden deshalb Anforderungen an den Verlauf des Hubes gestellt, die ganz ähnlich denen sind, die man an einen Regler stellt, der auf einen Führungssprung reagiert. Es wird außerdem von einer (über einen Hub) konstanten Sollkraft ausgegangen. Technologisch begründete zeitveränderliche Verläufe der Sollkraft sind denkbar, werden aber zunächst nicht betrachtet. Beim Auftreffen des Stößels auf Werkstück und Niederhalter steigt die Kraft sprunghaft an. (Die Kraft des Stößels liegt weit über der gewünschten Niederhalterkraft!) Durch Vorsteuerung am Regelventil wird der Stoß abgefangen bevor die Kraftregelung die Ventilsteuerung übernimmt. Der sich ergebenden Kraftverlauf wird nach folgenden Kriterien beurteilt, die unmittelbar die Qualität des Bauteils beeinflussen:

Stabilität: Nach einem Einschwingvorgang wird der Sollwert schwingungsfrei gehalten (bis zum unteren Totpunkt des Stößels \Rightarrow Rückkehr zur Lageregelung)

Kraftaufbauweg (s_{Fs}): Stößelweg (nach Auftreffen) bis zur (erstmaligen) Erreichung der Sollkraft

Maximalkraft (F_{max}): Maximalwert der Kraft über Sollwert (relativ zum Sollwert in %)

Minimalkraft (F_{min}): Höchste Kraftunterschreitung unter Sollwert (relativ zum Sollwert in %)

Krafteinschwingweg (s_{Fp}): Stößelweg, ab dem vorgegebene Soll-Ist-Toleranzen nicht mehr verletzt werden

2.2 Parametereinfluss

Das Simulink-Modell gestattet die Untersuchung der Auswirkung von Änderungen in den zu optimierenden Parametern. Die große Anzahl von wirksamen Parametern macht diesen Einfluß sehr unübersichtlich. Deshalb wurde zunächst durch Variation von lediglich 4 ausgewählten Einflußfaktoren die Auswirkung auf die obigen Kriterien untersucht.

Die methodische Basis für diese Betrachtungen sind die Arbeiten zu mehrdimensionalen Kennfeldern (SGR) [3]. Zunächst wurden die (sonstigen) Parameter des Prozesses auf praktisch relevante Werte gesetzt. Dabei wurden für die Hubzahl (Bewegungsgeschwindigkeit) verschiedene Varianten betrachtet ($n_H = 8 \dots n_H = 20$) und die als kritisch bekannte Ventilkennlinie des Regelventils zufällig geringfügig variiert. Auch für die Einstellparameter wurden bekannte technische Grenzen verwendet und schrittweise variiert. Folgende Parameter wurden für die Untersuchung ihrer Auswirkung auf die Bewertungskriterien ausgewählt:

K_P : Proportionalverstärkung des PID-Reglers

$K_I = \frac{K_P}{T_N}$: Integralverstärkung des PID-Reglers

Trim: Grad der Freigabe durch Vorsteuerung

F_{zu} : Kraftlevel für Start der Kraftregelung

Der D-Anteil wurde nicht verwendet, da sich mit ihm in diesem Fall erfahrungsgemäß keine Verbesserung zu erwarten sind.

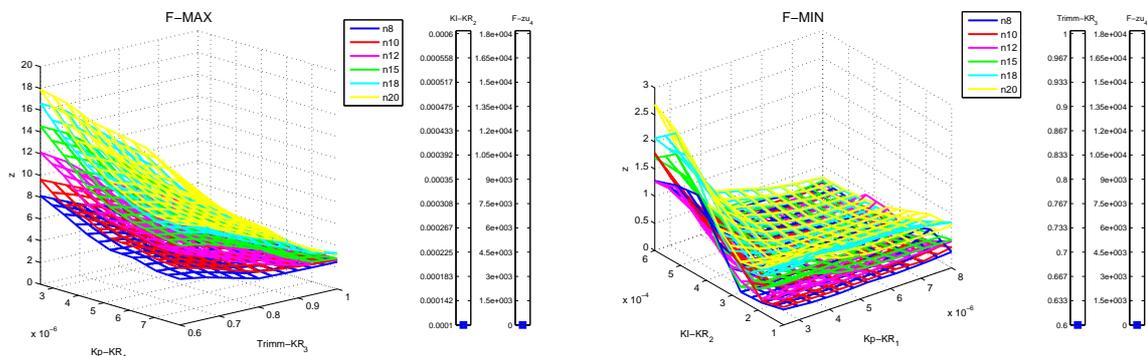


Bild 3: 4D-SGR-Modelle für Über- und Unterschwingweite

Mittels Simulation wurde eine Datenbasis generiert, durch die der Zusammenhang empirisch ermittelt wurde (Bild 3,4). Es wurden also für die 4 numerischen Kriterien jeweils 4-dimensionale Kennfelder berechnet. Auch die Abhängigkeit von der Hubzahl konnte so verdeutlicht werden. In den Abbildungen können die Abhängigkeiten natürlich nur 2D-dargestellt werden. Mittels Schieberegler können die restlichen Variablen verändert werden. Auf diese Weise und auch durch den Wechsel der Vordergrundvariablen ist eine umfassende visuelle Erfassung der Eigenschaften des Kissenreglers möglich. Werden

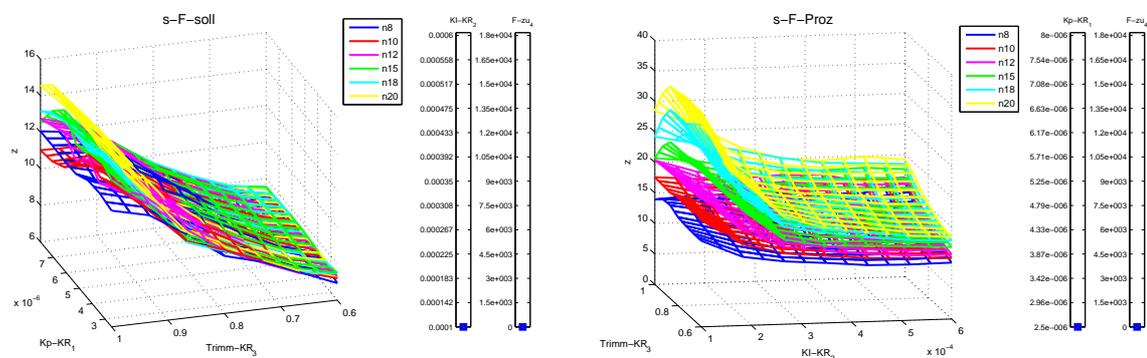


Bild 4: 4D-SGR-Modelle für Einschwingwege

die Werte aus der Datenbasis (soweit sie in der Nähe des dargestellten Ausschnitts der Funktion liegen) in die Darstellung aufgenommen, ist zum einen die Variationsbreite, verursacht durch die zufällige Störung der Ventilkennlinie, erkennbar (senkrechte Linie am Datenpunkt) und andererseits wird auch die (parametrisch einstellbare) Glättungseigenschaft der SGR-Methode deutlich (Bild 5,6). Die Untersuchung der 4 Zusammenhänge hat ergeben, dass in fast allen Fällen der 2D-Zusammenhang eine gute Näherung darstellt, d.h. in den Plots sind die Schieberegler für die Hintergrundvariablen nahezu wirkungslos.

Die Wirkungsweise der Parameter lässt sich qualitativ in Tabelle 1 verdeutlichen. Dabei wird die Wirkung positiv bezeichnet, wenn eine Vergrößerung des Parameters auch das Kriterium vergrößert (positiver Gradient). Die Zielstellung besteht generell in der Verringerung der Kriterien, woraus sich der Trend für die Korrektur der Parameter qualitativ

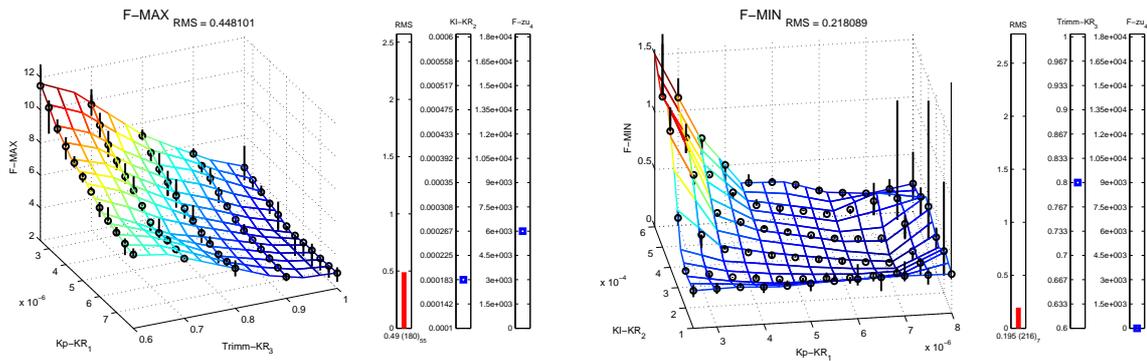


Bild 5: 4D-SGR-Modelle für Über- und Unterschwingweite bei $n_H = 12$ mit Daten

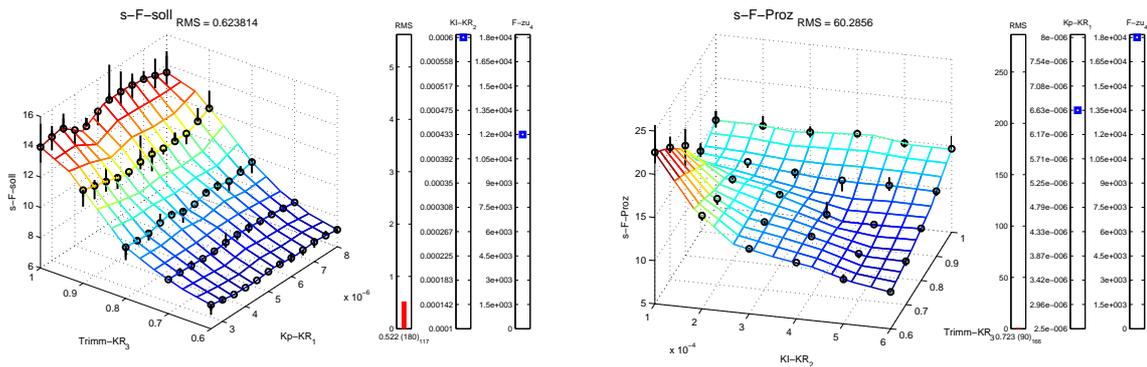


Bild 6: 4D-SGR-Modelle für Einschwingwege bei $n_H = 12$ mit Daten

(nicht ohne Widersprüche) ableiten läßt, solange es keine Stabilitätsprobleme gibt. Es kann festgestellt werden, dass zumindest in dem für die Simulation verwendeten Bereich das Kraftlevel für die Zuschaltung des Kraftreglers keine nennenswerte Wirkung hatte.

3 Optimierungskonzept

Die Modellierungsergebnisse lassen den Gedanken aufkommen, mit dieser relativ guten Kenntnis die Optimierung mittels eines Optimierungstools durchzuführen. Dagegen gibt es mehrere Gegenargumente:

- Auch bei sorgfältiger Simulation lassen sich die Ergebnisse nur bedingt auf die reale Presse übertragen.
- Die simulative Ermittlung des Zusammenhangs ist unter konkreten praktischen Bedingungen nicht möglich.

Parameter	F_{max}	F_{min}	s_{Fs}	s_{Fp}
ΔK_P	negativ stark	negativ schwach	gering	gering
ΔK_I	gering	positiv stark	gering	negativ schwach
$\Delta Trimm$	negativ stark	gering	positiv stark	gering
ΔF_{zu}	gering	gering	gering	gering

Tabelle 1: Wirkung der Einflussparameter

- Die Akzeptanz eines solchen Systems (wenn der Entwurf gelänge) wäre vermutlich sehr gering.

Aus diesem Grunde ist das Bestreben darauf ausgerichtet, ausgehend von einer stabilen Variante nach erfolgversprechenden Korrekturen der Einstellparameter zu suchen, also quasi dem ursprünglichen Fuzzy-Gedanken folgend, den Spezialisten durch ein lernfähiges System zu ersetzen. Der Spezialist wird üblicherweise nicht immer an allen Parameter gleichzeitig drehen, so dass neben der Frage nach der Höhe der Änderung auch die Auswahl, wann man welchen Wert ändern sollte, zur Diskussion steht.

Die Lernfähigkeit des Systems wird unter Berücksichtigung, welche Korrekturentscheidung in der Vergangenheit positiv oder negativ gewirkt hat, angestrebt. Die konkrete algorithmische Form dafür liegt noch nicht fest.

Da es sich bei der Suche nach optimalen Einstellparametern um ein mehrkriteriales Optimierungsproblem handelt und die Kriterien keineswegs voneinander unabhängig sind, ist eine klare Definition des Optimums nicht möglich. Für die Simulationsrechnung kann man mit den gewonnenen Daten versuchen, die Paretomenge anzunähern. Da dies im praktischen Fall nicht möglich ist, wird man sich damit begnügen müssen, für die Kriterien Zielbereiche anzugeben, die auch unter realen Verhältnissen realistisch erreichbar sind und die im Bedarfsfalle notfalls erweitert werden müssen. Wenn das so definierte Zielgebiet erreicht ist, werden weitere Korrekturen erst dann wieder vorgenommen, wenn der Prozeß das Zielgebiet wieder verläßt bzw. sich dem Rand nähert.

4 Systemstruktur

Das unter Einsatz von Fuzzy-Komponenten zu realisierende System soll folgende Zielstellungen erfüllen:

- Eine Einstellung, bei der ein Hub stabil erfolgt ist, d.h. aus der Beobachtung der Prozesssignale die Berechnung der Kriterien möglich war, ist notwendiger Ausgangspunkt
- Veränderungen der Parameter erfolgen zwischen den Hübten, d.h. beim Hub selbst sind die Parameter konstant.
- Aus der Auswertung der Kriterien werden fuzzylogisch Änderungsvorschläge abgeleitet.
- Es wird entschieden, welche Korrektur realisiert wird.
- Wenn der Prozess (unerwartet) instabil wird, wird die (falsche) Korrektur zurückgenommen und eine neue Entscheidung getroffen.
- Das Protokoll der vorherigen Korrekturen und ihrer erzielten Wirkungen wird zur Entscheidung über die neue Korrektur herangezogen.
- Die Erreichung des Zielbereichs terminisiert den aktiven Einsatz des Systems. Die weitere Überwachung (Auswertung der Kriterien) soll gegebenenfalls erneute Aktivität ermöglichen.

sind insbesondere bei der Überführung des Verfahrens auf eine reale Presse nicht auszuschließen (Tabelle 2). Es wird ein Erfolgsindikator E_x eingeführt, der für den Fall, dass der letzte Hub instabil war, hoch negativ gesetzt wird, ansonsten wird die Verbesserung der (direkt) abhängigen Kriterien (summiert) verwendet. Die Schrittfaktoren beschreiben,

Parameter	Einflußgrößen			Schrittfaktor
ΔK_P	E_{Kp}	F_{max}	F_{min}	a_{Kp}
ΔK_I	E_{KI}	F_{min}	sF_{proz}	a_{KI}
$\Delta Trimm$	E_{Trimm}	F_{max}	sF_{soll}	a_{Trimm}

Tabelle 2: Struktur der Teilmodelle

inwieweit mit der erfolgten (stabilen) Korrektur die erwartete Verbesserung der Kriterien eingetreten ist. Somit können die Definitionen der Zugehörigkeitsfunktionen während des Lernvorganges unverändert bleiben.

4.2.1 Fuzzifizierung der Einflußgrößen

Die Festlegung von linguistischen Werten orientiert sich an der praktischen Situation, dass sich aus Erfahrung Zielbereiche angeben lassen, die quasi das Optimierungsziel beschreiben. Für den ins Auge gefaßten Einsatzfall werden die in Tabelle 3 angegebenen Grenzwerte z_x verwendet:

$$\begin{aligned}
 F_{max} &< z_{max} = 10\% \\
 F_{min} &< z_{min} = 10\% \\
 sF_{soll} &< z_{Fs} = 10 \text{ mm} \quad (\text{max. } 15\text{mm}) \\
 F_{proz} &< z_{Fp} = 20 \text{ mm}
 \end{aligned}$$

Tabelle 3: Definition Zielgebiet (z_x)

Sie orientieren sich insbesondere an dem Wert des Kriteriums, der als Rand des Zielbereiches festgelegt wurde. Der Erfolgsindikator dient vor allem dazu, im Falle von instabilen Ergebnissen wieder in das Stabilitätsgebiet zurückzufinden. Für die Berechnung der Erfolgsindikatoren gibts es noch keine endgültigen Definitionen. Als Vorschlag für die ersten Untersuchungen wird folgende Berechnungsvariante untersucht:

$$E_{xx} = \frac{\sum_i \Delta F_{xi} / z_{xi}}{\Delta K_x / a_x} \quad (1)$$

Im Idealfall würde sich ein Wert nahe 1 ergeben. Die (Neu-)Berechnung ist jedoch nur möglich, wenn beim letzten Hub der Parameter geändert wurde.

Für den Fall von Instabilität (Kriterien nicht berechenbar) wird $E_{xx} = -10$ benutzt. Außerdem sollen durch seine Auswertung die Schrittfaktoren modifiziert werden (Lernstrategie). Er soll ausdrücken, inwieweit durch die letzte Korrektur der erwartete Effekt (Kriterienverbesserung) eingetreten ist. Es ist insofern eine Art Gradienteninformation, die sich längerfristig im Schrittfaktor niederschlägt. Da es sich um eine inkrementale Vorgehensweise handelt, bleibt die Festlegung einer Grundschriftweite im Raum der Kriterien, mit der Verbesserungen angestrebt werden noch als Verfahrensparameter offen.

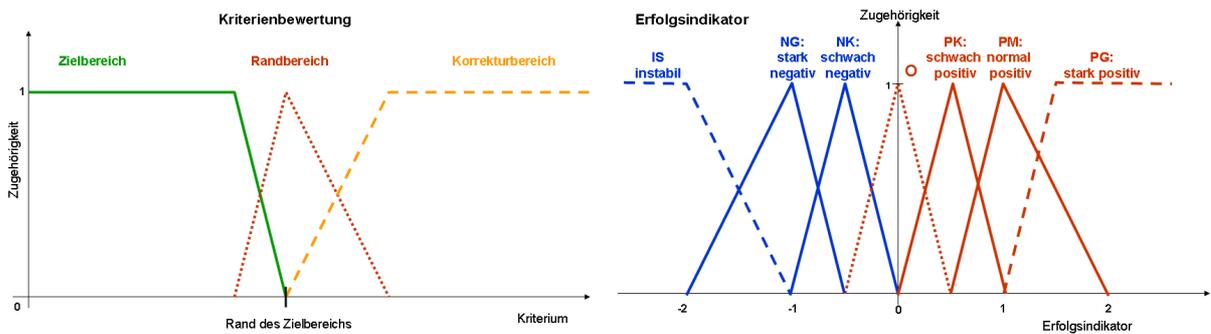
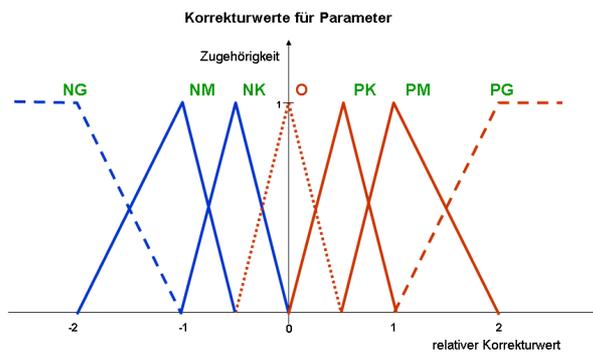


Bild 8: Definition linguistischer Werte - Input

4.2.2 Fuzzifizierung Parameterkorrekturen

Um unterschiedlichen Bedingungen einfach entsprechen zu können, werden für die 3 Parameter quasi Normalschritte für die Korrektur definiert, die im Fuzzy-System durch 1 repräsentiert werden. Mit den Schrittfaktoren, die zunächst vorgegeben werden und dann nachgeführt werden können, ergeben sich die tatsächlich am Regler zu verändernden Werte.



Die Festlegung des Parameters, der aktuell geändert wird, wird durch den jeweiligen Erfolgsindikator bestimmt.

Als Startreihenfolge wird $\Delta K_P \rightarrow \Delta Trimm \rightarrow \Delta K_I$ benutzt.

Bild 9: Definition Korrekturgrößen - Output

4.2.3 Aufbau der Wissensbasis

Die Aufstellung von Fuzzy-Regeln bezieht sich nur auf den Abschnitt des gesamten Kissenreglers, der sich mit der Berechnung von Korrekturwerten für den Regler befasst und also auch nur zwischen den Hülen wirksam wird. Die Regeln werden begründet durch die in Abschnitt 2 beschriebenen Simulationsergebnisse und das vorliegende Erfahrungswissen. Mit der Erprobung an einer Versuchspresse sind Änderungen zu erwarten. Die Regeln drücken aus, inwieweit eine Änderung des Parameters in Bezug auf eine „Normalschrittweite“ a_j erfolgen soll. Diese wiederum soll durch Trendbeobachtung am Prozeß mittels Lernstrategie nachgeführt werden.

5 Erprobung

Nach dem Test des Verfahrens mittels Simulation (Matlab/Simulink) wird es in einer Versuchspresse des IWU implementiert und erprobt.

Literatur

- [1] Niemann, M.; Adamy, J.; Nitzsche, H.-J.: *Modular aufgebaute Gießspiegelregelung für Stranggießanlagen* In: *Proc. 7. Workshop Fuzzy Systeme* (Kiendl, Knicker, Hg.), S. 133-148, UNI Dortmund. 1997.
- [2] Pfeiffer, B.-M. u.a.: *Erfolgreiche Anwendungen von Fuzzy Logik und Fuzzy Control* In. *Automatisierungstechnik* 50 Vol. 10/11 Oldenbourg Verlag 2002.
- [3] Priber, U.: *Smoothed Grid Regression*. In: *Proc. 13. Workshop Fuzzy Systeme* (Mikut, R.; Reischl, M., Hg.), S. 159-172, FZKA 6900. 2003.
- [4] Müller, J.: *Regeln mit SIMATIC*. Publicis Corporate Publishing. Erlangen 2004.

Wahrscheinlichkeitsbasierte globale Situationserkennung bei komplexen Systemen

Patrick Gerland¹, Horst Schulte², Andreas Kroll¹

¹Universität Kassel, FB Maschinenbau, FG Mess- und Regelungstechnik

²Bosch Rexroth AG, Hydraulics

E-Mail: patrick.gerland@mrt.uni-kassel.de

Der verschärfte internationale Wettbewerb und die sich ändernden gesetzlichen Bestimmungen fordern von Unternehmen eine wachsende Innovationsbereitschaft und -fähigkeit. Die angestrebten Verbesserungen wie die Steigerung der Energieeffizienz, Zuverlässigkeit und Kostenreduktion lassen sich nur durch zunehmende Automatisierung realisieren. In vielen Anwendungen geht dies mit einer systematischen Einbeziehung einer Vielzahl interagierender und vernetzter Einzelkomponenten einher, die über Rückkopplungsmechanismen untereinander sowie in Interaktion mit dem Bediener bzw. Fahrer und der sich verändernden Umwelt stehen. Damit werden Anforderungen an die Modellbildung gestellt, denen die traditionellen Modellbildungskonzepte im Allgemeinen nicht mehr gerecht werden.

In dieser Arbeit wird für komplexe technische Systeme (z.B. [6], [7]) ein Modellansatz bestehend aus zwei Sprachebenen entwickelt. Die erste Sprachebene beschreibt die physikalischen Zusammenhänge mittels mathematischer Modelle und die zweite als wahrscheinlichkeitsbasierte übergeordnete Ebene nutzt höhere Begriffe, mit denen die Eigenschaften des Gesamtsystemverhaltens beschrieben werden können. Anhand einer Fallstudie aus dem Bereich der Automatisierung mobiler Arbeitsmaschinen wird gezeigt, dass sich dieser Ansatz nutzbringend zur Situationserkennung bei wechselnden Umweltbedingungen einsetzen lässt.

1 Einleitung

Durch einen steigenden Automatisierungsgrad in Systemen wie Konsumgüter, mobile Systeme oder Fertigungsanlagen erlangt die Systemvernetzung und -integration derzeit eine wesentliche Bedeutung und stellt dabei ein enormes Innovationspotenzial dar. Das in vielen Anwendungsfällen nicht explizit formulierte Ziel besteht darin, durch eine Gesamtsystemoptimierung ungenutzte Potenziale aufgrund der Wechselwirkungen der Teilsysteme gezielt auszuschöpfen. Besonders deutlich ist dieser Trend im Automobilbereich zu erkennen: In den letzten Jahren haben eine Vielzahl verschiedener Fahrdynamikregelsysteme in den Domänen Fahrzeuglängs-, quer- und -vertikaldynamik Einzug in das Automobil genommen. Diese Systeme wurden stets nach dem Bottom-up-Prinzip entwickelt. Das klassische Vorgehen besteht dabei in der Aufstellung von mathematischen Modellgleichungen für die Komponenten und dem anschließenden Identifizieren der freien Modellparameter. Dabei werden die Systeme domäneorientiert oder nach einem speziellen Funktionsziel, d.h. bzgl. nur einer Funktion, ausgelegt.

Neuere Konzepte wie Global Chassis Control [10] oder Integrated Chassis Management [11] zielen auf die funktionale Zusammenfassung der domäneorientierten Teilsysteme ab, um die teilweise starken Wechselwirkungen untereinander gezielt auszunutzen. Beispielsweise bieten Vertikaldynamiksysteme das Potential zur Beeinflussung anderer Fahr-

zeugdynamiken und eröffnen damit die Möglichkeit der Verbesserung bzw. der Entwicklung ganz neuer Produkte und Funktionen.

Die wesentliche Herausforderung der Systemintegration besteht darin, dass mit zunehmendem Funktionsumfang gleichzeitig auch die Komplexität und somit die Fehleranfälligkeit des Systemverbundes stark ansteigt. Die Entwicklung und Absicherung des gesamten funktionalen Systemverbundes wird somit deutlich erschwert. Um die wachsende Komplexität und Heterogenität derartiger Funktionszuwächse und -umfänge beherrschen zu können, ist eine Betrachtung des Systemverbundes in seiner Gesamtheit zwingend notwendig.

Der weitgehende Stand der Technik innerhalb der Systemintegration ist das Konzept der „friedlichen Koexistenz“ [1]. Hierbei werden die Teilsysteme unter dem Aspekt von lokalen Optima individuell entworfen. Sollvorgaben und Systemzustände werden aufgrund einer dezentral organisierten Funktionsstruktur unabhängig voneinander interpretiert. Zur Vermeidung von störenden Wechselwirkungen werden übergeordnete Mechanismen zur Priorisierung hinzugefügt, die in vielen Fällen die Entwurfsziele der Teilsysteme nicht mehr berücksichtigen. Bedingt durch die Funktionsautonomie und durch die friedliche Koexistenz auf lokaler Funktionsebene können zueinander im Widerspruch stehende Funktionsziele (s.o) nur durch einen Kompromiss erreicht werden.

Die Qualität der Analyse und Synthese eines Systems oder Prozesses ist im starken Maße von der Qualität des verwendeten Systemmodells abhängig. Somit stellt die formale Erfassung eines komplexen technischen Systems die Grundlage für einen erfolgreichen modellbasierten Entwicklungsprozess dar. Die Qualität des Modells ist also dann als gut zu bezeichnen, wenn es alle für die Lösung der gestellten Aufgabe relevanten Systemeigenschaften enthält. Dabei ist die ganzheitliche Systembetrachtung für die Modellbildung eines komplexen technischen Systemverbundes zwingend notwendig. Dies erfordert eine Aufgabenverschiebung von der traditionell rein komponentenbasierten Modellierung nach dem Bottom-up-Prinzip hin zu einer Modellierung, die Sprachmittel bereitstellt, um sowohl die bekannten prozessnahen Zusammenhänge als auch die Strukturen und Sachverhalte, die das Gesamtsystem bestimmen, zu beschreiben.

Im folgenden zweiten Abschnitt dieser Arbeit wird zunächst der Komplexitätsbegriff näher umrissen. Hierbei werden problemangepasste charakteristische Eigenschaften aufgeführt, die im besonderen Maße die Fragestellungen der Automatisierung und Regelung technischer Systeme berücksichtigen. Im dritten Abschnitt werden die wahrscheinkeitsbasierten Methoden und Algorithmen vorgestellt, die zur Bildung der Sprachkonstrukte der übergeordneten Sprachebene bisher untersucht wurden. Im einzelnen werden dabei bekannte Methoden der Merkmalsselektion, Merkmalstransformation und die Klassifikation mit Bayes-Klassifikatoren beschrieben. Anhand einer aktuellen Aufgabenstellung zur automatisierten Fahrsituationserkennung bei mobilen Arbeitsmaschinen werden daraufhin im vierten Abschnitt die zuvor beschriebenen Methoden praktisch erprobt. Anschließend werden die Ergebnisse vorgestellt, interpretiert und das weitere Vorgehen zum formalen Entwurf der zweiten Sprachebene diskutiert.

2 Charakterisierung komplexer technischer Systeme

Der Komplexitätsbegriff spielt in der Mathematik eine große Rolle und es liegen viele Definitionen vor, mit denen versucht wird, diesen Begriff zu erfassen (Kolmogorov-Komplexität, Anzahl der Multiplikationen in einem Programm, nicht polynomial schnelle Lösbarkeit, nicht Separierbarkeit, etc.). Jedoch legen diese mathematischen Formulie-

rungen bezüglich relevanter praktischer Anwendungen oft viel zu weite Schranken fest. Ein Problem ist innerhalb des handlungsorientierten Kontext des Ingenieurs häufig schon nicht mehr beherrschbar, auch wenn es durch die mathematischen Begriffe als nicht komplex klassifiziert wird. Von der Praxis ausgehend wird in dieser Arbeit ein System oder ein Problem als komplex charakterisiert, wenn es innerhalb der Beschreibungssprache, in der es definiert ist, i.A. nicht beherrscht werden kann. Zu einem solchen System müssen daher zusätzlich zu den mathematischen Beschreibungen Heuristiken gefunden werden, um die von ihm aufgeworfenen Probleme zu lösen.

Reale komplexe technische Systeme lassen sich durch eine Reihe charakteristischer Eigenschaften beschreiben:

1. Das Gesamtsystem ist vielschichtig aufgebaut (z.B. Systemebene mit Sensorik und Aktorik, Funktionsebene, Applikationsebene).
2. Das Gesamtsystem besteht aus einer Vielzahl interagierender, vernetzter und teils stark heterogener Einzelsysteme mit komplizierten Zusammenhängen.
3. Das Gesamtsystem kann sehr ausgedehnte Betriebsbereiche (Betriebs- und Fehlersituationen) mit möglicherweise signifikant unterschiedlich dynamischem Verhalten annehmen (gezielte Variation: z.B. Reifenwechsel Sommer-/Winterreifen, systembedingte Variation: Einstecken eines Radladers in ein Haufwerk oder Steigungsfahrt, unerwünschte Variation: Sensorfehler und äußere Störgrößen).
4. Das Gesamtsystem steht über Rückkopplungsmechanismen in Interaktion mit dem Bediener und der variablen Umwelt.
5. Die wesentlichen Charakteristika des Gesamtsystemverhaltens werden durch die Verkettung der Teilsysteme geprägt. Beispielsweise können der Einfluß und die Auswirkung der Umgebung zu unvorhersehbaren Wechselwirkungen führen, die wiederum Fehlfunktionen im Betrieb zur Folge haben können.
6. Das Gesamtsystem wird gesteuert durch autonome oder teilautonome situationsangepasste Stelleingriffe der Teilsysteme (d.h. i.d.R. nur wenige oder keine Interaktion der Teilsysteme mit dem Menschen über einen längeren Zeitraum).

Gerade aufgrund der letztgenannten Eigenschaft ergibt sich die Notwendigkeit der automatischen Selbstüberwachung und Diagnose der Teilkomponenten (Aktoren, Sensoren, Hardware) für den sicheren und zuverlässigen Betrieb des Gesamtsystems.

Diese genannten Eigenschaften verdeutlichen, dass der entscheidende Schritt zur Beherrschung eines komplexen technischen Systems in dem robusten Erkennen und einer das ganze System berücksichtigenden Interpretation des aktuellen Systemzustandes liegt. Nur darüber kann eine situationsangepasste, das Gesamtsystem betreffende Entscheidungsfindung realisiert werden. Zur vollständigen formalen Erfassung eines komplexen technischen Systems sind dabei Modellrepräsentationen in zwei unterschiedlichen Beschreibungssprachen erforderlich.

In der ersten Beschreibungssprache werden die physikalisch-technischen Zusammenhänge prozessnah mittels einzelner mathematischer Modelle in Form von Differentialgleichungen beschrieben. Jedoch ist die Erkennung globaler Eigenschaften, Situationen und vorerst ungeklärter Sachverhalte wie bereits erwähnt mittels klassischer Methoden der Modellbildung nicht möglich bzw. ist der Aufwand nicht vertretbar. Zum Beispiel müssten

zeitvariante Umwelteinflüsse durch 3D-Repräsentation mittels aufwendiger Sensorik erfasst, ausgewertet und in die physikalische Beschreibungssprache überführt werden¹.

Die zweite Beschreibungssprache nutzt daher übergeordnete Sprachkonstrukte (höhere Begriffe, z.B. Steigungsfahrt), mit denen Eigenschaften, die das gesamte System betreffen, formulierbar sind. Über Korrespondenzregeln werden lokale Teilsysteme situativ parametrisiert und koordiniert. Diese strukturierte Funktionsorganisation bietet die wesentliche Grundlage, dass alle Aktionen der Teilsysteme aus einer globalen Zustandserkennung heraus synchronisiert und unter dem Aspekt des optimalen Gesamtsystemverhaltens realisiert werden können. Darüber lassen sich Uneindeutigkeiten und Zielkonflikte auf der Funktionsebene auflösen, Regeleingriffe koordinieren und ein Monitoring des Gesamtsystems durchführen.

Die verschiedenen Beschreibungssprachen zur Modellrepräsentation sind nicht direkt ineinander übersetzbar. Ansonsten könnte die eine Sprache auf die andere zurückgeführt und somit ersetzt werden. Die wesentliche Schwierigkeit besteht darin, dass im Gegensatz zur ersten Sprache, die durch die Physik vorgegeben ist, die Begriffe der zweiten Sprache erst noch definiert werden müssen.

Anhand einer aktuellen Aufgabenstellung aus dem Bereich der Automatisierung mobiler Arbeitsmaschinen, die in die hier betrachtete Klasse der komplexen technischen Systeme fallen, soll aufgezeigt werden, dass sich Methoden der Computational Intelligence zur Darstellung dieser zweiten Beschreibungssprache anwenden lassen.

3 Methoden

Die Modellierung und Erfassung des globalen Gesamtsystemverhaltens erfolgt in dieser Arbeit mittels eines Bayes-Klassifikators. Dieser Klassifikator ordnet sich in die Gruppe der statistischen Klassifikatoren ein. Notwendig ist die Verfügbarkeit von typischen Beispieldatensätzen, aus denen die Parameter des Klassifikators automatisch geschätzt werden können. Es sind daher keine komplexen physikalischen Modellgleichungen erforderlich, um die typischen Eigenschaften der betrachteten Daten zu beschreiben. Diese werden vielmehr automatisch aus den vorliegenden Trainingsdaten extrahiert. Bei dem Bayes-Klassifikator handelt es sich um einen überwacht lernenden Klassifikator, d.h. auch die Klasseneinteilung der Beispieldatensätze ist bekannt.

Das Ziel einer Klassifikation besteht darin, eine Trennung eines gegebenen Datensatzes in Klassen zu erreichen. Wesentliche Charakteristika der Daten sollen in möglichst kompakter Form zusammengefasst werden. Die Klassen sollen die folgenden Eigenschaften besitzen:

- Homogenität innerhalb der Klassen, d.h. die Daten innerhalb derselben Klassen sollten so ähnlich wie möglich sein.
- Heterogenität zwischen den Klassen, d.h. Daten unterschiedlicher Klassen sollen sich möglichst stark unterscheiden.

Klassifikationsverfahren bestimmen die Klassenzugehörigkeit eines Objektes oder eines Systemzustandes i.d.R. über eine nichtlineare Entscheidungsfunktion aus einer Menge

¹Dies ist ein aktuelles Forschungsgebiet, wobei es schon bei relativ klar definierten Umgebungen aus dem Bereich autonomer Fahrzeuge Problemstellungen gibt, die mittels aufwendiger Sensorik (Laserscanner, Radar, Lagesensoren) bisher nur bedingt lösbar sind.

von Informationen. Diese Informationen werden als Merkmale bezeichnet. Ein Merkmal ist eine informationstragende Zahl, die meist durch eine informationsverdichtende Maßnahme aus einer Zeitreihe oder Bildern gewonnen wird (nachfolgend werden nur noch Zeitreihen betrachtet). Diesen Schritt bezeichnet man als Merkmalsextraktion. Merkmale übernehmen eine Art Filterfunktion, da sie die wesentlichen Informationen einer großen Anzahl von verfügbaren Daten auf eine geringere Anzahl von Merkmalen abbilden. Merkmale sind immer problemspezifisch zu wählen. Die Festlegung geeigneter diskriminierender Merkmale ist das Standardproblem der Klassifikation.

Der Entwurf des Klassifikators erfolgt in dieser Arbeit vollautomatisch mittels der MATLAB-Toolbox Gait-CAD [4]. Alle im Folgenden aufgeführten Algorithmen und Begriffe sind ausführlich und sehr verständlich in [3] beschrieben. Bild 1 stellt den mehrstufigen, automatisierten Entwurfsprozess des Klassifikators grafisch dar.

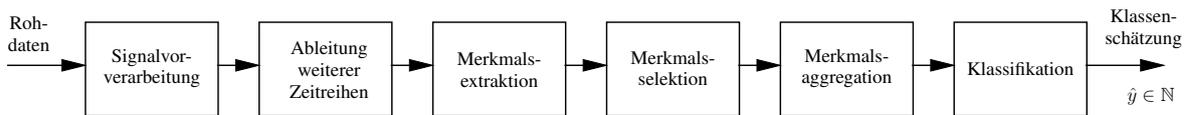


Bild 1: Automatisierter Entwurfsprozess des Klassifikators

In dem ersten Schritt werden die Rohdaten (Originalzeitreihen) des Beispieldatensatzes zur Ausreißerelimination typischerweise gefiltert, z.B. Tief-, Band- oder Hochpassfilterung. Daraufhin erfolgt die Ableitung weiterer Zeitreihen wie bspw. erste oder zweite Ableitung nach der Zeit aus den vorverarbeiteten Zeitreihen. Aus diesem Zeitreihensatz lässt sich jetzt ein Merkmalsatz \mathbf{x} mit s potentiell geeigneten und interpretierbaren informationstragenden Merkmalen extrahieren.

Merkmale sind vom Typ der Klassifikationsaufgabe abhängig zu wählen. Beispielsweise erfordert eine zeitkontinuierliche Klassifikation, bei der zu jedem Abtastpunkt eine Entscheidung getroffen wird, auch kontinuierliche Merkmale. Beispiele hierfür sind z.B. rekursiv berechenbare Größen wie gleitender Mittelwert, gleitende Streuungen oder Frequenzbänder. Die Verwendung zeitaggrierter Klassifikatoren, bei der die Entscheidungsfindung von mehreren Abtastpunkten abhängt oder sogar erst am Ende einer Sequenz erfolgt, ermöglicht die Berücksichtigung weiterer Merkmale wie bspw. minimale oder maximale Werte einer Zeitreihe oder auch Kurvensteigungen bzw. charakteristische Kurvenformen.

Alle N Tupel des s -dimensionalen Merkmalsatzes \mathbf{x} lassen sich für die folgenden Berechnungen in der Matrix \mathbf{X} zusammenfassen:

$$\mathbf{X} = \begin{pmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,s} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,s} \\ \vdots & \vdots & & \vdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,s} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{pmatrix} \quad (1)$$

Oftmals hängt der Erfolg zur Lösung der gestellten Klassifikationsaufgabe wesentlich von der Merkmalsextraktion und dem Auffinden von wichtigen bzw. diskriminierenden Merkmalen ab. Dieser Schritt wird als Merkmalsselektion bezeichnet. Das Ziel besteht darin, eine möglichst kleine und aussagekräftige Gruppe \mathcal{I} , zusammengesetzt aus s_m aller s Merkmalen aufzufinden. Diese Gruppe \mathcal{I} ist umso aussagekräftiger, je besser die zuvor genannten Eigenschaften Homogenität innerhalb und Heterogenität zwischen den Merkmalen der gleichen Klasse erfüllt werden. Zum Auffinden von Merkmalen, die diese

Forderungen erfüllen, kann die Multivariate Varianzanalyse (MANOVA [3]) angewandt werden. Dieses statistische Verfahren basiert auf der Annahme, dass die einzelnen Klassen näherungsweise durch eine mehrdimensionale Normalverteilung der Merkmale beschrieben werden können. Die Beurteilung bzw. der Vergleich verschiedener Kombinationen von s_m Merkmalen erfolgt durch das Likelihood-Quotienten-Maß

$$M_{\mathcal{I}} = 1 - \prod_{i=1}^{s_m} \frac{1}{1 + \lambda_i}, \quad (2)$$

wobei $M_{\mathcal{I}}$ Werte zwischen 0 (ungeeignete Merkmalskombination) und 1 (sehr geeignete Merkmalskombination) annimmt. Dieses Maß bewertet die s_m Eigenwerte λ_i , die man unter Anwendung des Streuungszerlegungssatzes $\mathbf{T} = \mathbf{W} + \mathbf{B}$ mit der Innerklassenvariationsmatrix \mathbf{W} und der Zwischenklassenvariationsmatrix \mathbf{B} aus der Lösung des Eigenwertproblems

$$(\mathbf{W}^{-1}\mathbf{B} - \lambda_i\mathbf{I})\mathbf{v}_i = \mathbf{0} \quad \text{mit} \quad \mathbf{B} = \mathbf{T} - \mathbf{W}, \quad \mathbf{W} = \sum_{c=1}^C N_c \cdot \mathbf{S}_c \quad (3)$$

erhält. Weiterhin sind in Gleichung (3) $\mathbf{T} = N \cdot \mathbf{S}$ die Gesamtvariationsmatrix, \mathbf{S} sowie \mathbf{S}_c Schätzungen der Kovarianz- und Klassenkovarianzmatrizen und \mathbf{I} die Einheitsmatrix. N_c ist der Umfang der Merkmale der jeweiligen Klassen $c = 1 \dots C$. Für den Sonderfall der Merkmalskombination bestehend aus lediglich einem Merkmal ($s_m = 1$) geht die MANOVA in die Univariate Varianzanalyse (ANOVA) über.

Durch Anwendung einer dimensionsreduzierenden Merkmalstransformation (Merkmalsaggregation) kann eine geringere Anzahl von s_d neuen, transformierten Merkmalen aus den zuvor ausgewählten s_m Merkmalen berechnet werden ($s_d < s_m$). Wichtig bei dieser Transformation ist, dass die Trennbarkeit der Klassen auch weiterhin bestehen bleibt. Als lineare Verfahren eignen sich bspw. die Hauptkomponentenanalyse (HKA) sowie die Diskriminanzanalyse (DA). Bei beiden Verfahren erfolgt die Berechnung der niederdimensionalen Merkmale \mathbf{X}_d durch die Multiplikation der zuvor ausgewählten Merkmale \mathbf{X}_m mit einer Transformationsmatrix \mathbf{V} ($\mathbf{X}_d = \mathbf{X}_m \cdot \mathbf{V}$). Die Transformationsmatrix $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_{s_d})$ setzt sich aus den s_d größten Eigenvektoren der Eigenwertprobleme

$$\text{HKA: } (\mathbf{T} - \lambda_i\mathbf{I})\mathbf{v}_i = \mathbf{0} \quad \text{bzw.} \quad \text{DA: } (\mathbf{W}^{-1}\mathbf{B} - \lambda_i\mathbf{I})\mathbf{v}_i = \mathbf{0} \quad (4)$$

mit den Matrizen wie in (3) zusammen. Die DA ist der HKA bei überwachten Klassifikationsproblemen mit bekannten Ausgangsklassen vorzuziehen, da sowohl Eingangsdaten (Merkmale) als auch Ausgangsdaten (deren Klassenzugehörigkeit) in die Berechnung der geschätzten Klassenkovarianzmatrizen \mathbf{S}_c eingehen. Allerdings ist wegen diesen Schätzungen auch ein umfangreicher Beispieldatensatz notwendig. Der wesentliche Nutzen dieser Dimensionsreduktion liegt zum Einen in der Laufzeitminimierung der Klassifikation und zum Anderen darin, dass die Ergebnisse nun durch den Entwickler visuell begutachtet werden können und dadurch die anschließende Klassifikation erleichtert wird. Durch die Transformation wird jedoch i.A. auch die Interpretierbarkeit der Merkmale beeinträchtigt.

Bei Verwendung des Bayes-Klassifikators fällt die Entscheidung \hat{y} zugunsten der Klasse $c = 1 \dots C$ mit der größten geschätzten A-posteriori-Wahrscheinlichkeit

$$\hat{y} = \underset{j}{\operatorname{argmax}}(\hat{p}(y = c|\mathbf{x}_d)) = \underset{j}{\operatorname{argmax}} \left(\frac{\hat{p}(y = c) \cdot \hat{p}(\mathbf{x}_d|y = c)}{\sum_{c=1}^C \hat{p}(y = c) \cdot \hat{p}(\mathbf{x}_d|y = c)} \right) \quad (5)$$

mit der jeweils geschätzten A-priori-Wahrscheinlichkeit $\hat{p}(y = c)$ und der mehrdimensionalen Wahrscheinlichkeitsdichte

$$\hat{p}(\mathbf{x}_d | y = c) = \frac{1}{(2\pi)^{\frac{s_d}{2}} \sqrt{|\mathbf{S}_c|}} \cdot e^{(-\frac{1}{2}(\mathbf{x}_d - \mu_c)^T \mathbf{S}_c^{-1} (\mathbf{x}_d - \mu_c))} \quad (6)$$

unter Annahme einer Normalverteilung der zuvor transformierten Merkmale \mathbf{x}_d . μ_c ist der Vektor der Erwartungswerte dieser Merkmale für die c -te Ausgangsklasse.

4 Automatisierte Fahrsituationserkennung bei mobilen Arbeitsmaschinen

Bau-, Land-, Forstmaschinen sowie Flurförder- und Kommunalfahrzeuge werden unter dem Begriff *mobile Arbeitsmaschinen* zusammengefasst. Während stationäre Arbeitsmaschinen im industriellen Bereich vorwiegend definierte, feste Arbeitsabläufe zyklisch durchlaufen, unterliegen mobile Arbeitsmaschinen vielfältigeren Anforderungen. Charakteristisch für mobile Arbeitsmaschinen ist dabei, dass sie für die Funktionen Fahren und Arbeiten ausreichend Leistung bereitstellen und den dazu notwendigen Energievorrat mitführen müssen. Die Fortbewegung ist zudem, anders als bei einem Zug, durch den Bediener frei wählbar. Aufgrund des vielfältigen Einsatzes ergeben sich extreme Umweltbedingungen hinsichtlich Temperatur, Feuchtigkeit, Bodenbeschaffenheit und Geländeänderungen. Die Einsatzfähigkeit muss unter diesen stark wechselnden Umweltbedingungen stets gewährleistet sein.

Immer mehr Arbeitsprozesse mobiler Arbeitsmaschinen sind durch automatisierte Vorgänge geprägt, in denen der Fahrer lediglich als übergeordnetes Kontrollorgan dient und die Funktionen den verschiedenen Automaten überlassen kann. Dabei werden automatisierte Vorgänge sowohl bei Funktionen im Fahrantrieb als auch bei der Arbeitshydraulik eingesetzt:

- Automatisierte hydrostatische Fahrantriebe entlasten den Fahrer und ermöglichen durch Entkopplung und zusätzliche Freiheitsgrade eine energieoptimierte Steuerung des gesamten Antriebsstrangs.
- Die Vorgewende-Management-Systeme bei Traktoren dienen der zeitgesteuerten automatisierten Ablaufsteuerung der Prozesse Absenken der Motordrehzahl, Pflugausheben, Pflugdrehen und Pflugeinsetzen.
- Voll- oder teilautomatisierte Ansteuerungen der hydraulischen Achsen bei Baggern und Forstmaschinen entlasten den Bediener und ermöglichen eine einfache Handhabung.

Aus dem Vergleich der im zweiten Abschnitt aufgeführten Eigenschaften zur Charakterisierung komplexer Systeme und die Beschreibung der Einsatzfälle wird die Aussage abgeleitet, dass die Problemstellung der Automatisierung von Arbeitsprozessen bei mobilen Arbeitsmaschinen in die hier betrachtete Klasse der komplexen technischen Systeme fällt. Unterstützt durch die Anforderungen der ausschließlichen Verwendung von Standardsensoren (Drehzahl-, Druck und Kraftsensoren) bestehender Teilfunktionen sowie die Einbeziehung vorhandener physikalischer Modellgleichungen zur Beschreibung der Fahrantriebsdynamik [8], weist diese Problemstellung ein hohes Potential zur Entwicklung einer problemangepassten Zweisprachenbeschreibung auf.

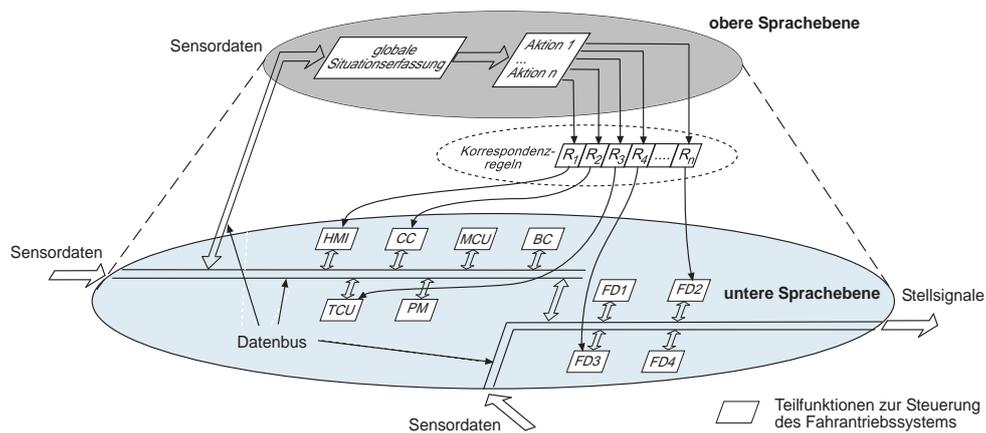


Bild 2: Architektur des hierarchisch strukturierten Systemmodells zur Steuerung des Fahrtriebssystems einer mobilen Arbeitsmaschine bestehend aus zwei Ebenen mit unterschiedlichen Beschreibungssprachen

Ein wesentlicher Schlüssel zur fortschreitenden Automatisierung der beschriebenen Abläufe ist die vollautomatisierte Erkennung der Fahrsituation. Durch den zunehmenden Wegfall manueller Überwachungsfunktionen und der fortschreitenden Substitution hydraulischer Rückfallebenen durch X-by-Wire Technologien ist zum einen das Wissen über diesen globalen Maschinenzustand für die Aufrechterhaltung und Steigerung der funktionalen Sicherheit notwendig. Zum anderen ergibt sich beim Entwurf zukünftiger Energiemanagementsysteme ein zusätzlicher Informationsbedarf, da sich die Motorantriebsleistung je nach der momentanen Fahrsituation aufteilt in den Fahrtrieb und die Arbeitshydraulik. Zur Steigerung der Energieeffizienz ist eine systematische Einbeziehung der situationsabhängigen Zugkraftanforderungen in die automatisierbaren Strategien ebenfalls vielversprechend. In dieser Arbeit beschränken wir uns jedoch auf die zuerst genannte Aufgabenstellung aus dem Bereich der sicherheitsgerichteten Überwachung. Konkret soll für automatisierte Steuerungen hydrostatischer Fahrtriebe eine modellbasierte Sensorredundanz aufgebaut werden. Zur Regelung der Zugkraft und für Getriebschutzfunktionen werden die hydrostatischen Drücke im geschlossenen Kreis gemessen. Zur Sicherstellung der erforderlichen Systemintegrität werden 2-kanalige Systeme zur Selbstprüfung nach [12] aufgebaut. Die Eingangsgrößen, in diesem Fall die Drucksignale, werden dabei durch analytische Redundanz über physikalische Prozessmodelle basierend auf [8] erfasst. Diese bilden die erste Sprachebene. Innerhalb dieser Sprachebene können die Gültigkeitsbereiche dieser Prozessmodelle jedoch nicht festgelegt werden. Aufgrund stark wechselnder Boden- und Geländebeschaffenheit wirken neben den Widerstandskräften (Roll-, Achs- und Luftwiderstand) zusätzliche Zugkräfte. Diese sind fahrsituationsabhängig und wirken zum Beispiel als Hangabtriebskräfte bei Bergauffahrten und als Reaktionskräfte beim Einstechen in ein Haufwerk. Aufgrund unbekannter und unsicherer Systemzustände sind zur Berechnung der hydrostatischen Drücke Störgrößenmodelle basierend auf der Druckaufbaugleichung und der Bewegungsgleichung nicht anwendbar. Nur wenn die Fahrsituation bekannt ist können die jeweils gültigen Prozessmodelle zur Beobachtung der Systemdrücke, wie in [9] abgeleitet, herangezogen werden. Innerhalb der zweiten Sprachebene muss nun entschieden werden, um welche Fahrsituation es sich aktuell handelt. Bild 2 zeigt die Architektur des hier vorgeschlagenen Zweisprachenmodells zur Steuerung des Fahrtriebssystems einer mobilen Arbeitsmaschine.

5 Ergebnisse aus der Anwendung des vorgeschlagenen Verfahrens

5.1 Problemstellung und Generierung des Beispieldatensatzes

Beispiele charakteristischer Fahrsituationen und Einsatzfälle eines Radladers als Vertreter mobiler Arbeitsmaschinen sind das Fahren in der Ebene, das Überwinden von Steigungen und Gefällen, das Einstechen in ein Haufwerk sowie das Abschieben und Verteilen von Material (siehe Bild 3).

Die einzelnen Fahrsituationen können aufgrund von Parameterabhängigkeiten wie Fahrgeschwindigkeit, Steigung, Bodenbeschaffenheit, Körnung und Feuchte des Haufwerksmaterials, etc. stark variieren.

Aus der Problemstellung der Fahrsituationserkennung mobiler Arbeitsmaschinen ergeben sich Anforderungen an den Klassifikator:

- Dem Klassifikator stehen zur Entscheidungsfindung ausschließlich die im Serienfahrzeug vorhandenen Sensorsignale zur Verfügung.
- Der Klassifikator soll in der Lage sein, in Echtzeit, d.h. ohne größere Verzögerungen auf einen Fahrsituationswechsel reagieren zu können.
- Es werden hohe Anforderungen bezüglich der Sicherheit von Entscheidungen des Klassifikators gestellt.
- Der Klassifikator sollte gute Generalisierungsfähigkeiten aufweisen.
- Der Klassifikator sollte Freiheitsgrade zur Anpassung an die gegebene Aufgabenstellung besitzen.

Aus der zweiten Anforderung ergibt sich die Notwendigkeit, dass der Klassifikator zur Entscheidungsfindung nur den aktuellen oder wenige Abtastpunkte verwenden sollte. Die oben genannten sowie weitere Fahrsituationen folgen i.d.R. direkt aufeinander und können sich u.U. auch überlagern, d.h. die Fahrsituationen bestehen nicht aus einer abgeschlossenen Sequenz. Zusätzlich kann jede Situation beliebig lang oder kurz auftreten. Hieraus folgt, dass sich keine charakteristischen Abschnitte wie Anfang, Mitte oder Ende der Fahrsituationen definieren lassen.

Basierend auf diesen Anforderungen wurde sich in dieser Arbeit entschieden, einen zeitkontinuierlichen Bayes-Klassifikator zu entwerfen. Es wurden daher nur zeitkontinuierlich berechenbare Merkmale zur Entscheidungsfindung verwendet (hier gleitende Mittelwerte von Originalzeitreihen und daraus berechneten Zeitableitungen).

Für die Untersuchungen dieser Arbeit standen Daten bestehend aus 11 Messfahrten mit jeweils 21 Zeitreihen zur Verfügung. Jede dieser Messfahrten enthielt verschiedene Sequenzen aufeinander folgender, systematisch durchgeführter Arbeitsprozesse und Überführungsfahrten. Die Beispieldatensätze wurden auf einem Testgelände mit einem Serienfahrzeug aufgenommen.

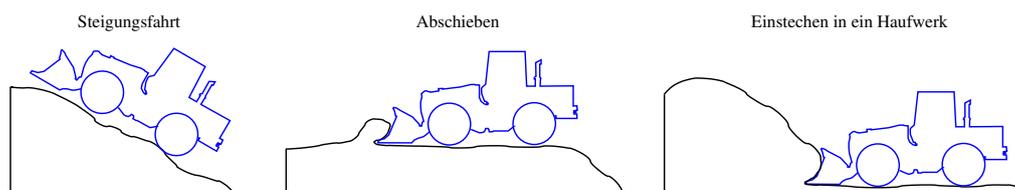


Bild 3: Beispiele von Einsatzfällen eines Radladers

Der automatische Anlernprozeß des Bayes-Klassifikators geschieht datenbasiert. Daher war es notwendig, aus den vorliegenden Daten einen Anlern- und Testdatensatz zu generieren. Hierfür war es notwendig, repräsentative Daten der verschiedenen Fahrsituationen zusammenzustellen und diese Daten verschiedenen Klassen zuzuordnen (die Daten zu labeln). Dieser Arbeitsschritt erwies sich als relativ schwierig, da wie in Abschnitt 2 beschrieben, die Begriffe bzw. die Klassen nicht vorgegeben und daher erst noch definiert werden mussten. Typische Fragen sind:

- Welche Fahrsituationen sind charakteristisch und sind daher zu unterscheiden?
- Welche Fahrsituationen können bzw. sollten zusammengefasst werden?
- In wieviele Fahrsituationen bzw. Klassen sollte der Arbeitsbereich des Radladers sinnvollerweise unterteilt werden?

Antworten darauf waren in einem ersten Schritt zu finden. Erst daraufhin konnten die Messdaten entsprechend gelabelt werden. Diese Labelung warf aufgrund der Tatsache, dass die verschiedenen Fahrsituationen direkt ineinander übergehen, eine weitere Frage auf:

- Wann beginnt bzw. endet eine Fahrsituation und welche dieser Daten sind charakteristisch für die definierte Klasse?

An den Lerndatensatz werden hohe Anforderungen an die Güte und Aussagekräftigkeit der Daten gestellt, um daraus einen brauchbaren Klassifikator anlernen zu können. Daher ist es wichtig, die Beispieldaten sorgfältig zu labeln und nicht repräsentive Daten sowie Ausreißer vorab aus dem Datenmaterial zu entfernen.

Zum Zeitpunkt der Erstellung dieser Arbeit wurden die folgenden sechs zu erkennenden Klassen festgelegt:

- K1: Stillstand
- K2: Fahrt in der Ebene vorwärts
- K3: Fahrt in der Ebene rückwärts (wobei diese Klasse auch das Herausziehen der gefüllten Schaufel aus dem Haufwerk beinhaltet)
- K4: Steigungsfahrt aufwärts
- K5: Steigungsfahrt abwärts
- K6: Einstechen in ein Haufwerk (wobei diese Klasse auch das Abschieben von Erdboden/Material beinhaltet)

Bild 4 zeigt eine Messung, in der Fahrsituationen der Klassen 1, 2, 3, und 6 auftreten und zum Anlernen des Klassifikators verwendet wurden.

5.2 Vorgehensweise

Die 21 zur Verfügung stehenden Originalzeitreihen wurden in einem ersten Schritt zum Glätten tiefpassgefiltert (die Wahl der Grenzfrequenz bedarf besonderer Aufmerksamkeit) und daran anschließend deren erste sowie zweite Zeitableitung berechnet. Dieser neue Datensatz wurde daraufhin in Gait-Cad Version 1.2 [4] importiert. Dort wurden die

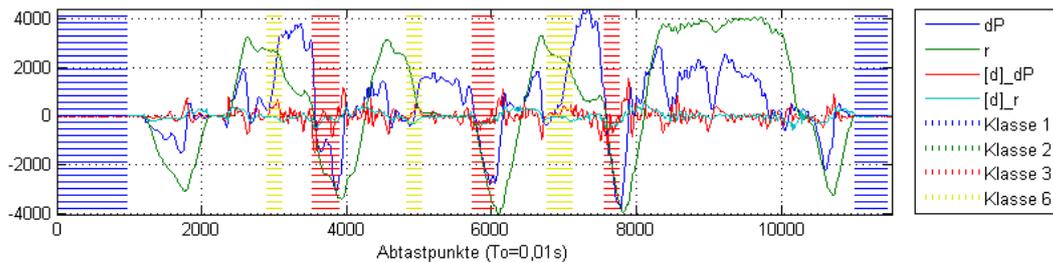


Bild 4: Beispielmessung mit typische Einsatzfällen eines Radladers (Klassen 1, 2, 3 und 6). Nur die schraffierten Abschnitte wurden für das Anlernen des Klassifikators verwendet.

Zeitreihen problemspezifisch weiterverarbeitet, so dass in einem nächsten Schritt Einzelmerkmale (hier Mittelwerte vieler sehr kurzer Zeitabschnitte) berechnet werden konnten². Basierend auf diesen Einzelmerkmalen wurden daraufhin verschiedene systematische Untersuchungen zum automatisierten Entwurf des Bayes-Klassifikators nach dem beschriebenen Entwurfsschema in Abschnitt 3 durchgeführt. In Bild 5 sind die Verteilungen der aus den zuvor mittels univariater Varianzanalyse ausgewählten zehn Merkmalen und mittels Diskriminanzanalyse auf 2 Dimensionen transformierten Merkmalsvektoren des Anlern Datensatzes abgebildet.

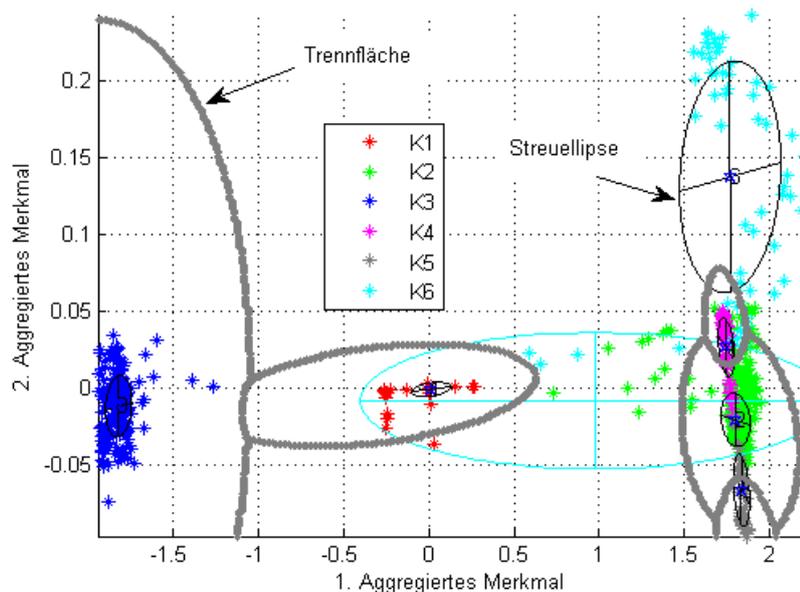


Bild 5: Darstellung des Klassifikationsergebnisses der Anlern Daten nach dimensionsreduzierender Merkmalstransformation mittels Diskriminanzanalyse von zehn auf zwei Merkmale.

Es ist deutlich zu erkennen, dass die Merkmalsvektoren der einzelnen Klassen dicht zusammen liegen und damit die Homogenitätseigenschaft aus Abschnitt 3 relativ gut erfüllt wird. Eine Ausnahme macht K6. Die Merkmalsvektoren streuen deutlich stärker als die der restlichen Klassen. Dies kann damit begründet werden, dass der Prozeß des Einstechens sehr komplexer Art ist und von vielen Parametern (s.o.) abhängt. Weiterhin wur-

²Dieser Ablauf stellt ein mögliches Vorgehen dar. Sämtliche Arbeitsschritte können auch direkt in Gait-Cad Version 1.2 durchgeführt werden.

den auch Abschiebesituationen in die Anlerndaten dieser Klasse aufgenommen³. Darüber hinaus ist zu erkennen, dass die Merkmalsvektoren von K1 und K3 in horizontaler Ausrichtung deutlich entfernt von denen der restlichen Klassen liegen. Da K1 und K3 die Fahrsituationen Stillstand und Fahrt in der Ebene rückwärts kodieren, kann das 1. aggregierte Merkmal als eine Art Geschwindigkeit oder Drehzahl interpretiert werden⁴.

Die Merkmalsvektoren der Klassen K2, K4 und K5 liegen insgesamt relativ kompakt zusammen und überlappen sich in den Randbereichen der vertikalen Ausrichtung. Weiterhin liegen die Merkmalsvektoren der Klasse K6 zum Teil in vertikaler Richtung noch deutlich oberhalb denen der zuvor genannten Klassen. Dies lässt die Interpretation zu, dass es sich bei dem 2. aggregierten Merkmal um eine Art Druck handelt. Es wird somit schnell klar, dass sich die einzelnen Klassen nicht ideal voneinander trennen lassen (Heterogenitätseigenschaft ist nicht erfüllt). Die Überlappungen im Merkmalsraum resultieren daher, dass die einzelnen Fahrsituationen kontinuierlich ineinander übergehen. Sie sind jedoch qualitativ voneinander trennbar. In Bild 5 sind zudem die Trennflächen des angelernten Klassifikators sowie die Streuellipsen der einzelnen Klassen abgebildet. Anhand der Trennflächen ist zu erkennen, dass K1 bis K5 durch relativ enge Gebiete abgegrenzt werden. Lediglich K6 ist nicht eng umgrenzt. Dies bedeutet jedoch, dass alle Merkmalsvektoren, die nicht innerhalb eines der engen Gebiete der Klassen K1 bis K5 liegen, immer K6 zugeordnet werden. Die Ursache hierfür liegt an der verhältnismäßig großen Streuung der Klasse K6 gegenüber denen der restlichen Klassen. Dies wird auch durch die Größenverhältnisse der Streuellipsen ausgedrückt.

5.3 Validierung

Zur Beurteilung der Güte des angelernten Klassifikators bieten sich die zwei Validierungstechniken 1. Kreuzvalidierung und 2. Testen über Testdaten an [5].

Bei der v -fachen Kreuzvalidierung wird der Anlerndatensatz zufällig in $v - 1$ Anteile zum Anlernen sowie einen Anteil zum Testen des Klassifikators aufgeteilt. Dieses Verfahren wird v -mal wiederholt, sodass sämtliche Daten genau einmal zum Testen verwendet wurden. Die mittlere Fehlklassifikationsrate bei 10-facher Kreuzvalidierung (90% Anlerndaten) beträgt 5,9% (Tabelle 1).

Fehlklassifikationsrate in %		
Minimum	Mittelwert	Maximum
3,5	5,9	10,3

Tabelle 1: Ergebnis der 10-fachen Kreuzvalidierung

Für das Testen über Testdaten wurden die Parameter des angelernten Bayes-Klassifikators auf eine Online-Version des Bayes-Klassifikators inklusive Merkmalstransformation übertragen. Dieser Online-Klassifikator trifft zu jedem Abtastzeitpunkt eine Entscheidung. Bild 6 (Mitte) zeigt die A-posteriori-Wahrscheinlichkeiten (Gl. (5)) für Messung Nr. 7 des vorliegenden Datensatzes. Aus Messung Nr. 7 wurden lediglich die Daten der Zeitpunkte $t=96...112s$, $t=148,5...162s$, $t=197,5...206s$ sowie $t=333...343s$ zum Anlernen der Klasse

³Begründung: Diese beiden Situationen konnten bis zum jetzigen Zeitpunkt auch in systematischen Untersuchungen basierend auf den vorliegenden Daten nicht deutlich voneinander getrennt werden. Daher wurde angenommen, dass sich diese Situationen charakteristisch stark ähneln.

⁴K2, K4, K5 und K6 weisen jeweils Geschwindigkeiten größer 0 auf.

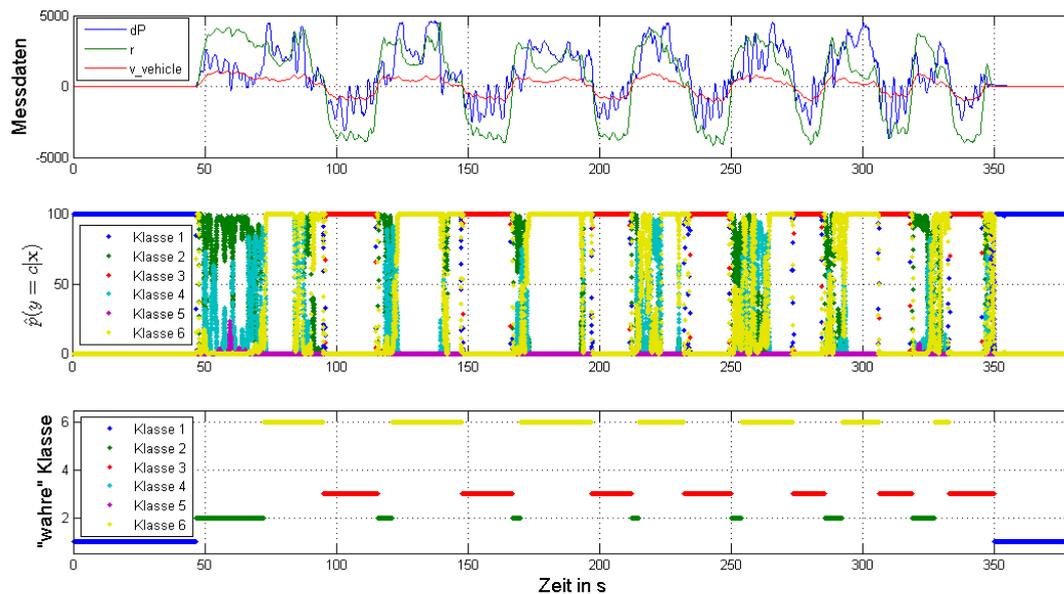


Bild 6: Online-Klassifikation eines Testdatensatzes

K3 verwendet. Die restlichen Daten waren nicht in dem Anlerndatensatz vertreten. Die Klassifikationsentscheidungen erfolgen sehr schnell und stimmen qualitativ mit denen der manuellen Labelung überein. Der Bayes-Klassifikator weist gute Generalisierungsfähigkeiten auf. Durch die Schätzung der klassenspezifischen Kovarianzmatrizen, der A-priori-Wahrscheinlichkeiten sowie der Möglichkeit, anstatt der Wahrscheinlichkeiten die Kosten von Fehlklassifikationen zur Entscheidungsfindung zu verwenden, stehen verschiedene Freiheitsgrade des Bayes-Klassifikators zur Anpassung an die jeweilige Aufgabenstellung zur Verfügung.

Die Klassifikationsentscheidungen stimmten jedoch nicht in allen Messungen mit der manuellen Labelung überein. Sehr häufig wurde z.B. zugunsten K6 entschieden. Die Gründe sind zum einen durch die zuvor diskutierten Punkte (Abschnitte 5.1, 5.2) erklärbar, zum anderen dadurch, dass der Anlerndatensatz noch nicht ausreichend repräsentative Daten enthält. Merkmalsvektoren liegen in Bereichen, die durch die Anlerndaten nicht abgedeckt werden. Im Laufe der Untersuchungen wurde bspw. festgestellt, dass die zwar systematisch durchgeführten Messungen im strengen Sinne doch noch nicht systematisch genug durchgeführt wurden. Es sind z.B. sämtliche Fahrsituationen bei unterschiedlichen Geschwindigkeiten durchzuführen. Dadurch erhöht sich sicherlich auch die Streuung der Merkmalsvektoren innerhalb der einzelnen Klassen. Ansatzweise ist dies in Bild 5 anhand der wenigen Merkmalsvektoren von K2, die nahe an denen von K1 liegen, zu erkennen.

5.4 Schwierigkeiten und weiteres Vorgehen

Als relativ schwierig und zeitaufwendig erwies sich die Zusammenstellung eines repräsentativen Anlerndatensatzes. Besondere Schwierigkeiten bestanden in der Definition der notwendigen und optimalen Anzahl der benötigten Klassen und der dazugehörigen Labelung der Anlerndaten. Dies gilt insbesondere auch für die restlichen Messdaten. Weiterhin lag eine wesentliche Schwierigkeit darin, dass die unterschiedlichen Fahrsituationen direkt ineinander übergehen und daher nicht unmittelbar klar ist, welche Daten

für einen repräsentativen Datensatz notwendig sind (dies hat einen direkten Einfluss auf die zu erwartenden Streuungen der Merkmalsvektoren). Ein Problem bestand auch darin, dass die Fahrsituationen zum Teil nur in einem bestimmten Geschwindigkeitsbereich durchgeführt wurden und somit den relevanten Arbeitsbereich nicht ausreichend abdeckten. All dies führte dazu, dass der Klassifikator relativ oft eine Entscheidung zu Gunsten K6 (Haufwerk) traf. Weiterhin führte die Merkmalsselektion mittels MANOVA in diesem Fall zu keinem befriedigenden Ergebnis. Der Grund lag darin, dass die Güten der besten Einzelmerkmale bereits sehr nahe an eins lagen und ein Abbruchkriterium dazu führte, dass nur sehr wenige Merkmale mit ebenso wenig Informationsgehalt für die weiteren Schritte selektiert wurden. Die Gründe sind noch genauer zu untersuchen.

Weitere systematische Untersuchungen sind anhand der Simulation eines hydrostatischen Fahrtriebes mit Fahrzeugmodell und Geländesimulation geplant. Verschiedene Erweiterungen des Klassifikators sind notwendig, um diesen erfolgreich auf Problemstellungen komplexer Systeme (vgl. Charakterisierungen in Abschnitt 2) anwenden zu können. Zum Beispiel kann dem Klassifikator eine prüfende Instanz angehängt werden, die eine Entscheidung zurückweist oder verwirft, wenn die maximale A-posteriori Wahrscheinlichkeit unterhalb einer problemspezifisch definierten Schwelle liegt [2] oder Merkmalsvektoren vorliegen, die nicht durch die Anlern Daten abgedeckt wurden.

6 Zusammenfassung und Ausblick

In dieser Arbeit wird für die Problemklasse der komplexen technischen Systeme ein Modellansatz bestehend aus zwei Sprachebenen entwickelt. Die erste Sprachebene beschreibt die physikalisch-technischen Zusammenhänge prozessnah mittels einzelner mathematischer Modelle. Durch eine Charakterisierung komplexer technischer Systeme wird aufgezeigt, dass die Modellierung globaler Systemeigenschaften dieser Problemklasse in der ersten Beschreibungssprache nicht möglich bzw. nicht mit vertretbarem Aufwand zu realisieren ist. Daher wird eine zweite Sprachebene eingeführt, mittels deren übergeordneter Sprachkonstrukte (z.B. das Wort Steigungsfahrt) Eigenschaften, die das gesamte System betreffen, formulierbar sind. Anhand einer Fallstudie zur automatischen Fahrsituationserkennung mobiler Arbeitsmaschinen werden wahrscheinkeitsbasierte Methoden zur Bildung dieser übergeordneten Sprachkonstrukte untersucht. Es wird gezeigt, dass sich verschiedene Fahrsituationen als Beispiele globaler Systemeigenschaften qualitativ durch Anwendung eines Bayes-Klassifikators unterscheiden lassen. Der Entwurf des Klassifikators erfolgt durch ein mehrstufiges statistisches Verfahren unter Anwendung der MATLAB-Toolbox Gait-CAD.

Folgenden Arbeitsschritte befassen sich mit der Erweiterung des Lern Datensatzes, mit dem Aufbau einer Simulation eines hydrostatischen Fahrtriebes mit Fahrzeugmodell und Geländesimulation sowie der Erweiterung des Klassifikators um Möglichkeiten, Klassifikationsentscheidungen in bestimmten Situationen zurückzuweisen oder zu verwerfen zu können.

Danksagung: Die Autoren danken den Herren Dr. R. Mikut, Dr. M. Reischl und O. Burmeister für ihre Literaturempfehlungen, Tipps und praktischen Hilfestellungen im Umgang mit Gait-Cad. Diese Unterstützung trug wesentlich zu einem zügigen Einstieg in das Gebiet des Data-Minings bei.

Literatur

- [1] Bertram, T.: Vernetzung von Längs-, Quer- und Vertikaldynamik-Regelung. *Fahrdynamik-Regelung. Modellbildung, Fahrerassistenzsysteme, Mechatronik*, hg. von Isermann, R., S. 345-363. Wiesbaden: Vieweg & Sohn, 2006.
- [2] Burmeister, O.: Analyse von Zeitreihen in der Medizin: Informationsgehalt, Klassifikation und Unsicherheit. *Proc., 16. Workshop Computational Intelligence*, Universitätsverlag Karlsruhe, 2006.
- [3] Mikut, R.; Peter, N.; Malberg, H.; Jäkel, J.; Gröll, L.; Bretthauer, G.; Abel, R.; Döderlein, L.; Rupp, R.; Schablowski, M.; Gerner, H.: *Diagnoseunterstützung für die instrumentelle Ganganalyse (Projekt GANDI)* Forschungszentrum Karlsruhe (FZKA 6613), 2001.
- [4] Mikut, R.; Burmeister, O.; Reischl, M.; Loose, T: Die MATLAB-Toolbox Gait-CAD. *Proc., 16. Workshop Computational Intelligence*, Universitätsverlag Karlsruhe, 2006, S. 114-124.
- [5] Reischl, M.: *Ein Verfahren zum automatischen Entwurf von Mensch-Maschine-Schnittstellen am Beispiel myoelektrischer Handprothesen*. Dissertation, Technische Universität Karlsruhe, Fakultät für Maschinenbau, 2006.
- [6] Richter, K.; Rost J.-M.: *Komplexe Systeme*. Fischer Verlag, Frankfurt am Main, 2004.
- [7] Schmidt, G.: Was sind und wie entstehen komplexe Systeme, und welche spezifischen Aufgaben stellen sie für die Regelungstechnik? *RT-Regelungstechnik*, S. 331-339, Jahrgang 30(10), 1982.
- [8] Schulte, H.: Control-Oriented Modeling of Hydrostatic Transmissions using Takagi-Sugeno Fuzzy Systems. *IEEE International Conference on Fuzzy Systems*, 23-26 Juli 2007, London, UK, S. 2030-2035.
- [9] Schulte, H.: Robuster Beobachterentwurf für Takagi-Sugeno Fuzzy Systeme zur Überwachung hydrostatischer Fahrtriebe. *Proc., 18. Workshop Computational Intelligence*, Universitätsverlag Karlsruhe, 2008.
- [10] Schwarz, R., Rieth, P.: Global Chassis Control - Systemvernetzung im Fahrwerk. *Autoreg 2002*, 14-16 April 2002, Mannheim, S. 415-451.
- [11] Vieler, H., Odenthal, D., Smakman, H., Richter, T., Köhn, P.: Integrated Chassis Management - ein Weg zur Integrierten Fahrdynamikregelung. *Autoreg 2008*, 12-13 Februar 2008, Baden-Baden, S. 37-48.
- [12] Norm DIN EN: *Funktionale Sicherheit sicherheitsbezogener elektrischer/elektronischer/programmierbarer elektronischer Systeme*. Norm DIN EN 61508. Berlin Beuth Verlag, 2001.

Data-Based Activity Analysis and Interpretation of Small Antibacterial Peptides

Ralf Mikut¹, Markus Reischl¹, Anne S. Ulrich², Kai Hilpert²

¹Forschungszentrum Karlsruhe GmbH, Institut für Angewandte Informatik,

²Forschungszentrum Karlsruhe GmbH, Institut für Biologische Grenzflächen (IBG2),

E-Mail: {ralf.mikut, markus.reischl}@iai.fzk.de, {anne.ulrich, kai.hilpert}@ibg.fzk.de

1 Introduction

Live threatening multi-drug resistant bacterial infections are one of the most serious challenges for human and animal health care. More and more pathogens are developing resistance against common antibiotics. Seeking novel antibiotics for treating multi-drug resistant bacterial infections has become one of the top priorities for many research teams.

Cationic host defense (also called antimicrobial) peptides represent a ubiquitous response in nature for overcoming microbial infections [1]. These peptides have emerged as central components of the innate defenses of both lower and higher organisms. They kill bacteria very rapidly compared with conventional bactericidal antibiotics [2]. Although they have an enormous variety of sequences and structures, they share certain common features. Cationic antimicrobial peptides are 12-50 amino acids in length with at least two excess positive charges due to lysine (K) and arginine (R) residues and contain around 50% hydrophobic amino acids, however they are structurally diverse. With the multitude of cationic peptide sources, structures and spectra of activity come a number of complex and controversial structure-function theories attempting to describe and explain peptide modes of interaction that kill bacteria (see e.g. [3]).

A novel peptide array technique of high-throughput screening and optimizing short broad-spectrum cationic antimicrobial peptides has developed recently [4]. This assay is based on the SPOT technology, a peptide synthesis strategy on cellulose, combined with a bacterium, where a luminescence gene cassette was introduced. Using this method, thousands of peptides can be screened per year.

This immense data amount needs to be analyzed in order to extract critical information to understand the "rules" to synthesize an antimicrobial peptide. A quantitative structure-activity relationship approach (QSAR) [5–9] can support this data analysis. QSAR analyses are influenced by the set of descriptors, choice of statistical learning technique, and the quality and quantity of the dataset. Up to now, hard to interpret descriptors and techniques (e.g. Artificial Neural Networks) are the most successful [10]. The success of such prediction is of great advantage when screening *in silico*.

Here, we focus on easy to interpret descriptors to learn more about the mode of action and allow to do knowledge-based drug design. This paper is organized as follows: In Section 2, the used datasets and analysis methods are presented. The results for feature relevances and regression models are shown in Section 3 and discussed in Section 4.

2 Material and Methods

2.1 Experiments and Datasets

Peptides are sequences of amino acids, the amino acids are coded by characters A (alanine) - Y (tyrosine) using a standardized system. The data used for this analysis is taken from [9]. All peptides are synthesized on cellulose and subsequently cleaved by ammonia according to the recently published protocol [11]. All following investigations base on three different Datasets A, B, and C with peptides of nine amino acids each in a sequence.

- The first Dataset A consists of 200 fully randomly generated peptides. Here, all amino acids A-Y except C have a nearly uniform distribution resulting in frequencies of amino acids around $\frac{1}{19}$.
- The Dataset B with 943 peptides is semi-randomly generated using a modified amino acid distribution generated by expert rules to increase the activity. These rules prefer amino acids like K, R, and W (tryptophane) and avoid amino acids like D (aspartic acid) and E (glutamic acid).
- The third Dataset C with 500 peptides is designed using the amino acid distribution of active peptides from Dataset B. The main results are increased frequencies of R and W in comparison to Dataset B.

The peptides were stepwise diluted and the antimicrobial effect was evaluated by using a bacteria strain (*Pseudomonas aeruginosa*) that contains a gene cassette enabling the bacteria to luminescence. This luminescence is dependent on FMN, a marker of the energy level of the bacteria. If the energy level of the bacteria is disturbed, the luminescence is reduced [4, 12].

The peptides are organized in plates with twelve peptides: ten peptides for investigation, at least one positive control peptide (Bac2A, k left in Fig. 1) and an optional negative control. For each of the twelve peptides, C different measurements ($i_c = 1, \dots, C$, here: $C = 8$) with a relative peptide concentration

$$c(i_c) = \begin{cases} \frac{1}{2^{i_c-1}} & i_c < C \\ 0 & i_c = C \end{cases} \quad (1)$$

are used resulting in relative concentrations $1, \frac{1}{2}, \frac{1}{4}, \dots, \frac{1}{64}, 0$. For each peptide ($n = 1, \dots, N$) and each measurement, the luminescence $L(c, n)$ is measured after four hours. The expected outcome is a decreasing luminescence $L(c, n)$ with increasing concentrations c . In the datasets, luminescence values L between 13 and 63669 occur.

2.2 Scenario

The chosen scenario for the interpretation and analysis of peptide activities is shown in Fig. 2. Firstly, a new dataset with promising peptides will be synthesized and the related luminescence experiment is performed.

The results will be analyzed leading to

- an activity model to evaluate unknown peptides,

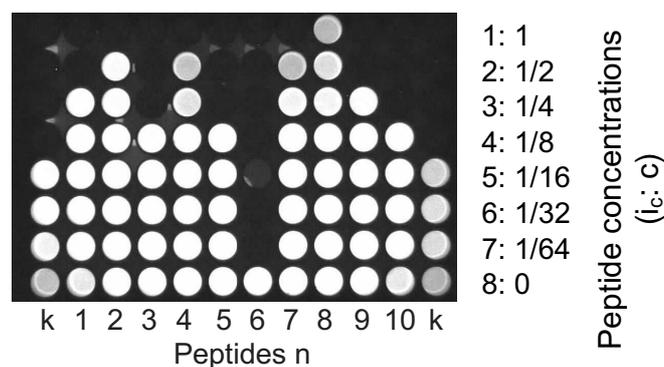


Figure 1: Luminescence values for different peptides and two control peptides with different concentrations

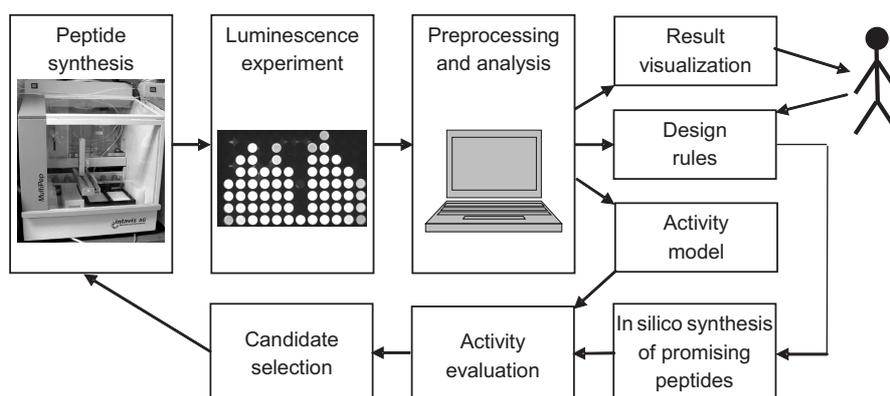


Figure 2: Scenario for the interpretation and analysis of peptide activities

- a comprehensive result visualization (e.g. lists of promising features, low-dimensional projections, results of regression models), and
- semi-automatically found design rules for more active peptides.

Most existing work for the analysis has been done for activity models [7–9]. They are sufficient for some design procedures e.g. by a stochastic in silico generation of peptide candidates and a synthesis of the most promising ones. For this strategy, a black-box behavior of the activity model with many input features and a complex processing is tolerable.

In this paper, we focus on visualization in low-dimensional feature spaces and a semi-automatic rule design. This information can support the understanding of mode of action of these cationic host defense peptides and lead to knowledge-based drug design. The aim is an enhanced interpretability of the results and the opportunity for an interactive design of new promising peptides. The necessary steps including a generation of corresponding sequences, derived sequences, segmentation, extracted features, their evaluation, and the regression model design are shown in Fig. 3. They are similar to extraction steps for time series (with the analogy sequence = time series) as proposed in [13].

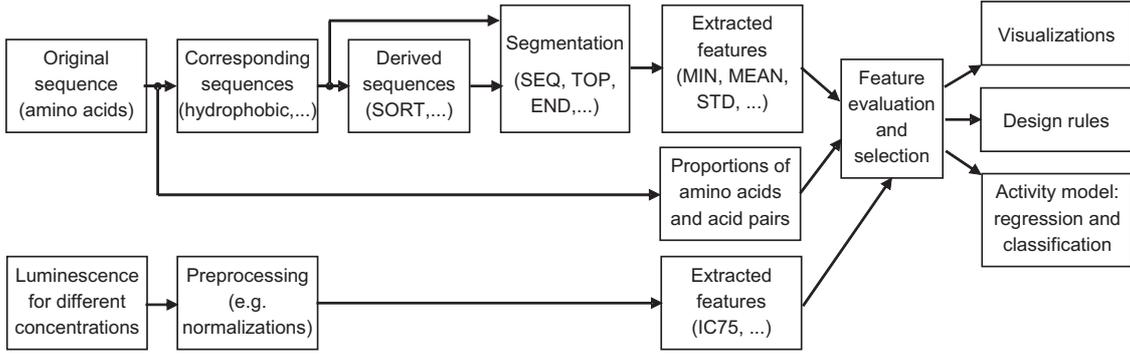


Figure 3: Steps for the preprocessing and analysis

2.3 Preprocessing

The experiments are very sensitive to small differences in bacteria and peptide concentration and purity, live conditions of bacteria (e. g. temperature, humidity) etc. Some preprocessing steps are done to exclude implausible experiments and to increase the data quality.

In a first step, peptides with unusual curve forms of the luminescence are excluded. As discussed before, the luminescence $L(c, n)$ should not increase with increasing concentrations c of a peptide (see examples for usual curves in Fig. 4a). We exclude all peptides n with at least one $i_c > 1$ with

$$L(c(i_c), n) > \alpha_{rel} \cdot L(c(i_c - 1), n) \text{ AND} \quad (2)$$

$$L(c(i_c), n) > \alpha_{abs} + L(c(i_c - 1), n). \quad (3)$$

The condition (2) excludes peptides with a relevant relative luminescence increase (here: parameter for relative increase $\alpha_{rel} = 1.3$) and the condition (3) guarantees a minimal difference (here: parameter for absolute increase $\alpha_{abs} = 300$) to ignore small oscillations for low luminescence values. With these conditions, 34 peptides (15/14/5 for the three datasets) are classified as luminescence artifacts and will be excluded from further investigations (Table 1).

The preprocessing of the luminescence experiment includes an estimation of the concentration c with a distinct proportion of the luminescence value $IC_x(n)$, e.g. $x = 75$: a reduction of 75% of the maximal luminescence value $L_{max}(n)$. The value is based on a normalization of the luminescence values L for each peptide ($n = 1, \dots, N$) to its maximal value $L_{max}(c, n)$ to avoid artifacts by different absolute values:

$$L(IC_x(n), n) = \frac{100 - x}{100} L_{max}(n) \text{ with } L_{max}(n) = \max_{i_c} L(c(i_c), n). \quad (4)$$

This value is computed by a regression model because only values for discrete $c(i_c)$ are known.

In a next step, the relative value of this concentration is computed in comparison to the positive control group in the same plate $IC_x(n_p)$:

$$RelIC_x(n) = \frac{IC_x(n)}{IC_x(n_p)}. \quad (5)$$

With this approach, each positive control peptide has a value of $RelIC_x(n) = 1$ and peptides with lower values are more active than the control peptide. The optimal value is zero.

Finally, the activity indication is subdivided into four classes (fully active, similar to positive control, worse than control, inactive) based on fix intervals for $RelIC_{75}$, e.g. fully active for $RelIC_{75} \leq 0.5$. Peptides with

$$\min_{i_c} L(c(i_c), n) > \alpha_{inactive} \cdot L_{max}(n) \quad (6)$$

are classified as inactive independent of their $RelIC_x$ value (here: parameter for minimal luminescence reduction $\alpha_{inactive} = 0.5$). This last preprocessing step avoids problems with relative weak reductions for some positive controls.

All resulting values for the datasets are summarized in Table 1 and Table 2.

Dataset	A	B	C
Numbers of peptides N	200	943	500
- Detected luminescence artifacts	15	14	5
- Consistent luminescence values	185	929	495
- Fully active: $RelIC_{75} \leq 0.5$	0 (0.0 %)	30 (3.2 %)	16 (3.2 %)
- Similar to positive control: $0.5 < RelIC_{75} \leq 2$	0 (0.0 %)	169 (18.2 %)	133 (26.7 %)
- Worse than control: $2 < RelIC_{75} \leq 20$	98 (53.0 %)	635 (68.4 %)	302 (61.1 %)
- Inactive: $20 < RelIC_{75}$ or inactive acc. to (6)	87 (47.0 %)	95 (10.2 %)	44 (8.9 %)
Positive control (Bac2A, $RelIC_{75} = 1$)	20	95	50
Negative control	20	32	50
Total	240	1140	600

Table 1: Number of peptides and class distribution of the Datasets A, B, and C

	Positive Control	Dataset A	Dataset B	Dataset C
IC_{75}	0.13 (0.05 – 0.29)	1.47 (0.31 – 2.98)	0.57 (0.01 – 1.18)	0.44 (0.03 – 7.90)
$RelIC_{75}$	1.00 (1.00 – 1.00)	13.09 (2.15 – 33.72)	3.57 (0.07 – 16.00)	4.26 (0.21 – 63.38)
$\log_{10} RelIC_{75}$	0.00 (0.00 – 0.00)	1.12 (0.33 – 1.53)	0.55 (-1.15 – 1.20)	0.63 (-0.68 – 1.80)

Table 2: Median, minimal and maximal values for IC_{75} and related values of the control peptides and the Datasets A, B, and C

The Dataset A does not contain any active peptides. Consequently, the chance for the generation of active peptides by means of a fully random library is rather small. The Datasets B and C show an increased number of active peptides. However, there is no clear improvement in Dataset C in comparison to Dataset B.

2.4 Feature Extraction

The main intention of the following feature extraction is the transformation of an amino acid sequence into functional descriptions. It is one possible solution to find clearer

characteristics for some functional sequence elements (e.g. based on charge or hydrophobicity) instead of apparently different distributions of functional similar amino acids.

For each peptide, corresponding sequences with different molecular descriptors are computed:

- the molecular mass and the isoelectric point of amino acids (2 sequences),
- z_j values for amino acids resulting from a principal component analysis (two versions with $j = 1, \dots, 3$ proposed by Hellberg [5] or $j = 1, \dots, 5$ proposed by Sandberg [14]) (8 sequences),
- different hydrophobicity and hydrophilicity scales proposed by Goldman-Engelman-Steitz, Hopp-Woods (see example in Fig. 4b), Kyte-Doolittle, Janin, Clothia and Eisenberg-Weiss (following the values from [15]), and the Interface, Octanol and Octanol-Interface scale from [16] (9 sequences), and
- a charge scale with a value of 1 for the amino acids arginine (R) and lysine (K), of 0.5 for histidine (H), and of -1 for aspartic acid (D) and glutamic acid (E).

In the next step, derived sequences are computed using a sorting by ascending values for each sequence (SORT). The intention is the search for special distribution characteristics.

To look for position dependent information, a segmentation is done by using the whole sequence (SEQ), the first two positions (TOP), the last two positions (END), or all positions in the middle (MID) of a sequence (3-7).

The feature extraction uses five features for the minimal (MIN), maximal (MAX), mean value (MEAN), the range (ROM) and the standard deviation (STD EM). With these features, a complete sequence or a segment is transformed to single features. These single features are more feasible for the following feature evaluation techniques. Because the values for SEQ are identical for the original and the sorted sequence, the features for SEQ SORT are not computed.

In addition, the proportions of the different amino acids (20 features) and of so-called 2-mers with neighbor pairs ($20 \cdot 20 = 400$) are computed directly from the original sequence. These features look for unique characteristics of single amino acids or amino acid pairs.

As a result, a total amount of

$$20 \cdot (5 \cdot 4 + 5 \cdot 3) + 20 + 400 = 1120 \quad (7)$$

features are computed resulting from 20 sequences, five features and four segments for the original and five features and three segments for the sorted sequences.

The main intention of this large set of candidate features is an overall evaluation of potentially relevant features.

2.5 Feature Evaluation and Selection

The aim of the feature evaluation and selection is the search for a small set of relevant features. An accuracy measure Q_{Acc} evaluates the solution of the classification or regression problem by measures resulting from a univariate (ANOVA) or multivariate (MANOVA) analysis of variance, and univariate and multivariate linear regression coefficients [17, 18].

All multivariate measures evaluate a new feature set generated by adding the l -th feature to a preselected feature set $\{\mathcal{I}_k\}$. This set is empty for univariate measures.

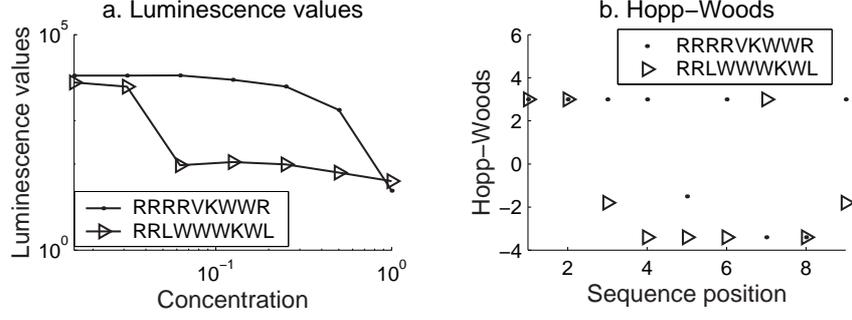


Figure 4: Example for two different peptides (Dataset B) (RRRRVKWWR, RRLWWWKWL):
a. luminescence $L(c(i_c), n)$ vs. the concentration c ,
b. hydrophobicity after Hopp-Woods vs. the amino acid number in the sequence

Each measure works on a set of selected features $\mathcal{I}_k = \{l_1, \dots, l_k\}$ from the original feature space with $l_i \in \{1, \dots, 1120\}$ resulting in a feature vector

$$\mathbf{x}_{\mathcal{I}_k}[n] = (x_{l_1}[n] \ \dots \ x_{l_k}[n])^T \quad (8)$$

and a feature matrix

$$\mathbf{X}_{\mathcal{I}_k} = \begin{pmatrix} \mathbf{x}_{\mathcal{I}_k}[1] \\ \dots \\ \mathbf{x}_{\mathcal{I}_k}[N] \end{pmatrix}. \quad (9)$$

ANOVA and MANOVA use the measure

$$Q_{Acc, \mathcal{I}_k, (M)ANOVA} = 1 - \frac{\det(\mathbf{W}(\mathbf{X}_{\mathcal{I}_k}))}{\det(\mathbf{T}(\mathbf{X}_{\mathcal{I}_k}))} \quad (10)$$

with

$$\mathbf{T}(\mathbf{X}_{\mathcal{I}_k}) = N \cdot \mathbf{S}(\mathbf{X}_{\mathcal{I}_k}) \quad \mathbf{W}(\mathbf{X}_{\mathcal{I}_k}) = \sum_{c=1}^C N_c \cdot \mathbf{S}_c(\mathbf{X}_{\mathcal{I}_k}) \quad (11)$$

with the determinant \det , the total variance matrix \mathbf{T} (or SSCP: sum of squares and cross products), the number of classes (here: $C = 4$), the total number of peptides N or the number of peptides in class c (N_c), the estimated covariance matrices for all peptides \mathbf{S} and the peptides in class c (\mathbf{S}_c). For a perfect model, the value $Q_{Acc, \mathcal{I}_k, (M)ANOVA}$ is one, a value of zero means that the model contains no information. For ANOVA, \mathcal{I}_k contains only one element l_1 .

The univariate and multivariate regression models measure the ability of one feature resp. a chosen group of features to predict the logarithmic value of the relative value of the IC_{75} concentrations from (5) using a linear model e.g. for a linear model with

$$\hat{y}[n] = a_0 + \sum_{l \in \mathcal{I}_k} a_l \cdot x_l[n]. \quad (12)$$

The vector \mathbf{y} consists of

$$\mathbf{y} = \begin{pmatrix} \log_{10} RelIC_{75}[1] \\ \dots \\ \log_{10} RelIC_{75}[N] \end{pmatrix}, \quad (13)$$

and its estimation is computed by

$$\hat{\mathbf{y}}(\mathbf{X}_{\mathcal{I}_k}) = \begin{pmatrix} \log_{10} \widehat{RelIC}_{75}(\mathbf{x}_{\mathcal{I}_k}[1]) \\ \dots \\ \log_{10} \widehat{RelIC}_{75}(\mathbf{x}_{\mathcal{I}_k}[N]) \end{pmatrix}. \quad (14)$$

The accuracy measure for the regression model (REG) computes the relative improvement with this estimation in comparison to a trivial model with the mean value of \mathbf{y} in $\bar{\mathbf{y}}$:

$$Q_{Acc, \mathcal{I}_k, REG} = R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{(\mathbf{y} - \hat{\mathbf{y}}(\mathbf{X}_{\mathcal{I}_k}))^T \cdot (\mathbf{y} - \hat{\mathbf{y}}(\mathbf{X}_{\mathcal{I}_k}))}{(\mathbf{y} - \bar{\mathbf{y}})^T \cdot (\mathbf{y} - \bar{\mathbf{y}})}. \quad (15)$$

Again, the value is one for a perfect model and zero if the model contains no information.

An additional preference measure $Q_{Pref,l}$ for each feature prefers features with a better interpretability [19]. For the peptide dataset, all z_j -values are downgraded by 0.7, all TOP and END values by 0.5, all MID values by 0.3 to prefer features with a better interpretability and robustness against disturbances as e.g. proportions or whole sequences.

The fusion is done by

$$Q_{Total, \{\mathcal{I}_k, l\}} = Q_{Total, \mathcal{I}_k} + (Q_{Acc, \{\mathcal{I}_k, l\}} - Q_{Acc, \mathcal{I}_k}) \cdot Q_{Pref, l}^\alpha \quad (16)$$

where the exponent $\alpha > 0$ (here a heuristically chosen value of $\alpha = 0.1$) tunes the compromise between accuracy and interpretability. A value of $\alpha = 0$ ignores interpretability aspects completely, larger values mean an increasing preference of interpretable solutions.

Some examples for corrected univariate features are

- STD EM SEQ Hopp-Woods with $Q_{Pref, l}^\alpha = 1^{0.1} = 1$ (no downgrades),
- MEAN SEQ z1 [Sandberg98] with $Q_{Pref, l}^\alpha = 0.7^{0.1} = 0.965$ (downgraded by the z_j -value),
- STD EM MID Hopp-Woods SORT ZR with $Q_{Pref, l}^\alpha = 0.3^{0.1} = 0.887$ (downgraded by the mid segment), and
- STD EM MID z1 [Sandberg98] with $Q_{Pref, l}^\alpha = (0.7 \cdot 0.3)^{0.1} = 0.856$ (downgraded by the z_j -value and by the mid segment).

The main effect is a slight preference for better interpretable and usually more robust features if many features have similar accuracy measures.

2.6 Activity Models by Regression

The generation of feasible classification models is extremely difficult due to the extremely overlapping classes. Consequently, the classification is only used for feature evaluation and visualization purposes. All following activity models bases on the regression approach.

For each dataset, the best three features are added stepwise using a multivariate regression analysis with (15) for (12). Using these features, regression models are designed by

- an Artificial Neural Network (Multi Layer Perceptron, one hidden layer with 5 neurons, tansig functions in the hidden layer, learning with the Levenberg-Marquardt method),

- a Nearest Neighbor Model (10 neighbors, fusion of the regression result by weighted sum of the output variable of all neighbors with weights one for the nearest and weight 0.1 for the most distant neighbor), and
- polynomial models of the degrees one, two (with x_l^2 terms), and three with (with x_l^3 terms) with a weight correction to prefer classes with lower probabilities and a maximal number of four terms.

The analysis was done using the open source MATLAB toolbox Gait-CAD [20]¹.

2.7 Evaluation of Activity Models

A first natural measure is the use of (15) for each activity model using the same dataset (recall) or datasets not used before (validation). An ideal model would forecast the real $\log_{10} RelIC_{75}$ values for each unknown peptide in the validation dataset.

Due to the high-throughput approach with a possible in silico generation of many peptide candidates, a rough estimation of this value would support the candidate selection (see Fig. 2). One possible strategy is to evaluate all candidates using the activity model and to select the best ones for a synthesis.

Other valuable measures base on characteristic values in the synthesized dataset for the best peptides after in silico generation. Two examples are

- the median value for the top 10 % peptides in a new dataset (here: Dataset C) \overline{RelIC}_{75} by applying an existing activity model (here: based on Dataset A+B) or
- the percentage of active peptides with $RelIC_{75} < 1$ under a predefined number of peptides sorted by the predicted activity [21].

Such measures give realistic outlines to the expected results for an in silico generation of future datasets.

3 Results

3.1 Feature Evaluation and Selection

The best ANOVA features for all datasets and the four class problem from Table 1 are shown in Table 3. For a MANOVA analysis, the following feature sets for $\{I_k\}$ are chosen by stepwise adding the best features:

- Dataset A+B: STD EM SEQ Hopp-Woods ($Q_{Total} = 0.27$), MEAN SEQ Isoelectric Point (0.32), all other features do not improve the MANOVA value significantly (<0.33),
- Dataset C: MEAN SEQ Interface scale [White99] (0.22), STD EM SEQ Molecular mass (0.29), STD EM SEQ Janin (0.34),
- Dataset A+B+C: STD EM SEQ Hopp-Woods (0.23), MIN SEQ Isoelectric Point (0.26), and MEAN MID z1[Hellberg87] SORT ZR (0.29).

The linear regression coefficients provide similar results for each dataset (Table 4). The following feature sets for $\{I_k\}$ are chosen by stepwise adding the best features with a multilinear regression approach:

¹<http://www.iai.fzk.de/projekte/biosignal/index.html> → Download

- Dataset A+B: STD EM SEQ Hopp-Woods ($Q_{Total} = 0.27$), MEAN SEQ Isoelectric Point (0.50), Proportion of R (0.54),
- Dataset C: MEAN SEQ Interface scale [White99] (0.28), MIN SEQ Molecular mass (0.36), MAX MID Hopp-Woods SORT ZR (0.42),
- Dataset A+B+C: STD EM SEQ Hopp-Woods (0.23), MIN SEQ Charge (0.25), and MEAN MID z1[Hellberg87] (0.27).

Feature	DS A+B	DS C	DS A+B+C
MEAN SEQ Interface scale [White99]	0.04 (241)	0.22 (1)	0.10 (33)
MEAN SEQ Molecular mass	0.18 (7)	0.16 (15)	0.17 (2)
MEAN SEQ Octanol scale [White99]	0.01 (514)	0.20 (3)	0.06 (135)
MEAN SEQ z1[Hellberg87]	0.03 (312)	0.21 (2)	0.08 (58)
STD EM MID Hopp-Woods SORT ZR	0.22 (3)	0.06 (114)	0.15 (4)
STD EM SEQ Eisenberg-Weiss	0.22 (2)	0.01 (664)	0.15 (3)
STD EM SEQ Hopp-Woods	0.27 (1)	0.17 (5)	0.23 (1)
Proportion of W	0.11 (58)	0.17 (6)	0.20 (6)
Proportion of WW	0.03 (279)	0.11 (40)	0.07 (84)

Table 3: Q_{Total} for the best features (upper part) and some other selected features (lower part) based on an ANOVA analysis, in parenthesis: feature ranking in the dataset

Feature	DS A+B	DS C	DS A+B+C
MEAN SEQ Interface scale [White99]	0.03 (304)	0.28 (1)	0.09 (25)
Proportion of W	0.13 (89)	0.24 (3)	0.14 (5)
STD EM SEQ Eisenberg-Weiss	0.36 (2)	0.00 (511)	0.15 (2)
STD EM SEQ Goldman-Engelman-Steitz	0.33 (3)	0.00 (660)	0.13 (8)
STD EM SEQ Hopp-Woods	0.38 (1)	0.25 (2)	0.21 (1)
STD EM SEQ Isoelectric Point	0.29 (7)	0.03 (242)	0.14 (3)

Table 4: Q_{Total} with $Q_{Acc,\{I\},REG} = R^2$ (15) for the best features based on univariate regression models to predict $\log_{10}RelIC_{75}$, in parenthesis: feature ranking in the dataset

In general, the most common features like proportion of amino acids or 2-mers (amino acid pairs) have lower relevances. The best features of this subset are Proportion of W and Proportion of WW (Table 3).

A two-dimensional feature space gives the best insights for a well-interpretable visualization of the class distribution for the Dataset A+B+C. Features with few discrete values (as e.g. the second MANOVA feature MIN SEQ Isoelectric Point) should be avoided because the symbols with the peptides might overlap very strongly. Instead of MIN SEQ Isoelectric Point, the first non-discrete feature MEAN SEQ z1[Hellberg87] (MANOVA: 0.25) was chosen (Fig. 5).

Fully active and rather active peptides show high values for STD EM SEQ Hopp-Woods and medial values for MEAN SEQ z1[Hellberg87]. Peptides of the classes "inactive" and "worse than control" can be found for lower values of STD EM SEQ Hopp-Woods. However, all classes overlap.

The datasets differ in the mean values and standard deviations of many features. As an example, the best feature STD EM SEQ Hopp-Woods has mean values and standard deviations of 1.71 ± 0.35 (DS A), 2.14 ± 0.27 (DS B), and 2.53 ± 0.26 (DS C). However,

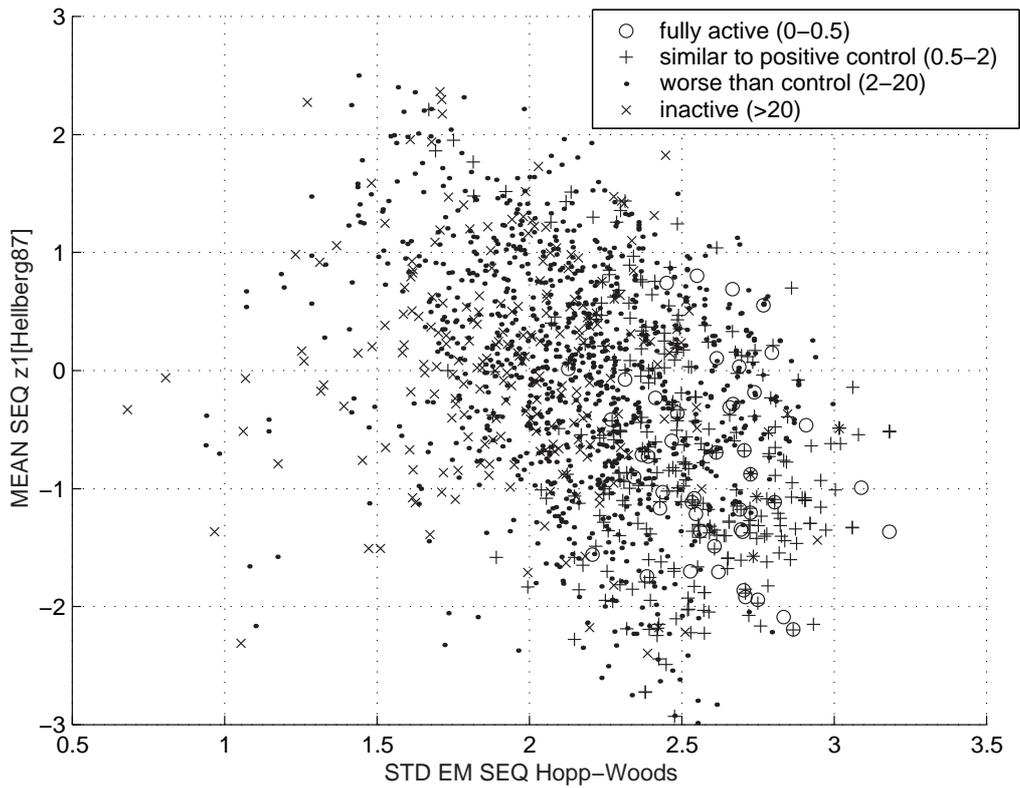


Figure 5: Best features after MANOVA for the Dataset A+B with labels for the five class problem

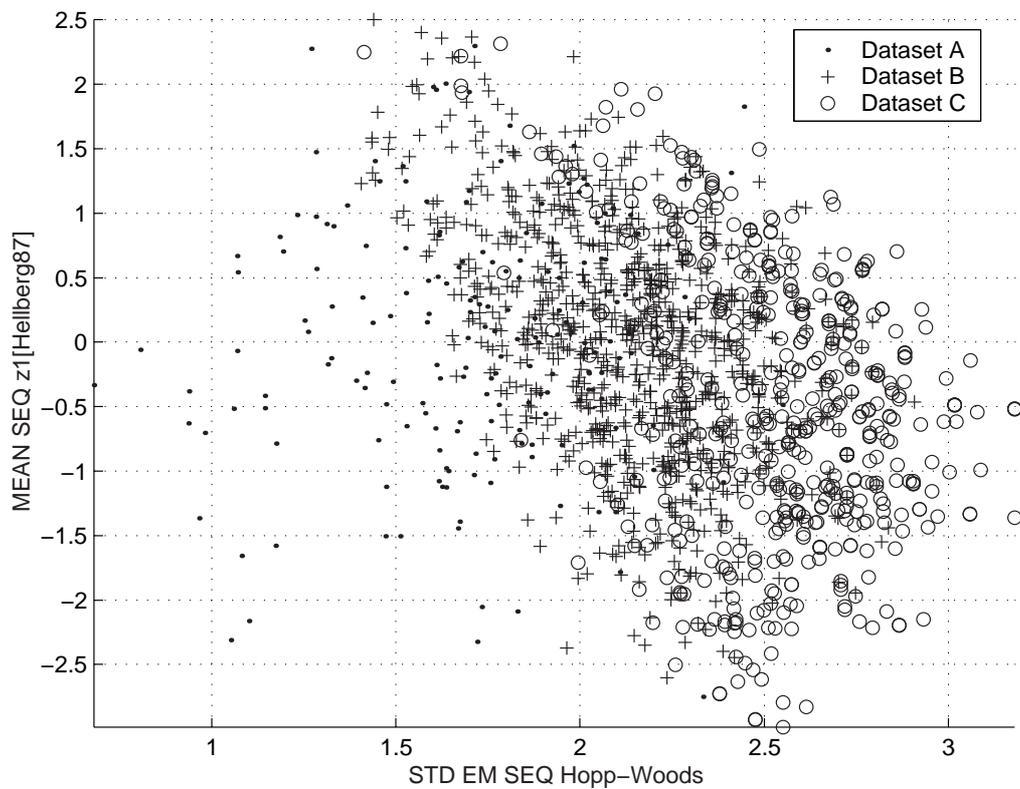


Figure 6: Best features after MANOVA for the Dataset A+B with labels for the five class problem

some but not all mean values and standard deviations for the class of fully active peptides differ as well. There are nearly identical values, e.g. STD EM SEQ Hopp-Woods with 2.59 ± 0.18 (DS B) vs. 2.63 ± 0.25 (DS C), but also large differences MEAN SEQ z1[Hellberg87] -0.60 ± 0.78 vs. -1.27 ± 0.60 (DS C). This effect is visible in Fig. 6 for the same features as in Fig. 5, but labeled for the different datasets instead of the different classes. A comparison between both figures shows that the region with a higher density of fully active peptides is especially covered by the Datasets B and C.

Such large differences explain the differences between the feature rankings. The typical assumption of data mining approaches is violated that different data sets are stochastic samples of the same distribution.

3.2 Regression models

The results of the regression models are shown in Table 5. The evaluation is done by the coefficient of determination R^2 and the median value of $RelIC_{75}$ for the top 10% of peptides in a dataset with the best values of the estimated activity model. These best values are computed by the lowest estimations for $\log_{10}RelIC_{75}$ in a regression model. All models improve the results of an estimation by chance given by a zero correlation for R^2 or the median value in the dataset. Polynomials of degree one are too simple, all other approaches give comparable results.

Model	A+B \rightarrow A+B		A+B \rightarrow C (Test)		C \rightarrow C		A+B+C \rightarrow A+B+C	
	R^2	\overline{RelIC}_{75}	R^2	\overline{RelIC}_{75}	R^2	\overline{RelIC}_{75}	R^2	\overline{RelIC}_{75}
Poly 1	0.53	0.99	0.11	1.79	0.42	0.90	0.27	1.46
Poly 2	0.58	0.99	0.15	1.49	0.42	0.91	0.27	1.40
Poly 3	0.58	0.99	0.15	1.46	0.45	0.90	0.27	1.40
k -NN	0.61	0.96	0.17	1.28	0.47	0.90	0.32	1.32
ANN	0.62	0.92	0.16	1.49	0.50	0.90	0.32	1.34
By chance	0.00	4.00	0.00	4.26	0.00	4.26	0.00	4.03

Table 5: $Q_{Acc, \{I_k\}, REG} = R^2$ from (15) of five different regression models for the prediction of $\log_{10}RelIC_{75}$ with the best three features by a multivariate regression model. Abbreviations: coefficient of determination R^2 , \overline{RelIC}_{75} : median value for the estimated top 10 % of the dataset, Poly n : polynomial model of order n , k -NN: k -Nearest Neighbor Model, ANN: Artificial Neural Network, A+B \rightarrow C: train on Dataset A+B and apply to Dataset C etc.

All models have problems to forecast the Dataset C from the model of Dataset A+B. The main reason is not an overfitting due to complex models but the structural difference between the datasets. An argument against the overfitting are the results for 10 trials of a 2-fold crossvalidation on Dataset A+B for Poly 3 ($R^2 = 0.54 \pm 0.00$), k -NN ($R^2 = 0.56 \pm 0.01$) and ANN ($R^2 = 0.47 \pm 0.02$).

An analysis of the simple univariate regression models in Fig. 7 gives a hint to understand this fact. On the left hand side, the model detects the lower values of \overline{RelIC}_{75} resp. higher activities with increasing values of STD EM SEQ Eisenberg-Weiss. On the right hand side, Dataset C is distributed on a smaller interval of values for STD EM SEQ Eisenberg-Weiss. The model is not wrong but does not improve the estimation in comparison to a mean value estimation relevantly.

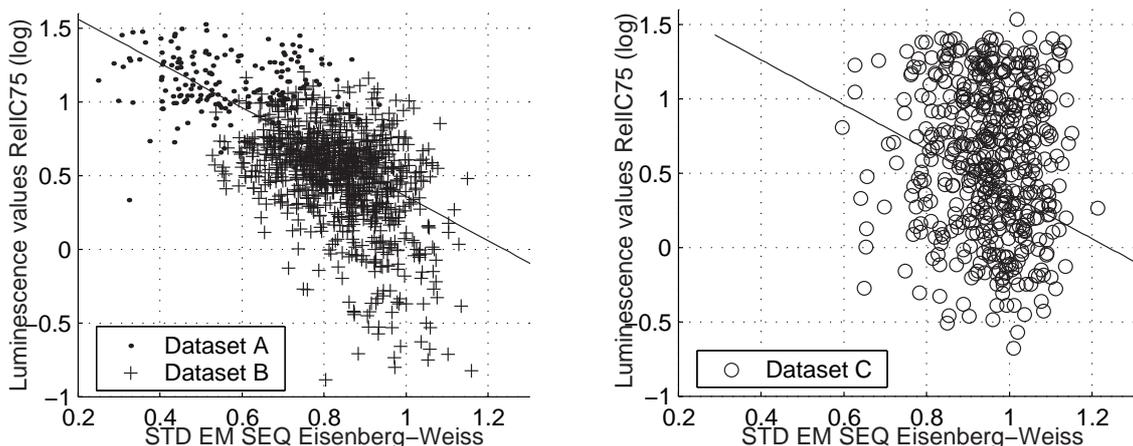


Figure 7: Results for a simple univariate linear regression model generated from Dataset A+B applied to Dataset A+B (left) and Dataset C (right)

The most interesting result for a newly generated dataset is the number of peptides with an improved activity (values of $RelIC75 < 1$) in comparison to the control peptide as discussed in Section 2.7. As shown in Table 6, the relatively bad regression models are able to identify many interesting candidates from the unseen validation dataset. As an example, 9 from 10 of the best rated peptides have a relevant antibacterial activity using the polynomial models of degree 2 or 3.

Model	Top 10	Top 25	Top 50	Top 100
Poly 1	70	56	32	26
Poly 2	90	56	38	27
Poly 3	90	60	38	27
k -NN	40	52	40	31
ANN	90	56	38	27
By chance	14	14	14	14

Table 6: Percentage of active peptides under the best rated peptides for Dataset C based on regression models for Dataset A+B

4 Discussion and Conclusion

All datasets contain relevant information for an activity model and give hints for future design rules. The main advantage of the proposed approach is a drastic reduction of the selected features (from 1120 to 2 or 3) for the regression model and the use of interpretable feature sets. A visual inspection and an interpretation of the results become much easier.

The most relevant features contain information about hydrophobicity values (e.g. Hopp-Woods). The best feature extraction is done by rather advanced methods like the standard deviation in a sequence. It measures the overall distribution of amino acids in a sequence instead of detailed position indicators. Such position information is completely discarded. However, different features with highly redundant information exist, e.g. molecular masses, or aggregated scales like z_j -values. Linear feature evaluation methods like ANOVA

or linear regression coefficients working with this feature set are sufficient. The compromise between loss of information and interpretability is tuned by preference measures of features.

With the given dataset, the prognosis of expected activities for unknown peptides is only possible with a remaining uncertainty. All activity models are based on regression approaches because a detailed analysis of the four class problem with acceptable classification accuracy is impossible. The analysis for adequate features is a first step to generate design rules for active features.

For the described high-throughput scenario, this uncertainty is acceptable. The model is able to increase the probability of active peptides for the next dataset as shown for the percentage of active peptides in Table 6. The expected result is a reduced amount of synthesized inactive peptides for future datasets and a faster search for active peptides.

Another advantage of the low-dimensional feature space is the possible search for "unusual" peptides by visual inspection of the plots or by a hierarchical clustering. Such peptides give hints for different activity principles. Unless such a small number of data points is not sufficient for a statistical analysis to characterize a sub-group, it might be interesting to explore similar peptides near these unusual ones in the next experiment.

However, some open questions remain. The order of amino acids in a sequence plays an important role in peptide activity. As an example, a permutation of an active sequence results in peptides with a large variety of activities [22]. However, the existing dataset seems to be too small for an automatic extraction of order information. Features like the proportion of amino acid neighbors like RW and WW might contain some relevant information. These effects should be evaluated by means of future experiments.

In general, a formal statistic evaluation of the results is difficult due to the large candidate feature set after feature extraction. Such an evaluation would require the use of Bonferroni-like corrections for multiple tests. Because the used features are strongly correlated, this leads to extremely conservative results from an overestimation of the degrees of freedom.

The strategy of generating a new dataset with new peptides characterized by an expected higher activity is a serious problem from a data mining point of view. It forces a generation of peptides outside the explored region, especially if the most active peptides are at the edges of this region. It leads to different mean values of relevant features and different feature relevances (see Table 3). As a consequence, the found activity model has to evaluate peptides outside the tested validity region of a model. This gives only valid results if the steps are not too large and the model has acceptable extrapolation properties. This effect is the main reason for the loss of prediction quality between different datasets.

One main activity for further research is the development of extended feature sets to improve the extraction of position and distribution of amino acids. This direction seems to be the most promising approach to transform the hidden sequence information into an interpretable feature space.

References

- [1] Ganz, T.: Defensins and other antimicrobial peptides: a historical perspective and an update. *Comb Chem High Throughput Screen* 8(3) (2005), pp. 209–17.

- [2] Steinberg, D. A.; Hurst, M. A.; Fujii, C. A.; Kung, A. H. C.; Ho, J. F.; Cheng, F. C.; Loury, D. J.; Fiddes, J. C.: Protegrin-1: A broad-spectrum, rapidly microbicidal peptide with in vivo activity. *Antimicrobial Agents & Chemotherapy* 41(8) (1997), pp. 1738–1742.
- [3] Brogden, K.: Antimicrobial peptides: pore formers or metabolic inhibitors in bacteria? *Nat Rev Microbiol* 3(3) (2005), pp. 238–50.
- [4] Hilpert, K.; Hancock, R. E. W.: Use of luminescent bacteria for rapid screening and characterization of short cationic antimicrobial peptides synthesized on cellulose by peptide array technology. *Nature Protocols* 2(7) (2007), pp. 1652–60.
- [5] Hellberg, S.; Sjoström, M.; Skagerberg, B.; Wold, S.: Peptide quantitative structure-activity relationships, a multivariate approach. *J. Med. Chem.* 30 (1987), pp. 1126–1135.
- [6] Perkins, R.; Fang, H.; Tong, W.; Welsh, W.: Quantitative structure-activity relationship methods: perspectives on drug discovery and toxicology. *Environmental Toxicology and Chemistry* 22 (2003), pp. 1666–79.
- [7] Cherkasov, A.: Inductive QSAR Descriptors. Distinguishing Compounds with Antibacterial Activity by Artificial Neural Networks. *Int. J. Mol. Sci* 6 (2005), pp. 63–86.
- [8] Frečer, V.: QSAR analysis of antimicrobial and haemolytic effects of cyclic cationic antimicrobial peptides derived from protegrin-1. *Bioorganic & Medicinal Chemistry* 14 (2006), pp. 6065–6074.
- [9] Cherkasov, A.; Hilpert, K.; Jenssen, H.; Fjell, C.; Waldbrook, M.; Volkmer-Engert, R.; Hancock, R.: Artificial intelligence delivers superb antibiotics for superbugs. *Nature Biotechnology* (2008). Submitted.
- [10] Weaver, D.: Applying data mining techniques to library design, lead generation and lead optimization. *Current Opinion in Chemical Biology* 8 (2004), pp. 264–270.
- [11] Hilpert, K.; Winkler, D. F. H.; Hancock, R. E. W.: Peptide arrays on cellulose support: SPOT synthesis - a time and cost efficient method for synthesis of large numbers of peptides in a parallel and addressable fashion. *Nature Protocols* 2(6) (2007), pp. 1333 – 1349.
- [12] Hilpert, K.; Volkmer-Engert, R.; Walter, T.; Hancock, R. E. W.: High-throughput generation of small antibacterial peptides with improved activity. *Nature Biotechnology* 23(8) (2005), pp. 1008–1012.
- [13] Wolf, S.; Loose, T.; Schablowski, M.; Döderlein, L.; Rupp, R.; Gerner, H. J.; Bretthauer, G.; Mikut, R.: Automated Feature Assessment in Instrumented Gait Analysis. *Gait & Posture* 23(3) (2006), pp. 331–338.
- [14] Andersson, P. M.; Sjöström, M.; Lundstedt, T.: Preprocessing peptide sequences for multivariate sequence-property analysis. *Chemometrics and Intelligent Laboratory Systems* 42(1-2) (1998), pp. 41–50.
- [15] Hildebrand, P.: *Zur Strukturvorhersage der Membranproteine*. Ph.D. thesis, Humboldt-Universität zu Berlin, Mathematisch-Naturwissenschaftliche Fakultät I. 2006.
- [16] White, S.; Wimley, W.: Membrane Protein Folding and Stability: Physical Principles. *Annual Reviews in Biophysics and Biomolecular Structure* 28(1) (1999), pp. 319–365.
- [17] Tatsuoka, M. M.: *Multivariate Analysis*. New York: Macmillan. 1988.
- [18] Mikut, R.: *Data Mining in der Medizin und Medizintechnik*. Universitätsverlag Karlsruhe. 2008.
- [19] Mikut, R.; Reischl, M.; Burmeister, O.; Loose, T.: Data Mining in Medical Time Series. *Biomedizinische Technik* 51(5/6) (2006), pp. 288–293.
- [20] Mikut, R.; Burmeister, O.; Braun, S.; Reischl, M.: The Open Source Matlab Toolbox Gait-CAD and its Application to Bioelectric Signal Processing. In: *Proc., DGBMT-Workshop Biosignalverarbeitung, Potsdam*, pp. 109–111. 2008.
- [21] Hilpert, K.; Fjell, C.; Cherkasov, A.: Short linear cationic antimicrobial peptides: screening, optimizing, and prediction. *Methods Mol Biol* 494 (2008), pp. 127–59.
- [22] Hilpert, K.; Elliott, M. R.; Volkmer-Engert, R.; Henklein, P.; Donini, O.; Zhou, Q.; Winkler, D. F. H.; Hancock, R. E. W.: Sequence Requirements and an Optimization Strategy for Short Antimicrobial Peptides. *Chemistry & Biology* 13 (2006) 10, pp. 1101–1107.

Combined Knowledge-based and Data-driven Modeling by Heterogeneous Mixture-of-Experts

Jörg Beyer^{1,2}, Kai Heesche¹, Werner Hauptmann¹, and Clemens Otte¹

¹Siemens AG, Corporate Technology, Information and Communications
Otto-Hahn-Ring 6, 81739 Munich, Germany

²School of Computer Science, Otto-von-Guericke-University, Magdeburg
{joerg.beyer.ext, kai.heesche, werner.hauptmann, clemens.otte}@siemens.com

Abstract

In this paper, a new method referred to as heterogeneous mixture-of-experts is introduced. The main objective of this approach is to jointly include knowledge-based and data-driven submodels in the modeling process. The integration of knowledge-based submodels is of particular interest, since they are able to provide information not contained in the data. On the other hand, data-driven models can complement the knowledge-based models with respect to performance or input space coverage. For this task of appropriately integrating the different models, a method for partitioning the input space for the given models is introduced. The introduced approach is compared with selected other modeling techniques for both a benchmark data set and a real-world application setting.

1 Introduction

Modern technical systems are characterized by an increasing complexity, making a structuring of the system indispensable. Furthermore, management and control of such systems demand more and more sophisticated combinations of knowledge-based and data-driven solutions. In the past, techniques like neuro-fuzzy systems have been applied with considerable success. However, due to their limitation to rule-based structures, these approaches are less suitable for applications where more comprehensive knowledge is available, e.g. in complex relational form. Here, the usage of heterogeneous information sources is required. Through the combined application of data-driven learning and knowledge-based structuring of systems undesirable effects can be avoided. Another advantage of using knowledge-based submodels is that they are also able to cover regions of the input space where the data is sparse or only represents few states of the problem. With purely data-driven models extrapolation beyond the data is required which can lead to undesired outputs. Therefore, methods are needed to provide reliable results.

The use of heterogeneous models is motivated by the paradigm that different partial models can complement each other avoiding the weakness of a single model or a combination of homogeneous models. The field of information fusion investigates the combination of information components from different sources in order to obtain more reliable and complete results with reduced uncertainty.

This contribution is organized as follows: First, in Section 2 a definition of information fusion is given and in Section 3 the mixture-of-experts (ME) approach is described. The heterogeneous mixture-of-experts (HME) model is introduced in Section 4. In Section 5, some experiments on a benchmark data set as well as a real-world application are outlined. Section 6 concludes the paper.

2 Information Fusion

Information Fusion (IF) is a commonly used technique in different application domains, such as sensor fusion [1], identity verification [2], intrusion detection [3], or signal and image processing [4]. Several definitions have been used to describe IF depending on the particular application domain. Consequently, there is no unified definition or terminology of IF. The terms information fusion and data fusion are usually accepted as interchangeable terms. According to Dasarathy [5] IF “encompasses the theory, techniques and tools created and applied to exploit the synergy in the information acquired from multiple sources (sensors, databases, information gathered by humans, etc.) in such a way that the resulting decision or action is in some sense better (qualitatively or quantitatively, in terms of accuracy, robustness, etc.) than would be possible if any of these sources were used individually without the synergy exploitation.” We adapt this definition as follows. IF describes the process of merging and integrating heterogeneous information components from multiple sources, for instance, in the form of sensors, human experts, symbolic knowledge, or physical process models. Each source represents different kinds of information. The combined solution is able to attain more accurate, transparent, and robust results since the different information sources can complement each other with respect to their strengths and weaknesses and their availability and validity.

In IF, we consider two kinds of fusion approaches according to the relationship among the information sources:

- Complementary: Each source provides information from a different region of the input space, i.e. their responsibilities do not overlap. These sources provide locally a high performance. However, outside their regions the results are not valid.
- Cooperative: The information is shared among several information sources in the same part of the input space and has to be fused for a more complete modeling of the underlying input/output relations. Information from overlapping regions of responsibility is treated in a cooperative way.

In [6], also redundant information sources are distinguished which are not considered in our work.

The following example describes a possible scenario of IF. Let us assume a globally valid source exists, e.g. a physical process model, providing a baseline performance in the entire input space. This source can be complemented by several local sources in order to optimize the global one. The responsibilities of the local sources must be zero outside those regions of the input space where these sources have been designed for.

The ME model described in the next section can be used for the purpose of IF. The information sources will be represented by homogeneous data-driven models. A separate data-driven model performs the partitioning of the input space and the fusion of the data-driven models.

3 Mixture-of-Experts

The mixture-of-experts model consists of a set of models that perform a local function approximation [7], [8]. ME models can be described as input-dependent mixture of models which solve problems by the divide-and-conquer strategy, i.e. they learn to decompose complex problems into simpler, easier to solve subproblems. This

decomposition is learned by a gate function by partitioning the input space and assigning experts to these regions. Figure 1 depicts a standard ME model. The ME model is a generating one, i.e. the data are generated by a set of M independent processes which are randomly selected. The introduction of a latent variable $Z = \{z_j^{(n)} : j = 1, \dots, M, n = 1, \dots, N\}$, where $z_j^{(n)}$ is 1 if input vector $x^{(n)}$ was generated by model j and 0 otherwise, and the data set $D = \{\bar{x}^{(n)} \in \mathfrak{R}^k, y^{(n)} \in \mathfrak{R}, n = 1, \dots, N\}$, allows the ME model to be trained with the Expectation-Maximization algorithm [9]. The conditional scalar output $y^{(n)}$ given the input vector $x^{(n)}$ is defined as:

$$P(y^{(n)} | \bar{x}^{(n)}, \Theta) = \sum_{j=1}^M P(z_j^{(n)} | \bar{x}^{(n)}, \theta_g) P(y^{(n)} | \bar{x}^{(n)}, \theta_j),$$

where Θ comprises the parameter of the gate, θ_g , and of the models, $\theta_j, j = 1, \dots, M$. The probability $P(y^{(n)} | \bar{x}^{(n)}, \theta_j)$ represents the conditional densities of target $y^{(n)}$ for model j and $P(z_j^{(n)} | \bar{x}^{(n)}, \theta_g)$ is the weighting coefficient of model j . The negative log likelihood function can be used as error function:

$$\ln L = - \sum_{n=1}^N \sum_{j=1}^M h_j^{(n)} \log P(z_j^{(n)} | \bar{x}^{(n)}, \bar{\theta}_g) - \sum_{n=1}^N \sum_{j=1}^M h_j^{(n)} \log P(y^{(n)} | \bar{x}^{(n)}, \bar{\theta}_j, z_j^{(n)}), \quad (1)$$

where $h_j^{(n)}$ represents the posterior probability of selecting model j for input vector $x^{(n)}$. The first term of the right-hand side of equation (1) is the part of the error that the gate contributes to the overall error. It can be interpreted as the entropy of the distribution of the input vectors among the models. The error of the gate is zero if only one of the mixture coefficients $P(z_j^{(n)} | \bar{x}^{(n)}, \bar{v})$ is equal to one. Thus, the gate prefers a solution where only one model is selected, otherwise the error is increased. The second term of equation (1) presents the error component which the individual models contribute. This is the cross-entropy among the posterior probability $h_j^{(n)}$ and the probability that model j has generated the target value $y^{(n)}$. To reduce the overall error a trade-off must be found between increasing the error by selecting more than one model and decreasing it due to a more accurate ME model output.

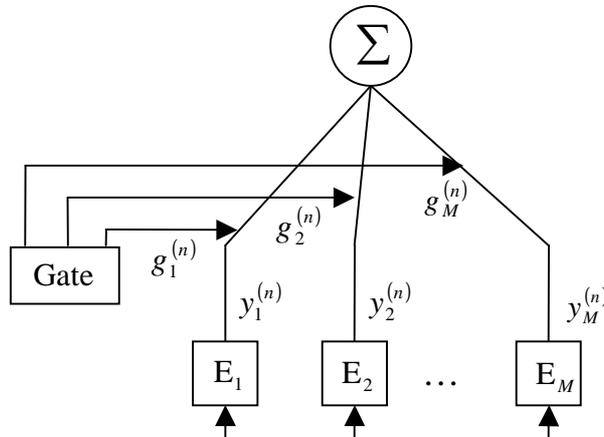


Figure 1: ME model with M experts.

4 Heterogeneous Mixture-of-Experts

We investigate the integration of knowledge-based models like characteristic maps or mathematical models in the ME framework. It is assumed that the knowledge-based models will only cover a part of the input space while data-driven models learn the remainder. Figure 2 shows the idea of the approach. The input space is divided into two types of regions: one represented by knowledge-based models and one represented by data-driven models. The transition between these regions can be soft, i.e. knowledge-based and data-driven models can share the responsibility for input data from the corresponding border.

For the appropriate integration of different submodels, a method for partitioning the input space for these given models has to be defined. The knowledge-based models are valid in domains of the input space they are designed for. Therefore, it is required to include the information about the validity of these models into the training process of the gate function. The data-driven models are trained to close the gaps between the knowledge-based models with respect to performance or input space coverage.

The main advantage of the HME model is the ability to replace training data by some kind of expert knowledge represented by knowledge-based models. In current approaches it is not possible to train the gate in regions of the input space where training data is not available. In such situations the assignment of input data to certain submodels by the gate is not plausible. In the HME model, the known validity ranges of the knowledge-based models can be applied to predefine the gate function in those regions of the input space where training data is sparse or not available. Human experts with knowledge concerning the responsibility of the given models can model the validity ranges in an appropriate representation. For each model j a mapping $v_j : \mathfrak{R}^n \rightarrow [0,1], \forall j = 1, \dots, M$ is defined that computes a validity value of the model for an input vector. These mappings can be completely or partly determined by human experts and can be modelled for example by mixture-of-gaussians or radial basis function networks. If the input space is sufficiently covered by training data, a simple method is to derive the validity ranges from the training data by density estimators, e.g. Parzen-window. In the approach described here, the validity has to be completely predefined by the user. To guide the partitioning of the gate the posterior probability $h_j^{(n)}$ is defined as follows:

$$h_j^{(n)} = \frac{v_j P(z_j^{(n)} | \bar{x}^{(n)}, \theta_g) P(y^{(n)} | \bar{x}^{(n)}, \theta_j)}{\sum_{i=1}^M v_i P(z_i^{(n)} | \bar{x}^{(n)}, \theta_g) P(y^{(n)} | \bar{x}^{(n)}, \theta_i)}. \quad (2)$$

The role of the gate's output and the mapping v_j can be interpreted as follows: Based on the performance, the gate's task is to assign a selection probability to the model whereas the mapping v_j evaluates the validity of the model for an input vector. By doing so, in (2) the gate will be enforced to decrease the weights of model outputs if the input vectors are located outside their domains. The particular amount of weight decrease is dependent on v_j . Additional optimizations are required if the validity ranges of submodels are to be adapted. For adaptive validity ranges of knowledge-based local models additional constraints are to be considered. In particular, these ranges must only scale down and not enlarge, since it has to be avoided to give some knowledge-based model responsibility for those parts of the input domain it has not been designed for. On the other hand, a reduction of the validity ranges may even lead to an improved

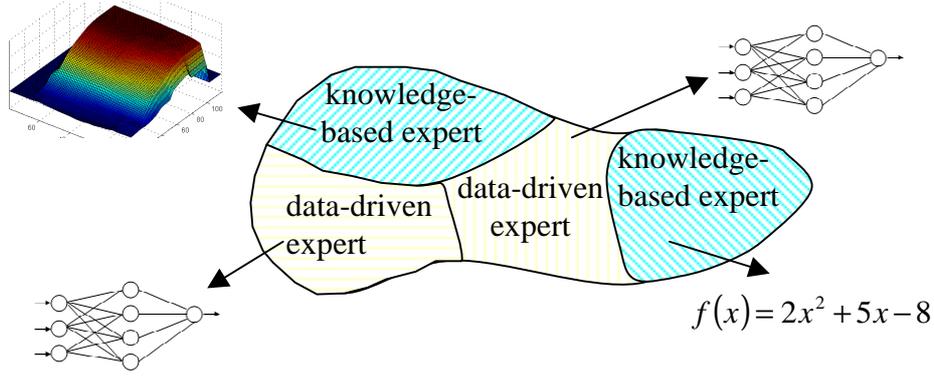


Figure 2: Partitioning of the input space.

behaviour of the overall model since a knowledge-based model can be locally complemented by data-driven models. By a corresponding local reduction of the validity ranges the data-driven submodel receives an increased gate activation.

In the following section, the HME is applied for both a benchmark data set and a real world application where different heterogeneous sources are available. It is shown that the HME is able to integrate different knowledge-based models and in this way achieves a superior performance.

5 Experiments

5.1 Experimental Setup

In this section, we describe an experimental comparison of the proposed method with a standard ME model, an MLP model, and a linear regression method on a common benchmark and an application data set. The data sets are randomly divided into a training data set (80% of the data) and a test data set (20% of the data). We used the mean absolute error (MAE) to estimate the predictive error:

$$\hat{e} = \frac{1}{N} \sum_{n=1}^N |y^{(n)} - f(\bar{x}^{(n)})|.$$

The whole experiment is performed ten times for each data set and the results are averaged.

5.2 Benchmark Data Set

The Body Fat data set¹ includes 13 attributes and 252 instances. The task is to estimate the percentage body fat. The setup of the models used for the purpose of comparison was the following:

- HME: two generalized linear models, more specifically, single layer perceptrons (SLPs) with *tanh* activation function, and a knowledge-based model were used as expert models. The knowledge-based model is represented by the following function²:

¹ <http://lib.stat.cmu.edu/datasets/>

² http://en.wikipedia.org/wiki/Body_fat_percentage

$$BF = \frac{-98.42 + 4.15 \cdot waist - 0.082 \cdot weight}{weight},$$

with *waist* as abdomen circumference (measured in inches) and *weight* as body weight (measured in lbs). Due to the lack of knowledge about the validity range of the function, its responsibility comprises the entire input space.

- ME: three generalized linear models, again SLPs, were applied as expert models and an MLP with three hidden units was used as gate.
- MLP: an MLP with six hidden units, *tanh* activation function, and back propagation training was used.
- Linear regression method: the robustfit method from Matlab was used.

The results of the experiment are summarized in Table 1 and Figure 3. Table 1 shows the MAE of the different models for the data set. The HME achieves good performance for the Body Fat data set. We conclude from the comparable performance of the HME that the knowledge-based model given by the relation above provides no or only few additional information about the target compared to the given data set. Figure 3 shows the distribution of the MAE around its median. The HME approach led to more robust results with respect to the lowest variance of the MAE. The MLP achieved the best performance due to its ability of nonlinear function approximation.

5.3 Simulation of Energy Flows in a Hybrid Electric Vehicle

In this application the energy flows in the powertrain of a hybrid electric vehicle are simulated. The hybrid electrical vehicle setup consists of a combustion engine, an

Table 1: Predictive error for each model for the Body Fat data set.

Regression model	Predictive error (std)	
	training	testing
HME	2.86 (0.08)	3.01 (0.12)
ME	2.99 (0.14)	3.09 (0.15)
MLP	1.98 (0.11)	2.51 (0.22)
robustfit	3.37 (0.09)	3.51 (0.25)

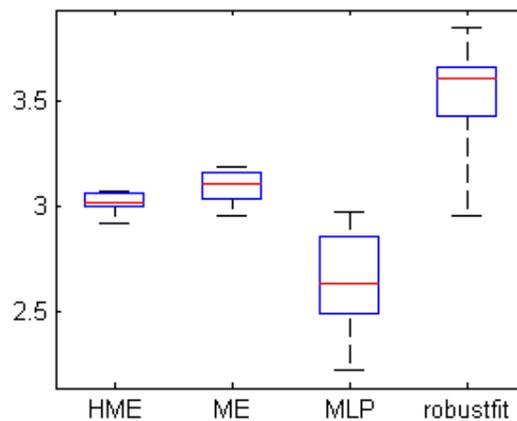


Figure 3: Distribution of model error for the Body Fat data set.

electric motor, and a battery. Charge and discharge energy flows of the battery can be distinguished. By applying the available expert knowledge four distinct driving modes can be defined: brake mode, drag mode, pure electric drive mode, and hybrid drive mode. In brake mode and drag mode, the electric motor is operating as a generator to regenerate the kinetic energy used for charging the battery. In pure electric drive mode and hybrid drive mode energy is provided by the battery to drive the electric motor. Furthermore, the battery must maintain certain chemical limits. These limits determine the maximum charge and discharge capabilities of the battery dependent on its state of charge and temperature. Using knowledge from domain experts, specific models for each mode were designed. These models represent complementary information sources because they are defined in different regions of the input space with each model providing information for different mutually exclusive driving modes.

In this application, the hybrid electric vehicle data set includes five input features and about 410.000 samples. Each mode has different input features. The models' setup of the hybrid data set was the following:

- HME: four expert models were used. Two characteristic maps and a mathematical model represent the pure electric drive mode, brake, and drag mode. However, since the hybrid drive mode is too complex to provide an adequate mathematical model, for this mode a data-driven model was learned. As gate, an MLP with four hidden units was applied.
- ME: four identical MLPs were used with three hidden units and an MLP with four hidden units as gate.
- MLP: an MLP was used with 10 hidden units and back propagation as training algorithm.
- Linear regression method: the robustfit method from Matlab was used.

The results for the hybrid vehicle data are shown in Table 2 and Figure 4. The HME has achieved superior performance due to the incorporation of knowledge-based models. In Figure 4, the distribution of the MAE around its median is shown. Three paired t-tests were performed, one for each paired comparison of HME with each of the other models, at a confidence-level of 95%. The test found that HME is significantly better than the

Table 2: Predictive error for each model for the hybrid electric vehicle data.

Regression model	Predictive error (std)	
	training	testing
HME	2.46 (0.02)	2.62 (0.05)
ME	3.25 (0.07)	3.47 (0.09)
MLP	3.12 (0.09)	3.32 (0.14)
robustfit	4.01 (0.04)	4.18 (0.08)

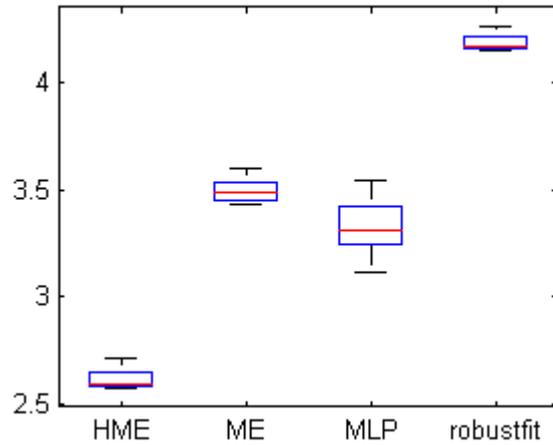


Figure 4: Distribution of the model error of the hybrid electric vehicle data set.

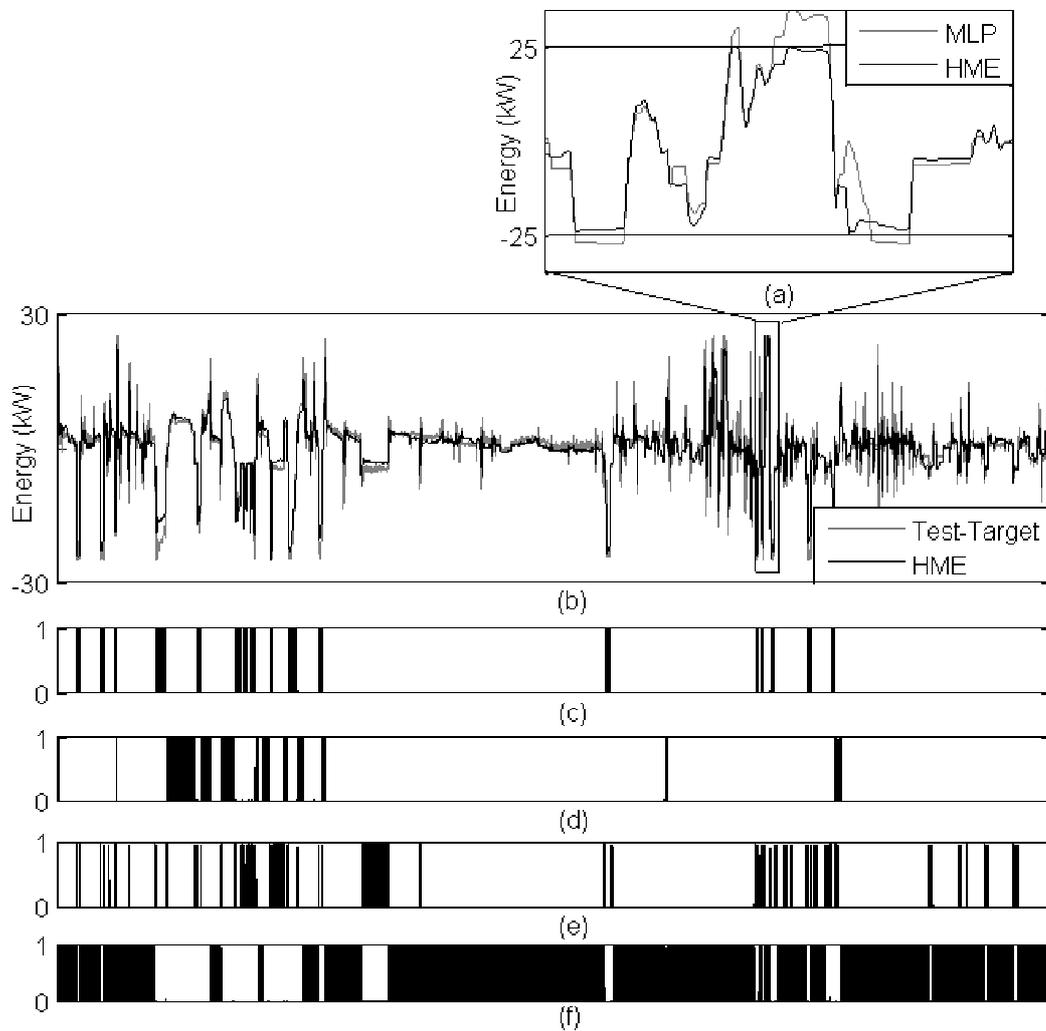


Figure 5: In figure (b) the prediction of the HME model and the target values for the energy flows are shown. Figure (a) depicts the output of the MLP and the HME for the detail of (b). The horizontal lines in (a) represent the chemical limits of the battery. It can be seen that the MLP violates these limits. Figures (c) to (f) show the gate activations for: (c) model for brake mode, (d) model for pure electric mode, (e) model for push mode, and (f) model for hybrid mode.

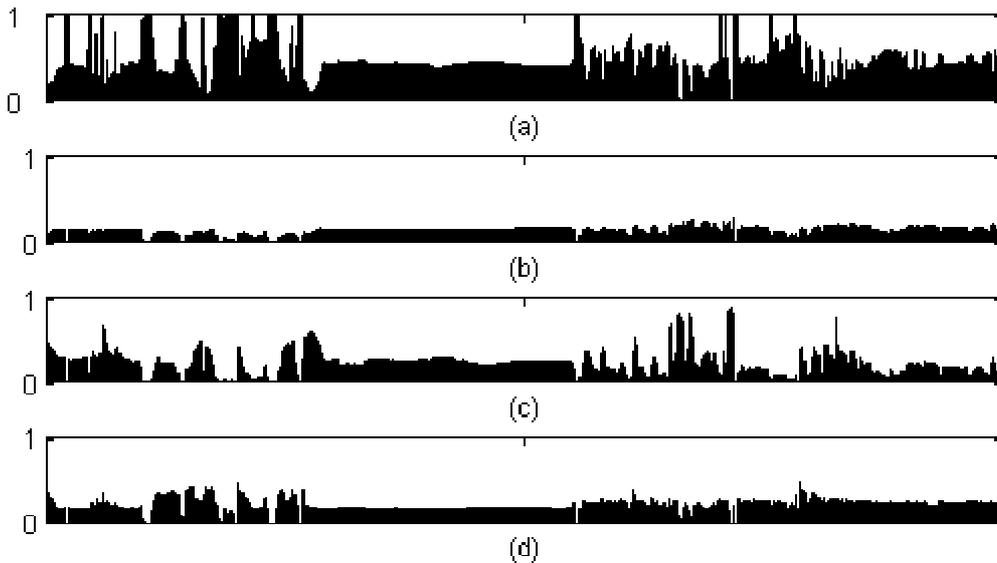


Figure 6: The figures (a) to (d) shows the gate activations of the ME for the sequence depicted in Figure 5 (b).

other models. The integration of the knowledge-based models did not only lead to a superior performance but also led to more robust results. Figure 5 (b) depicts a sequence of the predicted energy flows of the HME and the corresponding target values. The gate activations of the submodels of the HME for the input vectors are shown in Figures 5 (c) to (f). The gate treated the models as complementary information sources. This behaviour is consistent with the knowledge of the domain expert that each model was defined for different mutually exclusive modes. For the MLP, a detail of the sequence of its output is illustrated in Figure 5 (a). The violation of the maximum allowable chemical limits of the battery, illustrated as horizontal lines, can be seen. In order for the predicted energy flows to stay in the permitted region these limits must be considered as additionally inputs of the MLP, which increases its complexity. In contrast to the HME, the ME model was not able to distinguish the modes and partitioned the input space in a technically non-plausible way. This can be seen in Figures 6 (a) to (d) for the gate activations of the submodels of the ME for the sequence of figure 5 (b). The output of the ME model was obtained by combinations of the submodels' outputs. Except the HME model all other models violated chemical limitations because they predicted energy flows that cannot be provided by the battery. These conditions are implicitly contained in the knowledge-based models.

6 Conclusions

We proposed a heterogeneous mixture-of-experts approach for the fusion of multiple information sources of different kind. These sources are represented by knowledge-based or data-driven models. The HME model provides several advantages. As expected, the incorporation of knowledge-based models leads to a superior performance of the HME model compared to the ME model as shown in a real-world application example. The proposed HME is able to integrate the heterogeneous information sources and can thus exploit all available information. Further, the HME model provides reliable results because the knowledge-based models are responsible for regions they are defined for. By integration of knowledge-based models fewer data-driven models

are needed. This leads to a lower complexity of the HME model in comparison to the standard ME models.

Currently, different approaches for applying adaptive validity ranges for knowledge-based models are explored. The introduction of adaptive validity ranges enables data-driven models to be responsible for regions of the input space with high data density and can thus locally complement the knowledge-based models. Furthermore, computation of confidence measures based on the validity ranges will be explored.

References

- [1] Durrant-Whyte, H. F.: Sensor models and multisensor integration. *International Journal of Robotics Research*, 7(6):97-113, 1988.
- [2] Bengio, S.; Marcel, C.; Marcel, S.; Mariéthoz, J.: Confidence measures for multimodal identity verification. *Information Fusion*, 3(4):267-276, 2002.
- [3] Bass, T.: Intrusion detection systems and multisensor data fusion. *Communications of the ACM*, 43(4):99-105, 2000.
- [4] Bloch, I.: Information Fusion in Signal and Image Processing: Major Probabilistic and Non-Probabilistic Numerical Approaches. *John Wiley & Sons Inc.*, 2008.
- [5] Dasarathy, B. V.: Information fusion - what, where, why, when, and how? *Information Fusion*, 2(2):75-76, 2001.
- [6] Elmenreich, W.: An introduction to sensor fusion. Research Report 47/2001, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-1, 1040 Vienna, Austria, 2001.
- [7] Jacobs, R. A.; Jordan, M. I.; Barto, A. G.: Task decomposition through competition in a modular connectionist architecture: the what and where vision tasks. *Cognitive Science*, 15:219-250, 1991.
- [8] Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; Hinton, G. E.: Adaptive mixtures of local experts. *Neural Computation*, 3:79-87, 1991.
- [9] Jordan, M. I.; Jacobs, R. A.: Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181-214, 1994.

Adjusting Monitored Experiments to Real-World Cases by Matching Labeled Time Series Motifs

Christian Moewes and Rudolf Kruse

School of Computer Science, Otto-von-Guericke University of Magdeburg

Universitätsplatz 2, D-39106 Magdeburg, Germany

Tel: +49 (391) 67 ext. {11358, 18706}

Fax: +49 (391) 67 12018

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

Abstract

In this paper we devote ourselves to the difficulty of fitting human designed experiments to real-world cases. We decompose this problem into two smaller subproblems: 1.) The search of recurrent patterns in temporal sequences, so called motifs that are deemed to be discovered in both the experiments and the real observations and 2.) the matching of motifs to linguistic terms which are possibly available as domain knowledge. Therefore we describe an effective time series representation that enormously speeds up the search for these motifs. We present some approaches to adjust the designed experiments with the help of the discovered motifs. Finally, we conclude our work and give prospects to possible extensions.

Keywords: Multivariate Time Series Analysis, Motif Discovery, Labeling, Frequent Pattern Mining.

1 Introduction

Conducting field tests of complex systems to evaluate their behavior is usually expensive and time consuming. One requirement is that the designed tests should be as similar as the behavior of their pendants which are produced in series and used in the real world. Based on these experiments which quantitatively describe criteria (e.g., lifetime, errors, loadings), the quality of a system might be improved. For instance sources of error can be found and remedied in the next generation of systems.

In order to evaluate these criteria, sensor data are recorded over long periods of time from the test and the real objects, respectively. Existing data analysis methods are based on one-dimensional load spectra which are utilized to compare the tests to the reality. These methods are completely time-independent. However, the timely behavior of real-world systems might contain many different processes that have not been considered in the tests so far. Additionally, the tests may contain procedures that do not meet reality at all.

Data mining methods that consider time as additional variable are discussed to some extent in Section 2. By means of these methods to analyze time series we are able to discover interesting recurrent patterns, so-called *motifs*. We describe a very effective algorithm to find such motifs in Section 3.

One task then would be to match discovered motifs to test criteria of the time series. Every time series containing a motif thus can be labeled by at least one test criterion if the relevant test has been designed thoroughly. Hence a motif and its label can be regarded as

a discovered rule's antecedent and consequent, respectively. In Section 4 we explain how motifs are labeled by means of expert's knowledge.

Having identified a set of rules in all experiments, we try to retrieve a subset of it in real-world data. As a consequence unseen time series can be assigned and compared to the given experimental criteria. This part of our work is specified in Section 5.

Thence it is possible to adjust the experiments to the given real-world cases as follows. For example if the motifs of an unseen time series cannot be found in any experiment, then the tests should be adjusted such that every new motif occurs at least once. Then again motifs that only exist in test data and not in the real-world should be removed since they do not seem to be relevant. Section 6 clarifies further details of this approach. We finally summarize our work and give prospects to open questions and problems in Section 7.

2 Time Series Data Mining

In research and development, data mining in time series has gained incredibly big attention in the last years. Meanwhile time series are simply ubiquitous in areas such as finance, medicine, biometry, chemistry, astronomy, robotics, networks and industry. So-called time series databases save an additional time stamp for every stored dataset. A time series can be arbitrarily long and potentially involve several dimensions (attributes, channels, sensors). Then a time series is no longer called univariate but multivariate.

A big challenge is the search for useful information in time series. Today we distinguish between the following time series data mining tasks: Clustering [1], classification [2], motif discovery [3], rule generation [4], visualization and anomaly detection [5].

Due to the plenty of data, many of those problems can usually be broken down to the search for recurrent sequences in the time series that are similar to each other. In order to find these sequences one has to define a similarity measure that compares two sequences. Most publications in that field use the Euclidean distance

$$d(Q, C) = \sqrt{\sum_{i=1}^w (q_i - c_i)^2} \quad (1)$$

between two normally distributed sequences $Q = (q_1, \dots, q_w)^T$ and $C = (c_1, \dots, c_w)^T$ of length w as basis for the similarity measure. If we use (1) to measure the similarity, usually lots of comparisons must be performed to find some motifs. Moreover, the capacity of every fast main memory is most of the time too small to load all original data at once.

2.1 Memory-Efficient Representations

Owing to these many and especially slow accesses to the original data on the hard disc, one should use an approximation of every time series that fits into the main memory of a computer and contains all essential and interesting features. There are dozens of different kinds of time series approximations, e.g., discrete Fourier transform (DFT), discrete wavelet transform (DWT), piecewise linear models (PAA), piecewise constant models

(APCA), singular value decomposition (SVD), symbolic representations. The latter ones benefit by being applicable to algorithms that originate from text processing and bioinformatics, e.g., hashing, Markov models, suffix trees, etc.

In current research the symbolic representation of Lin and Keogh [6] wins out even over well-known approximations. Their symbolic aggregate approximation (SAX) transforms a univariate time series sequence into a word of defined length n over a chosen alphabet A with $|A| = a$. The SAX algorithm is rather simple but intuitive.

Firstly, the sequence is separated into n equal parts. Then the mean of every interval is computed as representative of all values in that interval. This method is also called piecewise aggregate approximation (PAA) [7].

After that step the essentially shorter sequence of mean values is discretized as follows. Every mean value of the PAA sequence is assigned to one of the a letters such that the occurrence of every letter in the sequence is equally probable. This is achieved by assuming that the PAA sequence's range of values is normally distributed. Furthermore, this distribution is split up into parts such that all parts share the area under the Gaussian curve. This assumption can be made due to the following fact. Long time series may not be normally distributed, but their short sequences certainly are to a high degree [6].

While other symbolic representations generate a word from time series data as well, SAX is yet one of a kind compared with them. It does not only compress the sequence. SAX also enables us to measure a distance $d^*(Q, C)$ between two SAX words which is a lower bound of the Euclidean distance (1) between the original sequences Q and C , formally

$$d^*(Q, C) \leq d(Q, C).$$

For the rest of the paper we assume that the similarity is determined by the Euclidean distance (1). So, a lower bound means that if two SAX words are dissimilar, then their original sequences are dissimilar as well. Consequently, algorithms that are based on SAX produce identical results compared to algorithms that work with the original data. Merely similar SAX words should be compared in the Euclidean space again. Fortunately, those accesses to the original data are only very rare since most of the comparisons are based on dissimilar sequences.

Having a memory-efficient representation we can concentrate ourselves on finding similar sequences efficiently. In the following we proceed from the assumption that every time series is approximated by SAX since the next algorithms are based on hashing.

3 Motif Discovery in Time Series

If we are able to find recurrent sequences that are similar to each other, then problems such as clustering or classification of time series are much easier to solve. These similar sequences are called *motifs* due to the vocabulary that is used in bioinformatics. This originates from the fact that in this domain, motifs correspond to recurrent strings (usually from a DNA).

In the article from Chiu et al. [8] SAX is associated with motif discovery in univariate time series for the first time. In order to find all motifs of a time series of length l , it is

separated by a sliding window with certain width w into $(l - w + 1)$ sequences. Every sequence is transformed into a SAX word and saved into a $(l - w + 1) \times n$ matrix which we call SAX matrix.

The positions of possible motifs are then guessed using the random projection algorithm proposed by Buhler and Tompa [9]. Actually, the positions are found by pairwise comparisons of the SAX words. So, for each of those $(l - w + 1)^2$ comparisons, we firstly reserve one entry in a collision matrix \mathbf{M} which can be implemented efficiently by a hash table. In the beginning, let every entry $\mathbf{M}(i, j)$ be zero for $1 \leq i, j \leq l - w + 1$.

Although usually $n \ll w$, it is not preferable to compare every single character of the saved SAX words in the matrix with each other. Buhler and Tompa rather had the idea that there exist so-called *don't care symbols* of which we do not know where they might be in the words. These symbols would correspond to, e.g., a noisy motif, a dilation/contraction of a temporal sequence.

Accordingly the SAX matrix is projected down to $1 \leq k < n$ randomly chosen columns. Afterwards all rows of the projected matrix are compared with each other. If two projected SAX words in the rows i and j are equal, then the value in $\mathbf{M}(i, j)$ is incremented by one.

The projection is repeated t times since one can assume that some of the hidden motifs will share one entry in \mathbf{M} after t iterations. Additionally, it is improbable that many random sequences will collide together with an already found motif. Therefore they would have to be identical to this motif in all k positions.

Since the algorithm cannot know if a collision entry in \mathbf{M} is a motif or not, the user must specify a threshold $1 \leq s \leq k$. All $\mathbf{M}(i, j) \geq s$ thus would be motif candidates. Remembering that we deal with temporal and not DNA sequences, the problem of motif discovery becomes harder as we find similar occurrences of the i -th sequence in its direct neighborhood. Those sequences which are named *trivial matches* [8] are heuristically removed from the set of potential motifs at the end of the discovery.

Although comparatively, many parameters have to be determined, i.e., n, a, w, k, t, s , random projection is robust against slight changes of the SAX parameters n and a as well as the projection size k [8]. Also the number of projections t can be set large enough in order to create some collisions. However, there are two questions left: How many and in particular what kind of motifs do we have to find?

If we set w and s too large on the one hand, then we may not find lots of “short” motifs. On the other hand, we will get completely different results if we set w and s too small. Then we will probably find many random consensuses that do not correspond to any real motif. Therefore, the choice of these two parameters should be made carefully. Expert’s knowledge may help in such a situation.

As a side remark, we would like to mention that Yankov et al. [10] extended time series random projection to a non-Euclidean distance measure, i.e., *uniform scaling*. With this method one can find motifs that are not exactly w time stamps long. This approach is indeed limited such that w must be chosen based on the respective application.

3.1 Subdimensional Motifs

The random projection to find time series motifs [8] was originally only designed for one-dimensional datasets. If we deal with multivariate time series, then there exist several

ways how to tackle this problem.

The simplest idea is to map the p dimensions down onto one and then use random projection. For instance, Tanaka et al. [11] have transformed the input dimensions by means of principal component analysis (PCA) into solely the first principal component. Finally, the approach from Chiu et al. [8] could be applied to the new univariate time series.

A first approach of Minnen et al. [12] is founded on the idea that p dimensions also generate p SAX words. These SAX words are then concatenated and treated like a SAX representation of a long univariate time series. As a consequence, the method from Chiu et al. [8] can be applied in this case as well.

Though notice that both approaches can only discover motifs that span all dimensions. This will be problematic in particular if we *a priori* do not know in which of the dimensions we can observe any motif. In practice it can also happen that a time series motif's attributes can differ quite from another one's attributes. Such multivariate time series motifs that do not span all dimensions are called *subdimensional*.

Formally, we denote a multivariate sequence as $w \times p$ matrix which stores w real values for each of the p attributes. We define the distance d_{mult} of two multivariate sequences $\mathbf{Q} = [Q_1, \dots, Q_p]$ and $\mathbf{C} = [C_1, \dots, C_p]$ by the Euclidean norm

$$d_{\text{mult}}(\mathbf{Q}, \mathbf{C}) = \|\mathbf{d}\|_2 = \sqrt{\sum_{j=1}^p |d_j|}$$

whereas $\mathbf{d} = (d_1, \dots, d_p)$ and $d_j \equiv d(Q_j, C_j)$ corresponds to the Euclidean distance (1) between Q_j and C_j for $1 \leq j \leq p$.

According to our literature research, there is so far only one approach that tries to discover subdimensional motifs. Minnen et al. [13] improve their original idea to concatenate the SAX words of every dimension. They increment the collision matrix \mathbf{M} *per attribute* at the appropriate entry for every projected SAX word that matches another one.

Afterwards all elements of \mathbf{M} that are greater than s are picked out and must be examined further. Note that although we have two positions for each pair of sequences, nonetheless we do not know its relevant dimensions. Furthermore, there is not any assignment of the pairs of sequences to the potential motifs yet.

Before we can perform this assignment, we have to extract the subdimensions of the sequences by means of the following naïve idea. For every pair of sequences we sort all distances d_1, \dots, d_p in an ascending order. Then the distance is accumulated in that order for every single dimension until a certain threshold r_{max} is exceeded. The attributes of the smallest distances thus correspond to the pair of sequences' relevant subdimensions.

These heuristics can be also improved by not regarding attributes having smallest distances, but using only probably relevant attributes to compute the distance [13]. Therefore one estimates the empirical frequency distribution $P(d_j)$ over the distances between some non-trivial matches for every dimension $1 \leq j \leq p$ by random sampling. Later on the distances d_1^*, \dots, d_p^* are computed for every entry $\mathbf{M}(i, j) \geq s$. If the value of the cumulative distribution function $P(d_j \leq d_j^*)$ is smaller than the dimension relevance r_{rel} which is specified by the user, then the j -th dimension be relevant.

Determined all pairs of subdimensional sequences, the trivial matches have to be eliminated as it was done in the univariate case of motif discovery. With this idea [13], motifs do not need to span all dimensions. This would be an asset compared to [11, 12] when the set of attributes does contain, e.g., very noisy signals, uninformative dimensions.

Disadvantages of this method for subdimensional motif discovery are the threshold parameter r_{\max} and r_{rel} , respectively. Both extremely depend on the sequence length w . So, if domain knowledge is present, then it is suitable to use r_{\max} as threshold. Otherwise one must estimate the distribution $P(d)$ and handle with r_{rel} .

4 Labeling Discovered Motifs

Having identified a set of subdimensional motifs, we merely found multivariate time series sequences of certain length w that recur at least twice. Note that we can find random motifs accidentally as well. Thus it is probable that a motif which recurs only twice might not be what we are looking for.

Yet, motifs that recur more often should be labeled meaningfully from mainly experts who designed the experiments. They usually possess the necessary knowledge to interpret both simple and complex curve progressions. This labeling can be done, e.g., by means of the test criteria.

If there is no expert's knowledge available, then one can fall back on methods from fuzzy set theory (FST) [14]. In FST one tries to model imprecise, vague or even uncertain concepts, e.g., sensor measurements, such that the human being obtains a better understanding of these concepts.

For instances, every attribute can be regarded as linguistic variable [15]. In doing so, the attribute's range of values is separated into a so-called fuzzy partition. Every partition is described by a fuzzy set A . Thus every value x can be assigned to a membership degree $\mu_A(x) \in [0, 1]$ of the fuzzy set A .

We consider the measured velocity v as an illustrating example. The velocity can be described by some linguistic terms, e.g., *fast*, *medium*, *slow*. Every expression corresponds to a fuzzy partition which then again is described by a fuzzy set, i.e., A_{fast} , A_{medium} , A_{slow} .

If we want to assign a discovered motif to a linguistic term, for example we can compute the mean \bar{v} of all velocity values in the respective sequence. The linguistic term with the highest of the three membership degrees $\mu_{A_{\text{fast}}}(\bar{v})$, $\mu_{A_{\text{medium}}}(\bar{v})$ and $\mu_{A_{\text{slow}}}(\bar{v})$ is then labeled to the motif.

If the experiments are designed thoroughly (i.e., they do not contain contradictory linguistic terms), then it is assumed that a time series which contains the labeled motif can be labeled in the same way. If this is not the case, we can firstly compute the relative frequencies of labeled motifs in a time series, and secondly assign several labels to this time series to a certain degree.

Every labeled motif and its linguistic term can thus represent the antecedent and the consequent of a rule, respectively. We can further hope that such a consequent corresponds to a test criterion. From the monitored experiments we finally obtain a set of rules which can be interpreted in terms of natural language by more or less great efforts.

5 Matching Labeled Motifs

So far we solely considered the data coming from the field tests. The assumption in Section 1 was that these trials are designed and performed very thoroughly. The system that needs to be tested may behave completely different in a real-world environment, e.g., when it is utilized by an end user. In this situation we face the problem that systems under real-world loads might not follow any designed schedule model.

Usually the only thing what remains to evaluate these systems is monitored sensor data that hopefully contains motifs similar to the ones from the experiments. These real-world data is foremost approximated memory-efficiently (see Section 2) before we try to find motifs in the data (cf. Section 3). Now we can try to label the newly discovered motifs with similar linguistic terms by means of the already labeled motifs from the field tests. In machine learning, this would correspond to classification that is based on unsupervised learning.

Remember that it is very important to choose an adequate distance measure in order to compare two motifs. For example Lin und Keogh [6] have developed not only SAX but the so-called MINDIST function which computes the distance between two SAX words. It is preferable to use this function since the sequences are stored as SAX words anyway. Of course, other distance measures, e.g. (1), could be used as well.

No matter which measures we choose, eventually every real-world time series can be matched with previously unknown criteria. Taking everything into account, we can state that a classification into different criteria is thus a trivial consequence. Nevertheless, we have to consider that this classification should be carried out rather fuzzy than crisp. Accordingly, the usage of fuzzy clustering methods [16] seems to be desirable.

6 Adjusting the Experiments

Having finally discovered all motifs of the real-world data and labeled them to the already existing ones, experts should have a closer look at the results of the matching. The goal should be to adjust the original experiments such that they will resemble the time series more than before.

In total, three different possibilities have to be distinguished. If an unseen motif (coming from any real-world case) could be matched easily with a motif from a field test, then we can assume that we found some important feature of the system behavior. At any rate, such characteristics should be kept in all experiments in the next generation of system tests.

Experts would probably react differently in the case that a motif is exclusively discovered in field tests and not in real-world case. Such a feature should most likely be removed from the experiments after expert opinion. It is clear that this type of motif does not matter at all.

If there are in turn motifs in unseen time series that do manifest themselves in any trial, then experts have to adjust at least one trial. After all, this motif seems to be a recurrent feature of the system which occurred either never or not often enough in the field tests.

When all motifs are examined and the test design is improved, the next generation of experiments can be performed. The gained knowledge about, e.g., loading, service live, which results from the tests should consequently be more consistent with the serial product used in the reality. Finally, these experiences can possibly provoke enhancements of the systems.

7 Conclusions and Future Work

In this paper, we dealt with the question how field tests of systems that are produced in series might be adjusted to real circumstances. We especially pursued the efficient analysis of multivariate time series. This is due to the fact that in practice there is usually nothing but monitored data from many sensors available.

We argue that symbolic representations (in particular SAX) are comparatively superior in the data analysis of time series. Furthermore we can find very efficient methods which allow us to find recurrent sequences, i.e., motifs, in multivariate time series very fast. Unfortunately, there is not any satisfying heuristics to handle the vast number of parameters which influence the search and thus the success of the application.

Some unseen time series can be matched with the field trials by means of linguistic terms and discovered motifs in both the tests and real-world cases. The linguistic terms may either result from human experts or be generated from temporal sequences. After the clustering of all time series, the same experts might adjust the original tests to the observed reality.

In our work, we did not cover the discovery of errors and anomalies [5] in the measured data. We are rather concerned how to support the optimization of test procedures. While we are looking for frequent patterns, anomalies and errors usually occur highly infrequent.

The idea was born during an external funded project that we are working on with an industrial partner. This partner supplies us with much data from experiments and matched test criteria which shall describe the experiments. Usually in the industry one-dimensional load spectra are used to compare experiments with the reality. Our work is meant as addition to existing methods for data analysis.

So far we proceeded in our efforts to the discovery of some meaningful motifs. Though given the nondisclosure of the project, we are not allowed to publish either a name, or results, or some visualization of the data.

The next step will be the application of fuzzy clustering methods to all found subdimensional motifs in order to generate rules for some of the time series. This should help us to analyze unseen time series. Moreover, we plan both to automate the labeling of motifs with the linguistic terms and to verify them with the given test criteria. In order to do so, we will not only restrict ourselves to local trends in sequences (e.g., mean values) but we will consider the variability and the length of a motif as well.

The latter measure will probably be hard to obtain since the length of any motif is unknown in principle. If we can develop an algorithm that discovers motifs of different lengths, then we could focus on the next challenge in this field. Approaches from [17, 10] may be helpful to solve this task.

References

- [1] Lin, J.; Vlachos, M.; Keogh, E. J.; Gunopulos, D.: Iterative Incremental Clustering of Time Series. In: *EDBT* (Bertino, E.; Christodoulakis, S.; Plexousakis, D.; Christophides, V.; Koubarakis, M.; Böhm, K.; Ferrari, E., Hg.), Bd. 2992 von *Lecture Notes in Computer Science*, S. 106–122. Springer. ISBN 3-540-21200-0. 2004.
- [2] Xi, X.; Keogh, E. J.; Shelton, C. R.; Wei, L.; Ratanamahatana, C. A.: Fast time series classification using numerosity reduction. In: *ICML* (Cohen, W. W.; Moore, A., Hg.), Bd. 148 von *ACM International Conference Proceeding Series*, S. 1033–1040. ACM. ISBN 1-59593-383-2. 2006.
- [3] Patel, P.; Keogh, E. J.; Lin, J.; Lonardi, S.: Mining Motifs in Massive Time Series Databases. In: *ICDM* [18], S. 370–377. 2002.
- [4] Höppner, F.: Discovery of Temporal Patterns: Learning Rules about the Qualitative Behaviour of Time Series. In: *PKDD* (De Raedt, L.; Siebes, A., Hg.), Bd. 2168 von *Lecture Notes in Computer Science*, S. 192–203. Springer. ISBN 3-540-42534-9. 2001.
- [5] Lin, J.; Keogh, E. J.; Lonardi, S.: Visualizing and Discovering Non-Trivial Patterns In Large Time Series Databases. *Information Visualization* 4 (2005) 2, S. 61–82.
- [6] Lin, J.; Keogh, E. J.; Lonardi, S.; Chiu, B. Y.: A symbolic representation of time series, with implications for streaming algorithms. In: *DMKD* (Zaki, M. J.; Aggarwal, C. C., Hg.), S. 2–11. ACM. 2003.
- [7] Keogh, E. J.; Chakrabarti, K.; Pazzani, M. J.; Mehrotra, S.: Dimensionality Reduction for Fast Similarity Search in Large Time Series Databases. *Knowledge and Information Systems* 3 (2001) 3, S. 263–286.
- [8] Chiu, B. Y.; Keogh, E. J.; Lonardi, S.: Probabilistic discovery of time series motifs. In: *KDD* (Getoor, L.; Senator, T. E.; Domingos, P.; Faloutsos, C., Hg.), S. 493–498. ACM. ISBN 1-58113-737-0. 2003.
- [9] Buhler, J.; Tompa, M.: Finding Motifs Using Random Projection. *Journal of Computational Biology* 9 (2002) 2, S. 225–242.
- [10] Yankov, D.; Keogh, E. J.; Medina, J.; Chiu, B. Y.; Zordan, V. B.: Detecting time series motifs under uniform scaling. In: *KDD* (Berkhin, P.; Caruana, R.; Wu, X., Hg.), S. 844–853. ACM. ISBN 978-1-59593-609-7. 2007.
- [11] Tanaka, Y.; Iwamoto, K.; Uehara, K.: Discovery of Time-Series Motif from Multi-Dimensional Data Based on MDL Principle. *Machine Learning* 58 (2005) 2-3, S. 269–300.
- [12] Minnen, D.; Starner, T.; Essa, I. A.; Isbell, Jr, C. L.: Improving Activity Discovery with Automatic Neighborhood Estimation. In: *IJCAI* (Veloso, M. M., Hg.), S. 2814–2819. 2007.
- [13] Minnen, D.; Isbell, Jr, C. L.; Essa, I. A.; Starner, T.: Detecting Subdimensional Motifs: An Efficient Algorithm for Generalized Multivariate Pattern Discovery. In: *ICDM*, S. 601–606. IEEE Computer Society. 2007.

- [14] Dubois, D.; Prade, H. (Hg.): *Fundamentals of Fuzzy Sets*. Boston, MA, USA: Kluwer Academic Publishers. ISBN 978-0-792-37732-0. 2000.
- [15] Zadeh, L. A.: The Concept of a Linguistic Variable and its Applications to Approximate Reasoning–I. *Information Sciences* 8 (1975) 3, S. 199–249.
- [16] Höppner, F.; Klawonn, F.; Kruse, R.; Runkler, T.: *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*. New York, NY, USA: John Wiley & Sons Ltd. ISBN 978-0-471-98864-9. 1999.
- [17] Oates, T.: PERUSE: An Unsupervised Algorithm for Finding Recurring Patterns in Time Series. In: *ICDM* [18], S. 330–337. 2002.
- [18] *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002), Maebashi City, Japan, December 9-12, 2002*. IEEE Computer Society. ISBN 0-7695-1754-4. 2002.

Mehrdimensionales Systemverhalten mit evolutionär verbesserten neuronalen Netzen modelliert

E. Kraft¹, M. Schneider¹, U. Lehmann¹, J. Krone¹

¹Institut für Computer Science, Vision und Computational Intelligence,
Fachhochschule Südwestfalen, Frauenstuhlweg 31, 58644 Iserlohn
Tel. (02371) 566-303 Fax (02371) 566-209
E-Mail: erwin_kraft@web.de
E-Mail: {MSchneider, Lehmann, Krone}@fh-swf.de

Abstract

Technische Anwendungen sind von einer guten Abbildung des Systemverhaltens abhängig. Die meisten realen Systeme sind von komplexer Natur und verhalten sich nichtlinear. Es ist schwierig, solche Systeme zu modellieren, da die zugrunde liegenden mathematischen Abhängigkeiten nicht bekannt sind. Es gibt jedoch die Möglichkeit, Messdaten zu generieren, welche das Systemverhalten beschreiben. Diese Daten können für das Training Künstlicher Neuronaler Netze (KNN) genutzt werden. Messdaten sind oft verrauscht und enthalten weitere Unsicherheiten. Die Verfügbarkeit von Messdaten/ Trainingsdaten ist zudem meist sehr beschränkt. In diesem Artikel wird eine Methode vorgestellt, um die Generalisierungsleistung Künstlicher Neuronaler Netze bei Fehlen von adäquaten Trainingsdaten durch evolutionäre Optimierung zu verbessern. Es werden reale Daten eines großflächig gekrümmten Bauteils aus der Flugzeuggruppmontage verwendet.

1 Einleitung

Viele Anwendungen, wie z. B. die Simulation und die Optimierung technischer Systeme, sind auf eine gute Abbildung des Systemverhaltens angewiesen. Das Systemverhalten wird oft als Relation zwischen Systemeingang und Systemausgang modelliert. Dies kann durch eine Abbildung $f_{trans}(\bar{x}): \mathbb{R}^n \mapsto \mathbb{R}^m$ eines Eingangsvektors \bar{x} ($\bar{x} \in X, X \subseteq \mathbb{R}^n$) auf einen Ausgangsvektor \bar{y} ($\bar{y} \in Y, Y \subseteq \mathbb{R}^m$) beschrieben werden. Im Falle einer Simulation wird die Frage untersucht: Welche Ausgangsdaten werden durch beliebige Eingangsdaten erzeugt? Optimierungsprobleme sind hingegen oftmals an der inversen Abbildung $f_{inv}(\bar{y}): \mathbb{R}^m \mapsto \mathbb{R}^n$ interessiert: Welche Eingangsdaten werden benötigt, um gewünschte Ausgangsdaten zu erzeugen? Falls es sich bei der Abbildung $f_{trans}(\bar{x})$ nicht um eine Bijektion (verschiedene Eingangsdaten können dieselben Ausgangsdaten erzeugen) handelt, sind die Ausgangsdaten $\bar{x}_* \in X_*$ der inversen Abbildung $f_{inv}(\bar{y})$ auf eine Untermenge $X_* \subset X$ mit $X \subseteq \mathbb{R}^n$ beschränkt. In beiden Fällen kann das Problem in unterschiedliche Abbildungen des Typs $f_i(\bar{x}): \mathbb{R}^n \mapsto \mathbb{R}$ (Simulation) und $g_j(\bar{y}): \mathbb{R}^m \mapsto \mathbb{R}$ (Optimierung) mit $i=1, \dots, m$ und $j=1, \dots, n$ zerlegt werden:

$$f_{trans}(\vec{x}) : \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \mapsto \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} f_1(x_1, x_2, \dots, x_n) \\ f_2(x_1, x_2, \dots, x_n) \\ \vdots \\ f_m(x_1, x_2, \dots, x_n) \end{pmatrix} \quad (1.1)$$

$$f_{inv}(\vec{y}) : \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} \mapsto \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} g_1(y_1, y_2, \dots, y_m) \\ g_2(y_1, y_2, \dots, y_m) \\ \vdots \\ g_n(y_1, y_2, \dots, y_m) \end{pmatrix} \quad (1.2)$$

Jeder Systemausgang wird hierbei unabhängig von anderen Systemausgängen betrachtet und jede der Funktionen $f_i(\vec{x})$ und $g_j(\vec{y})$ wird durch ein Künstliches Neuronales Netz (KNN) modelliert. Eine solche Zerlegung von $f_{trans}(\vec{x})$ und $f_{inv}(\vec{y})$ in Funktionen des Typs $f_i(\vec{x})$ und $g_j(\vec{y})$ ist jedoch nicht unbedingt immer die beste Lösung. Da die einzelnen KNN jeweils nur einen Ausgang besitzen, sind während des Trainings weniger Fehlerinformationen vorhanden. Die Anzahl der Verbindungen zwischen den Neuronen wird jedoch durch die Zerlegung reduziert, so daß weniger Trainingsdaten für das Anpassen der Verbindungsgewichte des KNN benötigt werden. Dieses Modell geht davon aus, daß sich die Ausgänge nicht gegenseitig beeinflussen und von den Eingängen abhängig sind.

Für viele reale Anwendungen wird die Menge der Trainingsdaten $P\{X, Y\}$, $X \subseteq \mathbb{R}^n$ und $Y \subseteq \mathbb{R}^m$ durch Messung erzeugt. Hierzu werden die Daten von verschiedenen Sensoren gemessen um zusammengehörige Eingangs- und Ausgangsdaten zu erzeugen. Dies kann eine komplexe Aufgabe sein, da u. U. verschiedene Messverfahren eingesetzt werden und die Daten entsprechend zu Synchronisieren sind. Die so generierten Trainingsmuster $p_i = (\vec{x}_i, \vec{y}_i)$ ($p_i \in P$, $i = 1, \dots, |P|$) sind i. d. R. verrauscht. Die Trainingsmenge P kann auch falsche Muster $\tilde{p}_i(\vec{x}_i, \vec{y}_i)$ enthalten, wobei $\vec{x}_i \not\propto \vec{y}_i$.

In vielen Fällen ist die Anzahl der Trainingsmuster begrenzt, da Messungen in einem komplexen System zeitaufwendig und kostenintensiv sind. Es ist schwierig für die KNN, das Systemverhalten aus einer begrenzten Anzahl von Trainingsmustern zu erlernen. KNN mit einer großen Anzahl von Neuronen in den versteckten Schichten erreichen i. d. R. eine hohe Performanz auf den Trainingsdaten, liefern aber schlechte Ergebnisse, wenn den KNN neue Daten präsentiert werden. Es sind nicht genügend Trainingsmuster vorhanden, um alle Parameter der KNN zu optimieren. KNN mit einer geringen Anzahl versteckter Neuronen sind dagegen oftmals nicht in der Lage, das Systemverhalten gut abzubilden.

In diesem Artikel wird ein Verfahren vorgestellt, mit welchem die Performanz der KNN durch evolutionäre Optimierung [8, 9] verbessert werden kann, insbesondere wenn die Anzahl der verfügbaren Trainingsmuster begrenzt ist. Im nächsten Abschnitt wird dazu eine Anwendung im Bereich der Optimierung, für die Montage einer Seitenschale eines Flugzeugruppfes vorgestellt. Alle Tests werden im Rahmen dieser Anwendung durchgeführt, um die Performanz des Verfahrens für eine reale Anwendung mit realen Daten zu evaluieren.

1.1 Testanwendung

Es werden Daten einer Anwendung zur Montage eines Flugzeugrumpfes verwendet. Hierbei ist eine Seitenschale einer Sektion des Flugzeugrumpfes mittels verfahrbaren Positionierern so zu platzieren, daß bestimmte Schlüsselpositionen der Seitenschale mit Schlüsselpositionen anderer Rumpfteile übereinstimmen. Die Rumpfsktion, welche untersucht wird, besteht im Wesentlichen aus vier Teilen: Der unteren Sektion, der linken und rechten Seitenschale, sowie der oberen Sektion. Zunächst wird die untere Sektion relativ zu einem Weltkoordinatensystem ausgerichtet. Anschliessend wird die linke bzw. rechte Seitenschale an sechs unabhängig voneinander verfahrbaren Positionierern aufgehängt. Diese Positionierer sind in y - und z - Richtung des Weltkoordinatensystems beliebig verfahrbar, in x - Richtung können sie sich nur gemeinsam bewegen. Die räumliche Position der Positionierer kann in Weltkoordinaten bestimmt werden.

Sieben Markierungen sind auf der Seitenschale installiert. Diese Markierungen werden als Targets bezeichnet. Die Position eines Targets kann in Weltkoordinaten ermittelt werden. Die Targets definieren so die Geometrie und räumliche Position der Seitenschale. Um den Montageprozess durchzuführen, wird jeder Positionierer relativ zur Eintaktposition (Position nach dem Aufhängen der Seitenschale) verfahren, so daß die Distanzen d_i der aktuellen Targetpositionen T_i und den Schlüsselpositionen T_i^{Key} kleiner als eine bestimmte Schwelle ε sind:

$$d_i = \|T_i - T_i^{Key}\|_2 < \varepsilon, \quad i = 1, \dots, 7 \quad (1.3)$$

Hierbei handelt es sich um ein Optimierungsproblem, bei welchem die Distanzen d_i minimiert werden. Die zu klärende Frage hierbei ist: Wie sind die Positionierer zu bewegen, sodass die Positionen der Targets mit den entsprechenden Schlüsselpositionen übereinstimmen? Um das Problem zu lösen, ist es notwendig zu wissen, welche Targetbewegungen durch das Verfahren der Positionierer verursacht werden. Hierbei handelt es sich um eine unbekannte nichtlineare Funktion. Das Verfahren eines einzelnen Positionierers kann eine Bewegung aller Targets zur Folge haben. Die Abhängigkeiten zwischen den Verfahrenswegen der Positionierer und den Bewegungen der Targets ist weitgehend unbekannt. Die Targetbewegungen \vec{a}_j sowie die Positioniererbewegungen \vec{t}_i werden durch die Differenzen zwischen den aktuellen Positionen T_i , A_j und den Eintaktpositionen T_i^{init} , A_j^{init} beschrieben:

$$\vec{t}_i = T_i - T_i^{init}, \quad i = 1, \dots, 7 \quad (1.4)$$

$$\vec{a}_j = A_j - A_j^{init}, \quad j = 1, \dots, 6 \quad (1.5)$$

Für das Optimierungsproblem werden die Targetbewegungen als Systemeingang und die Positioniererbewegungen als Systemausgang verwendet. Falls es möglich ist, die Abhängigkeiten zwischen Systemeingang und Systemausgang gut abzubilden, kann man die Vektoren, welche von den Eintaktpositionen T_i^{init} auf die Schlüsselpositionen T_i^{Key} zeigen, als Systemeingabe verwenden und die dazugehörigen Positioniererbewegungen \vec{a}_j vom Systemausgang verwenden, um die Targetpositionen möglichst nah an die Schlüsselpositionen heranzuführen. Jedes Target kann seine

Position in x -, y - und z - Richtung ($\vec{t}_i = (x_i, y_i, z_i)^T$) verändern. Im Gegensatz dazu kann ein Positionierer seine Position nur in y - und z - Richtung verändern, da ein Positionierer in x - Richtung nicht relativ zu den anderen Positionierern bewegt werden darf. Die Positioniererbewegungen werden deshalb durch Vektoren mit zwei Komponenten $\vec{a}_j = (y_j, z_j)^T$ modelliert. Die x - Komponente a_x , welche dieselbe für alle Positionierer ist, wird als ein einzelner Systemausgang behandelt. Das System besitzt somit 21 Eingänge und 13 Ausgänge. Die Eingangsvektoren \vec{i} enthalten die Targetbewegungen ($(7 \cdot 3) = 21$ Parameter) und die Ausgangsvektoren \vec{o} die Positioniererbewegungen ($(1 + 6 \cdot 2) = 13$ Parameter):

$$\vec{i} = \begin{pmatrix} \vec{t}_1 \\ \vdots \\ \vec{t}_7 \end{pmatrix} \quad (1.6)$$

$$\vec{o} = \begin{pmatrix} a_x \\ \vec{a}_1 \\ \vdots \\ \vec{a}_6 \end{pmatrix} \quad (1.7)$$

Es ist die Funktion $f(\vec{i}): \mathbb{R}^{21} \mapsto \mathbb{R}^{13}$ zu modellieren, welche Targetbewegungen \vec{i} auf Positioniererbewegungen \vec{o} abbildet. Wie bereits in Gleichung (1.1) und Gleichung (1.2) beschrieben, kann die Funktion $f(\vec{i})$ in 13 Abbildungen des Typs $f_j(\vec{i}): \mathbb{R}^{21} \mapsto \mathbb{R}$ zerlegt werden. Jede der Funktionen $f_j(\vec{i})$, $j=1, \dots, 13$ wird durch ein KNN modelliert. Die Anzahl der Trainingsmuster ist begrenzt, da das Erzeugen der Daten sehr kostenintensiv ist. Die Daten sind verrauscht und enthalten Trainingsmuster, bei welchen nicht eindeutig ist, ob die Eingangsvektoren und die Ausgangsvektoren zusammengehörig sind. Das Entfernen von Ausreißern in der Datenmenge gestaltet sich als schwierig, da nicht offensichtlich ist, welche Daten Ausreißer sind und welche nicht.

1.2 Verwandte Arbeiten

Rekurrente neuronale Netze [6, 7] (RNN) werden oft für die Vorhersage von Sequenzen verwendet. RNN stellen mächtige Konzepte dar, um Eingangssequenzen eines Systems auf Ausgangssequenzen abzubilden. Ein besonders interessantes RNN ist das sog. Echo State Network [7] (ESN), welches eine große Menge zufällig verbundener versteckter Neuronen mit festen Verbindungsgewichten besitzt. Während des Trainings werden lediglich die Verbindungen der Ausgangsneuronen mittels linearer Regression angepasst. Ein anderes RNN ist das sog. Long Short- Term Memory (LSTM). Kernstück der LSTM- Architektur ist die sog. Gedächtniszelle (Memory Cell), welche mittels eines Gradientenabstiegsverfahrens trainiert werden kann. Gedächtniszellen können mit anderen Neuronen verbunden werden, wodurch sich eine große Anzahl möglicher Netzarchitekturen ergibt.

Die Trainingsdaten für die beschriebene Anwendung in Abschnitt 1.1 lassen die Formulierung von Eingangs- und Ausgangssequenzen nicht zu, so daß RNN im Moment noch nicht eingesetzt werden können. Es werden deshalb einfache Feed Forward Netze (FFN) verwendet.

Evolutionäre Algorithmen [8, 9] (EA) in Verbindung mit dem Training von KNN wurden bereits erfolgreich eingesetzt. Einen sehr guten Überblick findet man in [1]. EA können verwendet werden, um die Verbindungsgewichte, die Architektur, Lernregeln sowie die Auswahl von Eingangsmerkmalen zu optimieren. Evolution stellt hierbei eine weitere Form der Adaption dar, zusätzlich zum Training der KNN. Viele frühere Ansätze fokussierten sich auf einzelne Aspekte von KNN und dem Training von KNN. So wird z. B. in [8] die Problematik der Initialisierung der Verbindungsgewichte behandelt, wohingegen Konzepte des aktiven Lernens (Active Learning) [10, 11, 12, 13] die für das Training optimalen Trainingsmuster ermitteln und in einer optimierten Trainingsmenge zusammenzufassen. EA hingegen ermöglichen die gleichzeitige Optimierung verschiedener Parameter eines KNN.

Ein häufiger Ansatz bei der Nutzung von EA um die Performanz von KNN zu verbessern ist, mittels EA die initialen Verbindungsgewichte des KNN zu ermitteln [2]. Die so gefundenen Verbindungsgewichte werden anschliessend mit Hilfe des Backpropagation- Algorithmus [13] angepasst. Gradientenabstiegsverfahren können in lokalen Minima hängen bleiben, insbesondere wenn die Fehleroberfläche stark zerklüftet ist. Das Absuchen des Raumes der Verbindungsgewichte mittels EA stellt eine Möglichkeit dar, lokale Minima zu vermeiden [1, 2, 6].

Die Netzarchitektur spielt eine zentrale Rolle beim Design von KNN. [3] und [4] nutzen mutationsbasierte Operatoren, um die Netzarchitektur, sowie die Verbindungsgewichte, zu optimieren.

2 Vorgeschlagenes Verfahren

Die Performanz von KNN basiert auf vielen Aspekten, wie z. B. der Anzahl und Qualität der vorhandenen Trainingsdaten, der Initialisierung der Verbindungsgewichte, der Netzarchitektur, dem Trainingsverfahren und der Vorverarbeitung der Daten. Das Ziel des Trainings von KNN ist es, Modelle zu generieren, die geringe Vorhersagefehler für neue Daten besitzen. Dies wird häufig als die Generalisierungsleistung eines KNN bezeichnet. Die Performanz eines KNN wird oftmals mithilfe von Testdaten ermittelt, welche nicht für das Training des KNN verwendet wurden. Das hier vorgestellte Verfahren ist auf drei Aspekte von KNN und dem Training von KNN ausgerichtet: Der Initialisierung der Verbindungsgewichte, der Netzarchitektur und einem Auswahlschema für Trainingsmuster. Alle Parameter werden dabei koadaptiv optimiert. Es ist schwierig, eine exakte Lösung für die Optimierung der Parameter zu liefern, wenn keine Informationen über den Gradienten vorliegen. Aus diesem Grund wurde ein evolutionäres Optimierungsverfahren gewählt, da hier lediglich die Ermittlung der Performanz eines KNN (Fitness) über ein statistisches Maß, wie z. B. dem mittleren absoluten Fehler oder dem maximalen Fehler, nötig ist.

Für die evolutionäre Optimierung ist die Kodierung der Parameter des KNN in genetische Informationen (Gene und Chromosomen) notwendig. Die Individuen des evolutionären Optimierungsprozesses besitzen jeweils drei Chromosomen. Jedes Chromosom kodiert die Parametermenge eines der drei oben genannten Aspekte. Neue Individuen werden mithilfe einer Crossover- Operation erzeugt, die auf die zugehörigen Chromosomenpaare der Eltern angewendet wird. Die Allele eines Gens im ersten Chromosom bezeichnen die Indizes der Trainingsmuster in der Trainingsmenge P , welche während des Trainings ausgewählt werden. Hierbei werden keine weiteren

Beschränkungen verwendet: Ein Trainingsmuster darf mehrfach ausgewählt werden, wohingegen andere Trainingsmuster nicht ausgewählt werden können. Für die Reproduktion des ersten Chromosomes werden die Chromosomen der Eltern an einer zufälligen Stelle getrennt und das neue Chromosom aus zufällig ausgewählten Teilen der Elternchromosomen zusammengesetzt.

Das zweite Chromosom kodiert die Netzarchitektur. Die Anzahl der Gene entspricht der Anzahl der versteckten Schichten. Ein Allel repräsentiert die Anzahl der Neuronen in einer Schicht. Für die Reproduktion werden die Allele des neuen Chromosoms aus dem zufällig gewichteten Mittelwert der Allele der Elternchromosomen bestimmt. Die Ergebnisse werden dabei auf die nächste Ganzzahl gerundet. Dies wird durch den Operator $C_R : (A \times A) \mapsto A$ beschrieben:

$$C_R(a_1, a_2) = R(a_1 \cdot r + a_2 \cdot (1-r)) \quad (1.8)$$

$A \subseteq \mathbb{R}$ beschreibt die Menge möglicher Allele, r ist eine zufällige Zahl aus dem Intervall $[0,1]$ und R beschreibt den Rundungsoperator, $a_1, a_2 \in A$.

Das dritte Chromosom kodiert die Verbindungsgewichte des KNN. Die Verbindungen zwischen zwei Schichten eines KNN können durch die Adjazenzmatrix W_k beschrieben werden:

$$W_k = \begin{pmatrix} w_{1,1} & \cdots & w_{1,N_k} \\ \vdots & \cdots & \vdots \\ w_{N_{(k-1)},1} & \cdots & w_{N_{(k-1)},N_k} \end{pmatrix} \quad (1.9)$$

N_k bezeichnet die Anzahl der Neuronen in der aktuellen Schicht und $N_{(k-1)}$ die Anzahl der Neuronen in der vorhergehenden Schicht, $w_{m,n}$ ist das Verbindungsgewicht zwischen dem n ten Neuron der aktuellen Schicht und dem m ten Neuron der vorhergehenden Schicht ($m \in N_{(k-1)}$, $n \in N_k$), $k = 2, \dots, K$, wobei k die Indizes der aktuellen Schicht bezeichnet und K die Anzahl der Schichten des Netzes (inklusive Eingabeschicht). Jede der Adjazenzmatrizen W_k repräsentiert ein Allel. W_k^1 und W_k^2 bezeichnen die Allele der jeweiligen Eltern und W_k^{Off} das Allel des Nachwuchses. Da Eltern und Nachwuchs unterschiedliche Netzarchitekturen besitzen können, ist die Größe der Matrizen W_k^1 , W_k^2 und W_k^{Off} nur in seltenen Fällen identisch. Zunächst wird deshalb die maximale Höhe und Breite der Matrizen bestimmt:

$$N_k^{Max} = \max(N_k^1, N_k^2, N_k^{Off}) \quad (1.10)$$

N_k^1 beschreibt die Breite der Matrix des ersten Elternteils, N_k^2 die des zweiten Elternteils und N_k^{Max} die Breite der Matrix des Nachwuchses. Die maximale Höhe wird auf dieselbe Weise ermittelt.

$$N_{(k-1)}^{Max} = \max(N_{(k-1)}^1, N_{(k-1)}^2, N_{(k-1)}^{Off}) \quad (1.11)$$

Nachdem die maximale Breite N_k^{Max} und Höhe $N_{(k-1)}^{Max}$ bestimmt wurden, werden die Verbindungsmatrizen W_k^1 und W_k^2 des ersten und zweiten Elternteils auf die neue Größe

$(N_{(k-1)}^{Max}, N_k^{Max})$ erweitert. Die neuen Koeffizienten der erweiterten Matrizen E_k^1 und E_k^2 werden dabei auf den Wert Null gesetzt:

$$E_k^1 = \begin{pmatrix} W_k^1 & 0 & \dots & 0 \\ 0 & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \vdots \\ 0 & \dots & \dots & 0 \end{pmatrix} \quad (1.12)$$

$$E_k^2 = \begin{pmatrix} W_k^2 & 0 & \dots & 0 \\ 0 & \dots & \dots & \vdots \\ \vdots & \dots & \dots & \vdots \\ 0 & \dots & \dots & 0 \end{pmatrix} \quad (1.13)$$

Dies ist identisch mit dem Hinzufügen neuer Neuronen zu der Schicht k , wobei die neuen Verbindungsgewichte auf Null gesetzt werden, damit die Berechnung der Ausgabe nicht beeinflusst wird. Da die Matrizen E_k^1 und E_k^2 nun dieselbe Größe besitzen, kann man die Verbindungsmatrizen E_k^{Off} des Nachwuchses ermitteln. Die neuen Verbindungsgewichte $w_{m,n}^{Off}$ werden dabei durch den zufällig gewichteten Mittelwert der entsprechenden Gewichte $w_{m,n}^1$ und $w_{m,n}^2$ der Matrizen E_k^1 und E_k^2 ermittelt. Dies wird mit Hilfe des Operators $C : (\mathbb{R} \times \mathbb{R}) \mapsto \mathbb{R}$ durchgeführt:

$$C(w_{m,n}^1, w_{m,n}^2) = w_{m,n}^1 \cdot r + w_{m,n}^2 \cdot (1-r) = w_{m,n}^{Off} \quad (1.14)$$

r bezeichnet eine zufällige Zahl aus dem Intervall $[0,1]$, $m = 1, \dots, N_{(k-1)}^{Max}$ und $n = 1, \dots, N_k^{Max}$, $w_{m,n}^1 \in E_k^1$, $w_{m,n}^2 \in E_k^2$ und $w_{m,n}^{Off} \in E_k^{Off}$. Um die Verbindungsmatrizen W_k^{Off} zu erhalten werden die Matrizen E_k^{Off} auf die Größe $(N_{(k-1)}^{Off}, N_k^{Off})$ reduziert.

Der Algorithmus für die genetische Optimierung kann nun wie folgt formuliert werden:

1. Initialisiere die Population der Individuen \wp . Jedes Individuum $i \in \wp$ repräsentiert ein KNN. Alle Parameter des KNN werden per Zufall initialisiert.
2. Jedes Individuum (KNN) wird für 20 Epochen mit dem Backpropagation-Algorithmus trainiert, wobei für die Anpassung der Gewichte das Levenberg-Marquardt-Verfahren eingesetzt wird. Die Trainingsmuster werden entsprechend den Indizes im ersten Chromosom der Individuen ausgewählt.
3. Für jedes Individuum $i \in \wp$ wird die Fitness ermittelt. Dies geschieht durch Berechnung der Vorhersagefehler $E_p = |o_p^{ANN} - o_p|$ für die Menge der Trainingsdaten P und der Menge der Testdaten T , wobei der maximale Fehler

E_{Max} als Fitnessindikator verwendet wird. E_p bezeichnet den Vorhersagefehler für ein Muster p , $p \in (P \cup T)$. o_p^{ANN} bezeichnet die Ausgabe des KNN für ein Muster p und o_p bezeichnet die gewünschte Ausgabe für p .
 $E_{Max} = \max(E_1, \dots, E_p, \dots, E_{|P|+|T|})$.

4. Die Elite λ der Population \wp , $\lambda \subset \wp$, wird ausgewählt. Dabei handelt es sich um die 10% der Individuen mit den niedrigsten maximalen Fehlern (beste Fitness)
5. Die Reproduktion für jedes Individuum $e \in \lambda$ wird durchgeführt. Dabei werden als Partner zufällige Individuen $i \in \wp$ aus der Population \wp ausgewählt. Der Nachwuchs wird wie in 2. beschrieben trainiert. Die unteren 10% der Individuen mit den größten maximalen Fehlern (schlechteste Fitness) wird durch den Nachwuchs ersetzt.
6. Bis die Anzahl der maximalen Generationen nicht erreicht ist, gehe zu 3.

3 Testergebnisse

Die verwendeten Trainingsdaten bestehen aus 164 Trainingsmustern und 15 Testmustern. Die Testmuster beschreiben Targetbewegungen außerhalb des bekannten Arbeitsbereiches der Trainingsmuster. Um geringe Vorhersagefehler für die Positioniererbewegungen zu erzielen, benötigen die KNN eine gute Generalisierungsleistung. Als erstes wurden die KNN ohne evolutionäre Optimierung trainiert. Es wurde dabei eine versteckte Schicht mit 15 Neuronen verwendet. Die Verbindungsgewichte wurden mit dem Nguyen- Widrow- Verfahren [10] initialisiert und jedes KNN 10 mal trainiert. Die KNN mit den geringsten maximalen Fehlern wurden für den Test ausgewählt. Abbildung 1 zeigt die Vorhersagefehler für die 15 Testmuster. Die Fehler übersteigen die Schwelle $\varepsilon = (a/2)mm$ mehrere Male. Anschliessend wurden die KNN mit dem evolutionären Optimierungsverfahren aus Abschnitt 2 trainiert. Die Population bestand aus 700 Individuen. Der Optimierungsprozess wurde nach 70 Generationen abgebrochen. Abbildung 2 zeigt die Vorhersagefehler der evolutionär optimierten KNN. Die Vorhersagefehler liegen hier unter der Schwelle $(a/5)mm$, bis auf den Positionierer 2:Y.

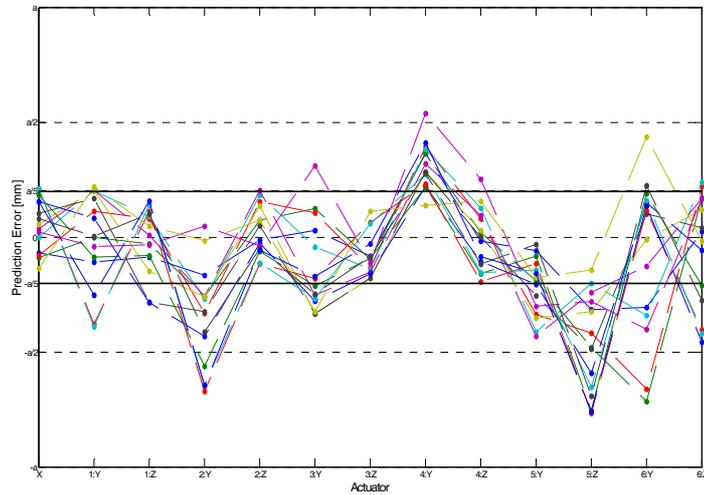


Abbildung 1: Vorhersagefehler für die KNN ohne evolutionäre Optimierung. Jedes KNN wurde 10mal trainiert. Für die Anpassung der Gewichte wurde das Levenberg- Marquardt- Verfahren verwendet. Die KNN mit den niedrigsten maximalen Fehlern wurden ausgewählt.

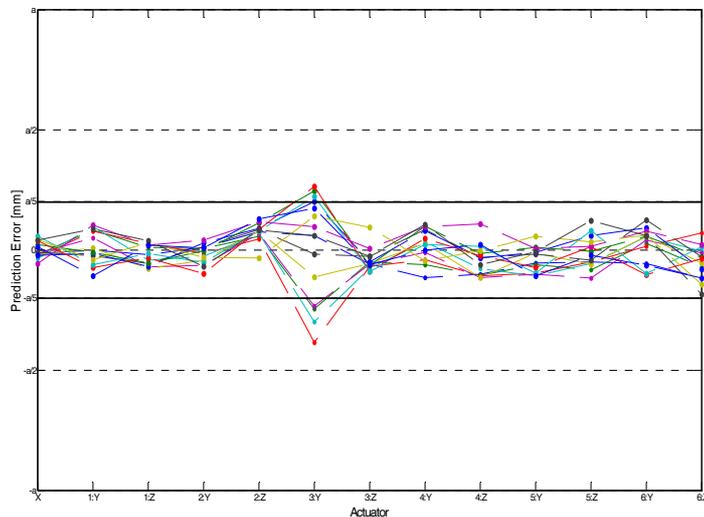


Abbildung 2: Vorhersagefehler für die Testdaten mit evolutionärer Optimierung. Die Population bestand aus 700 Individuen, der evolutionäre Optimierungsprozess wurde nach 70 Generationen abgebrochen.

4 Fazit

In diesem Artikel wurde eine Möglichkeit zur Modellierung komplexen Systemverhaltens mit Hilfe von KNN aufgezeigt. Eine reale Anwendung im Bereich der Flugzeugrumpfmontage wurde vorgestellt. 15 Testmuster dieser Anwendung wurden für die Evaluierung der Performanz der KNN genutzt. In Abschnitt 2 wurde ein Verfahren zur genetischen Optimierung der KNN vorgestellt. Der genetische Algorithmus optimiert die Trainingsmenge, die Netzarchitektur und die Verbindungsgewichte.

5 Danksagung

Das Institut für Computer Science, Vision und Computational Intelligence ist aus der Kompetenzplattform Computer Vision und Computational Intelligence (KOPF CV&CI) heraus gegründet worden. Die KOPF CV&CI wird vom Ministerium für Innovation, Wissenschaft, Forschung und Technologie des Landes Nordrhein-Westfalen gefördert.

Die Autoren wollen ihren aufrichtigen Dank an Herrn Dipl.-Ing. Wolfgang Brandt von der Airbus Deutschland GmbH für die Unterstützung dieser Forschungsarbeit ausdrücken.

6 Literaturverzeichnis

- [1] Yao, X. (1999), *Evolving Artificial Neural Networks*, Proceedings of the IEEE, Volume 87, Issue 9, pp. 1423-1447
- [2] Cantu-Paz, E., Kamath, C. (2005), *An Empirical Comparison of Combinations of Evolutionary Algorithms and Neural Networks for Classification Problems*, IEEE Transactions on Systems, Man, and Cybernetics, Part B, Volume 35, Issue 5, pp. 915 -927
- [3] Stanley, K. O., Bryant, B. D., Miikkulainen, R. (2003), *Evolving Adaptive Neural Networks with and without Adaptive Synapses*, The 2003 Congress on Evolutionary Computation, Volume 4, pp. 2557-2564
- [4] Gao, W. (2005), *New Evolutionary Neural Networks*, First International Conference on Neural Interface and Control, pp. 167-171
- [5] Konar, A. (2005), *Computational Intelligence*, pp. 167-195, Springer Verlag
- [6] Schmidhuber, J., Wierstra, D., Gagliolo, M., Gomez, F. (2007), *Training Recurrent Networks by Evolino*, Neural Computation, Issue 3, pp. 757-779
- [7] Jaeger, H. (2001), *The Echo State Approach to Analyzing and Training Recurrent Neural Networks*, Tech. Rep. No. 148, German National Research Center for Information Technology
- [8] Holland, J. H. (1992). *Adaptation in Natural and Artificial Systems*, The MIT Press
- [9] Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*, Frommann-Holzboog
- [10] Nguyen, D., Widrow, B. (1990). *Improving the Learning Speed of 2- Layer Neural Networks by Choosing Initial Values of the Adaptive Weights*, Proceedings of the International Joint Conference on Neural Networks, Vol. 3, pp. 21-26
- [11] Adejumo, A., Engelbrecht, A. P. (1999). *A Comparative Study of Neural Network Active Learning Algorithms*, Proceedings of the International Conference on Artificial Intelligence, pp. 32-35
- [12] Zhang, B. T. (1994). *Accelerated Learning by Active Example Selection*, International Journal of Neural Systems, Vol. 5(1), pp. 67-75
- [13] Fukumizu, K. (2000). *Statistical Active Learning in Multilayer Perceptions*, IEEE Transactions on Neural Networks, Vol. 11, Issue 1, pp. 17-26
- [14] Cohn, D., Ghahramani, Z., Jordan, M. (1996). *Active Learning with Statistical Models*, Journal of Artificial Intelligence Research, Vol. 4, pp. 129-145
- [15] Rumelhart, D., Hinton, G. (1986). *Learning Representations by Backpropagation Errors*. Nature, Vol. 323, pp. 533-536

A Single-Objective Fuzzy Clustering Model for Very High-Dimensional Data

Thomas A. Runkler
Siemens AG Corporate Technology
Information and Communications, CT IC 4
80200 München
Tel. 089/636-40010
E-Mail Thomas.Runkler@siemens.com

Abstract

One of the most popular models for finding clusters in data is the fuzzy c -means model. Recently, clustering high-dimensional data became an important research issue in the scientific community, mainly driven by applications in the field of bioinformatics. For very high-dimensional data, the fuzzy c -means model fails and produces so-called ignorant partitions that assign each object to each cluster with (almost) the same membership. Such trivial partitions contain only minimal information and are thus associated with very low values of the so-called partition coefficient. Therefore, it was recently suggested to combine the fuzzy c -means objective function with the partition coefficient and then to find a set of good cluster partitions by Pareto optimization using a genetic algorithm. This so-called Pareto fuzzy c -means clustering model produced good solutions for the considered experimental data sets but required a high computational effort because in each optimization step a whole population of solutions has to be evaluated. In this paper we suggest an approach with a considerably lower computational effort that we call high-dimensional fuzzy c -means. In this approach, the fuzzy c -means objective function and the partition coefficient are merged to a single-objective function using linear combination. For the special case of the fuzzifier $m = 2$ this leads to a very elegant formulation of the objective function. Experiments with the lung cancer data set show that this approach produces very good fuzzy partitions with low computational effort.

1 Introduction

Clustering high-dimensional data is a very active field of research, mainly driven by bioinformatics applications where microarrays measure very large numbers of features [5, 11]. Recently, it was shown that the *fuzzy c -means* (FCM) clustering model [2] is not suitable for very high-dimensional data because it tends to yield ignorant fuzzy partitions where each object is assigned to all clusters with (almost) the same membership [14]. This trivial solution can be avoided by considering not only the FCM objective function but at the same time also a cluster validity functional such as the *partition coefficient* [1]. The FCM objective function can be reformulated [8] and combined with the partition coefficient to form a multi-objective optimization problem [15] whose set of Pareto optimal solutions can be found using a genetic algorithm, for example the *non-dominated sorting genetic algorithm* NSGA-II [4]. In fuzzy clustering we are usually not interested in the complete set of Pareto optimal solutions but in individual solutions that reasonably balance between both clustering criteria. Hence, in this paper we propose a *single-objective*

clustering model using a linear combination of the FCM clustering model and the partition coefficient.

This paper is structured as follows: Section 2 gives a brief overview of the fuzzy c -means clustering model and illustrates its problems with very high-dimensional data. Section 3 reviews cluster validity measures, and particularly focuses on the partition coefficient. Section 4 shortly repeats the main findings of a previous paper where high-dimensional data were clustered based on a multi-objective objective function that was Pareto optimized by the multi-objective genetic algorithm NSGA-II. Section 5 presents an alternative single-objective approach that we call high-dimensional fuzzy c -means (HDFCM). Section 6 shows the properties of HDFCM based on some illustrative experiments. Section 7 finally summarizes the conclusions.

2 Clustering: Fuzzy c -Means

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

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

by minimizing the objective function

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

The necessary conditions for local extrema or saddle points of J_c specify the optimal locations of the cluster centers.

$$v_i = \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m}, \quad (3)$$

$i = 1, \dots, c$. Inserting (3) into (2) yields a *reformulated* fuzzy c -means objective function [8].

$$J_c(U; X) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m \left| x_k - \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \right|^2 \quad (4)$$

The necessary conditions for local extrema or saddle points of J_c also specify the optimal memberships.

$$u_{ik} = 1 / \sum_{j=1}^c \left(\frac{\left| x_k - \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} \right|}{\left| x_k - \frac{\sum_{k=1}^n u_{jk}^m x_k}{\sum_{k=1}^n u_{jk}^m} \right|} \right)^{\frac{2}{m-1}}, \quad (5)$$

$i = 1, \dots, c, k = 1, \dots, n$. Optimization of the FCM clustering model can be done by random initialization of the partition $U \in M_{fcn}$ and iterative update using (5) until an appropriate termination condition holds. In this paper we always use $m = 2$ and the Euclidean distance $|\cdot| = \|\cdot\|$.

The fuzzy c -means model has been successfully used in numerous applications. Recently, finding clusters in very high dimensional data sets such as gene expression data in bioinformatics [5, 11] has gained increasing interest. It was shown in [14], however, that the fuzzy c -means model is not adequate for such high dimensional data sets. To understand the reason for this inadequacy, imagine two p dimensional vectors generated by a p dimensional unitary random distribution, i.e. each dimension of each vector is a random value in the unit interval $[0, 1]$. For $p = 1$ the average distance between two such vectors (scalars in this case) is between zero and one, and it can be shown that the average distance is $1/3$. For arbitrary p the distance between two such vectors is between zero and \sqrt{p} , the length of the main diagonal in a p dimensional unit cube. The length between two arbitrary points in a p dimensional unit cube is mainly determined by the maximum distance along any of the p dimensions. If p becomes very large, then the probability increases that the distance along one of the p dimensions becomes close to one, for any arbitrary pair of points. This means that in a very high dimensional space the (Euclidean) distance between any arbitrary pair of points becomes almost equal. Of course this also holds in particular for an arbitrary pair of feature vector and cluster center, so the distances between all data points and all cluster centers become very similar. So, finally, if the memberships are computed using (5), each membership is very close to $u_{ik} = 1/c$. A partition matrix U whose memberships are all equal to $1/c$ is called *ignorant* or *indifferent*. Obviously, such matrices do not provide any nontrivial information. This confirms the previous statement that the fuzzy c -means model yields only ignorant (trivial) partitions for very high-dimensional data sets.

To avoid fuzzy clustering from producing ignorant partitions, extended clustering models were proposed that not only consider the cluster objective function but also cluster validity functionals. In the following sections we will illustrate this for the special case of the partition coefficient as a widely used cluster validity measure.

3 Cluster Validity: Partition Coefficient

Cluster validity measures quantify some kind of quality or some kind of information content of a partition matrix. One of the main purposes of cluster validity measures is to determine the number of fuzzy clusters c that are (naturally) contained in a data set X . This is done by the following algorithm: For $c = 2, \dots, c_{\max}, c_{\max} < n$, produce a fuzzy partition U_c with a given clustering algorithm and compute the validity $J_c = J(U_c)$. Then, pick the number of clusters that maximizes J_c : $c = \operatorname{argmax}\{J_2, \dots, J_{c_{\max}}\}$.

The list of cluster validity measures proposed in the literature includes the Davies–Bouldin index [3], the classification entropy [18], the proportion exponent [18], the cluster height [12], or the Xie–Beni index [19]. Here, we restrict to one of the most popular validity measures, the *partition coefficient (PC)* [1] defined as the average quadratic membership:

$$F_c(U) = \frac{1}{n} \sum_{k=1}^n \sum_{i=1}^c u_{ik}^2 \quad (6)$$

As we will see in chapter 5 the choice of the partition coefficient as a cluster validity measure leads to a very elegant formulation when linearly combined with the fuzzy c -means clustering functional for the special case of $m = 2$.

In the previous section we have pointed out that the fuzzy c -means model for very high-dimensional data yields the trivial partition $u_{ik} = 1/c$, $i = 1, \dots, c$, $k = 1, \dots, n$. For such an ignorant partition the partition coefficient becomes $F_{c \min} = 1/c$, which is the minimal partition coefficient achievable for any fuzzy partition. The maximum possible value of the partition coefficient is $F_{c \max} = 1$ which is achieved when U is a crisp partition, i.e. when each object is crisply assigned to exactly one cluster. Here, we are interested in a compromise solution between these two extremes: The memberships should not all be the same, i.e. F_c should be significantly higher than $1/c$; and the memberships should not be crisp, i.e. F_c should be significantly lower than 1. At the same time, of course, the cluster functional J_c should have a sufficiently high value. This means that we do not use cluster validity measures for an *a posteriori* rating of partitions, but include the validity measure directly into the clustering functional. More precisely, we want to find a good solution that is associated with a low but not necessarily globally minimal value of J_c and a high but not necessarily globally maximal value of F_c . There are several ways of combining such two criteria. One is a Pareto approach that explicitly searches for subsets of the (typically infinite) set of Pareto optimal solutions, and the other minimizes a linear combination of both criteria using a conventional (single-objective) optimization approach. The first option was pursued in [14], where the so-called *Pareto fuzzy c -means* (PFCM) clustering model was introduced. We briefly illustrate this approach in the next section. In the remaining sections of this paper we will present and discuss an approach for the second option.

4 Multi-Objective Approach: NSGA-II

In general, the set of Pareto optimal solutions is defined as the set of solutions for which none of a set of given criteria can be improved without deteriorating at least one other criterion. The Pareto fuzzy c -means model presented in [14] searches for a subset of the Pareto optimal set for the two criteria J_c and $-F_c(U; X)$. Notice that the negative partition coefficient is used, so that both criteria are minimized, not maximized. Pareto fuzzy c -means clustering then finds a set of partitions that is a finite subset of all Pareto optimal partitions

$$P_{JF_c} = \left\{ \begin{array}{l} U \in M_{fcn} \mid \\ \neg \exists U' \in M_{fcn} \\ J_c(U'; X) < J_c(U; X) \quad \wedge \\ -F_c(U'; X) < -F_c(U; X) \end{array} \right\} \quad (7)$$

Computing the set of Pareto optimal solutions is not a trivial task. Some of the methods frequently suggested in the literature for Pareto optimization are *particle swarm optimization* [16], *ant colony optimization* [13], or *genetic algorithms* such as the multi-objective genetic algorithm (MOGA) [6], the similar non-dominated sorting genetic algorithm NSGA [17], the strength Pareto evolutionary algorithm (SPEA) [21], its extension SPEA2

[20], or the fair evolutionary multi-objective optimizer (FEMO) [10]. In [14] a more efficient variant of NSGA called NSGA-II [4] is used. It was shown that NSGA-II produces reasonable clustering results also for very large data sets, but the computational effort is very high, especially because in each step a whole population of candidate solutions has to be evaluated. For finding clusters in the lung cancer data set [9] 1000 steps with a population of 100 individual solutions had to be performed, so in total $100 \times 1000 = 100,000$ solutions had to be evaluated. In order to reduce this computational effort, this paper suggests to combine both criteria to a single criterion by linear combination and then apply conventional (single-objective) optimization. This approach is illustrated in the following sections.

5 Single-Objective Approach: High-Dimensional Fuzzy c-Means

We combine the fuzzy c-means clustering functional J_c (2) and the (negative) partition coefficient $-F_c$ (6) by convex combination. In the general form, this yields the new clustering functional

$$J_{c^\circ} = a_0 + a_1 \cdot J_c - a_2 \cdot F_c \quad (8)$$

with three new scalar variables $a_0, a_1, a_2 \in \mathbb{R}$. The optimization that we will apply to minimize this clustering functional is not affected by adding arbitrary constants to the objective function, nor by multiplying the objective function with arbitrary constants. So, we can save two of the three variables and simply use one scalar variable $\alpha \in \mathbb{R}$ in the functional

$$J_c^* = J_c - \alpha \cdot n \cdot F_c \quad (9)$$

or, more explicitly,

$$J_c^* = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^m |x_k - v_i|^2 - \alpha \cdot \sum_{k=1}^n \sum_{i=1}^c u_{ik}^2 \quad (10)$$

In many applications of the fuzzy c-means algorithm the fuzzifier is set to $m = 2$. For this special case the new clustering functional J_2^* can be conveniently written as

$$J_2^*(U, V; X) = \sum_{i=1}^c \sum_{k=1}^n u_{ik}^2 (|x_k - v_i|^2 - \alpha) \quad (11)$$

We denote this as the objective function of the *high-dimensional fuzzy c-means* (HD-FCM) clustering model. The HD-FCM model is minimized using alternating optimization of the necessary conditions for extrema

$$v_i = \left(\sum_{k=1}^n u_{ik}^2 x_k \right) / \left(\sum_{k=1}^n u_{ik}^2 \right) \quad (12)$$

$$u_{ik} = 1 / \sum_{j=1}^c \left(\frac{|x_k - v_i|^2 - \alpha}{|x_k - v_j|^2 - \alpha} \right) \quad (13)$$

$i = 1, \dots, c, k = 1, \dots, n$. Notice that this scheme is equivalent to FCM with a specific *non-Euclidean* distance measure [7]. Since the distance is not an inner matrix norm, these equations may yield memberships outside the unit interval, so we truncate and renormalize the memberships at zero and one (first we set all $u_{ik} < 0$ to $u_{ik} = 0$ and set all $u_{ik} > 1$ to $u_{ik} = 1$; then we divide each u_{ik} by the sum of all $u_{jk}, j = 1, \dots, c$).

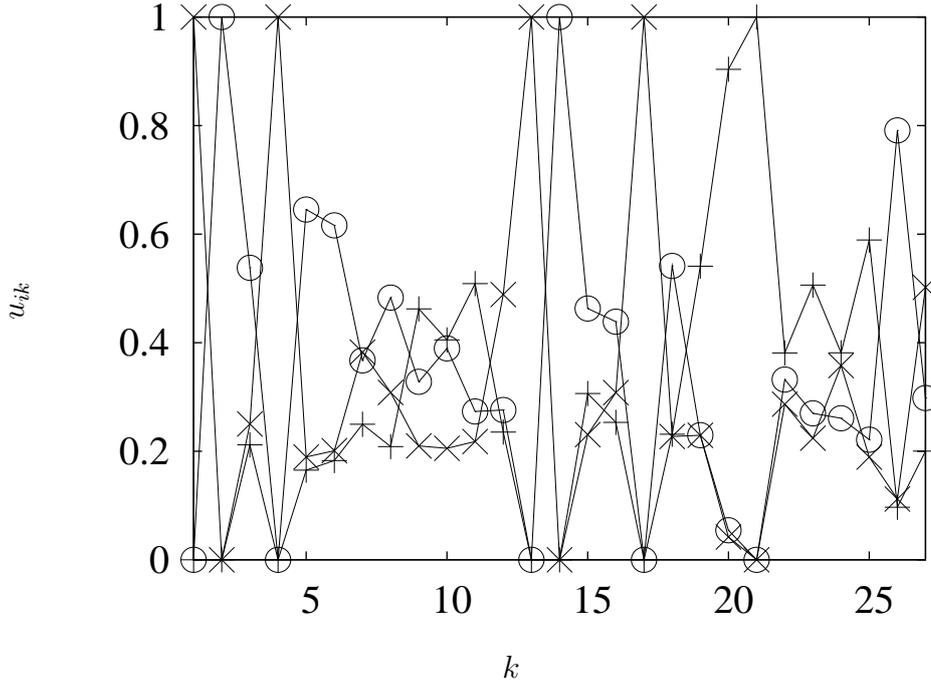


Figure 1: Memberships for the lung cancer data set.

6 Experiments

As an illustrative example we apply the HDFCM algorithm to the *lung cancer data set* [9] available at the *UCI Machine Learning Repository* which contains 32 vectors with 56 features associated with 3 different classes of pathological lung cancers. As shown in [14] (and theoretically discussed in Section 2) FCM always produces the trivial solution $u_{ik} = 1/3$ for all $i = 1, \dots, c, k = 1, \dots, n$, for this data set. As an alternative to FCM we run HDFCM for 100 steps with the parameters $c = 3, \alpha = 9$, and obtain the memberships shown in Fig. 1. Notice that the number of steps (100) is very low, and in most cases very good solutions are already found after about 10 steps; this compares with 1000 steps at a population size of 100 of the Pareto fuzzy c -means model as in [14], corresponding to a total of $100 \times 1000 = 100,000$ steps; so the computational effort of HDFCM is lower than that of PFCM by a factor of 1000. This partition is obviously not trivial, but clearly provides significant representatives with membership one for each cluster, and clearly recognizes objects that are only partially assigned to two or three clusters. In order to analyze how well this single-objective approach finds solutions close to the Pareto front we run the same algorithm for all values of $\alpha = \{6, 6.01, 6.02, \dots, 12\}$. The corresponding values of the FCM objective function and the negative partition coefficient are displayed in the scatter plot at Fig. 2. A comparison with the results in [15] shows that these solutions are a very good approximation of the Pareto front. Figs. 3 and 4 show how the resulting values of the FCM objective function and the negative partition coefficient depend of the algorithmic parameter α representing the weight of the partition coefficient in the joint single-objective function of HDFCM. As the weight of the negative partition coefficient α increases, lower values of the negative partition coefficient are achieved (Fig. 3). On the other hand, increasing α decreases the effective weight of the FCM objective

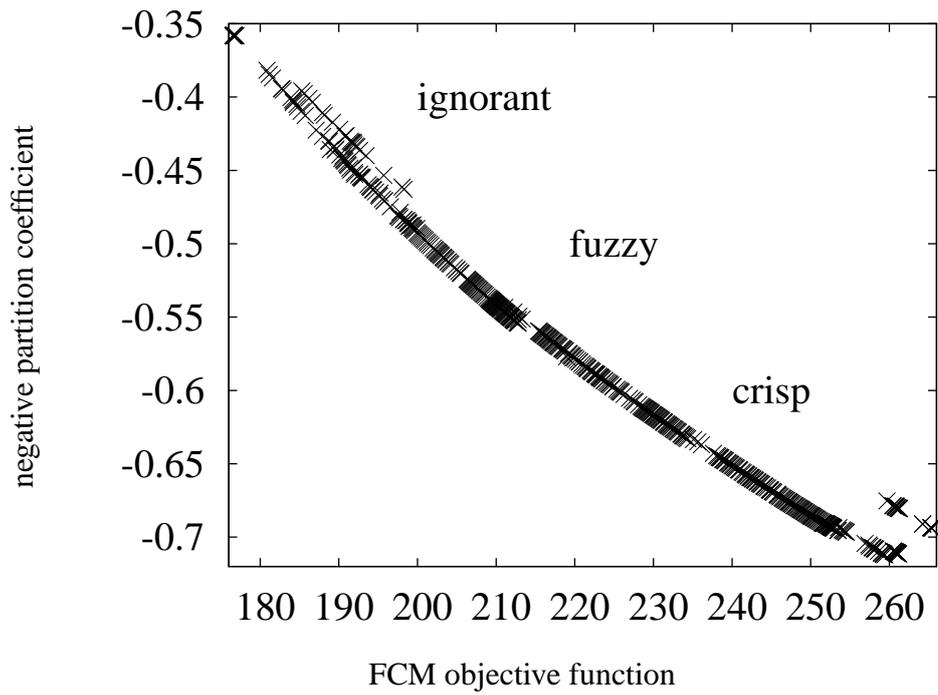


Figure 2: Approximation of the Pareto front.

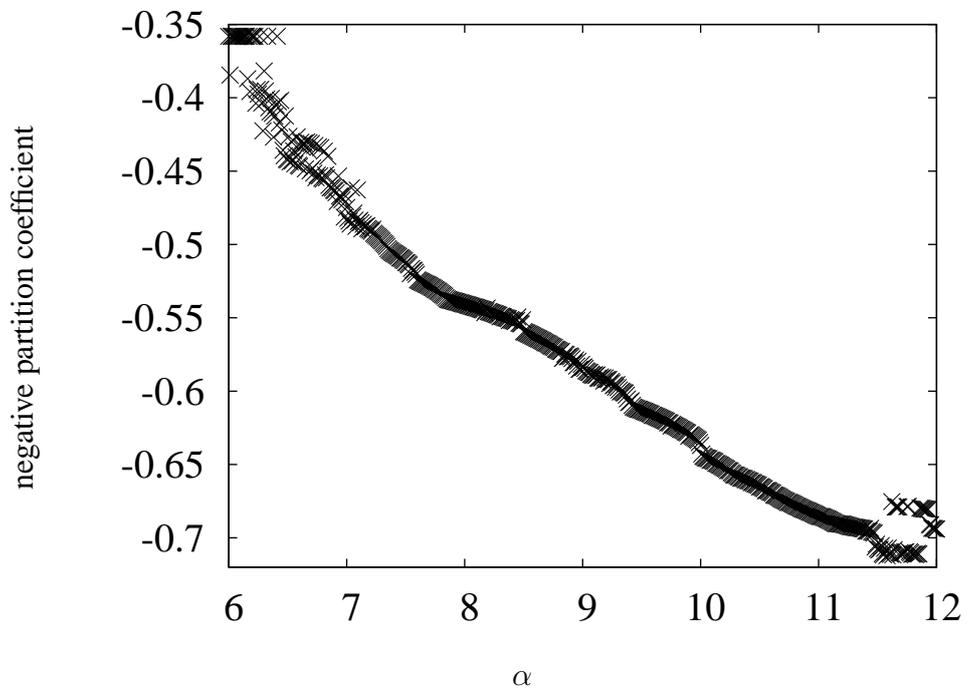


Figure 3: Negative partition coefficient versus α .

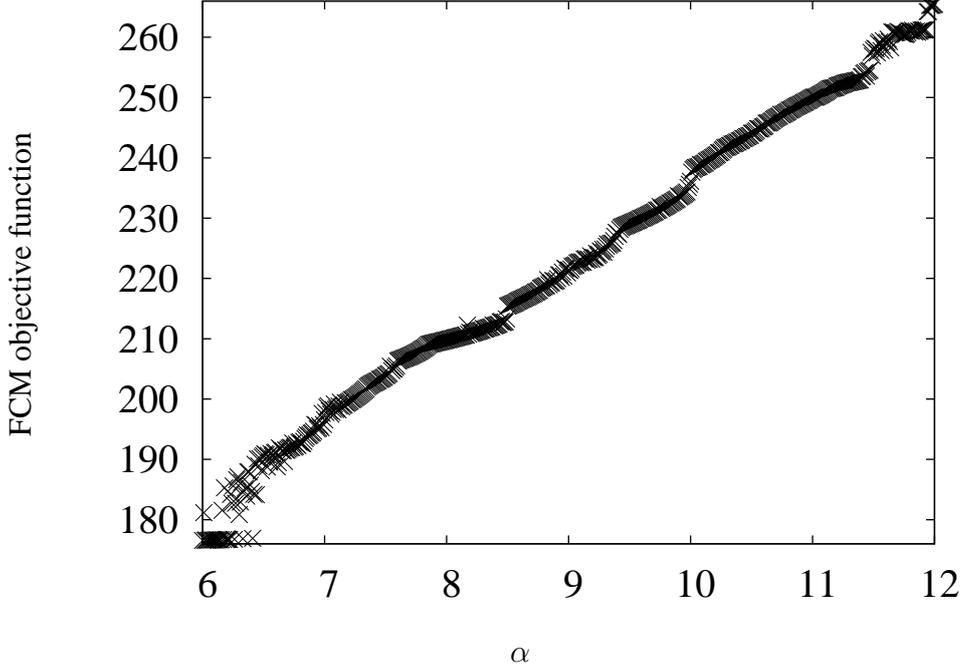


Figure 4: FCM objective function versus α .

function, so higher values of the FCM objective function are achieved (Fig. 4). Thus, low values of α lead to more ignorant partitions (top left in Fig. 2), and high values of α lead to crisper partitions (bottom right in Fig. 2). Reasonable fuzzy partitions (center in Fig. 2) are achieved when the effect of both components of the objective function, J_2 and $-F_2$ is almost the same. This can be achieved when the value of α is set to the average quadratic distance between data points and cluster centers

$$\alpha \approx \frac{1}{c} \frac{1}{n} \sum_{i=1}^c \sum_{k=1}^n |x_k - v_i|^2 \quad (14)$$

If the cluster centers are uniformly distributed in the feature space, then we can roughly approximate the average quadratic distance between data points and cluster centers by the average quadratic distance between pairs of data points divided by c , the number of clusters. Inserting this into (14) finally yields an equation for a reasonable choice of α

$$\alpha \approx \frac{1}{c} \frac{1}{n^2} \sum_{j=1}^n \sum_{k=1}^n |x_k - x_j|^2. \quad (15)$$

7 Conclusions

The results obtained in the experiments with the lung cancer data set show that HDFCM yields good compromise solutions between the FCM objective function and the partition coefficient. As opposed to FCM and HCM, HDFCM yields reasonable non-trivial fuzzy partitions for high dimensional data sets. As opposed to Pareto optimization of a multi-objective functional, HDFCM very efficiently finds individual solutions close to the Pareto

front. Thus, HDFCM is a very efficient and powerful model for finding clusters in high-dimensional data.

References

- [1] J. C. Bezdek. Numerical taxonomy with fuzzy sets. *Journal of Mathematical Biology*, 1(1):57–71, 1974.
- [2] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.
- [3] D. L. Davies and D. W. Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1:224–227, 1979.
- [4] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. In *Parallel Problem Solving from Nature*, pages 849–858, Paris, France, 2000.
- [5] M. Dejori and M. Stetter. Identifying interventional and pathogenic mechanisms by generative inverse modeling of gene expression profiles. *Journal of Computational Biology*, 11(6):1135–1148, 2004.
- [6] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *International Conference on Genetic Algorithms, Urbana–Champaign, USA*, pages 416–423, June 1993.
- [7] R. J. Hathaway and J. C. Bezdek. NERF c-means: Non-Euclidean relational fuzzy clustering. *Pattern Recognition*, 27:429–437, 1994.
- [8] R. J. Hathaway and J. C. Bezdek. Optimization of clustering criteria by reformulation. *IEEE Transactions on Fuzzy Systems*, 3(2):241–245, May 1995.
- [9] Z. Q. Hong and J. Y. Yang. Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Pattern Recognition*, 24(4):317–324, May 1991.
- [10] M. Laumanns, L. Thiele, E. Zitzler, E. Welzl, and K. Deb. Running time analysis of multi-objective evolutionary algorithms on a simple discrete optimization problem. In *Parallel Problem Solving From Nature, Granada, Spain*, pages 117–120, 2002.
- [11] M. Popescu, J. M. Keller, and J. Mitchell. Fuzzy measures on the gene ontology for gene product similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(3):263–274, July 2006.
- [12] T. A. Runkler. Improved cluster validity measures for the fuzzy c-elliptotype algorithm. In *European Congress on Intelligent Techniques and Soft Computing*, pages 1331–1335, Aachen, August 1995.
- [13] T. A. Runkler. Ant colony optimization of clustering models. *International Journal of Intelligent Systems*, 20(12):1233–1261, December 2005.

- [14] T. A. Runkler. Pareto optimality of cluster objective and validity functions. In *IEEE International Conference on Fuzzy Systems*, pages 79–84, London, July 2007.
- [15] T. A. Runkler. Pareto optimization of the fuzzy c–means clustering model using a multi–objective genetic algorithm. In *GMA/GI Workshop Fuzzy Systems and Computational Intelligence, Dortmund, Germany*, pages 229–241, December 2007.
- [16] T. A. Runkler and C. Katz. Fuzzy clustering by particle swarm optimization. In *IEEE International Conference on Fuzzy Systems*, pages 3065–3072, Vancouver, July 2006.
- [17] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.
- [18] M. P. Windham. Cluster validity for the fuzzy c–means clustering algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-4(4):357–363, July 1982.
- [19] X. L. Xie and G. Beni. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(8):841–847, August 1991.
- [20] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength Pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Zurich, Switzerland, 2001.
- [21] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

Anwendungsfälle für elektrische Energiespeicher und deren Dimensionierung mit Hilfe von Datenanalysen

Christian Arnold, Steffen Nicolai, Peter Bretschneider

Fraunhofer Anwendungszentrum Systemtechnik (FhG AST)

Am Vogelherd 50, 98693 Ilmenau

Tel.: (03677) 461-170; Fax.: (03677) 461-170

Email: {and, nic, bsr}@ast.iitb.fraunhofer.de

1 Einleitung

Die Voraussetzung für die Stabilität eines Elektroenergiesystems ist die Deckung des Bedarfs durch zeitgleiche Erzeugung. Im Allgemeinen wird diese Aufgabe mittels Regelung der bedarfsfolgenden Erzeugung erfüllt, welche zusätzlich durch die Prognose des zu deckenden Bedarfs unterstützt wird. Im liberalisierten Energiemarkt wird elektrische Energie flexibel von verschiedenen Erzeugern bezogen und über die elektrischen Netze verteilt. Hieraus resultieren komplexe Optimierungsprobleme, welche neben der bedarfsfolgenden Erzeugung auch die netzgebundene Energieverteilung berücksichtigen müssen. In der Regel befinden sich Erzeugung und Bedarf weder geografisch noch netztopologisch in unmittelbarer Nähe. Aufgrund mehrerer Transport- und Verteilmöglichkeiten ergeben sich somit Fragestellungen der optimalen Verteilung elektrischer Energie. (vgl. [1])

Um zeitliche und räumliche Differenzen der Energiebilanz zu kompensieren, stehen mehrere Möglichkeiten zur Verfügung. Die konventionelle Methode ist der Bezug von Regenergie, was bedeutet, dass aus technischer Sicht die Erzeugung abweichend vom prognostizierten Bedarf nachgeführt wird. Eine Alternative bietet das *Demand-Side-Management*, indem zum Ausgleichen der Differenz eine Nachführung von beeinflussbaren Lasten erfolgt. Die dritte Methode ist der Ausgleich über speichernde Elemente, welche als steuerbare Lasten und als steuerbare Erzeuger fungieren können.

Für die Speicherung elektrischer Energie gibt es verschiedene Ansätze und Verfahren wie z.B. Spulen und Kondensatoren für die direkte Speicherung. Als allgemeiner Nachteil ist die geringe Speicherfähigkeit elektrischer Energie zu nennen. Demgegenüber erweist sich die hohe Wandlungsfähigkeit elektrischer Energie als großer Vorteil. Eine Umwandlung in nahezu alle anderen Energieformen mit entsprechender Rückwandlung ist möglich, so dass neben der direkten auch indirekte Speicherungen erfolgen können und hieraus vielfältige Speichertechnologien resultieren.

Pumpspeicherkraftwerke und chemische Batterien sind bereits erfolgreich im Einsatz. Erstere erfordern hohe Investitionskosten (vgl. [2]) und sind an bestimmte geografische Anforderungen gebunden, so dass sie meist nur in großen Leistungsklassen anzutreffen sind (vgl. [3]). Herkömmliche chemische Batterien sind überwiegend für kleinere Leistungen vorgesehen. Vergrößerungen des Leistungsbereichs sind generell möglich, erfordern jedoch Verschaltungen einzelner Batterien. Zudem ist die Speicherung nur bedingt skalierbar, da Energieinhalt und Leistung in einem Verhältnis stehen und nur in Grenzen entkoppelt dimensioniert werden können (vgl. [2]).

Aktuelle Forschungs- und Entwicklungsarbeiten untersuchen moderne Speichertechnologien und stellen die Marktreife von Speichern der mittleren Leistungsklasse im Bereich von wenigen MW in Aussicht. Hierzu zählen insbesondere Wasserstoffspeicher, Redox-Flow-Batterien und Druckluftspeicher, deren Technologie- und Wirkprinzipien in [4] erläutert sind. Für die elektrischen Energiespeicher stellt sich sowohl für die Entwicklung als auch im Fall ihrer Verfügbarkeit die Frage, wie diese für unterschiedliche Anwendungen in das Netz zu integrieren, zu betreiben und zu dimensionieren sind.

Basis für die folgenden Betrachtungen ist ein technologieunabhängiges, verallgemeinertes Speichermodell (Abbildung 1). Alle Energiespeicher haben Stellglieder im Be- (*charge path* CP) und Entladeweg (*discharge path* DP). Diese können Umformungen elektrischer Größen oder Energiewandlungsprozesse beinhalten, welche stets verlustbehaftet ablaufen. Zudem kann Energie in gespeicherter Form durch Selbstentladung verloren gehen. Die Integration der Leistungsbilanz in der zentralen Speichereinheit (*central storage*, CS) lässt auf den Ladezustand schließen. Die Modellierung kann durch Abbildung einzelner technologiespezifischer Komponenten in dieser Struktur erfolgen.

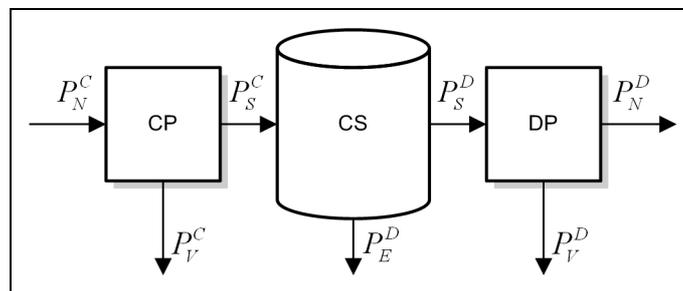


Abbildung 1: Verallgemeinertes Modell eines Energiespeichers

2 Anwendungsfälle für elektrische Energiespeicher

Zur Verdeutlichung der technischen und wirtschaftlichen Ziele der Energiespeicherung folgt ein grober Überblick möglicher Anwendungen des Einsatzes elektrischer Energiespeicher. Zwei interessante Anwendungsfälle zeigen exemplarisch die Abhängigkeiten und Problematiken der Speicherdimensionierung.

2.1 Überblick

Kunden mit einem hohen Elektroenergiebedarf wird neben der elektrischen Energie (elektrische Arbeit) auch die (jährlich) bereitgestellte Maximalleistung vom Lieferanten in Rechnung gestellt, wonach sich auch der Preis für die Netznutzung bemisst (vgl. [1]). Eine Vermeidung von Kosten ist möglich, indem eine Reduktion der maximalen Leistung erfolgt. Energiespeicher könnten hierbei verwendet werden, um in Schwachlastzeiten die Energie zu beziehen, welche später zu Spitzenlastzeiten verbraucht wird und eine Absenkung der bezogenen maximalen Leistung bewirkt.

Preise elektrischer Energie sind lieferanten- und zeitabhängig. Sondervertragskunden können die Flexibilität im Bezugsverhalten ausnutzen und wirtschaftlich optimierte Strombezugsverträge abschließen. Der Kunde meldet seinen künftigen Bedarf in Form von Fahrplänen. Der Energiespeicher könnte aus Kundensicht verwendet werden, um neue Freiheitsgrade bei der Bezugsplanung zu schaffen, indem Bezug und tatsächlicher Verbrauch zeitlich entkoppelt werden. Der Speicher wird in solchen Anwendungen parallel zur Last betrieben (Abbildung 2 links).

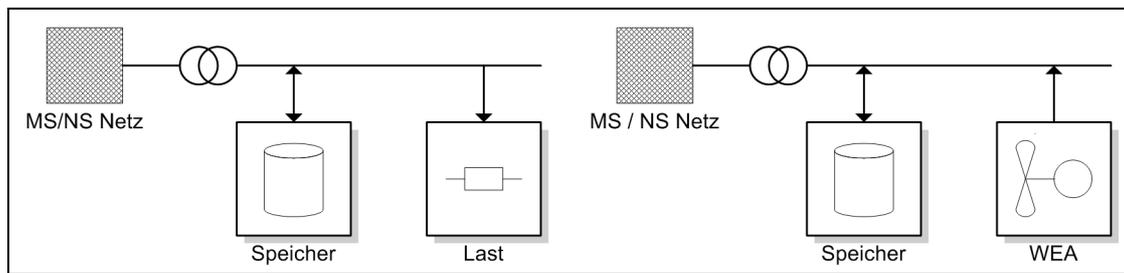


Abbildung 2: Speicher in Kombination mit Last (links) und Erzeuger (rechts)

Im Netz sind Erzeuger vorhanden, deren Bereitstellung elektrischer Energie nicht gesteuert werden kann, die Erzeugung ist nicht bedarfsabhängig. Hierzu zählen besonders regenerative Erzeugungen in Form von Photovoltaik- (PVA) und Windenergieanlagen (WEA). Wenn die Erzeugung solcher Anlagen nicht gesteuert werden kann, stellt sich die Frage, inwieweit die Erzeugung zumindest planbar ist. Die Einspeisung von PVA besitzt deterministische Komponenten, welche sich aus Sonnenstand und Anlagenaufstellung ergeben (vgl. [5]). Diesem planbarem Signalanteil sind stochastische Signale in verschiedenen Zeitbereichen überlagert, welche durch Abschattungen (z.B. durch Wolkenzüge) erzeugt werden. Im Signal der WEA-Erzeugung sind keine Muster aufzufinden. Flauten und Böen rufen in unterschiedlichen Zeitbereichen stochastische Signalanteile hervor. Die Fluktuationen von PVA und WEA, sind wegen zeitlichen und räumlichen Ausgleichseffekten (vgl. [6]), von Größe und Anzahl der Anlagen sowie den betrachteten Zeitbereichen abhängig. Die eingeschränkte Planbarkeit kann jedoch durch Kombination der Prognose für die stochastische Erzeugung mit einem Energiespeicher gesteigert werden, indem Fehler durch den parallelbetriebenen Speicher (Abbildung 2 rechts) abgefangen werden. Betreiber solcher Anlagen könnten dann die Einspeisung in Form eines Fahrplans anmelden oder gesondert verkaufen (Direktvermarktung).

Zu den Zielen der Verteilung elektrischer Energie zählen die Minimierung der Transportverluste, die Erhöhung von Netzstabilität und Ausfallsicherheit sowie die Steigerung der Lebensdauer von Betriebsmitteln. Weiterhin werden durch minimierte Nutzung vorgelagerter Netze Verlustleistungen, Nutzungsgebühren und Netzausbauten verringert oder gar vermieden. Um solche Aufgaben zu erfüllen, ist die Steuerung der Lastflüsse erforderlich. Da elektrische Energiespeicher sowohl die Funktion der Last als auch die der Erzeugung übernehmen können, ist es vorstellbar, dass Energiespeicher hierbei einen Beitrag leisten können. Erste Ansätze sind hierzu in [7] zu finden. Eine mögliche Systemstruktur ist in Abbildung 3 dargestellt.

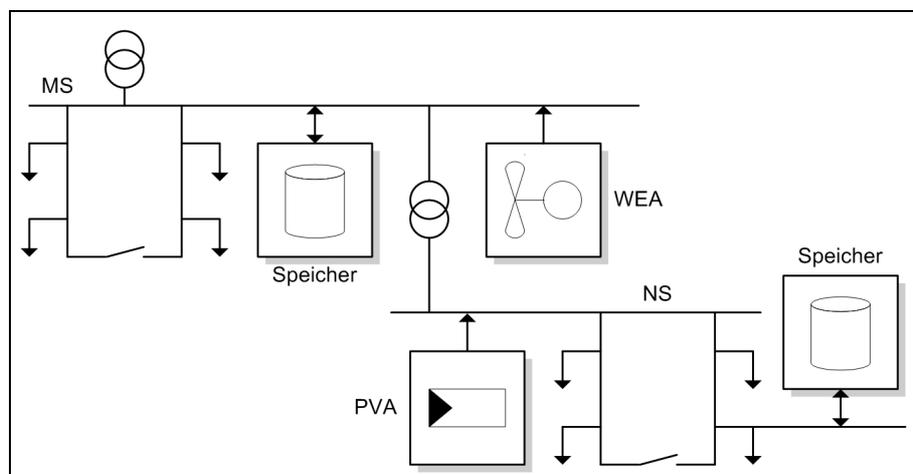


Abbildung 3: Speichereinsatz im Verteilnetz

2.2 Anwendungsbeispiel: Reduktion der Spitzenlast von (Sonder-)Kunden

Ziel der Anwendung ist die Reduktion der jährlichen Maximalbezugsleistung eines Industriekunden. Abbildung 4 links zeigt den normierten Verlauf des Leistungsbedarfs über ein Jahr. Leistungsspitzen werden reduziert, wenn in Spitzenlastzeiten die Energie aus dem Speicher bezogen wird. Die neue Maximalbezugsleistung P_{red} ist um die Speichernennleistung kleiner als die Ursprüngliche. In Abbildung 4 rechts ist die geordnete Leistungskurve des Bedarfs dargestellt. Aus der Speicherleistung resultiert ein Bereich der Kurve, um welchen die Kurve reduziert wird. Im Bereich unterhalb der neuen Maximalbezugsleistung muss der Speicher die Energiemenge aufnehmen, welche zur Spitzenglättung benötigt wird, zuzüglich der durch den Speicher hervorgerufenen Verluste.

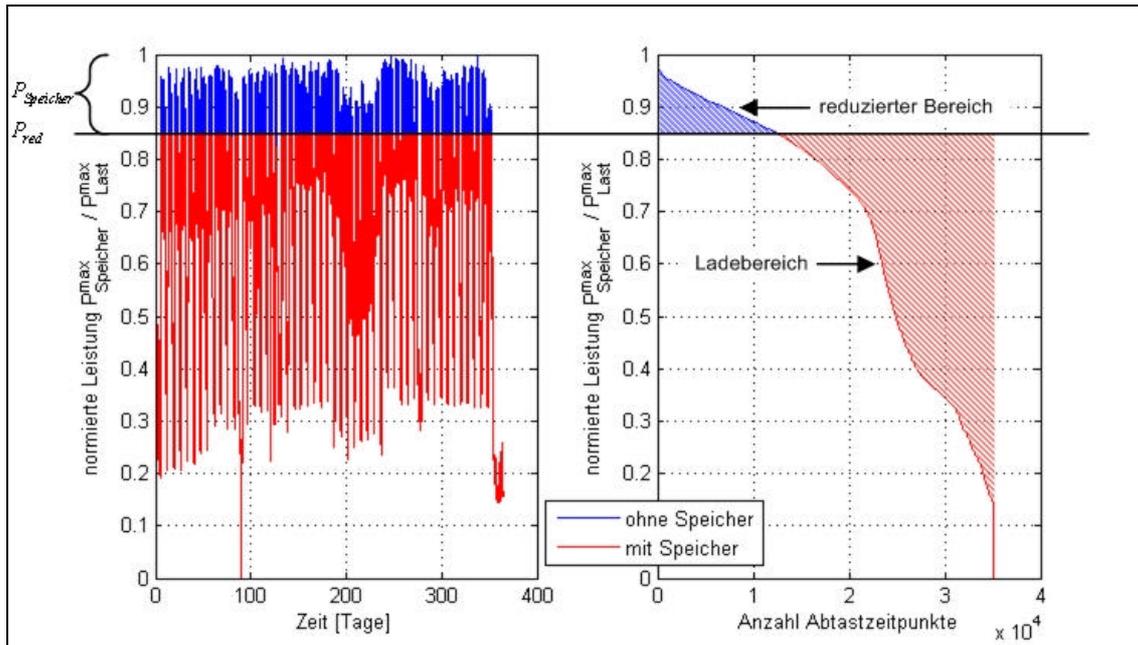


Abbildung 4: Verlauf der Bezugsleistungen und der geordneten Leistungskurve

Die Ladephasen können im Ladebereich frei gewählt werden. Folgende Betrachtungen gehen davon aus, dass der Speicher bis zum Initialfüllstand geladen wird, sobald eine definierte Grenze P_{load} unterschritten wird. Diese wird über einen Faktor zur maximalen Bezugsleistung in Beziehung gesetzt und dient als Entwurfparameter der Betriebsführung ($c = P_{red} / P_{load}$). Das Speichervolumen wird überdimensioniert, so dass nach einer Simulation die Differenz zwischen maximalem und minimalem Speicherfüllstand (*state of charge, SoC*) als notwendiges Speichervolumen interpretiert wird. In Abhängigkeit der Speicherleistung und dem Entwurfparameter ergeben sich die notwendigen Speichervolumina (Abbildung 5 rechts). Die triviale Betriebsführung zeigt bereits, dass sie einen signifikanten Einfluss auf die Dimensionierung hat. Abbildung 5 links ist zu entnehmen, dass der Lastgang Rhythmen aufweist. Tagsüber wird mehr Leistung bezogen als nachts, zudem fällt am Wochenende der Leistungsbezug auf bis zu 30% der Maximalleistung ab. Aufgrund der Rhythmen wird angenommen, dass der künftige Leistungsbedarf vorhergesagt werden kann (vgl. [8]). In alternativen Betriebsführungsstrategien können daher zusätzliche Freiheitsgrade zur Festlegung der Ladephasen in unterschiedlichen Schwachlastzeiten geschaffen werden. Die Dimensionierung umfasst also die Analyse von Betriebsführungen, die Generierung von Bewertungskriterien und entsprechende Optimierungen. Aus der Analyse von Simulationsdaten können Bewertungskriterien herangezogen werden, beispielsweise notwendige Speicherfüllstände.

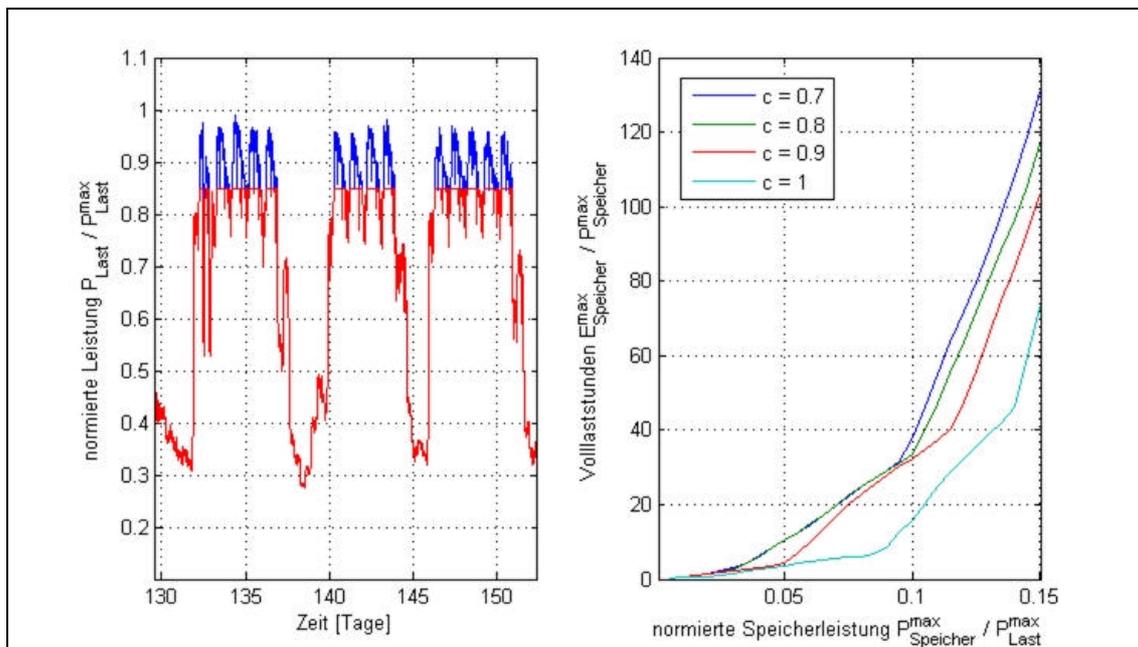


Abbildung 5: Verlauf des Leistungsbezuges (Zoom) und Kennlinienfeld der Speichervolumen

2.3 Anwendungsbeispiel: Steigerung der Planbarkeit von Windkraft

Bestehende Vorhersagesysteme für WEA- und PVA-Einspeisungen basieren auf Wettervorhersagen durch Erdbeobachtungssysteme, wie z.B. Satellitenbilder. Die Vorhersage der Wetterlage wird im Bereich weniger Stunden durch Extrapolation der Beobachtungsdaten und im Bereich weniger Tage mittels statistischer und numerischer Wettermodelle erstellt, mit physikalischen Anlagenmodellen wird auf den Verlauf der Einspeisung geschlossen (vgl. [9]). Für einzelne Anlagen sind Prognosen wegen der groben zeitlichen und räumlichen Auflösung der Beobachtungsdaten entsprechend fehlerbehaftet. Ausgleichseffekte treten auf, wenn die Gesamteinspeisung mehrerer Anlagen betrachtet wird, so dass zu diesem Zweck eine Steigerung der Prognosequalität erfolgt. Es ist somit möglich Prognosen für ganze Regelzonen zu erstellen, wie z.B. in [10]. Im Folgenden werden Erzeuger- und Prognosedaten für eine Regelzone mit stündlicher Auflösung herangezogen (Abbildung 6 oben, Datenquelle: [11]).

Eine naheliegende Anwendung wäre nun, den Prognosefehler durch den Einsatz eines Energiespeichers zu kompensieren und somit eine planbare Windenergieeinspeisung zu realisieren. Durch Ein- und Ausspeichervorgänge treten zusätzliche Effekte auf, wie Verluste beim Ein- und Ausspeichern. Die damit verbundene Planungs- und Betriebsführungsaufgabe muss daher folgende Größen beachten:

- Windenergieprognose und Windenergieprognosefehler
- Speicherfüllstände
- Verluste bei Speicher- und Lagerprozessen (siehe Abbildung 1)
- Dynamik bei Speicherprozessen

Hierzu wurde im Ansatz bisheriger Arbeiten das Ziel verfolgt, am Ende eines Speicherhorizontes einen definierten Speicherfüllstand zu erreichen (z.B. $SoC(t+H)=0,5$). Die Speicherkapazität wird erneut überdimensioniert und die tatsächlich erforderliche Speicherkapazität nach der Simulation aus verwendeten Speicherbereichen ermittelt.

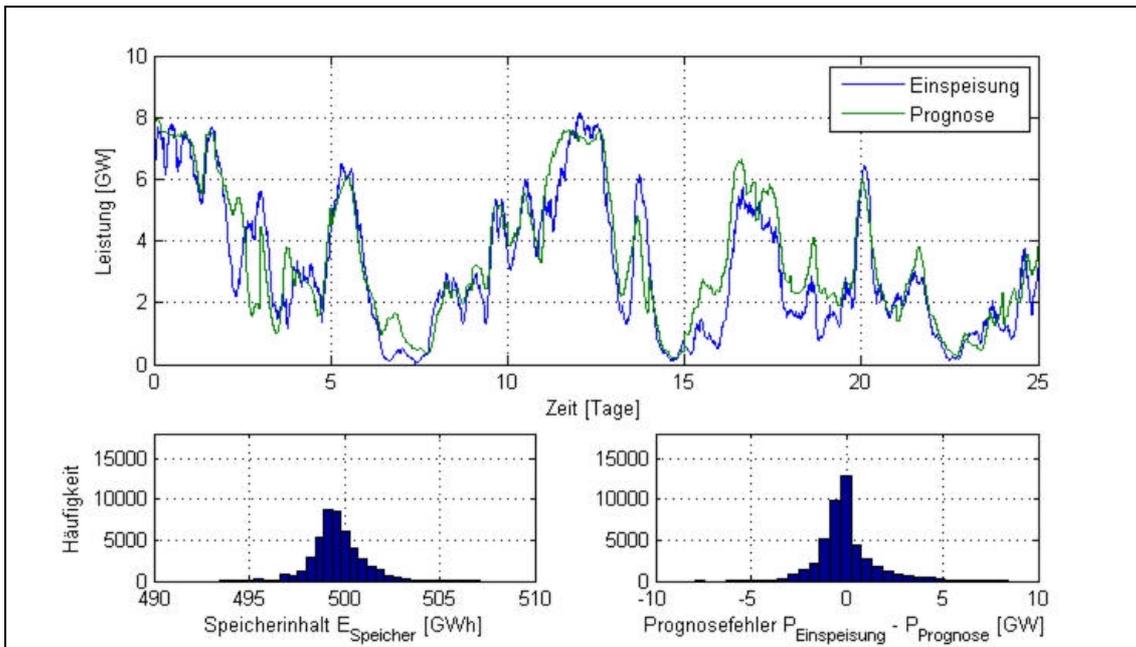


Abbildung 6: Windenergie (oben), Histogramme: Speicherfüllstände & Prognosefehler (unten)

Als verwendeten Speicherbereich wird das Volumen zwischen maximalem und minimalem Speicherfüllstand betrachtet. Um den Einfluss der Prognosegüte zu veranschaulichen wird ein Faktor eingeführt, welcher die Annäherung an den realen Verlauf kennzeichnet ($c = 0$ für ideale Prognose, $c = 1$ für reale Prognose). Abbildung 7 links zeigt die Kapazitäten bei verschiedenen Prognosegüten. Das Histogramm der Speicherfüllstände in Abbildung 6 unten links zeigt, dass der hier definierte Bereich der Speicherkapazität nur selten genutzt wird. Es handelt sich daher sicherlich nicht um die optimale Dimensionierung. Kriterien können für die Speicherdimensionierung entwickelt werden, indem die Regelernergie (Abbildung 7 rechts) zur Fahrpläneinhaltung sowie die Speicherleistung und -kapazität bewertet werden. Die verwendete Betriebsführung enthält keine variablen Parameter, da sie sehr trivial gewählt wurde. Alternative Betriebsführungsstrategien lassen sicherlich zu optimierende Parameter offen.

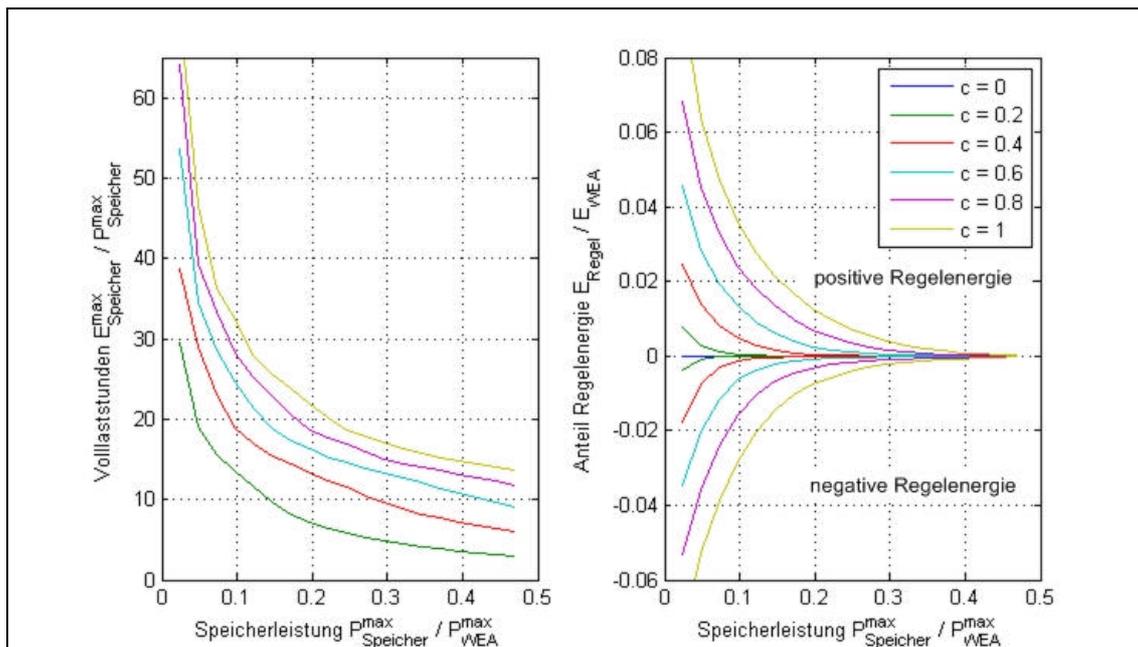


Abbildung 7: Einfluss der Prognoseunsicherheit auf die Speicherkapazität und Regelernergie

3 Verallgemeinerte Beschreibung des Dimensionierungsproblems

Aus den Anwendungsfällen und Beispielen des vorangegangenen Abschnittes wurde deutlich, dass die Dimensionierung elektrischer Energiespeicher von unterschiedlichen Rahmenbedingungen, wie z.B. von der Güte der Windenergieprognose oder von der zu realisierenden Betriebsführung, abhängig ist. Für die Speicherdimensionierung ergibt sich ein Prozess, den Abbildung 8 veranschaulicht. Zunächst wird der Zweck des Speichereinsatzes ermittelt, welcher aus ökonomischen oder technischen Zielen des Anwendungsfalles hervorgeht. Durch mathematische Formulierungen (wie Kriterien oder Zielfunktionen) ist es möglich, das gewählte Speichersystem in konkreten Szenarien, bestehend aus Speicher und Betriebsführung, zu bewerten. Die Modellierung des Speichers kann unabhängig vom Anwendungsfall über einen verallgemeinerten oder technologie-spezifischen Ansatz erfolgen. Die Ableitung von Betriebsführungsstrategien, die Auswahl der Simulationsdaten sowie die Kriterien zur Bewertung des Speichereinsatzes werden hingegen vom Anwendungsfall geprägt.

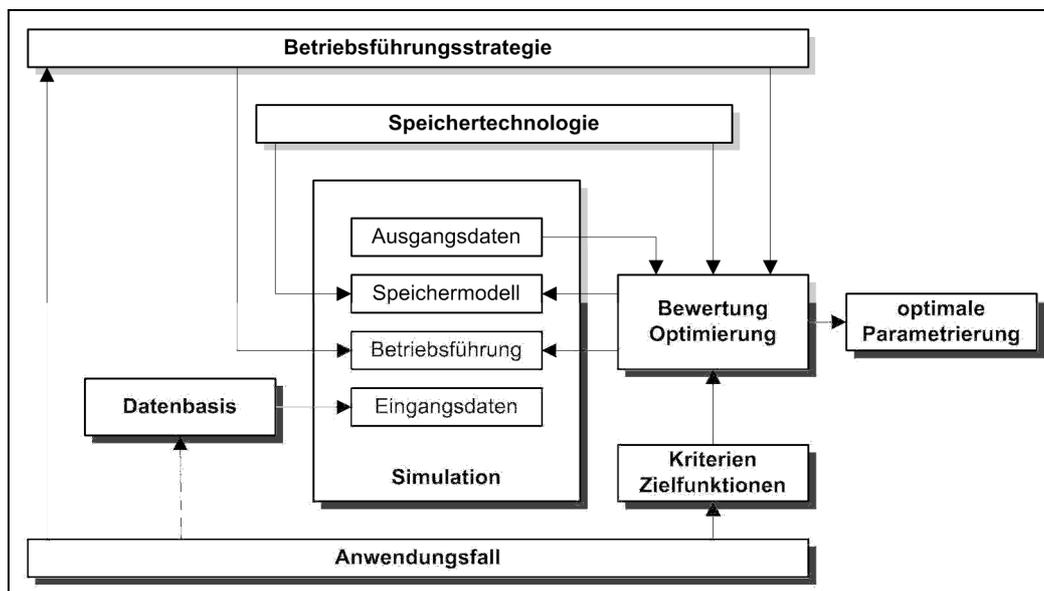


Abbildung 8: Komponenten der Speicherdimensionierung

Änderungen von Parametern der Betriebsführung und des Speichers können zum Erreichen des Ziels beitragen, so dass mittels Bewertung des Speichersystems im Szenario eine Optimierung der Parameter erfolgen kann. Unter der Dimensionierung des Speichers wird im Wesentlichen die Festlegung von Leistung und Energie verstanden. Da es sich hierbei um Parameter des Speichers handelt, umfasst die angesprochene Optimierung auch die Speicherdimensionierung. Viele Anwendungsfälle setzen voraus, dass unterschiedliche Signalverläufe vorhersagbar sind. Parameter des Speichersystems können daher von der Prognosegüte und der Prognosefehlerverteilung einzelner Signale abhängig sein. Es ist somit erforderlich, nicht nur Untersuchungen mit vorliegenden Daten durchzuführen, sondern auch Folgen möglicher stochastischer Einflüsse zu analysieren, beispielsweise mittels statistischer Simulationen.

Für konkrete Szenarien kann die Prognosequalität erheblichen Einfluss auf die Varianz der Systemparameter haben und somit entscheidend für den zu verwendenden Speichertyp sein. Zu den relevanten Kriterien der Speichertechnologieauswahl zählen Leistung und Energieinhalt, so dass die in Frage kommenden Technologien von den konkreten Szenarien abhängig sind. Bei der Dimensionierung für Anwendungsfälle mit stochastischen Signalkomponenten sollte somit auch eine Analyse der Robustheit erfolgen.

4 Ausblick: Generierung synthetischer Zeitreihen

Grundlage bisheriger Betrachtungen ist die Verfügbarkeit der notwendigen Datenbasis. Eine Entwicklung des Speichersystems erfordert entsprechend gemessene Zeitreihen, welche für konkrete Szenarien nur teilweise oder gar nicht vorhanden sind. Zudem ist für eine robuste Speicherdimensionierung eine große Datenbasis von Zeitreihen mit entsprechenden stochastischen Komponenten wünschenswert, welche jedoch in der Regel nicht gegeben sind. Um von exemplarischen Anwendungsbeispielen auf verallgemeinerte Erkenntnisse schließen zu können, ist eine ggf. künstliche Generierung einer solchen umfassenden Datenbasis notwendig – insbesondere, um die Fragestellungen bezüglich Betriebsführung und Dimensionierung besser beantworten zu können.

Im Rahmen weiterer Arbeiten ist die Erzeugung von Zeitreihen durch Signalgeneratoren angedacht, um die erforderlichen Daten bereit zu stellen. Wie allgemein bekannt, stehen zur Modellierung von Zeitreihen deterministische und stochastische Signalmodelle zur Verfügung, beschrieben in [8], [12] und [13]. Deterministische Signalanteile können durch analytische Zusammenhänge über die Zeit vollständig beschrieben werden (vgl. [8]). Stochastische Signale hingegen basieren auf Theorien der Statistik und bilden Zufallsprozesse ab, so dass zur Generierung Zufallsgrößen erzeugt werden müssen.

Es sind zunächst quellspezifische Signalmodellstrukturen zu identifizieren, welche durch die Vorgabe von Parametersätzen die gewünschten realitätsnahen Zeitreihen generieren. Die Signalmodellstruktur ist abhängig von dem zu generierenden Zeitreihentyp, so dass keine verallgemeinerte Darstellung der Signalgeneratoren erfolgt. Einige Signaltypen werden von exogenen Größen beeinflusst, beispielsweise sind Zeitreihen der Windenergie (Abbildung 9) unter anderem von Windrichtung, Windgeschwindigkeit und Anlagenverfügbarkeit abhängig. Die einzubeziehenden exogenen Größen selbst können wiederum aus anderen Signalen erzeugt werden, so dass eine gestaffelte Zeitreihenmodellierung erfolgt. In den unterschiedlichen Instanzen werden Zeitreihen entweder erzeugt oder über parametrierbare Systemmodelle der Anlagen miteinander verknüpft. Stochastische Effekte werden durch die Angabe statistischer Kenngrößen, wie beispielsweise Verteilungen, und entsprechenden Signalmodellen berücksichtigt. Es ist dann möglich, die betrachtete Zeitreihe mehrmals unter Berücksichtigung der entstehenden Zufallsprozesse zu erzeugen, so dass statistische Simulationen, wie die Monte-Carlo-Simulation durchgeführt werden können

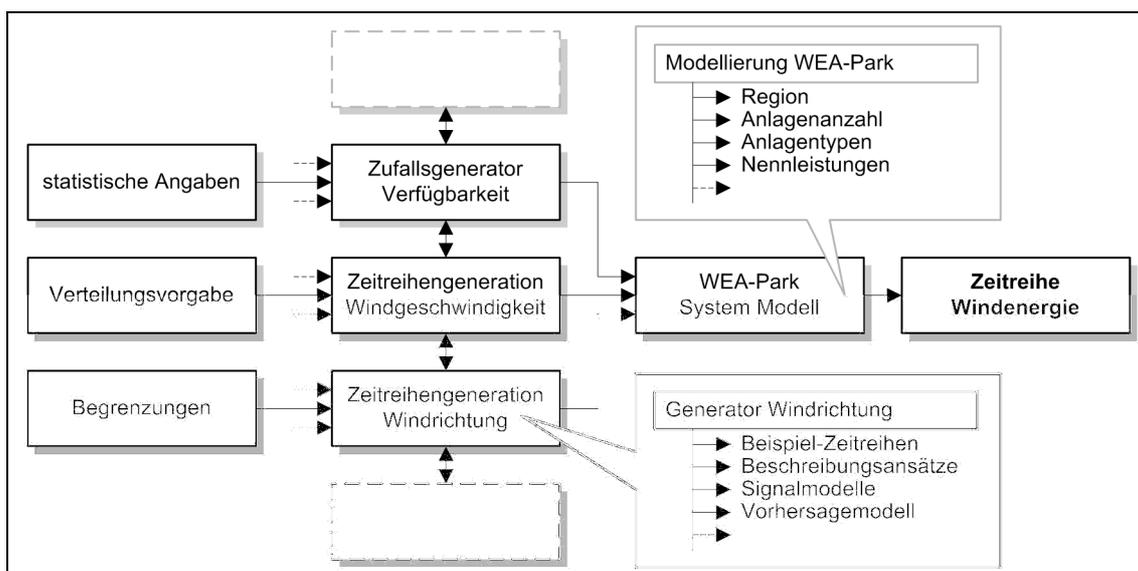


Abbildung 9: Signalmodellstruktur des Signalgenerators für Windenergiezeitreihen

Neben der Erzeugung von Signalverläufen ist es mit entsprechenden Modellen denkbar auch Prognosen nachzubilden, etwa indem die betrachtete Zeitreihe durch die entsprechende Prognosefehlercharakteristik verfälscht wird. Exemplarisch werden die in Abschnitt 2.3 verwendeten Daten betrachtet. Der Signalgenerator erzeugt unter Berücksichtigung ermittelter, grundsätzlicher Eigenschaften verschiedene Varianten des Prognosesignals. Zur Robustheitsanalyse entsprechender Anwendungsfälle ist es möglich, erforderliche Zeitreihen von Windenergieprognosen flexibel zu erzeugen, dargestellt in Abbildung 10. Nach dem in Abschnitt 3 vorgestellten Schema sind für jede generierte Zeitreihe Simulationen zu betreiben, so dass die Ausgangsdaten mit statistischen Methoden analysiert werden.

Die Anwendungsfälle erfordern unterschiedlichste Zeitreihen, wie Lastgänge, Einspeiseverläufe oder der Verlauf exogener Größen. Ungleiche Eigenschaften und Merkmale einzelner Zeitreihentypen können unterschiedliche Modellansätze erfordern, so dass die nötigen Parametersätze untereinander abweichen werden. Es bietet sich daher an, mehrere quellenspezifische Signalgeneratoren zu entwickeln, welche nach Eigenschaften in eine strukturierte Bibliothek gruppiert werden, um flexible Szenarien aufzubauen.

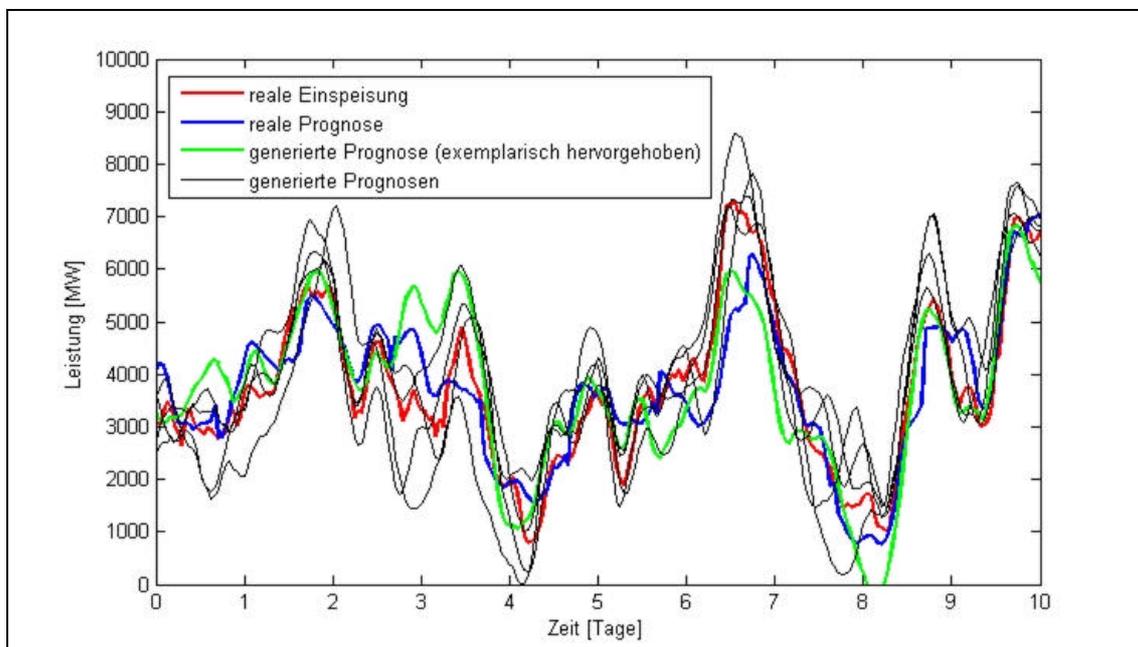


Abbildung 10: Varianten des Prognosesignals

5 Zusammenfassung

Eine Methode zum Ausgleichen der Energiebilanzen in einem Elektroenergiesystem ist die Pufferung über speichernde Elemente. Unterschiedliche Technologien elektrischer Energiespeicher stehen zur Verfügung, wobei die Frage der Netzintegration und der Betriebsführung für interessante Anwendungsfälle noch ungeklärt ist. Die Speicherdimensionierung ist Teil einer Optimierungsaufgabe, welche mit der Anpassung von Betriebsführungsparametern gekoppelt ist. In mehreren Anwendungsfällen ist die Betrachtung fehlerbehafteter Prognosen erforderlich. Das Ausmaß der auftretenden Unsicherheiten beeinflusst die Speicherdimensionierung. Statistische Simulationen und Datenanalysen sind daher zur Bewertung von Simulationen erforderlich. Eine mangelnde Datenbasis erschwert die Dimensionierung des Speichers für einen konkreten Anwendungsfall, zudem ist die erforderliche Datenbasis für eine robuste Speicherdimensionierung im Allgemeinen nicht vorhanden.

Zur Erstellung erforderlicher Zeitreihen und zur Modellierung von Unsicherheiten in Zeitreihen werden diese von einem Signalgenerator erzeugt, so dass auch bei schwacher Ausgangsdatenbasis eine Dimensionierung erfolgen kann. Die Generierung synthetischer Zeitreihen basiert auf Signalmodellen und Zufallsprozessen, welche aus bekannten Methoden der Datenanalyse und der Signaltheorie hervorgehen. Verschiedene Methoden der *computational intelligence* finden bereits in Prognoseverfahren und Systemidentifikationen Anwendung, beispielsweise der Einsatz künstlich neuronaler Netze als nichtlineares Signalmodell (vgl. [8]) oder die Mustererkennung unter Verwendung der Fuzzy-Technologie (vgl. [14]). Da die Erzeugung synthetischer Zeitreihen auf ähnlichen Ansätzen basiert, ist die Anwendung erwähnter Methoden zu erwarten.

Literatur

- [1] Schwab, A.: *Elektroenergiesysteme*. Springer Verlag, Berlin, Heidelberg; 2006
- [2] Garcke, J.: *Elektrochemische Energiespeicher: Stand, Probleme, Perspektiven*. Energiewirtschaftliche Tagesfragen (56), Heft 10, S. 61-66; 2006
- [3] Bodach, M.: *Energiespeicher im Niederspannungsnetz zur Integration dezentraler, fluktuierender Energiequellen*. Dissertation, TU Chemnitz; 2006
- [4] Sauer, D.-U.: *Optionen zur Speicherung elektrischer Energie in Energieversorgungssystemen mit regenerativer Stromerzeugung*. Solarzeitalter (18), Heft 4, S. 12-34; 2006
- [5] Quaschnig, V.: *Regenerative Energiesysteme*. Hanser Verlag, 4. Auflage, München, Wien; 2006
- [6] Hasche, B.: *Analysen von Prognosen der Windgeschwindigkeit und Windstrom-einspeisung*. Teilbericht im Projekt NetMod, IER Universität Stuttgart; 2007
- [7] Westermann, D.; Nicolai, S.; Bretschneider, P.: *Energy Management for Distribution Networks with Storage Systems - A Hierarchical Approach*. IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, S. 1-6; 2008
- [8] Bretschneider, P.: *Ein Beitrag zur Vorhersage musterbasierter nichtlinearer stochastischer Signale*. Dissertation, TU Ilmenau, 2002
- [9] Heinemann, D.: *Energiemeteorologie: Ein Überblick*. FVS Workshop Energiemeteorologie, Berlin, S. 16-21; 2006
- [10] Rohrig, K.: *Das Windpower Management System – Integration großer Windleistungen in die elektrische Energieversorgung*. 8. Kassler Symposium Energiesystemtechnik, Kassel, S. 141-149; 2003
- [11] N.N.: *Netzkennzahlen, Vattenfall*. Download August 2008
Link:http://www.vattenfall.de/www/trm_de/trm_de/178444netzk/index.jsp
- [12] Schlittgen, R.; Streitberg, B. H. J.: *Zeitreihenanalyse*. Oldenbourg Verlag, 9. Auflage, München, Wien; 2001
- [13] Wernstedt, J.: *Experimentelle Prozessanalyse*. Verlag Technik, Berlin; 1983
- [14] Bretschneider, P.; Rauschenbach, T.: *Vorhersage musterbasierter stochastischer Signale unter Verwendung von Fuzzy-Technologien*. Proceedings 11. Workshop Fuzzy Control des GMA-FA 5.22., S. 73-74, Dortmund; 2001

Electronic-Banking Kernel mit neuronalen Validierungsmechanismen auf Basis des FinTS 4.0 Standards

S. Schäfer, S. Turck, M. Schneider, W. Roth

Institut für Computer Science, Vision
and Computational Intelligence

Frauenstuhlweg 31

58644 Iserlohn

Tel. (02371) 566-153

Fax (02371) 566-377

E-Mail: {Schaefer.S, Turck, MSchneider, Roth}@fh-swf.de

1 Zusammenfassung

Im Rahmen des Projektes MoneyPenny [1] wurde an der Fachhochschule Südwestfalen ein Electronic-Banking Kernel zur Abwicklung bankfachlicher Geschäftsvorfälle entwickelt, in welchen als innovatives Sicherheitsfeature eine optionale clientseitige Validierung der Transaktionsdaten mittels künstlicher neuronaler Netze (KNN) integriert wurde. Dieser Kernel, nachfolgend auch als FinTS-Kernel bezeichnet, stellt die Schnittstelle zwischen einer Homebanking-Applikation und dem Kreditinstituts-server dar. Die aktuelle Version des FinTS-Kernels unterstützt die vom Zentralen Kreditausschuss (ZKA) veröffentlichten Banking-Standards HBCI (Home Banking Computer Interface) bzw. FinTS (Financial Transaction Services) [2], einschließlich der aktuellen Version 4.0.

2 Homebanking

Als Homebanking werden generell Verfahren bezeichnet, die es ermöglichen, bankfachliche Geschäftsvorfälle von zu Hause aus durchführen zu können, anstatt eine Filiale des Kreditinstitutes aufsuchen zu müssen. Bis zum heutigen Tage wurde bereits eine Vielzahl von verschiedenen Homebanking-Verfahren entwickelt, wie zum Beispiel das Postbanking, das Telefonbanking oder das Online-Banking. Diese zeichnen sich durch eine entscheidende Gemeinsamkeit aus, da alle Homebanking-Verfahren eine schwierige Gratwanderung zwischen der Benutzerfreundlichkeit und der Sicherheit vollbringen müssen. Die Kreditinstitute müssen ihren Kunden entsprechende Verfahren anbieten, welche sowohl die Vertraulichkeit als auch die Integrität und die Authentizität der Daten ausreichend gewährleisten. Es muss also im Rahmen dieser drei Sicherheitsaspekte sichergestellt werden, dass die Informationen ausreichend gegen das Ausspähen und Manipulieren durch Dritte geschützt sind und die Identität des Urhebers eindeutig bestimmbar ist. Um diese sicherheitsrelevanten Eigenschaften der Homebanking-Verfahren zu wahren, ist ein hohes Maß an Komplexität erforderlich, was gleichzeitig mit Einschnitten in der Anwenderfreundlichkeit verbunden ist.

2.1 Homebanking

Die vom Zentralen Kreditausschuss herausgegebene FinTS-Spezifikation ist eine Weiterentwicklung des HBCI-Standards und definiert eine automatisiert nutzbare, multibankfähige Online-Banking-Schnittstelle [2]. Der Standard soll eine bundesweite Vereinheitlichung der Schnittstelle zwischen Kunden und Kreditinstituten schaffen und versteht sich als Baukastensystem (siehe Abb. 1), bei dem die Legitimationsverfahren [3, 4], Geschäftsvorfälle [5] und Finanzdatenformate [6] jeweils einzeln und unabhängig vom zugrunde liegenden Übertragungsprotokoll beschrieben werden. Darüber hinaus werden in der Schnittstellenspezifikation allgemeine Festlegungen zu Kommunikationsszenarien und -verfahren [7] getroffen. Mit dem Wechsel auf die aktuelle Version 4.0 geht der FinTS-Standard neue Wege. Neben weiteren Rollenmodellen und der Einführung eines asynchronen Kommunikationsverfahrens steht vor allem der Wechsel der Nachrichtensyntax zu dem vom W3-Konsortium spezifizierten XML-Standard [8] im Mittelpunkt. Seitens der Kreditinstitute werden heutzutage überwiegend die Standards HBCI 2.2 und FinTS 3.0 angeboten, während die aktuelle Version 4.0 noch nicht offiziell bereitgestellt wird.



Abbildung 1: Modularer Aufbau der FinTS-Spezifikation

3 Neuronale Validierung

Im Rahmen der Validierung wird geprüft, ob eine durchzuführende Transaktion mit dem gewöhnlichen Verhalten des Benutzers übereinstimmt. Zu diesem Zweck werden in dem FinTS-Kernel künstliche neuronale Netze eingesetzt, welche das Nutzerverhalten abbilden. Da der aktuelle Entwicklungsstand noch keine individualisierten neuronalen Netze bietet, wird dem Benutzer ein vorgefertigtes Profil zugewiesen,

welches sein Nutzungsverhalten am besten approximiert. Zum jetzigen Zeitpunkt werden fünf verschiedene Profile angeboten, welche den Bereich vom Normalnutzer bis hin zu international agierenden Firmenkunden abdecken. Zu den validierungsrelevanten Parametern der Transaktionen zählen unter anderem der Betrag und das Ziel sowie die Anzahl der bisher getätigten Transaktionen innerhalb eines bestimmten Zeitraumes. Die Ausgabe der neuronalen Validierung ist als relative Wahrscheinlichkeit zu interpretieren, mit welcher die Transaktion zu dem Nutzerverhalten passt.

3.1 Training

Da es sich bei den bankfachlichen Transaktionsdaten um extrem sensible Daten handelt, konnten keine realen Datensätze für das Training herangezogen werden. Aus diesem Grund wurden für jedes Profil hundert statistisch gleichverteilte Datensätze mit Hilfe einer Zufallsfunktion generiert und folgendermaßen aufgeteilt:

Trainingsdatensätze	70
Validierungsdatensätze	20
Testdatensätze	10
<hr/>	
Gesamt	100

Das Erstellen und Trainieren der Netze wurde mit Hilfe des Programms JavaNNS [9] durchgeführt. Als Trainingsverfahren kam die Implementierung des Backpropagation-Algorithmus [10] zum Einsatz. Für jedes Profil wurden die Netze mit mehreren Initialisierungswerten trainiert, was zu unterschiedlich erfolgreichen Trainingsresultaten führte (siehe Abb. 2).

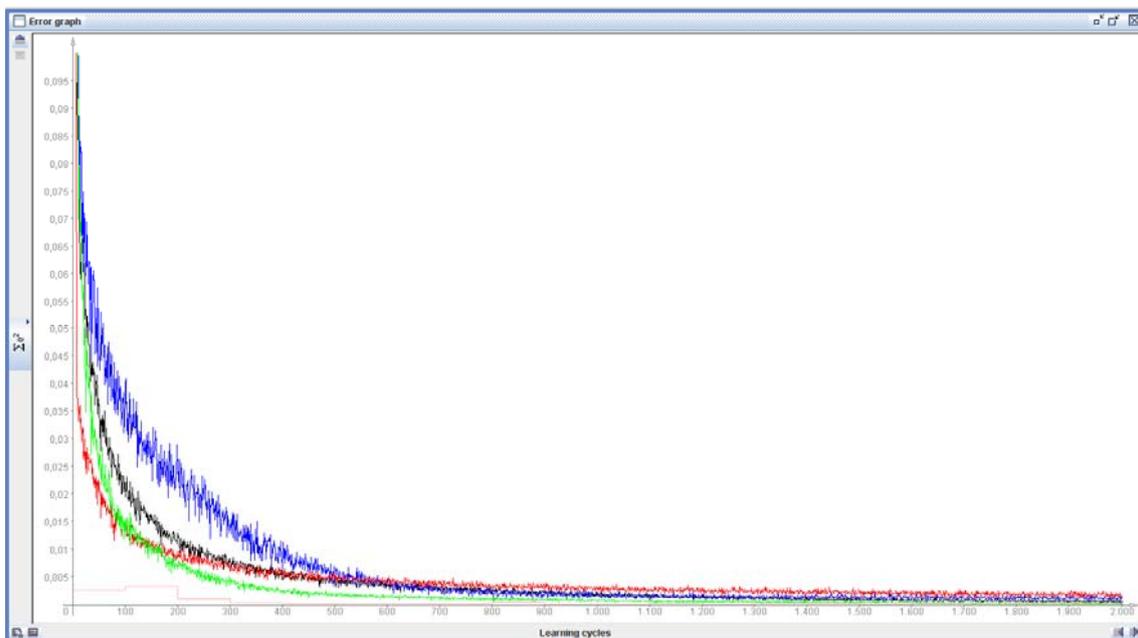


Abbildung 2: Error Graph aus JavaNNS für vier Trainingszyklen mit unterschiedlichen Initialisierungswerten

3.2 Ergebnis

Die zuvor generierten Testdatensätze wurden eingesetzt um den Trainingserfolg und somit die Qualität der künstlichen neuronalen Netze zu kontrollieren. Die Testdaten wurden durch die Netze propagiert und die jeweilige Netzausgabe mit der Soll-Ausgabe verglichen. Anhand dieser Abweichung wurde für jedes Profil das Netz mit dem geringsten mittleren Fehler aus den unterschiedlichen Trainingszyklen ausgewählt. Die größte aufgetretene Abweichung beträgt 8,29‰, der durchschnittliche Testfehler liegt bei 1,5‰. Das nachfolgende Diagramm zeigt beispielhaft den Fehler für die Testdaten des Profils für kleine Unternehmen.

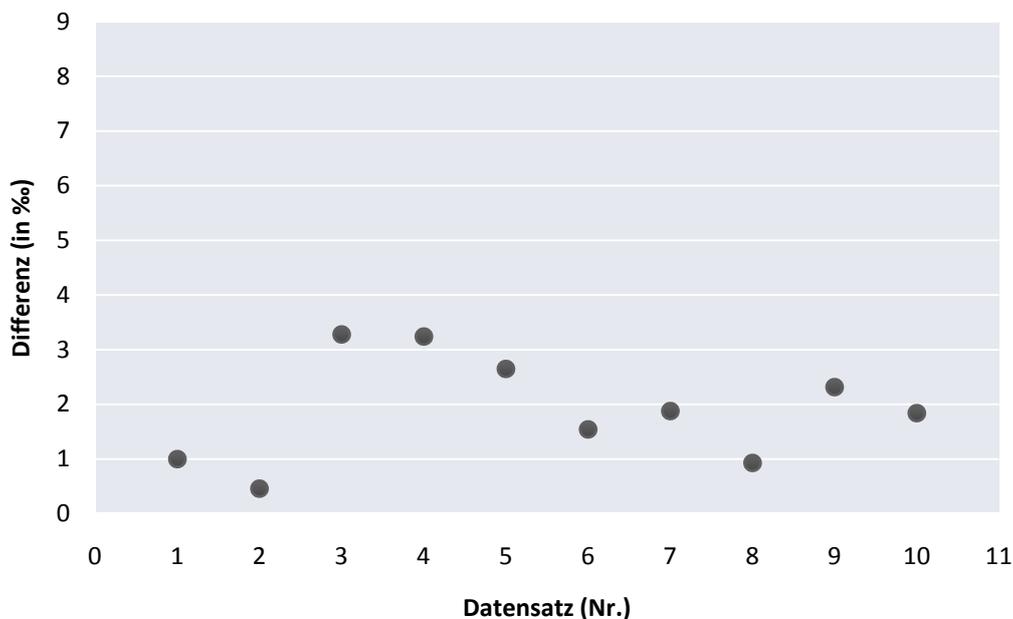


Abbildung 3: Testfehler des Profils Small-Business

3.3 Integration und Einsatz

Die trainierten Netze wurden mit dem JavaNNS-Kommandozeilentool `sns2c` in einen C-kompatiblen Quellcode konvertiert und direkt in den Kernel integriert. Da die Netzarchitektur für jedes Profil identisch ist, wurden die trainierten Gewichtungen und Bias-Werte in eine Konfigurationsdatei ausgelagert um eine dynamische Initialisierung des Netzes zur Laufzeit und eine einfache Möglichkeit zur Erweiterung um zusätzliche Profile zu schaffen.

Bevor die eigentliche Überweisung innerhalb des Kernels eingeleitet wird, lassen sich die Transaktionsdaten optional mittels des neuronalen Netzes validieren. Zu diesem Zweck muss das Netz mit dem gewünschten Profil und einer Warnschwelle initialisiert werden. Anhand dieses Grenzwertes lässt sich festlegen, ab welcher Netzausgabe eine Transaktion als gültig anzusehen ist. Ein negatives Validierungsergebnis durch das künstliche neuronale Netz führt in keinem Fall zu einer kernelseitigen Ablehnung der Transaktion. Das weitere Vorgehen ist in diesem Fall dem Entwickler der Homebanking-Applikation überlassen.

4 Ausblick

Im weiteren Verlauf des Projekts steht neben dem Ausbau des FinTS-Kernels um weitere bankfachliche Geschäftsvorfälle vor allem die Erweiterung der neuronalen Validierung im Vordergrund. Diese sieht sowohl die Einbeziehung weiterer Parameter zur Auswertung vor, als auch die Entwicklung eines Online-Training-Verfahrens um die Validierung für jeden Benutzer individualisieren zu können. Auf Basis des Kernels soll weiterhin ein FinTS-Server erstellt werden, welcher die kreditinstitutsseitige Verarbeitung von FinTS-Nachrichten umsetzt und eine serverseitige Validierung der Transaktionsdaten bietet.

5 Literatur

- [1] Fachhochschule Südwestfalen (Hrsg.): Projekt MoneyPenny. Internet Publikation: <http://www3.fh-swf.de/fbin/roth/moneypenny.htm>. Stand: 22.02.2006.
- [2] Zentraler Kreditausschuss (Hrsg.): FinTS – Financial Transaction Services. Internet Publikation: <http://www.hbci-zka.de>. Stand: 26.07.2007.
- [3] Zentraler Kreditausschuss (Hrsg.): Financial Transaction Services (FinTS). Security. Sicherheitsverfahren HBCI. Internet Publikation, Schnittstellenspezifikation: http://www.hbci-zka.de/dokumente/spezifikation_deutsch/FinTS_4.0_Security_HBCI.pdf. Erstellt: 09.07.2004.
- [4] Zentraler Kreditausschuss (Hrsg.): Financial Transaction Services (FinTS). Security. Sicherheitsverfahren PIN/TAN. Internet Publikation, Schnittstellenspezifikation: http://www.hbci-zka.de/dokumente/spezifikation_deutsch/FinTS_4.0_Security_PINTAN.pdf. Erstellt: 09.07.2004.
- [5] Zentraler Kreditausschuss (Hrsg.): Financial Transaction Services (FinTS). Messages. Multibankfähige Geschäftsvorfälle. Internet Publikation, Schnittstellenspezifikation: http://www.hbci-zka.de/dokumente/spezifikation_deutsch/FinTS_4.0_Messages_Geschaeftsvorfaelle.pdf. Erstellt: 09.07.2004.
- [6] Zentraler Kreditausschuss (Hrsg.): Financial Transaction Services (FinTS). Messages. Belegungsrichtlinien für Finanzdatenformate der deutschen Kreditwirtschaft. Internet Publikation, Schnittstellenspezifikation: http://www.hbci-zka.de/dokumente/spezifikation_deutsch/FinTS_4.0_Messages_Finanzdatenformate.pdf. Erstellt: 09.07.2004.
- [7] Zentraler Kreditausschuss (Hrsg.): Financial Transaction Services (FinTS). Formals. Grundsätzliche Festlegungen. Internet Publikation, Schnittstellenspezifikation: http://www.hbci-zka.de/dokumente/spezifikation_deutsch/FinTS_4.0_Formals.pdf. Erstellt: 09.07.2004.
- [8] World Wide Web Consortium (Hrsg.): Extensible Markup Language (XML) 1.0 (Fourth Edition). Internet Publikation: <http://www.w3.org/TR/2006/REC-xml-20060816>. Erstellt: 16.08.2006. Geändert: 29.09.2006.

- [9] Zell, Andreas: JavaNNS: Java Neural Network Simulator. Internet Publikation: <http://www-ra.informatik.uni-tuebingen.de/software/JavaNNS>. Geändert: 05.10.2007.
- [10] Zell, Andreas: Simulation neuronaler Netze. 4., unveränderte Auflage. München: Oldenbourg Wissenschaftsverlag GmbH 2003.

Client/Server-Architektur eines verteilten ANN mit XML-basierter Ablage

Rainer Bartz

Fachhochschule Köln; Fak. IME
Betzdorfer Str. 2
Tel. (0221) 8275-2473
Fax (0221) 8275-7-2473
E-Mail: rainer.bartz@fh-koeln.de

1 Abstract

Dieser Beitrag beschreibt die Arbeiten zu ANNCOLOGNE, einem Client/Server basierten Simulator für neuronale Netze, sowie zu der neu entworfenen XML-Sprache ANNSL.

Für manche Aufgabenstellungen insbesondere im Rahmen des Trainings eines ANN (artificial neural network) ist der Einsatz der bisher verfügbaren monolithischen ANN-Implementierungen nicht optimal. Wenn es z.B. darum geht, das Ergebnis rechenzeitintensiver Aufgaben für mehrere Varianten einer initialen Konfiguration zu bestimmen, wären zentral gesteuerte aber parallel ablaufende Prozesse wünschenswert. Aus diesem Grund wurde ANNCOLOGNE als Client/Server System konzipiert und implementiert. Wesentliche Teilschritte sowie die Architektur von ANNCOLOGNE werden in diesem Beitrag erläutert.

Im Zuge dieser Arbeiten zeigte sich, dass eine geeignete moderne und leistungsfähige Ablage für Daten und Konfigurationen im ANN-Umfeld nicht verfügbar war. Daher wurde mit ANNSL eine XML-Sprache definiert, die alle diesbezüglich wesentlichen Informationen darstellen kann. Ihrer Struktur widmet sich der zweite Teil dieses Beitrags.

2 Client/Server Architektur des NN-Simulators ANNCOLOGNE

2.1 Motivation und Übersicht

ANNs (artificial neural networks) sind derzeit vornehmlich als monolithische Applikationen implementiert; sie belegen die Ressourcen des Systems auf dem sie gestartet wurden und können nicht miteinander kommunizieren.

Gerade in der Trainingsphase ergibt sich häufig die Notwendigkeit, eine optimale Konfiguration zu finden, indem derselbe Trainingssatz für unterschiedliche ANN-Start-Konfigurationen eingesetzt wird; dies führt für jede Start-Konfiguration zu einem trainierten ANN. Die sich ergebenden unterschiedlichen trainierten ANNs können dann verglichen und das Beste kann für den produktiven Einsatz ausgewählt werden.

Allerdings erfordert ein solches Vorgehen entweder, dass die Trainingsläufe sequentiell nacheinander gestartet werden (was einen zeitlich vervielfachten Aufwand verursacht), oder dass sie parallel auf demselben System laufen (was die Performance der einzelnen Läufe herabsetzt), oder dass die monolithischen Applikationen unabhängig voneinander jeweils auf unterschiedlichen Systemen gestartet und bedient werden; in allen Fällen ergibt sich dann noch ein zusätzlicher Aufwand für die Zusammenführung und

Auswertung der Ergebnisse. Eine Client/Server-Architektur für ein ANN kann sich hier als vorteilhaft erweisen und die Basis dafür bilden, dass Ergebnisse einfach zueinander in Beziehung gesetzt und viele Teilschritte automatisiert abgewickelt werden können.

Eine dafür geeignete System-Konfiguration ist in Bild 1 skizziert; sie stellt das Ziel der derzeit laufenden Entwicklungen dar.

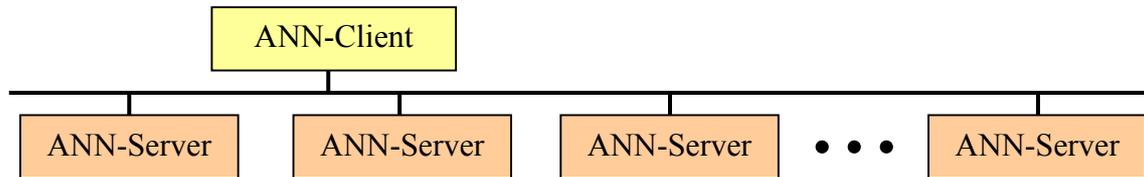


Bild 1: Angestrebte C/S-Architektur für den parallele Einsatz von ANNs

Dieser erste Teil des Beitrags stellt die Architektur des ANN ANNCOLOGNE dar und erläutert die wesentlichen Entscheidungen der bisherigen Entwicklungsschritte. Die Persistenz von Daten und Konfigurationen erfolgt auf Client-Seite auf Basis der im zweiten Teil beschriebenen ANNSL (artificial neural network specification language). Eine ausgewogene Schnittstelle zwischen Client und Server ist der Schlüssel zu vertretbarer Performance; hier ist insbesondere das Verwalten der Trainingsdaten, das Logging und die Aktualisierung der GUI von Bedeutung.

2.2 Erster Ansatz: Untersuchung von JavaNNS auf Client/Server Eignung

Wenn frei verfügbare ANN-Implementierungen eingesetzt werden, ist die Wahrscheinlichkeit groß, das seit Jahren existierende und funktional reiche SNNS [1] oder auch die später entstandene Java-Version JavaNNS anzutreffen. SNNS beinhaltet eine C-basierte Programmierschnittstelle, über die ein großer Teil der SNNS Funktionalität erreichbar ist. Mit JavaNNS wird eine Java-Klasse bereitgestellt, die selber über das Java Native Interface auf die SNNS-Programmierschnittstelle zugreift und deren Funktionalität einer Java-Applikation verfügbar macht.

Da die programmiertechnische Grundausbildung an der FH Köln auf Java aufbaut, und somit Studierende eher Java als C++ Erfahrungen mitbringen, bot es sich an, JavaNNS und die darin befindliche Klasse zum SNNS-Zugriff zu untersuchen. Dies erfolgte im Rahmen einer studentischen Arbeit [2]; die Zielsetzung war, festzustellen inwieweit und unter welchen Randbedingungen die mit JavaNNS verfügbaren Schnittstellen geeignet sind, eine verteilte ANN-Applikation zu erstellen, bei der mehrere SNNS-basierte Server über ein Netzwerk von einer Client-Applikation gesteuert werden.

Als Ergebnis lieferte diese Arbeit eine Client/Server-Applikation, die mittels RMI-Kommunikation in der Lage ist, einen entfernten SNNS-Server zu konfigurieren, und die Ergebnisse am lokalen Client zu visualisieren. Obwohl damit gezeigt war, dass eine Client/Server-Verteilung auf Basis von JavaNNS prinzipiell möglich ist, stellten sich einige Hindernisse im Zusammenhang mit dem erforderlichen Datenmanagement heraus.

Die angestrebte C/S Architektur mit mehreren Servern, die einem Client zuarbeiten, erfordert ein Datenmanagement auf Client-Seite. JavaNNS überlässt den Zugriff auf Konfigurations- und Daten-Dateien jedoch dem SNNS-Kernel; es besitzt keine eigenen Lese- und Schreibfunktionen. Damit müssen derartige Dateien auch in einer C/S-Architektur im Zugriffsbereich des Rechners stehen, auf dem der SNNS-Kernel läuft.

Ein Client-seitiges Datenmanagement mit einem verteilten JavaNNS lässt sich zwar auf mehrere Arten lösen, die jedoch jeweils deutliche Nachteile zeigen:

- Ein Verzeichnis des Client-Rechners, in dem die Dateien verwaltet werden, kann freigegeben und als virtuelles Laufwerk bei allen Server-Rechnern eingerichtet werden. Dies erfordert manuelle Konfigurationsarbeit an jedem beteiligten System und entsprechende Zugriffsberechtigungen auch ggf. über Firewall-Grenzen hinweg.
- Ein Verzeichnis auf jedem der Server-Rechner kann freigegeben und als virtuelles Laufwerk bei dem Client-Rechnern eingerichtet werden; dieser kopiert dann die jeweils erforderlichen Daten auf die Server-Rechner. Auch dies erfordert manuelle Konfigurationsarbeit und entsprechende Zugriffsberechtigungen.
- Die Applikationen auf Client und Server werden so erweitert, dass sie miteinander über entsprechende Protokolle einen Filetransfer vornehmen. Dies führt (insbesondere für die Behandlung diverser Fehlerfälle) zu einem nicht unerheblichen Programmieraufwand.
- Auf Client-Seite werden entsprechende Parser und Writer für die SNNS Fileformate implementiert. Die Client-Applikation selber liest und interpretiert die SNNS-Dateien und benutzt lediglich die Programmierschnittstelle um mit dem SNNS-Kernel zu kommunizieren. Dies wäre ein noch erheblicherer Programmieraufwand, da die SNNS Formate nicht einfach strukturiert sind.

Hinzu kommt bei den Lösungen, die mit Dateien auf Server-Seite arbeiten, dass Server-Applikationen oder Server-User die Dateien löschen, verändern, oder auch einfach nur einsehen können, was ggf. unerwünscht ist.

Weiterhin zeigte sich, dass die SNNS Datenablage nicht alle in Zukunft absehbaren interessanten Informationen aufnehmen kann, so dass parallel die Arbeiten für eine neue XML-basierte ANN-Persistenz begannen. Vor diesem Hintergrund machte eine Investition in die Unterstützung von SNNS-Formaten wenig Sinn.

Daher wurden die Arbeiten an einer C/S-fähigen Variante von JavaNNS zunächst eingestellt.

2.3 Vorüberlegungen zu ANNCOLOGNE

Bereits zu einem recht frühen Zeitpunkt begannen erste Ideen zu einem Design eines eigenen NN-Simulators. Der Grundstein der Implementierung wurde im Rahmen einer Projektarbeit eines Studententeams gelegt, das ein Neuron mit Backpropagation-Lernverfahren und einem geeigneten API sowie eine über das API zugreifende Testapplikation implementierte [3]. Aufbauend auf diesem Neuron-Server wurde durch ein zweites Studententeam ein neuronales Netz mit entsprechendem API und eine zugehörige ANN-Applikation implementiert [4]. Beide Implementierungsergebnisse besaßen bereits eine rudimentäre XML-Persistenz.

Bevor in einem nächsten Schritt eine Verteilung als C/S Architektur in Angriff genommen wurde, wurde in einer weiteren studentischen Arbeit [5] eine geeignete Zwischenstufe der o.g. ANN-Applikation sowohl über RMI als auch über CORBA in einen Client- und einen Server-Teil zerlegt. Beide Technologien zeigten sich handhabbar und recht performant, wobei RMI sowohl in Bezug auf den Implementierungsaufwand als auch auf die Performance leichte Vorteile aufwies (was aufgrund des nicht notwendigen Language-Mapping zu erwarten war).

Zur groben Quantifizierung dieser Vorteile wurde ein Test-Server mit den vier Grundrechenarten als minimalen API-Funktionen und ein Test-Client implementiert, der diese Funktionen in hoher Folge aufrief. Tests mit zwei PCs (2,0GHz Dual-Core,

2,8GHz, jeweils WindowsXP) über 10MBit/s bzw. 100MBit/s LAN ergaben einen Performance-Vorteil von 35-50% für RMI.

Da eine Sprachunabhängigkeit derzeit nicht erforderlich ist und ein Umstieg von RMI auf CORBA (bzw. bevorzugt eine Erweiterung auf CORBA) mit erträglichem Aufwand möglich ist, wurde für das entstehende ANNCOLOGNE als Kommunikationstechnologie RMI gewählt.

Mit dieser Entscheidung wurde im Folgenden für ANNCOLOGNE ein sinnvolles C/S-Interface definiert und die Implementierung in einen Client- und einen Serverteil aufgetrennt. Die im folgenden Kapitel beschriebene Persistenz ANNSL wurde als zwei unabhängige Repositories implementiert und auf Client-Seite in ANNCOLOGNE eingebunden. Damit liegt eine verteilte ANN-Implementierung vor, für die erste Performance-Tests erfolgreich abgeschlossen werden konnten.

2.4 Architektur von ANNCOLOGNE

Nach den im vorigen Abschnitt bereits erläuterten Entwicklungsschritten stellt Bild 2 die Kommunikationsaspekte der ANNCOLOGNE Architektur dar.

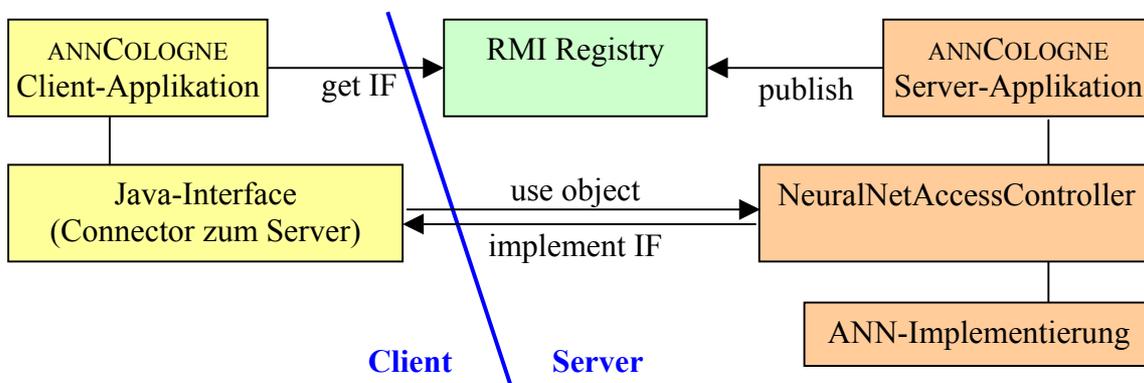


Bild 2: Kommunikationsaspekte von ANNCOLOGNE

Aus Implementierungssicht ist gegenüber einer monolithischen Applikation lediglich

- ein Java Interface zu definieren, das dann auf Server-Seite implementiert wird.
- Server-seitig die implementierende Klasse in einer RMI Registry zu veröffentlichen
- Client-seitig ein geeignetes Angebot in der Registry zu identifizieren und zu instanzieren; mit dieser Instanz können nun alle Methoden verwendet werden, die das Serverobjekt bereitstellt.

Das Interface zwischen Client und Server umfasst derzeit etwa 40 Methoden, wird aber entsprechend der Anforderungen zukünftig sicher noch erweitert.

Bild 3 gibt einen Einblick in die Architektur der ANNCOLOGNE-Serverseite.

Der Zugriff auf die Implementierungen wird sowohl beim Neuron als auch beim Neural Net über einen Access Controller gesteuert. Bei der Konfigurierung können Neurontyp, Art der Aktivierungs- und Output-Funktion, zu verwendender Learner usw. eingestellt werden. Auf eine einfache Erweiterbarkeit wurde besonderer Wert gelegt; dadurch können weitere Neurontypen, Learner, ... mit vertretbarem Aufwand ergänzt werden.

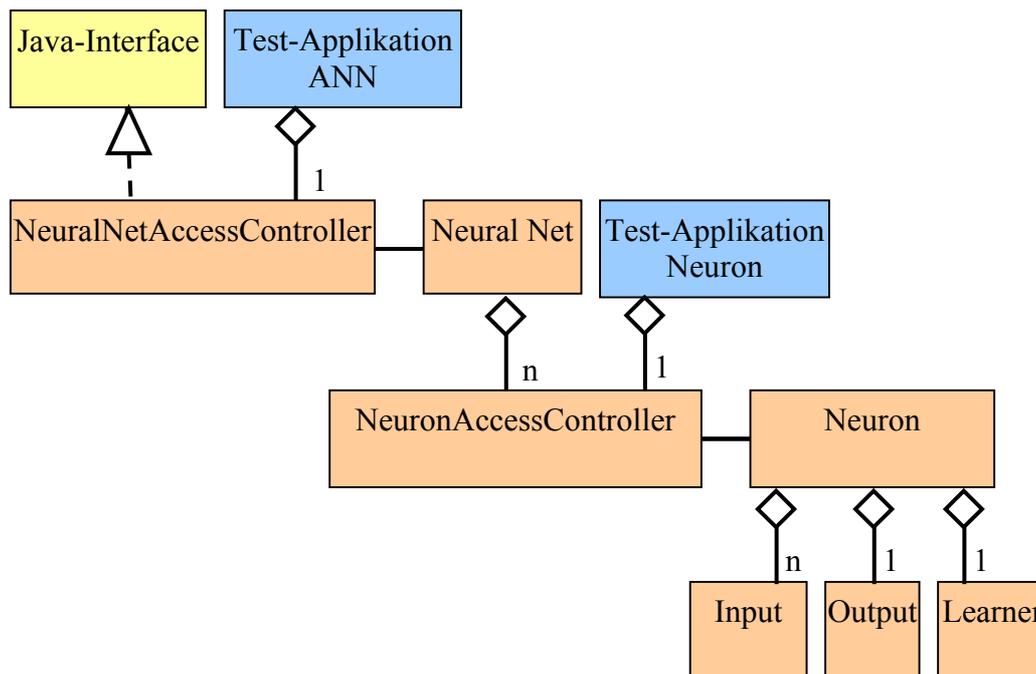


Bild 3: Vereinfachte Übersicht über die serverseitige Architektur von ANNCOLOGNE

2.5 Ergebnisse und Ausblick

Die bisher entstandene Applikation bietet bereits alle Grund-Funktionalität, die für einen Einsatz als ANN erforderlich ist. So kann man

- ein neues ANN definieren, Gewichte und Verbindungen anpassen und die Parameter jedes einzelnen Neurons modifizieren,
- eine ANN-Konfiguration speichern und auch wieder laden,
- Input- und ggf. Target-Daten laden,
- die Output-Daten des ANN für geladene Input-Daten bestimmen und speichern (inkl. evtl. Fehlerinformation bei vorhandenen Target-Daten),
- das ANN aufgrund geladener Input- und Target-Daten trainieren; dies kann im Einzelschritt, unter Vorgabe einer maximalen Epochenzahl oder unter Vorgabe eines maximal zulässigen Fehlers (mit einer oberen Zeitschranke) erfolgen. Die Ergebnisse während des Trainings lassen sich mit diversen Filtereinstellungen in unterschiedlicher Granularität speichern.

Ein Beispiel für das GUI von ANNCOLOGNE zeigt Bild 4.

Für die aktuelle Client/Server Implementierung wurden einige Performance-Messungen durchgeführt. Diese Messungen wurden sowohl für die Client- als auch für die Server-Applikation auf den bereits oben beschriebenen PCs durchgeführt.

Es wurden zwei ANN Konfigurationen erzeugt: eine 'breite' Konfiguration mit 400 Layern zu je 3 Neuronen, und eine 'tiefe' Konfiguration mit 20 Inputs, 20 Output-Neuronen und drei Hidden Layer mit je 100 Neuronen. In beiden Fällen lag eine vollständige Feedforward-Vernetzung vor.

Untersucht wurden vier Konfigurationen: ANNCOLOGNE als monolithische Applikation (mit innerer Bindung), als Client/Server-Applikation mit lokalem Server (localhost)

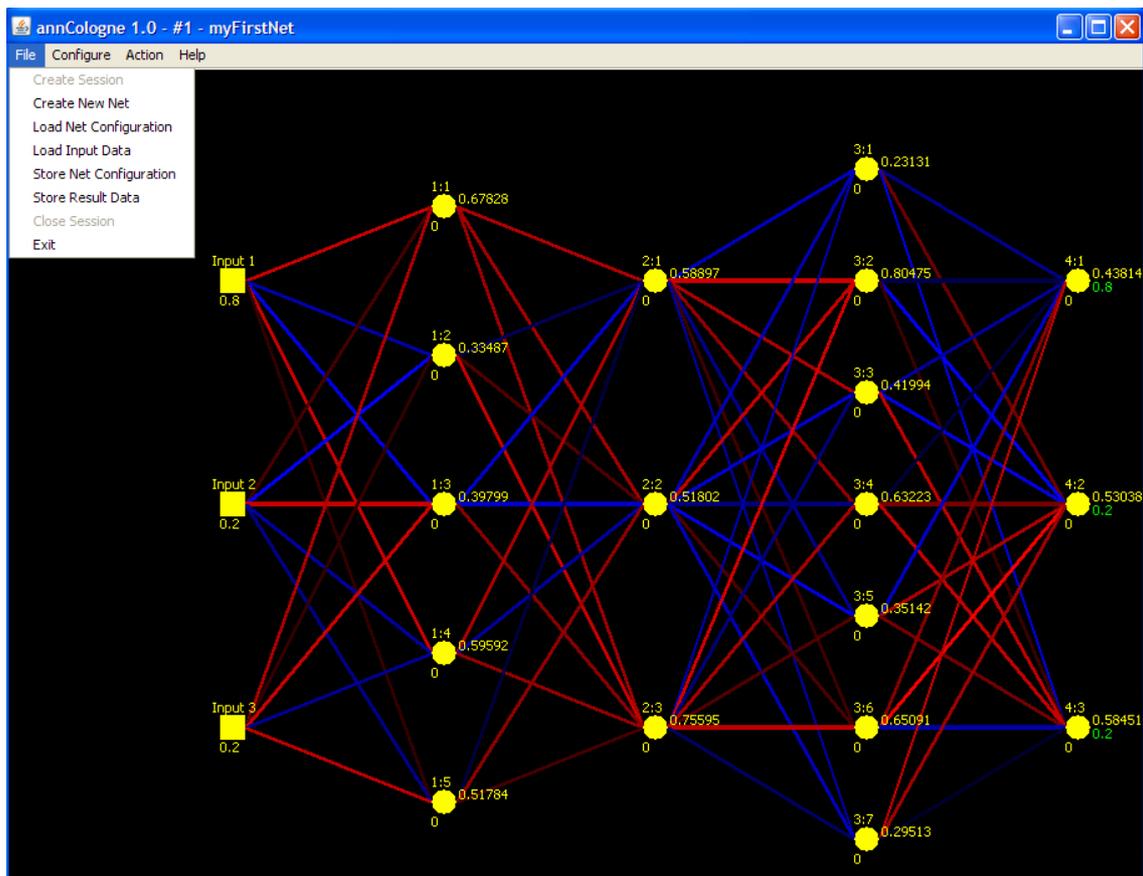


Bild 4: Darstellung eines ANN mit Input-Werten, Identifier (oberhalb), Bias (unterhalb), und Output-Werten der Neuronen, sowie Target-Outputs

sowie als Client/Server-Applikation mit remote Server über ein 100MBit/s und ein 10MBit/s Netzwerk.

Gemessen wurde der Zeitbedarf zum Generieren (Create) des ANN, zum Speichern (Store) des ANN in eine ANNSL-Datei und zum Laden (Load) des ANN aus einer ANNSL-Datei. Die Messergebnisse sind in Tabelle 1 dargestellt.

Tabelle 1: Performance von ANNCOLOGNE in verschiedenen Konstellationen

Zeitbedarf	Monolith	localhost	remote 100MBit/s	remote 10MBit/s
Create 'breit'	< 1s	< 1s	~ 1s	~ 3s
Create 'tief'	~ 2s	~ 7s	~ 21s	~ 37s
Store 'breit'	< 1s	< 1s	~ 1s	~ 1s
Store 'tief'	< 1s	~ 4s	~ 10s	~ 18s
Load 'breit'	< 1s	~ 1s	~ 3s	~ 4s
Load 'tief'	~ 2s	~ 11s	~ 31s	~ 55s

Ein stichprobenartiger Vergleich ergab, dass JavaNNS in der Performance zwischen dem ANNCOLOGNE Monolith und der localhost-Variante liegt. Ähnliches gilt für das hier nicht dargestellte Zeitverhalten im Training, bei dem der Zeitbedarf für ein einfaches ANN bei 10000 Epochen (je 10 Datensätze) verglichen wurden.

Die Konzeption und initiale Implementierung von ANNCOLOGNE kann als abgeschlossen betrachtet werden. Optimierungen mit Blick auf den Zeitbedarf beim Training finden derzeit statt. Modulare Erweiterungen werden nach und nach

entsprechend dem Bedarf eingeführt. Mit dieser Architektur ist es möglich, von einem Client aus mehrere ANN-Server auf unterschiedlichen Systemen zu steuern, und die Ergebnisse zu sammeln und auszuwerten. Durch Verwendung von wenig ausgelasteten Systemen als Plattform für ANN-Server kann insbesondere die Trainingsphase deutlich verkürzt werden; alternativ können bei gleichem Zeitaufwand eine größere Anzahl von Start-Konfigurationen untersucht werden. Mit ersten realen Aufgabenstellungen wird in Kürze gestartet.

3 XML-basierte Ablage von ANN-Daten und -Konfigurationen

3.1 Motivation und Übersicht

ANNs (artificial neural networks) stehen in Beziehung zu zwei Arten von Informationen. Zum einen ist ihr Verhalten beschrieben durch ihre jeweilige Konfiguration (Anzahl von Ein- und Ausgängen, Layern, Neuronen in den Layern; Typen und Eigenschaften der Neuronen; Verbindungen; ...). Zum Anderen interagieren sie mit der Außenwelt über Daten: Während des Trainings werden sie mit Eingangsdaten und Sollwerten für die Ausgänge gespeist und erzeugen Ausgangsdaten sowie veränderte Konfigurationen; im produktiven Einsatz verarbeiten sie Eingangsdaten und erzeugen Ausgangsdaten.

Sowohl die Konfiguration eines ANN als auch die auftretenden Daten werden in der Regel gespeichert; diese Speicherung wird meist von der Applikation übernommen, die das ANN implementiert und erfolgt häufig auf proprietäre Art. Bei einem Applikations-Wechsel oder bei paralleler Verwendung verschiedener Applikationen erfordert dies eine entsprechende Konvertierung. Eine kompatible Ablage wäre hier wünschenswert.

Dieser zweite Teil des Beitrags stellt zunächst typische use cases bei der Verwendung von ANNs zusammen und klassifiziert die auftretenden Informationen. Daraufhin werden die Möglichkeiten der bekannten, vorwiegend XML-basierten Lösungen zur Speicherung ANN-bezogener Daten kurz beschrieben. Es zeigte sich, dass diese Lösungen in Teilaspekten gute Ansätze boten, aber Schwächen in jeweils anderen Aspekten aufwiesen. Daher wurde mit ANNSL (artificial neural network specification language) ein Vorschlag für eine standardisierte Sprache vorgestellt, die nahezu alle Aspekte abdeckt.

3.2 Use Cases im Umfeld von ANNs

Im Einsatz von ANNs findet man typischerweise folgende Use Cases:

➤ Konfigurierung eines ANN

Hierbei wird entweder auf Basis eines schon bestehenden ANN oder (häufiger) ohne Voreinstellung ein ANN definiert. Im ersten Fall ist diesem Use Case eine ANN-Konfiguration zu übergeben. In beiden Fällen wird eine ANN-Konfiguration produziert.

➤ Training eines ANN

Für diesen Use Case ist dem ANN eine Konfiguration sowie eine ausreichende Zahl Trainingsdatensätze zur Verfügung zu stellen. Jeder Trainingsdatensatz besteht aus einem Inputvektor und einem Outputvektor, der als Zielvorgabe interpretiert wird.

In jeder Trainingsepoche produziert das ANN daraufhin für jeden Trainingsdatensatz jeweils einen Outputvektor und (je nach Trainingsstrategie) eine einzelne oder eine Sequenz von Konfigurationen. Jeder der sich ergebenden Outputvektoren kann mit der zugehörigen Zielvorgabe verglichen werden; aufgrund der beobachteten Abweichungen werden (entsprechend der gewählten Fehlerdefinitionen) Werte für den aktuellen Fehler des ANN ermittelt.

➤ **Bewertung eines ANN**

Ein konfiguriertes ANN kann im Hinblick auf seine Eignung für eine bestimmte Aufgabe bewertet werden. Dies erfolgt in der Regel im Anschluss an ein Training, kann aber auch in bestimmten Abständen während eines produktiven Einsatzes vorgenommen werden (Qualitätssicherung).

Für die Bewertung werden dem ANN eine Konfiguration und eine Anzahl Bewertungsdatensätze zur Verfügung gestellt. Ein Bewertungsdatensatz besteht (wie ein Trainingsdatensatz) aus einem Inputvektor und einem Outputvektor, der als Zielvorgabe interpretiert wird. Als Ergebnis einer Bewertung wird für jeden Bewertungsdatensatz ein Outputvektor produziert, aus dem (entsprechend der gewählten Fehlerdefinitionen) Werte für den aktuellen Fehler des ANN und damit für dessen Eignung ermittelt werden.

➤ **Produktiver Einsatz eines ANN**

In diesem aus Datensicht recht einfachen Fall wird dem ANN eine feste Konfiguration sowie nachfolgend sukzessiv Inputvektoren bereitgestellt, aus denen es jeweils zugehörige Outputvektoren produziert.

Daneben findet man weitere häufige Use Cases, die mit ANN-relevanten Informationen zu tun haben, allerdings ohne den Einsatz eines ANN:

➤ **Vorbereitung des Trainings oder der Bewertung eines ANN**

Zur Vorbereitung eines Trainingslaufes oder einer ANN-Bewertung müssen eine ausreichende Anzahl Paare aus Input- und Outputvektoren (Bewertungs- bzw. Trainingsdatensätze) erzeugt werden. Die Outputvektoren stellen dabei die jeweils gewünschten Ausgaben an den ANN-Outputs dar.

➤ **Vorbereitung des produktiven ANN Einsatzes**

Voraussetzung zur Verwendung eines ANN im produktiven Einsatz ist die Verfügbarkeit von Inputvektoren. Im online-Einsatz wird durch einen separaten Prozess jeder Inputvektor unmittelbar bei seiner Verfügbarkeit an das ANN weitergegeben; im offline-Einsatz müssen diese Input-Vektoren gesammelt werden, um dann im Nachgang gemeinsam dem ANN übergeben zu werden.

➤ **Auswertung von ANN-bezogenen Informationen**

Die in den vorigen Use Cases bereits aufgetretenen Informationen (ANN-Konfiguration, Inputvektoren, Outputvektoren als Zielvorgaben und als Ist-Ergebnisse, Fehlerwerte, ...) können je nach Fragestellung auf sehr unterschiedlicher Art und in diversen Kombinationen ausgewertet werden.

3.3 Klassifikation ANN-bezogener Informationen

Die in den verschiedenen Use Cases in unterschiedlicher Zusammensetzung aufgetretenen Informationen können wie folgt klassifiziert werden:

Klasse "ANN Configuration": Sie beinhaltet (unter Anderem) Informationen über

- die Anzahl der Inputs und Outputs,
- die Anzahl der Layer und der Neuronenzahl in jedem Layer
- den Typ der einzelnen Neuronen
- die Verbindungen zwischen den Neuronen (feedforward/feedback, shortcuts, ...)
- die Gewichte (bzw. center-Referenzen) der Neuron-Eingänge und ggf. Bias-Werte
- den Typ und ggf. die Parameter von Net-, Activation- und Output-Function
- die Sequenz der internen Verarbeitung
- das Lernverfahren und die ggf. erforderlichen Parameter

Klasse "Net Input Data": Sie beinhaltet

- beschreibende Informationen zu den Inputvektoren (z.B. Dimension)
- eine Anzahl Inputvektoren

Klasse "Target Output Data": Sie beinhaltet

- beschreibende Informationen zu den Outputvektoren (z.B. Dimension)
- eine Anzahl Outputvektoren, die typ. als Zielwerte interpretiert werden

Klasse "Training Result Data": Sie beinhaltet

- beschreibende Informationen zu den Outputvektoren (z.B. Dimension) und zum Trainingslauf (z.B. Trainer)
- Informationen über die anfängliche ANN-Konfiguration (embedded oder als Referenz)
- Ergebnisse jeder einzelnen Trainingsepoche; diese beinhalten (z.T. optional)
 - die ANN-Konfiguration die sich jeweils aufgrund des Trainingsschrittes oder der gesamten Epoche ergeben hat (embedded oder als Referenz)
 - die Outputvektoren, die das ANN jeweils erzeugt hat
 - die aus dem Vergleich mit den Zielwerten sich ergebenden Fehlerwerte nach jedem Trainingsschritt und die zugrundeliegenden Fehlerdefinitionen.

Klasse "Net Output Data": Sie beinhaltet (z.T. optional)

- beschreibende Informationen zu den Outputvektoren (z.B. Dimension) und dem ANN-Lauf (z.B. Operator)
- die Outputvektoren, die typ. als reale Ergebnisse eines ANN interpretiert werden
- Informationen über die ANN-Konfiguration, mit der diese Outputvektoren erzeugt wurden (embedded oder als Referenz)
- die aus dem Vergleich mit den Zielwerten sich ergebenden Fehlerwerte und die zugrundeliegenden Fehlerdefinitionen (sofern Zielwerte bekannt sind).

Mit dieser Klassifizierung können die Use Cases und die darin involvierten Informationsklassen entsprechend folgender Tabelle in Beziehung gesetzt werden.

Tabelle 2: verwendete (used) und erzeugte (produced) Informationen in den Use Cases, dargestellt als Paar "used | produced"; manche Bestandteile sind optional.

Use Case	Klasse	Configuration	Input	TargetOutput	TrainingResult	NetOutput
Konfigurierung		✓ ✓	-- --	-- --	-- --	-- --
Training		✓ ✓	✓ --	✓ --	-- ✓	-- ✓
Bewertung		✓ --	✓ --	✓ --	-- --	-- ✓
Prod.Einsatz		✓ --	✓ --	-- --	-- --	-- ✓
Vorbereitung T/B		-- --	-- ✓	-- ✓	-- --	-- --
Vorbereitung P		-- --	-- ✓	-- --	-- --	-- --
Auswertung		✓ --	✓ --	✓ --	✓ --	✓ --

3.4 Bekannte XML-Sprachen zur Speicherung ANN-bezogener Informationen

In den vergangenen Jahren wurden verschiedene XML-Sprachen vorgestellt, die Informationen mit ANN-Bezug speichern können (so z.B. [6] bis [12]). Sie wurden u.a. anhand von Beispieldateien, Literaturstellen und Internet-Informationen analysiert.

Die meisten Vorschläge beschäftigen sich mit Informationen zur Konfiguration eines ANN. Nur wenige behandeln die Ablage von Input-, Target- und Output-Daten und ihren Bezug zur entsprechenden ANN Konfiguration. Die Ablage von Zwischenergebnissen eines Trainingsprozesses ist mit keiner der untersuchten Vorschläge möglich.

Manche Vorschläge erlauben nur homogene ANN-Strukturen; die Verbindungen müssen z.T. vollständige und reine Feedforward Struktur aufweisen, und die Neuronen müssen in ihrem internen Verhalten häufig identisch sein.

Manche Vorschläge sind in ihrer Sprachstruktur unvorteilhaft skaliert; ein ungünstiges Verhältnis zwischen Anzahl und Länge der erforderlichen XML-Tags und dem jeweiligen XML-Content erzeugt sehr große XML-Dateien, was dann für große ANNs Nachteile in Performance und Speicherbedarf entsprechender Applikationen mit sich bringt.

Viele der vorgeschlagenen Lösungen gehen von einer fest vorgegebenen Definition der Net-, Activation- und Output-Funktion aus; in einigen Fällen ist es möglich, diese durch einen aus einer Liste wählbaren Namen zu spezifizieren. NNML fällt hier durch die Möglichkeit auf, diese Funktionen mit einer eigenen Syntax zur allgemeinen Beschreibung mathematischer Funktionen frei zu definieren.

Ebenso gehen die meisten Vorschläge von einem (Backpropagation) oder einer kleinen Menge vordefinierter und wohlbekannter (und dann durch Namen spezifizierter) Lernalgorithmen aus.

Allerdings findet man in mehreren der vorgeschlagenen Sprachen die Möglichkeit, bestimmte Elemente durch eigene XML-Elemente zu erweitern; dies erlaubt zwar die Ergänzung fehlender Informationen, ist allerdings eine individuelle Lösung, die, wenn nicht abgestimmt, den Informations-Austausch zwischen Tools verhindert.

Aufgrund der weiten Verbreitung wurde zudem auch noch das Ablageformat von SNNS [1] betrachtet. Es ermöglicht eine flexible Speicherung verschiedenster ANN-Typen; allerdings sind auch hier Einschränkungen festzustellen. Trainingsbezogene (Zwischen-) Ergebnisse können nicht abgelegt werden. Zudem handelt es sich dabei nicht um eine XML-Sprache sondern um ein z.T. tabellenorientiertes ASCII-Format, das die Erstellung entsprechender Parser deutlich aufwändiger macht.

Für die in Zukunft beabsichtigten Anwendungen konnte aus den betrachteten Ablageformaten keines als geeignet befunden werden. Aus diesem Grund wurde auf Basis der oben dargestellten Informationsklassen mit ANNSL (artificial neural network specification language) ein Vorschlag für eine neue, erweiterte, und universeller einsetzbare XML-Sprache erarbeitet, deren Grundzüge im Folgenden dargestellt wird.

3.5 Konzept von ANNSL

Mit den zuvor beschriebenen Informationsklassen lassen sich zum Einen die Konfiguration eines ANN und zum Anderen die Daten die in diesem Zusammenhang auftreten, repräsentieren. Ihre Beziehungen untereinander sind vom Use Case abhängig.

Allerdings lässt sich feststellen, dass die Daten-Klassen untereinander enger gekoppelt sind als zur Konfigurations-Klasse; TargetOutput Daten können nur zusammen mit Input Daten verwendet werden; beide sind notwendig um TrainingResult Daten vollständig zu interpretieren und ggf. nachzuvollziehen, und auch NetOutput Daten erfordern Input und ggf. TargetOutput Daten. In vielen Fällen handelt es sich dabei um 1:1 Beziehungen. Hingegen können Konfigurationen unabhängig von Daten existieren,

und dieselbe Konfiguration wird z.B. im produktiven Einsatz für eine Vielzahl verschiedener Datensätze verwendet.

Diese und weitere Überlegungen führten schließlich dazu, dass ANNSL eine XML Sprache für Dokumente beschreibt, die entweder eine Konfiguration oder eine (weitgehend flexible) Kombination von Daten-Klassen beinhalten. Die höchste Strukturebene ist in Bild 5 dargestellt (dieses und die weiteren XML Strukturbilder wurden mit Liquid XML Studio erstellt, einem freien XML Schema- und Dokumenten-Editor; www.liquidtechnologies.com).

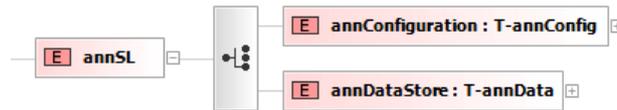


Bild 5: Oberste Strukturebene von ANNSL

Während Informationen der vier Daten-Klassen in ein XML-Element `annDataStore` eingebettet werden, sind ANN Konfigurations-Informationen in `annConfiguration` enthalten. Beide XML-Elemente werden in den nächsten Abschnitten kurz vorgestellt.

3.6 Konzept der ANNSL Datenablage

Das Element `annDataStore` kann eine oder mehrere der Daten-Klassen beinhalten.

Jedes der vier möglichen inneren Element hat eine ihm eigene Strukturierung. Während `NetInputData`, `TargetOutputData` und `NetOutputData` sich recht ähnlich sind, ist `TrainingResultData` aufgrund der Eigenart seiner Information deutlich komplexer strukturiert; es enthält u.a. die anfängliche ANN-Konfiguration und ein oder mehrere `EpochResult` Elemente mit Ergebnissen der einzelnen Trainingsepochen.

Um eine kompakte Darstellung zu ermöglichen, wird für jeden Input- bzw. Output-Vektor nur jeweils ein XML-Element verwendet; seine Daten werden als Whitespace-separierte Zahlenwerte gespeichert. Ein Bezug zu einer ANN-Konfiguration erfolgt entweder durch Einbettung einer ANNSL Konfigurationsablage oder durch die Referenzierung eines ANNSL-File, in dem dann die Konfiguration des ANN spezifiziert ist.

Bild 6 zeigt den inneren Aufbau von `annDataStore` sowie eines Teils seiner inneren Elemente. Aus Platzgründen muss hier auf weitere Erläuterungen verzichtet werden.

3.7 Konzept der ANNSL Konfigurationsablage

Das Element `annConfiguration` beinhaltet die Beschreibung einer ANN-Konfiguration. Der Kern dieser Konfigurationsinformation ist in den `Layer` Elementen enthalten. Bild 7 zeigt den inneren Aufbau von `annConfiguration` sowie eines Teils seiner inneren Elemente.

Für den Fall, dass alle Neuronen eines Layers hinsichtlich bestimmter Eigenschaften gleiches Verhalten zeigen, können diese Eigenschaften zentral beim Layer spezifiziert werden.

Für die Verbindungen zwischen Neuronen wurde kein eigenes Element auf oberer Ebene eingeführt; da jeder Input von genau einer Quelle gespeist wird, werden sie stattdessen direkt im Element `Inputs` definiert.

Beide Maßnahmen sorgen dafür, dass in den meisten Fällen eine sehr kompakte XML Beschreibungsdatei entsteht.

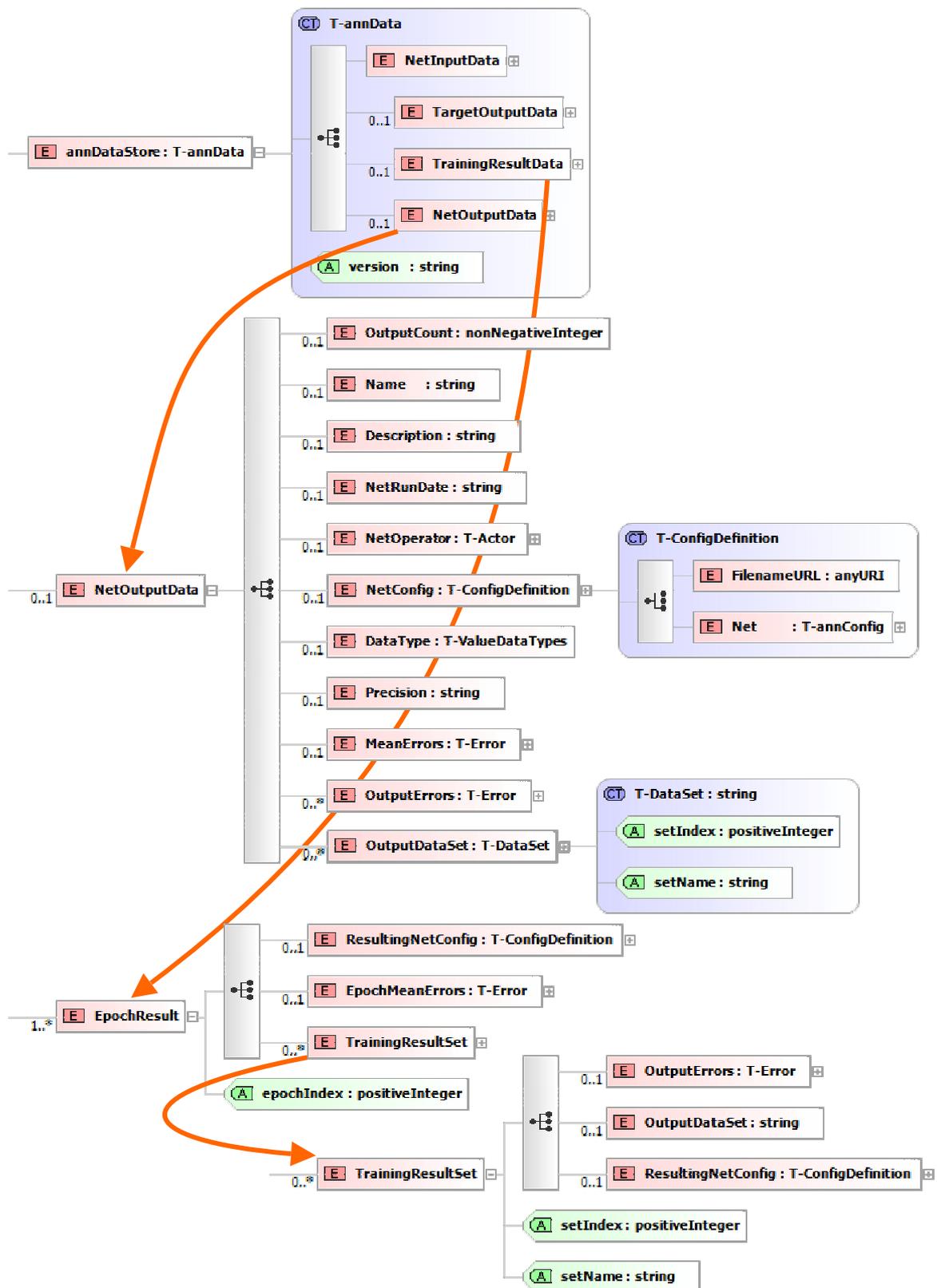


Bild 6: Der innere Aufbau von `annDataStore` mit Darstellung des darin enthaltenen Elementes `NetOutputData` sowie eines Teils des Inhaltes von `TrainingResultData`

Für spezielle Anforderungen erlaubt ANNSL die Definition eigener Funktionen als Net-, Activation- und Output-Funktion. Dazu wird jedoch (anders als in NNML) keine eigene

Beschreibungssprache für mathematische Ausdrücke eingeführt, sondern MathML- oder OpenMath-kompatible XML-Syntax zugelassen ([13],[14]). Dies geschieht innerhalb des Elementes `Definition` in den einzelnen Funktionsspezifikationen.

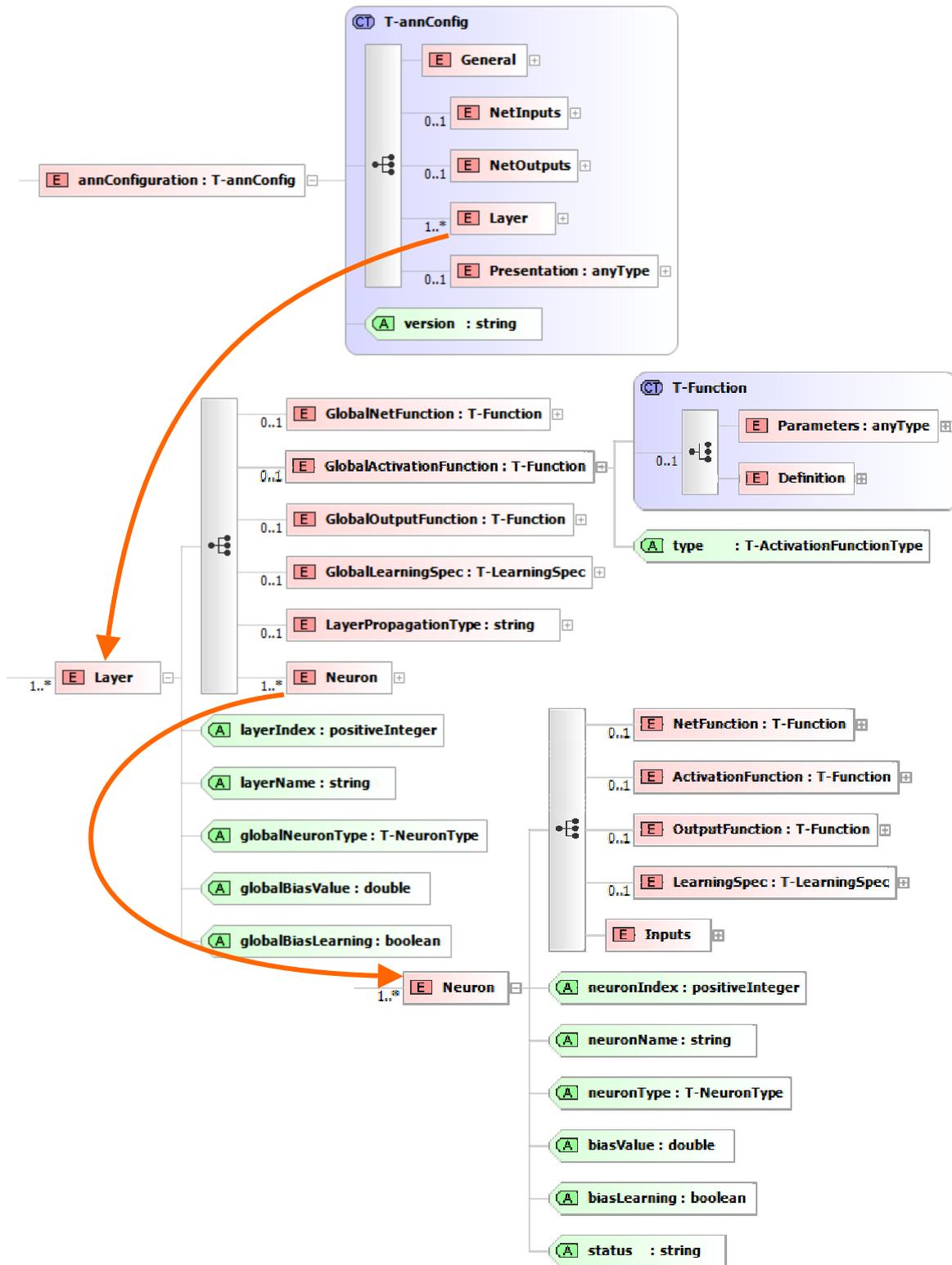


Bild 7: Der innere Aufbau von `annConfiguration` mit Darstellung des darin enthaltenen Elementes `Layer`

3.8 Aktueller Stand und Ausblick

Die Entwicklung von ANNSL ist weitgehend abgeschlossen. Die Spezifikation dieser Sprache übernimmt ein entsprechendes XML Schema (so dass eine formale Validierung vorgenommen werden kann), ergänzt um weitere textuell formulierte Regeln. Da die Struktur der beiden Hauptelemente als Schema-Typ definiert ist, können ANNSL Beschreibungen ihrerseits einfach in andere XML-Sprachen eingebettet werden. Die Dokumentation von ANNSL findet sich in [15]. Ein Beispiel für jeweils eine einfache ANNSL-Datei mit Daten- sowie mit Konfigurationsinformation gibt Bild 8.

Eine erste Implementierung der wesentlichen Funktionalität ist erfolgt. Ein Java Package ist verfügbar, das es erlaubt, ein Configuration Repository und ein Data Repository zu erzeugen, aus XML-Dateien zu füllen und in XML-Dateien zu speichern. Ein umfangreiches API unterstützt den Zugriff auf den Repository-Inhalt, so dass für darauf aufbauende Applikationen ein Parsen der XML-Dateien nicht mehr erforderlich ist. Performance-Messungen mit künstlich erzeugten Testdaten zeigten sehr gute Ergebnisse. So konnte ein vollständig Feedforward-verbundenes ANN mit 200 Inputs, 1000 Neuronen in einem Hidden Layer und 200 Output-Neuronen jeweils in weniger als 1s vom Repository geschrieben bzw. in das Repository gelesen werden (2GHz Dual-Core PC; Windows XP); die dabei entstandene XML-Datei hatte eine Größe von etwa 10MB.

```
<?xml version="1.0" encoding="UTF-8"?>
<annSL>
  <annConfiguration version="1.0">
    <General>...</General>
    <Layer layerIndex="1" ... >
      ...
      <Neuron neuronIndex="1">
        <Inputs>
          <FromLayers>0</FromLayers>
          <FromNeurons>1</FromNeurons>
          <InputWeights>-0.8475</InputWeights>
        </Inputs>
      </Neuron>
      <Neuron neuronIndex="2">...</Neuron>
      <Neuron neuronIndex="3">...</Neuron>
    </Layer>
    <Layer layerIndex="2" ... >
      ...
      <Neuron neuronIndex="1">
        <Inputs>
          <FromLayers>1 1 1</FromLayers>
          <FromNeurons>1 2 3</FromNeurons>
          <InputWeights>
            0.2 -0.21 0.6</InputWeights>
        </Inputs>
      </Neuron>
      <Neuron neuronIndex="2">...</Neuron>
    </Layer>
  </annConfiguration>
</annSL>

<?xml version="1.0" encoding="UTF-8"?>
<annSL>
  <annDataStore version="1.0">
    <NetInputData>
      <InputCount>4</InputCount>
      <Name>Iris_UCI_scaledIn</Name>
      ...
      <InputDataSet setIndex="1">
        -0.901 1.032 -1.341 -1.313</InputDataSet>
      <InputDataSet setIndex="17">
        -0.537 1.958 -1.398 -1.050</InputDataSet>
      <InputDataSet setIndex="71">
        0.069 0.338 0.592 0.791</InputDataSet>
      <InputDataSet setIndex="103">
        1.523 -0.125 1.218 1.185</InputDataSet>
    </NetInputData>
    <TargetOutputData>
      <OutputCount>1</OutputCount>
      <Name>Iris_UCI_scaledTarg</Name>
      ...
      <TargetDataSet setIndex="1">
        0.2</TargetDataSet>
      <TargetDataSet setIndex="17">
        0.2</TargetDataSet>
      <TargetDataSet setIndex="71">
        0.5</TargetDataSet>
      <TargetDataSet setIndex="103">
        0.8</TargetDataSet>
    </TargetOutputData>
  </annDataStore>
</annSL>
```

Bild 8: Auszug aus einer Beispieldatei für Konfigurationen (links) und für Daten (rechts; sie sind eine skalierte Repräsentation des korrigierten Iris-Datensatzes [16])

Bisher nicht durch die Implementierung abgesichert ist der Bereich der selbst-definierten Funktionen mittels MathML oder OpenMath. Hier können u.U. Ergänzungen der ANNSL Spezifikation erforderlich werden.

Die nächsten Schritte sind die vollständige Einbindung dieser Repositories in ANNCOLOGNE, sowie Weiterentwicklungen aufgrund auftretender Anforderungen.

4 Zusammenfassung

ANNCOLOGNE und die darunter liegende und durch ANNSL definierte Persistenzschicht sind zwei neue Ergebnisse der Arbeiten im CI-Bereich der Fakultät IME an der FH Köln. Sie bilden eine Basis, die in den kommenden Jahren sukzessiv weiter ausgebaut wird und zur Unterstützung entsprechender Master-Angebote sowie zur Untersuchung industrieller und wissenschaftlicher Fragestellungen dienen soll.

Die ANNSL-Spezifikation und das Java-Package können ("as is") von der entsprechenden Webseite geladen werden [15]. Feedback sowie Ideen für gemeinsame Projekte sind jederzeit gerne willkommen.

5 Literatur

- [1] A. Zell et al., "SNNS Stuttgart Neural Network Simulator user manual," Technical Report, University of Stuttgart, 1992
- [2] M. Noubough, "Design und Implementierung einer verteilten Anwendung zur Simulation neuronaler Netze mit Java-RMI basierend auf JavaNNS", Diplomarbeit FH Köln, 2008
- [3] Ph. Richter, M. Frey, "Programmierung eines Neurons in Java (als Building Block eines neuronalen Netzes)", Projektarbeit FH Köln, 2006
- [4] M. Reifferscheidt, A. Stec, M. Schlebusch, M. Fehre, "Neuronales Netz / Daten-Management für ein neuronales Netz", Projektarbeiten FH Köln, 2007/2008
- [5] R. Saidi, "Netzwerkprogrammierung eines neuronalen Netzes mittels RMI und CORBA", Bachelor-Thesis FH Köln, 2008
- [6] D. Rubtsov, S. Butakov, "A unified format for trained neural network description," Proc. of the 2001 Int. Joint Conf. on Neural Networks, Washington, pp.2367-2372
- [7] A.R. de Soto, et al. "Fuzzy Systems and Neural Networks XML Schemas for Soft Computing," *Mathware & Soft Computing*, (Vol. X, Núm. 2-3, Part B: Applications. Special issue on fuzzy sets in computer vision), 2003
- [8] P. Makhfi, "NNDEF Document Type Definition," published online, www.makhfi.com/nndef_v1_0.htm, 2003
- [9] T.S. Hussain, "Attribute Grammar Encoding based upon a Generic Neural Markup Language: Facilitating the Design of Theoretical Neural Network Models," Proc. of the 2004 Int. Joint Conf. on Neural Networks, pp.235-241
- [10] A. Madhusudanan, "NXML - Introducing an XML Based Language To Perform Neural Network Processing, Image Analysis, Pattern Detection Etc.," www.codeproject.com/KB/dotnet/nxml.aspx, 2006
- [11] S.L. Thaler, "Neural Network Development and Data Analysis Tool," US Patent Application 11/375630
- [12] Data Mining Group (DMG), "The Predictive Model Markup Language (PMML)," version 3.2, www.dmg.org/pmml-v3-2.html, May 2007
- [13] W3C, "Mathematical Markup Language (MathML)," version 2.0, www.w3.org/TR/MathML2, published online, 21-Oct 2003
- [14] The OpenMath Society, "The OpenMath Standard," version 2.0, www.openmath.org/standard, published online, Jun. 2004
- [15] R. Bartz, "The Artificial Neural Network Specification Language ANNSL," www.nt-rt.fh-koeln.de/research/annSL/index.html, v1.0, 2008
- [16] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, UCI, School of Information and Computer Science, 2007, archive.ics.uci.edu/ml/datasets/Iris

Daten, Information, Wissen – und die Computational Intelligence. Ein Beitrag zu 60 Jahren Informationstheorie

Rudolf Seising

Friedrich-Schiller-Universität Jena - Ernst-Haeckel-Haus
Institut für Geschichte der Medizin, Naturwissenschaft und Technik
Berggasse 7, D-07745 Jena

Abstract

Die drei im Titel dieses Beitrags stehenden Begriffe charakterisieren eine Thematik, die nach den elektro- und computertechnischen Entwicklungen im 20. Jahrhundert aufkam.

In den verschiedenen Wissenschaftsdisziplinen werden die Begriffe "Daten", "Information" und "Wissen" seither unterschiedlich diskutiert, voneinander abgegrenzt und interpretiert.

Diese Problematik geht auf den 1948 erschienenen Artikel *The Mathematical Theory of Communication* des US-amerikanischen Mathematikers und Elektrotechnikers Claude E. Shannon (1916-2001) zurück, der seine große Bekanntheit einer allgemeinverständlichen Darstellung seines Inhaltes verdankt, die der Mathematiker und damalige Director of Natural Sciences der *Rockerfeller Foundation* Warren Weaver (1894-1978) für die Zeitschrift *Scientific American* 1949 schrieb. Beide Abhandlungen wurden dann gemeinsam im Jahre 1949 als Buch veröffentlicht.

Sowohl im historischen als auch in epistemologischer Hinsicht lassen sich Bezüge der von Shannon und Wiener noch als „A Mathematical Theory of Communication“ genannten Informationstheorie zum Forschungsgebiet der *Computational Intelligence* herstellen. Die Informationstheorie gilt als eine der Wurzeln der gesamten Informatik und Artificial Intelligence und Lotfi Zadeh, der Begründer der Fuzzy Set Theorie – ein zentraler Bestandteil der Computational Intelligence – engagierte sich in den 1950er Jahren bei der Etablierung der Information Theorie. Die Schichtung des Informationsbegriffs in syntaktische, semantische und pragmatische Information einerseits und die in den letzten Jahrzehnten immer wieder thematisierte und kontrovers diskutierte Unterscheidung der Begriffe *Daten*, *Information* und *Wissen* sowie die Möglichkeit der Fuzzifizierung insbesondere des Informationsbegriffs geben Anlass, diese Begriffe zum 60. Jubiläum unter dem Aspekt der Computational Intelligence zu beleuchten.

Ant Colony Optimization for Cash Distribution

Stephan A. Hartmann, Thomas A. Runkler, Ralph Grothmann and
Joachim Bamberger

Siemens AG, Corporate Technology
Information and Communications, CT IC 4
Otto-Hahn-Ring 6, 80200 Munich - Germany
E-Mail: {stephan.hartmann.ext, thomas.runkler, ralph.grothmann,
joachim.bamberger}@siemens.com

Abstract

In this paper we address the problem of optimizing the replenishment of automated teller machines (ATMs) with respect to the sum of backorder, transportation and money storage costs. We transform this problem into a special case of the well known multi-day inventory routing problem (IRP) with one incapacitated vehicle. For the first time an algorithm based on ant colony optimization (ACO) is proposed to tackle the IRP. A comparative analysis shows that our approach is superior to genetic algorithms and local search approaches on a set of benchmark instances.

1 Introduction

Problem description. In this paper we address the problem of finding a near-optimal strategy for the replenishment of *automated teller machines* (ATMs). ATMs are assigned to special facilities or a depot where the money cartridges are filled and prepared for delivery. In contrast to most other commodities (liquids, gases, etc.) the filling process itself already takes place at the depot, i. e. there is no direct access to the bills anymore after the cartridges have left the depot. An armored car then picks up these cartridges and distributes them to the respective ATMs that are scheduled to be refilled on that day. At each ATM site a filled cartridge replaces the one that has been sitting in the machine, thus a direct refill of the commodity in the traditional sense does not take place. Most likely these return cartridges still hold some cash that is returned to the depot. The total cost of the refilling process can be distinguished in storage, travel and backorder costs.

The storage cost for cash in ATMs mainly comprises inflation and loss of interest due to frozen assets. The travel costs are the operational costs for the car and depend on the distance traveled. If an ATM runs empty additional backorder costs arise proportional to the amount of cash that could not be disbursed to customers. The backorder cost factor quantifies the dissatisfaction of customers. In contrast to other similar problems backorders are not treated as negative storage level, i. e. withdrawals cannot be delayed.

In order to reduce overall costs the supply chain practice *vendor managed inventory* (VMI) is used where the inventory is monitored and managed by the vendor on behalf of the customer. Instead of sending purchase orders, customers send demand information to the supplier. The supplier generates replenishment orders for the customer based on this demand information and takes full responsibility for maintaining an agreed level of inventory. This approach enables the vendor to combine deliveries and thus make more efficient use of resources.

The theoretical counterpart to VMI is the *inventory routing problem* (IRP). The IRP is derived from the well known *vehicle routing problem* (VRP). The VRP is solved on a daily basis, i. e. the optimal delivery routes are computed for each day independently based on the customer orders for each day. When applying the VMI strategy the vendor is responsible for replenishment thus also sets the replenishment dates and amounts for each customer. Thus IRPs have to deal with a longer horizon when looking at the optimality of a planning since each decision also has impact on future actions. For example, if one of two nearby customers is replenished on one day and the other customers on the next day this may result in higher transportation costs.

The IRP in general deals with the distribution of a single commodity from a central depot to a number of customers on a repeated basis over a given planning horizon. The inventory of each customer as well as the capacity of each of the homogenous distribution vehicles is limited. The objective is to minimize the average distribution costs during the planning period. Thus they combine two aspects of the supply chain: vehicle routing and inventory control.

IRPs already appeared in the 1970s but extensive research started around 1990. Since then many variations have been studied and many different approaches and algorithms have been developed. In [1] Moin and Salhi give a full overview of problem variations and recent approaches.

IRPs can be distinguished in single period models [2], multi-period models [3] and infinite time horizon models [4]. Like most of the publications for multi-periodic approaches [5] we also suggest a 2 phase approach where in phase 1 the clients and the fill amounts for each day are selected and in phase 2 the vehicle routing problem is solved.

The cash distribution problem that we consider in this paper is a special variant of the inventory routing problem. The fleet consists of only one vehicle thus the VRP is reduced to a traveling salesman problem. The vehicle capacity can be assumed to be infinite since the maximum number of storage slots for cartridges in the armored car is large enough for this problem. The demand data is forecasted and provided by the bank, so we deal with a so called deterministic IRP. In contrast to the general IRP the customer (ATM) is part of the same company as the vendor, thus we also include storage costs that need to be minimized in the objective function. IRPs have been optimized by integer programming [5] and genetic algorithms [6]. In this paper we present an algorithm based on ant colony optimization (ACO) for solving this problem. ACO has been applied to the VRP in [7], however, to the best of our knowledge ACO has never been used for solving IRPs.

Both, the assignment in phase 2 as well as the routing in phase 2 can be represented by matrices. This makes ACO a preferable choice for solving the problem since it uses a pheromone matrix for the construction of a solution.

This paper is structured as follows: Section 2 gives a short introduction to the ACO paradigm. Subsequently we present the adaption of ACO to IRP as well as appropriate guiding heuristics. After that we introduce a local search algorithm and a genetic algorithm suitable for this problem. In Section 6 we present computational results of the ACO adaption as well as comparisons to other approaches. At the end we give conclusions and an outlook to future work.

2 Ant Colony Optimization

Ant colony optimization (ACO) [7] is a bio inspired multi-agent meta-heuristic, belonging to a special group that attempts to emulate behaviors characteristic of social insects, in this case ant colonies. In ACO, the behavior of each agent in the optimization mimics the behavior of real life ants and how they interact with each other in order to find resources and carry them to the colony efficiently. To put it briefly, during a walk each ant deposits *pheromones*. Other ants are sensitive to these pheromones encouraging them to follow the trail, with higher or lower intensity depending on the concentration of the pheromone. After a period of time, the pheromones on the shortest path are saturated since this path is visited more frequently. This concentration is aided by the evaporation over time of the pheromones deposited on the paths, which makes less used and longer paths less and less attractive to the ants. In this way, through involuntary collaboration, ants can find the shortest path between a food source and their colony. This type of coordination is generally known as *stigmergy*.

In ACO algorithms the set of agents (artificial ants) probabilistically construct solutions S using the *pheromone matrix* τ that resembles the colony's collective memory. Besides the pheromones, ACO agents can be and are usually given some degree of "intelligence", in the form of a problem specific *heuristics* η which help guiding the search.

Since 1992 ACO has been applied to various discrete optimization problems, such as logistic problems [8] [9], maximum satisfaction problems [10], Bayesian network optimization problems [11] and many more.

3 ACO adaption for the IRP

Our cash distribution problem described in Section 1 is split into 3 sub-problems:

Problem 1 Which customer should be served on which day?

Problem 2 How much should be delivered to a customer when served?

Problem 3 Which delivery route should be used?

In order to apply ACO to this optimization problem, one needs to define: an appropriate representation of the problem, a problem specific heuristics η_{ijq} to help guide the ants, a way of updating the pheromone track τ_{ijq} , and the probabilistic rule p_{ijq} that moves the ants to the next stage. We used a modified version of the *MAX-MIN Ant System* (MMAS) [12].

Representation of the problem. Each solution of Problem 1 can be represented by a binary matrix Ψ of size $i_{max} \times j_{max}$ where $\psi_{ij} = 1$ if customer i is scheduled for day j , and 0 otherwise. The decision of ψ_{ij} being assigned or not is independent from all other decisions made for other customers and/or days. When Ψ is fixed the fill amounts $r_{ij} \in R$ for all $\psi_{ij} = 1$ can be computed by summing up the forecasted demand until the next replenishment is scheduled. Note that shortages can occur as the refill amount is limited by the customer's storage capacity which result in high backorder costs.

Pheromone update. The artificial pheromones are placed onto the elements ψ_{ij} of Ψ . If the customer-day-tuples (i, j) are stored in a matrix of size $i_{max} \times j_{max}$, then the dimension of the pheromone matrix is $i_{max} \times j_{max} \times 2$, since it separately stores the pheromones for the assignment values 0 and 1, thus $q \in \{1, 2\}$. The pheromone matrix contains knowledge of how good the results were when either 0 or 1 was chosen for each tuple in the creation of former solutions. After each iteration the pheromone matrix is updated by evaporating all pheromones

$$\tau_{ij} \leftarrow \tau_{ij} \cdot (1 - \rho) \quad (1)$$

where ρ is the evaporation coefficient. Then pheromones representing the iteration's best solution Φ_{IB} as well as the so far global best solution Φ_{GB} are then increased depending on the solution quality:

The update is done by applying evaporation to τ_{ij} followed by pheromone deposit by adding the inverse of total cost $v(\Phi)$ of the assignment Φ multiplied by the constant u , to the positions in the pheromone matrix corresponding to values of ψ_{ij} :

$$\tau_{ijk}(l+1) = \tau_{ijk}(l) \cdot (1 - \rho) + \delta_{ijk} \quad (2)$$

where δ_{ijk} is defined here as $\delta_{ijk} = u/v(\Phi)$. The total cost $v(\Phi)$ consists of travel costs ($v_t(\Phi)$), storage costs ($v_s(\Phi)$) and backorder costs ($v_b(\Phi)$) and u is a constant parameter.

Heuristics. Given a day d_j we want to determine an assignment for all $\phi_{i\hat{j}}$. Define d_{min} , d_{max} and d_{avg} as the minimum, maximum and average distance between all customers, respectively. The heuristic information $\eta_{i\hat{j}k}$ for $\phi_{i\hat{j}}$ depends on the cities which the decision of being assigned to day d_j or not has already been made for and it is stored in matrix E of size $i_{max} \times j_{max} \times 2$ which is initialized with d_{avg} . Whenever ψ_{ij} is set to 1, all nearby customers should also be more likely to be assigned on day d_j and for distant customers vice versa. On the other hand if ψ_{ij} is set to 0, all distant customers should also be more likely to be assigned and vice versa. If $\psi_{i\hat{j}}$ is set to $b \in \{0, 1\}$ then all entries in E for day \hat{j} are updated as follows:

$$\eta_{i\hat{j}b} = \eta_{i\hat{j}b} + |dist(\hat{i}, i) - d_{min}| \quad (3)$$

$$\eta_{i\hat{j}\bar{b}} = \eta_{i\hat{j}\bar{b}} + |dist(\hat{i}, i) - d_{max}| \quad (4)$$

where $dist(\hat{i}, i)$ returns the distance between city \hat{i} and i . Everytime a decision is made for $\psi_{i\hat{j}}$ all heuristic values $\eta_{i\hat{j}k}$ for day \hat{j} must be updated.

Probabilistic rule. The probability of customer c_i being served on day d_j is given by

$$p(\phi_{ij} = 1) = \frac{\tau_{ij1}^\alpha \cdot \eta_{ij1}^\beta}{\tau_{ij0}^\alpha \cdot \eta_{ij0}^\beta + \tau_{ij1}^\alpha \cdot \eta_{ij1}^\beta} \quad (5)$$

where α and β are parameters that balance the importance of pheromone and heuristics in $p(\phi_{ij})$, respectively.

After the assignment Ψ has been made an additional local search algorithm can be applied (see Section 4). In order to determine the travel cost v_t a traveling salesman problem (TSP) [13] has to be solved on each day \hat{j} in order to determine the shortest round trip to all customers i with $\psi_{i\hat{j}} = 1$ and back. We use an ACO based approach that was proposed in [14]. Since we repeatedly have to solve the TSPs for each ant and each iteration we make the routing pheromone matrices $\tilde{\tau}_j$ for each day j globally visible and reuse it whenever an ant constructs a routing solution for that day. Algorithm 1 sketches the proposed algorithm design.

Algorithm 1: ACO algorithm for solving the IRP

```
Set for every pair  $(i, j)$ :  $\tau_{ij} = \tau_0$ ;  
Initialize TSP pheromones  $\tilde{\tau}_j$  for every day  $d_j$  as suggested in [14];  
for  $it = 1$  TO  $it_{max}$  do  
    for  $k = 1$  TO  $k_{max}$  do  
        Set  $C = \{(c_i, d_j) \mid \forall i, j\}$  ;  
        while  $C \neq \emptyset$  do  
            randomly choose  $(\tilde{i}, \tilde{j}) \in C$ ;  
            ant  $a_k$  chooses  $\psi_{\tilde{i}, \tilde{j}}$  with  $p$  from (5) using  $\eta_{\tilde{i}, \tilde{j}q}$ ;  
            update  $E$  with (3) and (4) respectively;  
            compute total storage costs  $v_s$  and backorder costs  $v_b$ ;  
            (optional: perform local search (see Section 4))  
        Find best ant  $a_{IB}$  of iteration  $k$  that minimizes the total cost;  
        Evaporate  $\tau$  using (1);  
        Update  $\tau$  using (2) with  $\Phi_{IB}$  found by  $a_{IB}$  and also by global best  $\Phi_{GB}$  ;  
        Update all TSP pheromones  $\tilde{\tau}_j$  with the according daily routes of  $\Phi_{IB}$ ;  
return Best solution
```

4 Local Search

The local search approach (LS) is a heuristics for improving any given solution [15]. For our problem we propose a LS algorithm that first tries to swap all nonzero entries of Ψ to zero and then vice versa. During the second phase each refill assignment is moved to a neighboring day if no refill is planned for that day yet. Each change is only applied if it reduces the total cost. The city and day are chosen in a random order. These two phases are repeated until no improvement was made during the last repetition. LS is a very fast heuristics thus it can be applied iteratively though its scope of finding optimal solutions is limited (see Section 6). The heuristics is sketched in Algorithm 2.

5 Genetic Algorithm

In this section we briefly introduce a genetic algorithm (GA) ([16] [17] [18]) for the IRP that is a modified version of the GA presented in [6]. Modifications were necessary because the vehicle capacity is infinite. Therefore, we ignore the parts that deal with constructing a valid solution in terms of the vehicle capacity but simply consider any generated solution feasible. We do not use the chromosomes to store the fill amount information, but compute the fill amounts as suggested in Section 3. In the chromosomes we only encode whether a customer is visited or not. Every possible assignment Ψ is valid (though maybe arbitrarily bad), therefore the GA is initialized with a random Ψ . We use the same bitwise mutation and two-point-crossover operations as suggested in [6]. Also the routing is done in the same way as suggested in [14] by Stützle and Hoos.

6 Computational results

Benchmark algorithms. In order to show the performance of the proposed algorithm we have carried out a comparison with a local search algorithm (LS) (see Section 4) and a

Algorithm 2: LS for improvement of city-day-assignment

Input: $\Psi = \{\psi_{ij} | \psi_{ij} \in \{\text{true}, \text{false}\}\}$

```
1 repeat
2   Set  $U = \{(i, j) | \psi_{ij} = \text{false}\}$ ;
3   while  $U \neq \emptyset$  do
4     randomly choose  $(\tilde{i}, \tilde{j}) \in U$ ;
5     if  $\text{cost}(\neg\psi_{\tilde{i}\tilde{j}}) < \text{cost}(\psi_{\tilde{i}\tilde{j}})$  then
6       |  $\psi_{\tilde{i}\tilde{j}} \leftarrow \text{true}$ ;
7       |  $U = U \setminus (\tilde{i}, \tilde{j})$ ;
8   Set  $V = \{(i, j) | \psi_{ij} = \text{true}\}$ ;
9   while  $V \neq \emptyset$  do
10    randomly choose  $(\tilde{i}, \tilde{j}) \in V$ ;
11    if  $\text{cost}(\neg\psi_{\tilde{i}\tilde{j}}) < \text{cost}(\psi_{\tilde{i}\tilde{j}})$  then
12      |  $\psi_{\tilde{i}\tilde{j}} \leftarrow \text{false}$ ;
13      |  $V = V \setminus (\tilde{i}, \tilde{j})$ ;
14   Set  $W = \{(i, j) | \psi_{ij} \neq \wedge \psi_{i,j-1}, j \geq 2\}$ ;
15   while  $W \neq \emptyset$  do
16     randomly choose  $(\tilde{i}, \tilde{j}) \in W$ ;
17     if  $\text{cost}(\neg\psi_{\tilde{i}\tilde{j}} \wedge \neg\psi_{\tilde{i}\tilde{j}-1}) < \text{cost}(\psi_{\tilde{i}\tilde{j}} \wedge \psi_{\tilde{i}\tilde{j}-1})$  then
18       |  $\psi_{\tilde{i}\tilde{j}} \leftarrow \neg\psi_{\tilde{i}\tilde{j}}$ ;  $\psi_{\tilde{i},\tilde{j}-1} \leftarrow \neg\psi_{\tilde{i},\tilde{j}-1}$ ;
19       |  $W = W \setminus (\tilde{i}, \tilde{j})$ ;
20
21 until cost did not improve ;
22 return improved Assignment  $\Psi$ 
```

genetic algorithm (GA) (see Section 5) as well as combinations of them:

- Local Search (LS)
- Genetic Algorithm (GA)
- Genetic Algorithm with local search (GA+LS)
- Ant Colony Optimization (ACO)
- Ant Colony Optimization with local search (ACO+LS)

Parameter setting. Due to an intensive study we figured out the optimal parameter set for the ACO algorithm. The values in brackets represent the set of tested values: Number of iterations $it_{max} = 300$, number of ants $a_{max} = 20$ (5, 10, 15, 20, 25), decay factor $\rho = 0.1$ (0.01, 0.05, 0.1, 0.2), initial pheromones $\tau_{ijk}^0 = 1$ (0.1, 1, 5, 10); minimum pheromone $\tau_{min} = 0.001$ (0.0001, 0.001, 0.01, 0.1), maximum pheromone $\tau_{max} = 100$ (10, 50, 100, 150), pheromone update constant $u = 7$ (1, 5, 7, 9, 11, 15), weights $\alpha = 1$ and $\beta = 2$ (1, 2, 3).

For the GA we used the relevant parameters presented in [6]: $gen_{max} = 240$, size of each generation $wp_{max} = 25$, number of elitist individuals $e = 10$, number of children

$wc_{max} = 50$, crossover probability $p_c = 0.1$ and mutation probability $p_m = 0.01$; both algorithms construct a total of 6000 solutions.

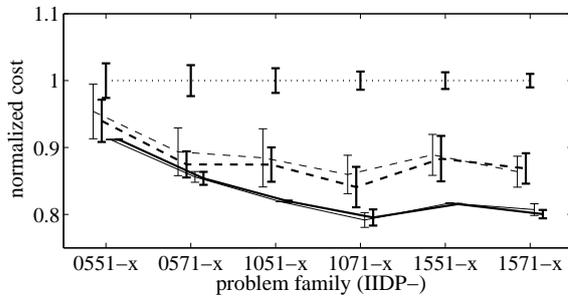
Test data. For this comparison we use the data set that was introduced in [19]. The customers in these test problems are spread over an area of 20×20 distance units. The travel cost per distance unit is 2. The number of customers are 5, 10 and 15 and the horizon is 5 and 7 days respectively. Capacity and backorder and storage costs vary for each customer within ranges additionally the demand is not constant. costs are also individual for each customer. Although these data sets are small they are perfectly suitable for our application and can be used to clearly show the different performances of the different approaches. Since our problem only contains one incapacitated vehicle, we ignore the given vehicle capacity and only take data sets generated for exactly one vehicle.

The naming convention of the test problems is as follows: They start with the letters ‘IIDP’, the first two numeric digits stand for the number of customers, the next one for the length of the planing period and the forth for the number of vehicles. The number after the dash is the instance number of that particular parameter set. Each instance is randomly generated. More detailed information about the generation process can be found in [19].

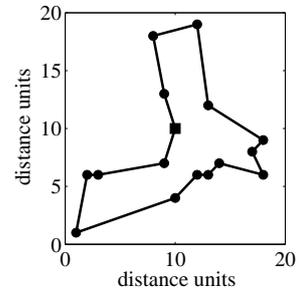
Table 1: Normalized average cost results. The standard deviation (S. D.) is $\times 10^{-2}$.

Problem	LS		GA		GA+LS		ACO		ACO+LS	
	Avg.	S. D.	Avg.	S. D.	Avg.	S. D.	Avg.	S. D.	Avg.	S. D.
IIDP-0551-1	1.00	(1.90)	0.99	(5.47)	0.96	(2.73)	0.94	(0.00)	0.94	(0.00)
IIDP-0551-2	1.00	(2.76)	0.93	(4.47)	0.93	(3.83)	0.89	(0.00)	0.89	(0.00)
IIDP-0551-3	1.00	(2.81)	0.94	(3.47)	0.94	(5.00)	0.90	(0.00)	0.90	(0.00)
IIDP-0551-4	1.00	(2.59)	0.90	(3.95)	0.88	(0.17)	0.88	(0.19)	0.88	(0.17)
IIDP-0551-5	1.00	(2.84)	0.99	(3.02)	0.97	(4.01)	0.92	(0.00)	0.92	(0.00)
IIDP-0571-1	1.00	(2.22)	0.86	(3.16)	0.84	(0.84)	0.82	(0.37)	0.83	(0.46)
IIDP-0571-2	1.00	(2.46)	0.89	(6.83)	0.90	(3.56)	0.85	(2.81)	0.83	(1.69)
IIDP-0571-3	1.00	(1.56)	0.88	(2.15)	0.86	(1.96)	0.84	(0.00)	0.84	(1.54)
IIDP-0571-4	1.00	(2.54)	0.90	(1.82)	0.88	(2.26)	0.88	(0.40)	0.88	(0.29)
IIDP-0571-5	1.00	(2.77)	0.91	(3.89)	0.87	(1.15)	0.87	(0.44)	0.87	(0.87)
IIDP-1051-1	1.00	(1.42)	0.83	(3.44)	0.81	(0.44)	0.80	(0.00)	0.80	(0.49)
IIDP-1051-2	1.00	(1.90)	0.86	(6.03)	0.85	(5.97)	0.79	(0.00)	0.79	(0.00)
IIDP-1051-3	1.00	(1.05)	0.94	(3.79)	0.91	(2.21)	0.84	(0.00)	0.84	(0.00)
IIDP-1051-4	1.00	(2.68)	0.92	(4.40)	0.93	(1.72)	0.85	(0.00)	0.85	(0.00)
IIDP-1051-5	1.00	(2.16)	0.84	(4.10)	0.84	(2.46)	0.80	(0.07)	0.80	(0.00)
IIDP-1071-1	1.00	(0.93)	0.86	(3.61)	0.82	(3.90)	0.77	(0.79)	0.78	(1.23)
IIDP-1071-2	1.00	(2.17)	0.84	(3.05)	0.86	(3.78)	0.78	(0.31)	0.78	(0.01)
IIDP-1071-3	1.00	(1.32)	0.86	(3.77)	0.84	(2.54)	0.82	(1.73)	0.83	(1.70)
IIDP-1071-4	1.00	(1.27)	0.83	(2.22)	0.81	(1.33)	0.77	(1.37)	0.76	(0.80)
IIDP-1071-5	1.00	(1.09)	0.87	(1.77)	0.85	(3.53)	0.80	(1.31)	0.81	(2.26)
IIDP-1551-1	1.00	(0.86)	0.88	(2.09)	0.87	(3.15)	0.80	(0.00)	0.80	(0.00)
IIDP-1551-2	1.00	(0.84)	0.88	(1.02)	0.89	(1.76)	0.86	(0.14)	0.85	(0.00)
IIDP-1551-3	1.00	(0.75)	0.93	(3.23)	0.92	(4.28)	0.86	(0.05)	0.86	(0.00)
IIDP-1551-4	1.00	(2.20)	0.84	(5.21)	0.83	(4.74)	0.74	(0.00)	0.74	(0.00)
IIDP-1551-5	1.00	(1.55)	0.88	(3.72)	0.88	(3.01)	0.81	(0.00)	0.81	(0.00)
IIDP-1571-1	1.00	(1.24)	0.89	(0.87)	0.89	(1.23)	0.84	(0.41)	0.84	(0.32)
IIDP-1571-2	1.00	(1.16)	0.86	(3.08)	0.87	(2.48)	0.82	(0.83)	0.81	(1.23)
IIDP-1571-3	1.00	(0.85)	0.84	(1.46)	0.82	(2.16)	0.78	(1.10)	0.77	(0.30)
IIDP-1571-4	1.00	(1.22)	0.86	(4.27)	0.88	(2.30)	0.77	(1.01)	0.77	(1.08)
IIDP-1571-5	1.00	(0.57)	0.86	(1.80)	0.86	(3.19)	0.79	(1.00)	0.78	(0.18)
total avg.	1.00	(1.72)	0.89	(3.37)	0.88	(2.72)	0.83	(0.48)	0.83	(0.49)

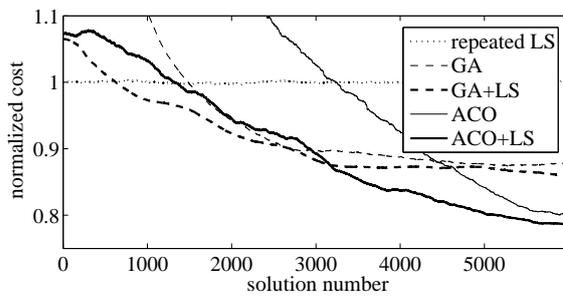
For each problem and algorithm the computation was repeated 10 times in order to find out the average behavior (*avg.*) of the algorithms. In Table 1 we present the results of all different approaches for the given data sets which show a clear difference in the performance of the analyzed algorithms. The result values for each problem are normalized by the average LS solution and the according standard deviation (*S. D.*) is given in brackets. The last line in the table is the average of all test instances.



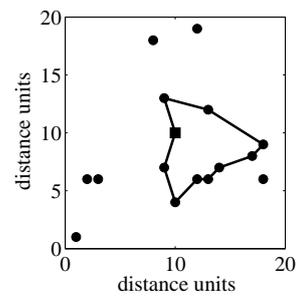
(a) Average solutions for for each solution family, x stands for instances 1-5



(c) Route for day 1 of problem IIDP-1571-2



(b) Typical convergence for problem IIDP-1071-1



(d) Route for day 3 of problem IIDP-1571-2

Figure 1: (a) Average normalized cost (lines) and standard deviation (vertical intervals) for each problem family. The legend of (b) also applies here. (b) Change of the solution quality with respect to the number of constructed solutions. The lines are smoothed using the moving average of 500 solutions. (c) and (d) show parts of a possible solution for problem IIDP-1571-2.

The LS is only close to the best found solutions in the smallest problem instances. The GA outperforms the LS by about 12% and performs comparable to ACO on some data sets, however the average solution quality is worse. In combination with the GA the local search does not significantly improve the solution value, only the standard deviation is reduced by 0.65. The ACO approach computes the best solutions for all test problems and there is no significant difference between the results of ACO and ACO+LS. On average, ACO is about 5% better than GA. In Figure 1(a) the average normalized performance of each algorithm is shown for each of the six problem families.

Figure 1(b) shows a typical example of how the normalized solutions for the different approaches evolve in time during the run of the algorithm. Repeated LS with random initialization of course does hardly show any change (since it is normalized with the best

result of LS), whereas the GA and the ACO are both converging. When used together with ACO and GA, LS improves the solution qualities for the earlier solutions but it does not diminish the iteration number when the optimum was reached. Although the GA based algorithms result in better solutions at the beginning the ACO based approach returns the best solutions at the later stages as it does not get stuck in local optima so often. Figure 1(c) and (d) show day 1 and day 3 of a solution for problem IIDP-1571-2 (for day 2 no refilling was scheduled).

7 Conclusions and Future Work

We applied the ACO paradigm to the ATM replenishment problem which can be reduced to an inventory routing problem. Our proposed algorithm splits the problem into the two subproblems of determining how much to deliver to which customer first and then finding the optimal route. In this preliminary study both problems are solved using different adaptations of the ACO meta-heuristics. The suggested approach outperforms other GA based approaches as well as local search. The necessary withdrawal amounts for each ATM are considered to be given, however, in our real life application we forecast the demand using recurrent neural networks.

In future work we want to extend the problem by introducing several depots with their respective customers (ATMs) as well as a central site where the vehicles are based. Doing so we also consider the inter-depot empty drives as well as from the base to the depots.

References

- [1] Moin, N. H.; Salhi, S.: Inventory routing problems: A logistical overview. *Journal of the Operational Research Society* 58 (2007) 9, S. 1185–1194.
- [2] Chien, T. W.; Balakrishnan, A.; Wong, R. T.: An integrated inventory allocation and vehicle routing problem. *International Journal of Transport Science* 23 (1989), S. 67–76.
- [3] Dror, M.; Trudeau, P.: Cash flow optimization in delivery scheduling. *European journal of operational research* 88 (1996) 3, S. 504–515.
- [4] Viswanathan, S.; Mathur, K.: Integrating routing and inventory decisions in one warehouse multiproduct distribution systems. *Management Science* 43 (1997), S. 294–312.
- [5] Campbell, A. M.; Savelsbergh, M. W. P.: A Decomposition Approach for the Inventory-Routing Problem. *Transportation Science* 38 (2004) 4, S. 488–502.
- [6] Abdelmaguid, T. F.; Dessouky, M. M.: A genetic algorithm approach to the integrated inventory-distribution problem. *International Journal of Production Research* 44 (2006) 21, S. 4445–4464.
- [7] Dorigo, M.; Stützle, T.: *Ant Colony Optimization*. MIT Press. 2004.

- [8] Silva, C. A.; Sousa, J. M.; Runkler, T. A.; Sá da Costa, J.: Distributed Optimization in Supply-Chain Management using Ant Colony Optimization. *International Journal of Systems Science* 37 (2006) 8, S. 503–512.
- [9] Hartmann, S. A.; Runkler, T. A.: Online Optimization of a Color Sorting Assembly Buffer Using Ant Colony Optimization. *Proceedings of the Operations Research 2007 conference* (2007), S. 415–420.
- [10] Pinto, P.; Runkler, T. A.; Sousa, J. M. C.: Ant Colony Optimization and its application to Regular and Dynamic MAX-SAT problems. *Advances in biologically inspired information systems: Models, methods, and tools* (2007), S. 283–302.
- [11] Pinto, P.; Nägele, A.; Runkler, T. A.; Dejori, M.; Sousa, J. M. C.: Structure learning of Bayesian Models of genetic Networks by a Local Discovery Ant Colony Algorithm. *IEEE Congress on Evolutionary Computation, Hong Kong, June 2008, accepted for publication* .
- [12] Stützle, T.; Hoss, H. H.: $MAX-MIN$ Ant System. *Future Generation Computer Systems* 16 (2000) 8, S. 889–914.
- [13] Applegate, D. L.; Bixby, R. E.; Chvátal, V.; Cook, W. J.: *The Traveling Salesman Problem: A Computational Study*. Princeton University Press. 2007.
- [14] Stützle, T.; Hoos, H. H.: The MAX-MIN Ant System and Loach Search for the Travelling Salesman Problem. *Proceedings of the IEEE International Conference on Evolutionary Computation* (1997), S. 309–314.
- [15] Aarts, E. H. L.; Lenstra, J. K.: *Local Search in Combinatorial Optimization*. Princeton University Press. 2003.
- [16] Goldberg, D. E.: *The Design of Innovation (Genetic Algorithms and Evolutionary Computation)*. Kluwer Academic Publishers. 2002.
- [17] Baeck, T.: *Evolutionary Algorithms in Theory and Practice*. Oxford University Press. 1996.
- [18] Brady, R. M.: Optimization Strategies Gleaned from Biological Evolution. *Journal of Nature* (1985) 317, S. 804–806.
- [19] Abdelmaguid, T. F.: *Heuristic approaches for the integrated inventory distribution problem*. Dissertation, University of Southern California. 2004.

Ein selbstanpassender und prozessspezifischer Ansatz in der Sensordiagnose

M. Grube¹, R. Mikut¹, Th. Alber², M. Jagiella², G. Bretthauer¹

¹ Forschungszentrum Karlsruhe, Institut für Angewandte Informatik

Postfach 3640, D-76021 Karlsruhe

Tel.: (07247) 82-5773

Fax: (07247) 82-2602

E-Mail: markus.grube@iai.fzk.de

² Endress+Hauser Conducta GmbH

Dieselstr. 24, D-70839 Gerlingen

Tel.: (07156) 209-186

Fax: (07156) 209-152

E-Mail: thomas.alber@conducta.endress.com

Insbesondere in Prozessumgebungen bestimmt der tatsächliche Zustand der eingesetzten Sensoren die Qualität der Produkte und die Sicherheit des Personals und der Umwelt. Durch die Integration dieser Zustandsinformation in ein Plant-Asset-Management-System kann die Effizienz gesteigert werden, was zu einer Reduktion der Produktionskosten führt. Aus diesem Grund wurde der Bedarf nach Diagnosesystemen, die den Zustand eines Feldgerätes bestimmen, in die Roadmap für Prozesssensoren der NAMUR aufgenommen [1]. Bislang wurden Systeme mit der Fähigkeit, den Zustand eines Sensors zu bestimmen, mit hohem Aufwand entwickelt. Deshalb besteht großer Bedarf nach standardisierten Systemen und Algorithmen in der Diagnose. Diese können besonders dort wichtig sein, wo das Wissen über Alterungseffekte für einen gewöhnlichen modellbasierten Ansatz nicht ausreicht oder andere Konzepte nicht eingesetzt werden können. Oft werden jedoch Alterungsheuristiken verwendet, die auf den gemessenen Prozessvariablen basieren, aber lediglich eine Vereinfachung für viele denkbare Prozesse darstellen.

In diesem Beitrag wird eine neue Methode für die Diagnose elektrochemischer Sensoren in Prozessumgebungen vorgeschlagen. Zuerst werden Richtlinien gegeben, die helfen, die Probleme in der Sensordiagnose zu standardisieren. Weiter wird gezeigt, wie Prozesskarten direkt aus statischen Prozessdaten gewonnen werden. Danach wird die Auswertung und Klassifikation der Daten dynamischer Prozesse durch Expertenwissen, welches in stochastischen Modellen repräsentiert ist, vorgenommen. Als Ergebnis werden Kalibrierintervalle für die Sensoren abgeleitet und es wird gezeigt, wie Parameter, die Alterung repräsentieren, automatisch ausgewählt werden können. Schließlich wird eine Bewertung des Zustandes des Sensors während der Kalibrierung vorgenommen. Damit kann die verbleibende Lebensdauer abgeschätzt werden. Es werden Algorithmen gezeigt, mit denen dieses System realisiert werden kann und in Hinblick auf ihre Effizienz verglichen.

Literatur

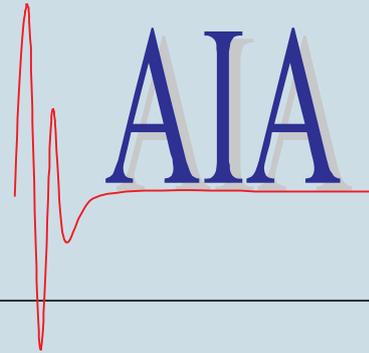
- [1] NAMUR; GMA: *Abschlussbericht Technologie-Roadmap: Prozess-Sensoren 2005 - 2015*. 2005

**Bereits veröffentlicht wurden in der Schriftenreihe des
Instituts für Angewandte Informatik / Automatisierungstechnik im
Universitätsverlag Karlsruhe:**

- Nr. 1: BECK, S.: Ein Konzept zur automatischen Lösung von Entscheidungsproblemen bei Unsicherheit mittels der Theorie der unscharfen Mengen und der Evidenztheorie, 2005
- Nr. 2: MARTIN, J.: Ein Beitrag zur Integration von Sensoren in eine anthropomorphe künstliche Hand mit flexiblen Fluidaktoren, 2004
- Nr. 3: TRAICHEL, A.: Neue Verfahren zur Modellierung nichtlinearer thermodynamischer Prozesse in einem Druckbehälter mit siedendem Wasser-Dampf Gemisch bei negativen Drucktransienten, 2005
- Nr. 4: LOOSE, T.: Konzept für eine modellgestützte Diagnostik mittels Data Mining am Beispiel der Bewegungsanalyse, 2004
- Nr. 5: MATTHES, J.: Eine neue Methode zur Quellenlokalisierung auf der Basis räumlich verteilter, punktwiser Konzentrationsmessungen, 2004
- Nr. 6: MIKUT, R.; REISCHL, M.: Proceedings – 14. Workshop Fuzzy-Systeme und Computational Intelligence: Dortmund, 10. - 12. November 2004, 2004
- Nr. 7: ZIPSER, S.: Beitrag zur modellbasierten Regelung von Verbrennungsprozessen, 2004
- Nr. 8: STADLER, A.: Ein Beitrag zur Ableitung regelbasierter Modelle aus Zeitreihen, 2005
- Nr. 9: MIKUT, R.; REISCHL, M.: Proceedings – 15. Workshop Computational Intelligence: Dortmund, 16. - 18. November 2005, 2005
- Nr. 10: BÄR, M.: μ FEMOS – Mikro-Fertigungstechniken für hybride mikrooptische Sensoren, 2005
- Nr. 11: SCHAUDEL, F.: Entropie- und Störungssensitivität als neues Kriterium zum Vergleich verschiedener Entscheidungskalküle, 2006
- Nr. 12: SCHABLOWSKI-TRAUTMANN, M.: Konzept zur Analyse der Lokomotion auf dem Laufband bei inkompletter Querschnittlähmung mit Verfahren der nichtlinearen Dynamik, 2006
- Nr. 13: REISCHL, M.: Ein Verfahren zum automatischen Entwurf von Mensch-Maschine-Schnittstellen am Beispiel myoelektrischer Handprothesen, 2006
- Nr. 14: KOKER, T.: Konzeption und Realisierung einer neuen Prozesskette zur Integration von Kohlenstoff-Nanoröhren über Handhabung in technische Anwendungen, 2007
- Nr. 15: MIKUT, R.; REISCHL, M.: Proceedings – 16. Workshop Computational Intelligence: Dortmund, 29. November - 1. Dezember 2006
- Nr. 16: LI, S.: Entwicklung eines Verfahrens zur Automatisierung der CAD/CAM-Kette in der Einzelfertigung am Beispiel von Mauerwerksteinen, 2007

- Nr. 17: BERGEMANN, M.: Neues mechatronisches System für die Wiederherstellung der Akkommodationsfähigkeit des menschlichen Auges, 2007
- Nr. 18: HEINTZ, R.: Neues Verfahren zur invarianten Objekterkennung und -lokalisierung auf der Basis lokaler Merkmale, 2007
- Nr. 19: RUCHTER, M.: A New Concept for Mobile Environmental Education, 2007
- Nr. 20: MIKUT, R.; REISCHL, M.: Proceedings – 17. Workshop Computational Intelligence: Dortmund, 5. - 7. Dezember 2007
- Nr. 21: LEHMANN, A.: Neues Konzept zur Planung, Ausführung und Überwachung von Roboterarbeiten mit hierarchischen Petri-Netzen, 2008
- Nr. 22: MIKUT, R.: Data Mining in der Medizin und Medizintechnik, 2008
- Nr. 23: KLINK, S.: Neues System zur Erfassung des Akkommodationsbedarfs im menschlichen Auge, 2008

Die Schriften sind als PDF frei verfügbar, eine Nachbestellung der Printversion ist möglich.
Nähere Informationen unter www.uvka.de.



Dieser Tagungsband enthält die Beiträge des 18. 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 3. - 5. Dezember 2008 im Haus Bommerholz bei Dortmund stattfindet.

Der GMA-Fachausschuss 5.14 „Computational Intelligence“ entstand 2008 aus den bisherigen Fachausschüssen „Neuronale Netze und Evolutionäre Algorithmen“ (FA 5.21) sowie „Fuzzy Control“ (FA 5.22). Der Workshop steht in der Tradition der bisherigen Fuzzy-Workshops, hat aber seinen Fokus in den letzten Jahren schrittweise erweitert.

Die Schwerpunkte sind Methoden, Anwendungen und Tools für

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

sowie der Methodenvergleich anhand von industriellen und Benchmark-Problemen.

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

ISSN: 1614-5267

ISBN: 978-3-86644-282-5