

SOFT COMPUTING IN MODELLBILDUNG UND SIMULATION

DATAENGINE - ECANSE - LABVIEW – MATLAB

Proceedings Seminarbericht S51

F. Breitenecker
M. Lingl
Herausgeber

ARGESIM Report No. 12

SOFT COMPUTING IN MODELLBILDUNG UND SIMULATION

DATAENGINE - ECANSE - LABVIEW – MATLAB

Proceedings Seminarbericht S51

F. Breitenacker
M. Lingl
Herausgeber

ARGESIM Report No. 12

Reihe „ARGESIM Reports“
Reihenherausgeber: F. Breitenecker

ARGESIM Report No. 12:
Soft Computing in Modellbildung und Simulation
Proceedings Seminarbericht S51
Herausgeber: F. Breitenecker, M. Lingl

Redaktion des Bandes: Shabnam Rahmi, Klaus Seits

© 1998 ARGESIM
ARGE Simulation News (ARGESIM)
Technische Universität Wien
Wiedner Hauptstraße 8-10
A-1040 Wien, Österreich
Tel.: +43-1-58801-11452
Fax: +43-1-58801-11499
Email: seminar@argesim.org
WWW: <http://www.argesim.org>

SOFT COMPUTING IN MODELLBILDUNG UND SIMULATION

DataEngine - ECANSE - LabVIEW – MATLAB

Inhaltsverzeichnis

Inhaltsverzeichnis	1
Über dieses Seminar	2
Impressum.....	2
Programm	3
Seminare zu Modellbildung und Simulation.....	5
 Vortragsunterlagen.....	9
Anwendung von Soft Computing im Bereich der Biomedizinischen Technik	
B. Bracio, ARC Seibersdorf / TU Clausthal	11
ECANSE - Soft Computing Studio - Einführung, Features, Anwendungen	
M. Sturm, Siemens Österreich	
Modellierung von Finanzmärkten mit ECANSE	
M. Kühner, Siemens Österreich	15
Energy Prognose - Simulation in ECANSE	
A. Mladenka, Siemens Österreich	
Lawinenprognose mit ECANSE	
K. Kleemayr, Inst. f. Wildbach und Lawinenschutz, Boku Wien	27
Fuzzy Entscheidungen in diskreter Modellbildung und Simulation	
M. Lingl, Abt. Simulationstechnik, TU Wien	33
DataEngine - Beschreibung von Fuzzy Systems, Neuralen Netzen, Entscheidungsbäumen	
K. Lieven, MIT, Aachen	51
DataEngine - Einbindung in Simulationsumgebungen	
K. Lieven, MIT, Aachen	
Soft Computing in LabVIEW	
E. Hajrizi, Inst. f. Fertigungstechnik, Abt. Intelligent Manufacturing Systems, TU Wien	99
MATLAB/SIMULINK - Neural Nets Toolbox - Anwendungen in der Bildverarbeitung	
H. Bischof, Abt. Mustererkennung und Bildverarbeitung, TU Wien	117
MATLAB/SIMULINK - Fuzzy Control Toolbox - Elektrodenregelung eines Drehstrom - Lichtbogenofens	
G. Görtler, Inst. für Maschinen- u. Prozeßautomatisierung, TU Wien	137
ARGESIM C9 - Fuzzy Control - Lösungen und Auswertungen	
F. Breitenacker, M. Lingl, J. Scheikl, ARGESIM, TU Wien	141
 Anhang.....	157
Informationen zu ECANSE	
Teilnehmerliste	
Hinweis auf folgende Seminare	

Seminare zu Modellbildung und Simulation

Über dieses Seminar

S51

SOFT COMPUTING IN MODELLBILDUNG UND SIMULATION
DATAENGINE - ECANSE - LabVIEW - MATLAB
Do, 3. Dez. 1998

Auf vielen Modellbildungs- und Simulationstagungen ist derzeit die Forderung nach einem „Paradigmenwechsel“ zu hören, darunter u.a. der Ruf nach Einbindung des Soft Computing in die Simulationstechnik. Gemeint ist dabei die Erweiterung der klassischen Modellbildungsgrundlagen (etwa Differentialgleichungen für kontinuierliche Anwendungen, oder Ereignissysteme für diskrete Simulationen) um Methoden des Soft Computing, hauptsächlich Fuzzy Systems, Neural Nets und Symbolic Computation.

Dieser Entwicklung trägt das gegenwärtige Seminar S51, "Soft Computing in Modellbildung und Simulation - DataEngine - ECANSE - LabVIEW - MATLAB", Rechnung. Die Annäherung an das Thema erfolgt zunächst über die Beschreibung und Vorführung einer Integration von Fuzzy Systems und Neural Nets in die kontinuierliche und diskrete Modellbildung an einigen Anwendungen. Es werden dabei über das simple Fuzzy Control hinausgehende Implementierungen vorgeführt und mit ECANSE, DataEngine und LabVIEW neben MATLAB/SIMULINK drei weitere dazu geeignete Software Tools vorgestellt.

ECANSE ("Environment for Computer Aided Neural Software Engineering") erlaubt Modellbildung und Simulation mit Hilfe von neuronalen Netzen in verschiedensten Anwendungen mit unterschiedlichen Zugängen. DataEngine kann Fuzzy Systeme (regelbasiert und clustered), neuronale Netze (MLP, Kohonen und Fuzzy Kohonen) und auch Entscheidungsbaumverfahren modellieren. Die dabei entstehenden DataEngine Modelle können in andere Softwareumgebungen, wie eben auch Simulatoren eingebunden werden. LabVIEW bietet dem Anwender die Möglichkeit, von Datenerfassung über eine Vorverarbeitung und dem Einsatz von Fuzzy Methoden bis hin zur Ergebnispräsentation in einer einzigen Entwicklungsumgebung zu arbeiten, die, da auf C- Basis programmiert, vollständig unabhängig von der benutzten Plattform ist.

Herausgeber:

ARGESIM
c/o EDV-Zentrum TU Wien
Wiedner HStr. 8-10
A-1040 Wien

Ch. Beistciner, Tel: +43-(0)1-58801-42021, oder
Prof. Dr. F. Breitenecker, Tel: +43-(0)1-58801-11452 bzw. 11456,
Fax: +43-(0)1-58801-42089
Email: seminar@argesim.org
WWW: <http://www.argesim.org/seminare>

Seminarorganisation und Zusammenstellung der Unterlagen:

Prof Dr. Felix Breitenecker, Tel: +43-(0)1-58801-11452, Email: felix.breitenecker@tuwien.ac.at
Shabnam Rahmi, Tel: +43-(0)1-58801-11456, Email: e9425764@fhma.tnwien.ac.at
Klaus Seits, Tel: +43-(0)1-58801-11456, Email: kseits@osiris.tuwien.ac.at

Seminare zu Modellbildung und Simulation

SOFT COMPUTING IN MODELLBILDUNG UND SIMULATION

DataEngine - ECANSE – LabVIEW - MATLAB

Donnerstag, 3. Dez. 1998

**Technische Universität Wien, Gußhausstraße 27 - 29, A-1040 Wien
Kontaktraum, 6. Stock**

Programm

- | | |
|---|--|
| 0900 Begrüßung, Vorstellung der Seminarreihe
F. Breitenacker, ARGESIM, A. Blauensteiner, ZID, TU
Wien | 1400 DataEngine: Einbindung in
Simulationsumgebungen
K. Lieven, MIT, Aachen |
| 0915 Soft Computing: Grundlagen - Anwendungen in
der Biomedizinischen Technik
B. Bracio, ARC Seibersdorf / TU Clausthal | 1430 Soft Computing in LabVIEW
E. Hajrizi, Inst. f. Fertigungstechnik, TU Wien |
| 1000 ECANSE – Soft Computing Studio - Einführung,
Features, Anwendungen
M. Sturm, Siemens Österreich | 1500 Kaffeepause |
| 1040 Modellierung von Finanzmärkten mit ECANSE
M. Kühner, Siemens Österreich | 1515 MATLAB/SIMULINK: Neural Net Toolbox –
Anwendungen in der Bildverarbeitung
H. Bischof, Abt. Mustererkennung u. Bildverarbeitung,
TU Wien |
| 1110 Kaffeepause | 1600 MATLAB/SIMULINK: Fuzzy Control Toolbox –
Elektrodenregelung eines Drehstrom -
Lichtbogenofens
G. Görtler, Inst. für Maschinen- u.
Prozeßautomatisierung, TU Wien |
| 1120 Energy Prognose - Simulation in ECANSE
A. Mladenka, Siemens Österreich | 1630 ARGESIM Comparison C9 – Fuzzy Control
- Lösungen und Auswertungen
F. Breitenacker, M. Lingl, J. Scheidl, ARGESIM, TU Wien |
| 1145 Lawinenprognose mit ECANSE
K. Kleernayr, Inst. f. Wildbach- und Lawinenschutz,
Boku Wien | 1715 Abschlußdiskussion
F. Breitenacker, M. Sturm, K. Lieven |
| 1215 Mittagspause | |
| 1245 Fuzzy Entscheidungen in diskreter Modellbildung
und Simulation
M. Lingl, Abt. Simulationstechnik, TU Wien | |
| 1315 DataEngine - Beschreibung von Fuzzy Systems,
Neuralen Netzen, Entscheidungsbäumen
K. Lieven, MIT, Aachen | |

Das Seminar wird von Firma MIT Management Intelligenter Technologien, Aachen, Firma Siemens Österreich, Wien, Firma Scientific, Aachen und ARGESIM gesponsert. Die Teilnahme ist kostenlos.

Seminare zu Modellbildung und Simulation

Entwicklung

Auf Initiation von I. Husinsky (Abt. „Institutsunterstützung“ des EDV-Zentrums (**IU-EDVZ**) der TU Wien) und F. Breitenecker (Abt. „Simulationstechnik und Regelungsmathematik“ (**SIMTECH**) der TU Wien) wurde Ende 1991 die Seminarreihe **Seminare aus Simulation** ins Leben gerufen, die zunächst als Unterstützung für Anwender verschiedener Simulationssoftware an der TU Wien gedacht war.

Aufgrund des Erfolges und des sichtlichen Bedarfes entwickelte sich die Reihe zu den **Seminaren zu Modellbildung und Simulation** weiter, die heute allen Interessenten, freilich auch jenen von außerhalb des Universitätsbetriebes, offenstehen. Dabei erweiterte sich auch der Themenkreis von einer reinen Vorstellung und Besprechung einzelner Simulationssoftwaretools hin zu allgemeinen Themen in Modellbildung und Simulation und zu Engineering Software.

Heute werden die Seminare in einer Zusammenarbeit der Abteilung „Simulationstechnik und Regelungsmathematik“ (**SIMTECH**) der TU Wien, der Abteilung „Institutsunterstützung“ des EDV-Zentrums (**IU-EDVZ**) und der „Arbeitsgemeinschaft Simulation News“ (**ARGESIM**) zu verschiedensten Themen in Modellbildung und Simulation und verwandten Gebieten organisiert und durchgeführt. Seit Ende 1997 finden einzelne Seminare auch unter Teilnahme der Fachgruppe „Simulationssoftware und Hardware“ der „Arbeitsgemeinschaft Simulation“ (**ASIM**) statt.

Die halb- oder ganztägigen Seminare werden teils von Firmen gesponsert, teils über Simulationsprojekte finanziert (Projekte der **ARGESIM**).

Ziel und Inhalt der Seminare

Die Intention ist, in Theorie und Praxis sowohl verschiedene Lösungsansätze in einzelnen Problemkreisen vorzustellen, dabei den Zusammenhang von Modellbildung und Simulation mit anderen Bereichen, wie z.B. der Symbolic Computation, dem Einfluß objektorientierter Sichtweisen, etc. zu diskutieren, wie auch die Anwendung spezieller Softwaretools näherzubringen, über deren Einsatzmöglichkeiten zu informieren und Erfahrungen auszutauschen. Pausen mit bereitgestelltem Buffet bzw. Kaffee fördern die Kommunikation und den Erfahrungsaustausch und Diskussionen zwischen den Seminarteilnehmern.

Konzentriert sich ein Seminar auf eine bestimmte Software, so werden neben Anwendervorträgen auch Informationen über Stand und Weiterentwicklung des betreffenden Simulators vermittelt. Oftmals bieten sich dabei Mitarbeiter der Herstellerfirmen bzw. des jeweiligen Distributors zu ebensolchen Vorträgen an. Zudem erfolgen Softwarepräsentationen und Vorführungen am Rechner.

Für Seminare mit Grundlagenthemen bzw. mit Übersichtscharakter gelingt es oft, bekannte Simulationsfachleute zu Vorträgen einzuladen.

Für das Wintersemester 1998/99 sind vier Seminare vorgesehen. Schwerpunkte werden dabei verstärkt auf die Anwendung verschiedenster Methodologien für spezielle Problembereiche in Ansätzen der Modellbildung und Simulation als rein in der Besprechung verschiedenster Merkmale einzelner Softwareprodukte gesetzt. Damit soll ein Trend vorgezeichnet werden, der sich in Zukunft verstärkt fortsetzen soll.

Unterlagen

Um die schriftlichen Unterlagen, die zu Beginn des Seminars aufgelegt werden, in ausreichender Zahl bereitstellen und diesen schon eine Teilnehmerliste, in der die Kontaktadressen der einzelnen Teilnehmer angeführt sind, beizufügen zu können und sonstige organisatorische Belange der Seminausrichtung zu erleichtern, werden die Teilnehmer nach Möglichkeit um eine vorherige Anmeldung gebeten.

Soweit diese von den einzelnen Vortragenden zur Vervielfältigung zur Verfügung gestellt wurden, erhält so jeder angemeldete Teilnehmer schriftliche Unterlagen zu den Vorträgen eines Seminars. Diese Unterlagen sind in Form von **Seminarberichten** zusammengefaßt und durch das Sponsoring durch oben erwähnte Organisationen kostenlos.

Bei größeren Seminaren werden die Unterlagen zu Tagungsunterlagen zusammengefaßt, die in der Reihe der **ARGESIM Reports** mit entsprechenden ISBN-Nummern erscheinen. Die **Seminarberichte** zu den Vorträgen können gegen einen Unkostenbeitrag von ATS 50,- (DM 7,-) pro Exemplar und mit einem Versandkostenpauschalpreis von ATS 20,- (DM 3,-) auch nachträglich über Internet angefordert werden. Die zugehörige Webseite mit Bestellmöglichkeit findet sich als "Liste der bisherigen Seminare" unter <http://www.argesim.org/seminare>.

Bisherige Seminare

Bisher wurden 49 Seminare in der Seminarreihe Modellbildung und Simulation abgehalten. Drei dieser Seminare fanden in Verbindung mit einem COMETT - Kurs in englischer Sprache im Rahmen eines EU Projektes statt.

Bisher wurden Seminare zu den folgenden Themen angeboten:

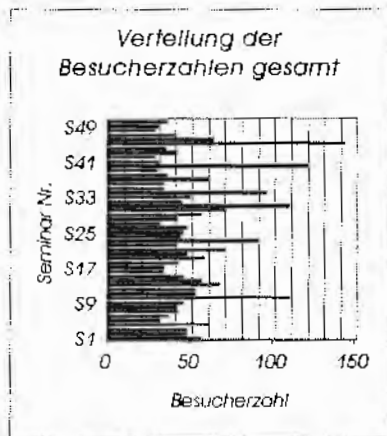
S01	1991-04-23	ACSL - kontinuierliche Simulation
S02	1991-06-04	CTRL_C, XANALOG - kontinuierliche Simulation
S03	1991-10-22	SIMUL_R - kontinuierliche Simulation
S04	1992-05-05	ACSL - kontinuierliche Simulation
S05	1992-05-06	MicroSaint - diskrete Simulation
S06	1992-06-17	Objektorientierte Modellbeschreibung und qualitative Simulation (Gast: F. Cellier, University of Arizona)
S07	1992-07-01	Diskrete Simulation und Analyse (Gast: D. Kelton, University of Minnesota)
S08	1992-10-23	GPSS/H - diskrete Simulation (Gast: T. Schriber, University of Michigan)
S09	1992-12-10	SIMPLE - objektorientierte diskrete Simulation
S10	1993-02-02	MATLAB und SIMULINK
S11	1993-03-25	Modellbildung mit Bondgraphen (Gast: D. Karnopp, University of California)
S12	1993-05-24	MicroSaint - diskrete Simulation
S13	1993-06-22	ACSL - kontinuierliche Simulation
S14	1993-10-21	XANALOG, SIMNON - kontinuierliche Simulation
S15	1993-10-22	GPSS/H - ds (Gast: T. Schriber, University of Michigan)
S16	1993-11-11	IDAS - kontinuierliche Simulation
S17	1993-12-07	SIMPLE++ - objektorientierte diskrete Simulation
S18	1993-12-14	Petrinetze - D_SIM für diskrete Simulation (Gast: R. Hohmann, Magdeburg)
S19	1994-02-04	Modellbildung und Simulation in der Lehre
S20	1994-03-14	GPSS/H und Proof Animation (Gast: T. Schriber, University of Michigan)
S21	1994-04-13	ACSL -, Kontinuierliche Simulation
S22	1994-05-10	SIMUL_R - kontinuierliche Simulation
S23	1994-11-22	MATLAB/SIMULINK
S24	1994-12-14	SIMPLE++ - objektorientierte diskrete Simulation
S25	1995-01-31	Parallele Simulation - der parallele Simulator mosis
S26	1995-03-28	ACSL - kontinuierliche Simulation
S27	1995-03-29	MicroSaint - diskrete Simulation
S28	1995-06-13	COMETT Course Simulation and Automatisisation Part 1: Introduction into Discrete Simulation
S29	1994-06-28	COMETT Course Simulation and Automatisisation Part 2: Simulation and Automatisisation with Object-Oriented Tools
S30	1995-09-07/8	COMETT Course Simulation and Automatisisation Part 3: Fuzzy Systems and Control
S31	1995-09-13	Simulation in der Didaktik
S32	1995-12-18	Diskrete Simulation - Moderne Grundlagen und ausgewählte Anwendungen

S33	1996-01-23	Symbolic Computation - Entwicklung und Anwendungen (Gast: B. Buchberger, RISC Linz)
S34	1996-03-08	MATLAB und SIMULINK - Entwicklung und Anwendungen
S35	1996-03-19	Parallele Simulation und Simulatorkopplung
S36	1996-05-09	Diskrete Simulation - Einführung in GPSS/H
S37	1996-11-26	Simulation mit ACSL
S38	1996-12-19	Diskrete Simulation - Verkehrssimulation - SIMPLE++
S39	1997-02-06	Modellbildung und Simulation in der Lehre - "Didaktiktag"
S40	1997-03-11	MATLAB und SIMULINK - Entwicklung und Anwendungen
S41	1997-04-10	Diskrete Simulation mit Micro Saint - Fuzzy Logic, Gleichzeitigkeit
S42	1997-05-06	Diskrete Simulation mit Micro Saint - Fuzzy Logic, Gleichzeitigkeit
S43	1997-11-25	Simulation in der Medizin - Regelungsmathematische Modellbildung in der Medizin - Präsentation eines Beratungssystems für Gefäßoperationen
S44	1997-01-12	Kontinuierliche Modellbildung und Simulation mit ACSL
S45	1998-03-12	MATLAB/SIMULINK/STATEFLOW - Neuerungen, Anwendungen
S46	1998-04-21	Computer Algebra Systeme (CAEs) –Einsatz, Anwendungen, Entwicklungen
S47	1998-04-22	MicroSaint, POWERSIM - Graphische Modellbildung und Simulation diskreter Prozesse
S48	1998-05-27	Modellbildung und Simulation diskreter Prozesse - (GPSS/H, Textuelle Modellbildung, Animation)
S49	1998-06-25	MATLAB/SIMULINK/STATEFLOW - Diskrete und kombinierte Modellbildung und Simulation mit Zustandsgraphen und Flowcharts, Spreadsheet-Simulation
S50	1998-11-12	Hybride Modellbildung und Simulation – Simulation mit ACSL

Teilnehmer

Die Teilnahme an den Seminaren steht allen Interessierten an Modellbildung und Simulation und der Anwendung verschiedenster Engineering Tools offen. Die je nach Seminar zwischen etwa 30 bis 110 Teilnehmer sind zum großen Teil Hörer und wissenschaftliches Personal der TU Wien und anderer Universitäten, erfreulicher Weise kommen Vortragende wie Zuhörer auch aus dem Bereich der Industrie, der Dienstleistungen und Verwaltung. Bei den bisherigen Seminaren war ein Anteil an Teilnehmern aus dem Bereiche der Industrie von etwa 20% zu beobachten.

All jene, die bereits an Seminaren der Seminarreihe teilgenommen haben, werden automatisch zu weiteren Seminaren ihres Interessensgebietes eingeladen.



Information

ARGESIM

p.A. Prof. Dr. F. Breitenecker
Abt. Simulationstechnik
Technische Universität Wien
Wiedner Hauptstr. 8-10
A-1040 Wien

Tel: +43-(0)1-58801-11452 bzw. -11456

Fax: +43-(0)1-58801-11499

E-Mail: seminar@argesim.org

WWW: <http://www.argesim.org/seminare>

VORTRAGSUNTERLAGEN



Soft Computing

Grundlagen und Anwendungen in der Biomedizinischen Technik

Boris Romanus Bracio
Austrian Research Centres Seibersdorf
Medizin- und Rehabilitationstechnik
Technische Universität Clausthal
Institut für Informatik

EINFÜHRUNG

Die Methoden der Modellbildung und der Simulation sind als mögliche Vorgehensweisen zur Gewinnung von Wissen über komplexe Prozesse seit langem etabliert. Als Ergänzung zu den klassischen Vorgehensweisen, die in der Regel ausgehend von einem Experiment, einer anschließenden Datenanalyse, über eine Abstraktion und Vereinfachung zu einem Differentialgleichungssystem führten, hat sich in den letzten Jahren ein neuer Forschungsschwerpunkt etabliert: Soft Computing (SC). Dieser Oberbegriff steht für alle die Algorithmen und Verfahren die mit unsicherem oder vagem Wissen beziehungsweise Daten arbeiten. Dazu gehören insbesondere die Neuronalen Netzen, Fuzzy Logik oder Evolutionsalgorithmen.

Die Methoden des Soft Computings sind dann einsetzbar, wenn die herkömmlichen Techniken, zum Beispiel auf Grund einer zu hohen Komplexität des zu untersuchenden Systems, versagen.

EINFÜHRUNG IN DIE VERFAHRENEN

FUZZY LOGIK

Schon 1973 wurde von Zadeh das Konzept einer Kombination von mathematischer Modellbildung und linguistischer Beschreibung zum Entwurf von Expertensystemen als Fuzzy Logik vorgestellt [Zah73]. Kern

der Fuzzy Systeme ist eine Regelbasis in der Abhängigkeiten verbal beschrieben werden können:

wenn die Konzentration von A hoch ist und die Konzentration von B normal **dann** erhöhe den Durchfluß

Ausgehend von einem realen Prozeß mit exakten Meßwerten müssen diese zu einer Bearbeitung mittels Fuzzy Logik einem *Fuzzy Set* (z. B. niedrig, normal, hoch) zugeordnet werden. Nach dieser *Fuzzyfizierung* wird mittels des Regelwerkes durch *Inferenz* eine Lösung ermittelt. Diese muß abschließend durch *Defuzzyfizierung* in eine exakte, für den realen Prozeß konforme Größe umgewandelt werden. Für die einzelnen Stufen eines solchen Fuzzy Controllers wurden unterschiedliche Methoden entworfen [YZ94].

NEURONALE NETZE

In Analogie zu den biologischen Nervensystemen wurden mit den Neuronalen Netzen (NN) Algorithmen entworfen, die zum einen aus signalverarbeitenden Elementen (Neuronen) und zum anderen aus verbindenden Komponenten bestehen. Jede Verbindung ist durch einen wichtenden Faktor charakterisiert, der den Anteil der Aktivität bestimmt, der von einem sendenden Neuron zu einem empfangenden Neuron gelangt. Ein Neuron kann mehrere Verknüpfungen auf der Ein- wie auch auf der Ausgangsseite haben. Die empfangenen Signale werden summiert und liefern

mittels einer spezifischen Aktivierungsfunktion ein neues Ausgangssignal. Die Architektur eines Neuronalen Netzes wird geprägt von der Anzahl der Neuronen in einer Verarbeitungsschicht, der Anzahl der verbundenen Schichten, der Wichtung der einzelnen Verbindungen und den implementierten Aktivierungsfunktionen. Ein besonderes Kennzeichen ist die Lernfähigkeit der Neuronalen Netze durch eine Variierung der Wichtungen auf der Basis von gegebenen Eingangs-/Ausgangsmustern. Die beim Trainieren verwendeten Verfahren (z. B. überwachtes oder unüberwachtes) sind ebenfalls charakterisierend für das Verhalten eines Neuronalen Netzes.

EVOLUTIONSALGORITHMEN

Unter den Begriff Evolutionsalgorithmen (EA) beziehungsweise evolutionary computation methods (EC) fallen alle die Algorithmen, welche auf dem grundlegenden Prinzip der Evolutionstheorie nach Darwin (survival of the fittest) basieren. Dazu gehören Verfahren wie

- genetic algorithms (GA),
- evolutionary programming,
- evolution strategies
- genetic programming

oder hybride Ansätze dieser Paradigmen.

Es handelt sich dabei um stochastische Optimierungsverfahren zur Auffindung einer bestmöglichen Lösung innerhalb einer potentiellen Zielmenge. Im allgemeinen wird eine Menge von möglichen Lösungen solange einem Selektions-, Mutations- und Auswahlverfahren unterzogen, bis ein bestimmtes Gütekriterium erfüllt ist [DM97].

ANWENDUNGEN

Soft Computing Verfahren haben sich in den letzten Jahren in vielen technischen und nicht-technischen Gebieten etabliert. Auf Grund des häufig biologischen Ursprungs der Methoden sollen exemplarisch biomedizinische Anwendungen vorgestellt werden.

Die Analyse von EKG-Daten im Hinblick auf die Detektion von Arrhythmien ist ein typisches Gebiet in dem Fuzzy Methoden und, oder in Kombination mit, Neuronalen angewendet werden. Zur Klassifikation werden die Spektren der EKG-Daten mittels eines Neuronalen Netzes analysiert. Um ein unüberwachtes Lernen zu ermöglichen werden beispielsweise Fuzzy Clustering Verfahren eingesetzt.

Ein anderes Gebiet bei der EKG Analyse ist die Unterscheidung von fetalen Herzrhythmen von denen der Mutter.

Bildgebende Verfahren wie Ultraschall-, CT- beziehungsweise MRI-Untersuchungen werden durch Fuzzy Systeme und Genetische Algorithmen erfolgreich verbessert. Im Bereich der Eliminierung von Bewegungsartefakten werden sie zur Bildrekonstruktion eingesetzt und sowie bei der Detektion von Tumor- oder Organkonturen.

Weiterhin ist das Gebiet der automatisierten Diagnose prädestiniert für den Einsatz von Soft Computing Methoden. Exemplarisch ist dafür die Überwachung von Diabetikern oder die Früherkennung von Brustkrebs zu nennen.

ZUSAMMENFASSUNG

In dem Beitrag wurden die grundlegenden Verfahren des Soft Computings kurz vorgestellt und einige Anwendungen im Bereich der biomedizinischen Technik skizziert. Gerade die komplexen Fragestellungen in diesem Bereich, die oftmals auch nur vage formuliert werden können, sind prädestiniert für die Anwendung von Soft Computing Methoden zur Modellbildung, Simulation und Signalverarbeitung.

LITERATUR

- [DM97] D. Dasgupta, Z. Michalewicz; *Evolutionary Algorithms in Engineering Applications*; Springer Verlag, Berlin, 1. Auflage, 1997
- [YZ94] R. R. Yager, L. A. Zadeh; *Fuzzy sets, Neural Networks and Soft Computing*; Van Nostrand Reinhold, New York, 1994
- [RBKZ95] G. Rau, K. Becker, R. Kaufmann, H.-J. Zimmermann; *Fuzzy Logic and Control: Principal Approach and Potential Application in Medicine*; Artif Organs, Vol. 19, No. 1, 1995, 105-112
- [HB98] J. Heitkoetter, D. Beasley; *The Hitch-Hiker's Guide to Evolutionary Computation: A list of Frequently Asked Questions (FAQ)*; USENET: comp.ai.genetic, Available via anonymous FTP from ftp://rtfm.mit.edu/pub/usenet/news.answers/ai-faq/genetic/About, 1998
- [NEI97] O. Nelles, S. Ernst, R. Isermann; *Neuronale Netze zur Identifikation nichtlinearer, dynamischer Systeme*; at Automatisierungstechnik, Vol. 45, No. 6, 1997, 251-261
- [Sch94] B. Schneider; *Neuronale Netze in der Medizin*; it+ti Informationstechnik und Technische Informatik, Vol 36, No. 6, 1994, 59-65

[Zah73] L. Zadeh; *Outline of a new approach to the analysis of complex systems and decision processes*; IEEE Trans.Syst. Man Cybernet. SMC-2, 1973, 28-44

AUTOR

Boris Romanus Bracio hat an der Technischen Universität Clausthal Maschinenbau mit Schwerpunkt Elektrische Informationstechnik studiert. Während seines anschließenden Promotionsstudiums war er wissenschaftlicher Angestellter am Institut für Informatik – Lehrstuhl für Technische Informatik der TU Clausthal. 1998 wurde er für das Hertha-Firnberg Stipendium ausgewählt und absolviert dieses zur Zeit am Österreichischen Forschungszentrum Seibersdorf. Sein besonderes Interesse liegt im Bereich der biomedizinischen Technik.

ADRESSE:

ARCS Ges.m.b.H.
Medizin- und Rehabilitationstechnik
Dipl.-Ing. Boris R. Bracio
Prof. Dr. Stefan-Koren Str. 8/3
A-2700 Wiener Neustadt
phone: +43 (0) 2622 6929014
email: boris.bracio@arcs.ac.at



Software Initiative

PowerProphet

Dr. Martin Kührer
PSE NLT AP
(+431) 1707 46360
martin.kuehrer@siemens.at



Software & Engineering
Siemens

Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kührer, PSE NLT2,
Page 1

PowerProphet



Software Initiative

- **Software tool to forecast**
 - Stock indices
 - Exchange rates
 - Interest rates
- **Excel Add-in**
- **Three different forecast methods
with Neural Networks and Fuzzy Logic**

PowerProphet

Software Initiative

- **Forecast horizon**
 - day
 - week
 - month
- **Menu to choose forecast method**
 - Neural Networks
 - Fuzzy Logic
- **Possibility to insert opinions**

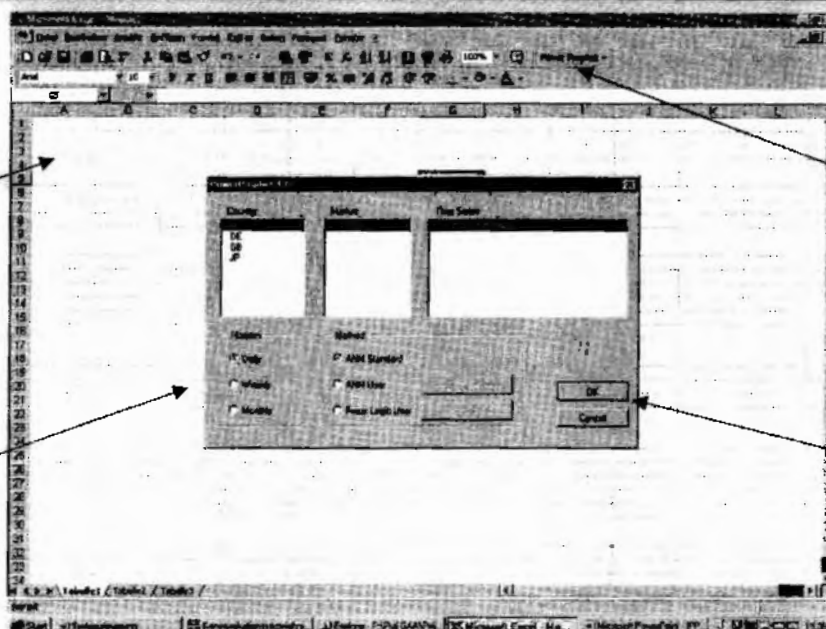
Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kührer, PSE NL T2,
Page 3

PowerProphet User Interface Excel

1.
Select a
cell.

3.
Select a
forecast.



2.
Click on the
Function Wizard
button.

4.
OK
and the forecasted
value is written
into the cell.

Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kührer, PSE NL T2,
Page 4

PowerProphet User Interface e.g.: Stock indices



Select a
country.

Select an
index.

Select a
market.

Select a
forecast
horizon.

Select a forecast method.

Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kührer, PSE NL12,
Page 5

PowerProphet Forecast Methods



Select one of
three different
forecast methods.

PowerProphet ANN Standard

Select
ANN Standard

Press OK

Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kührer, PSE NL12,
Page 7

PowerProphet ANN User

Select
ANN User

Choose Time Series

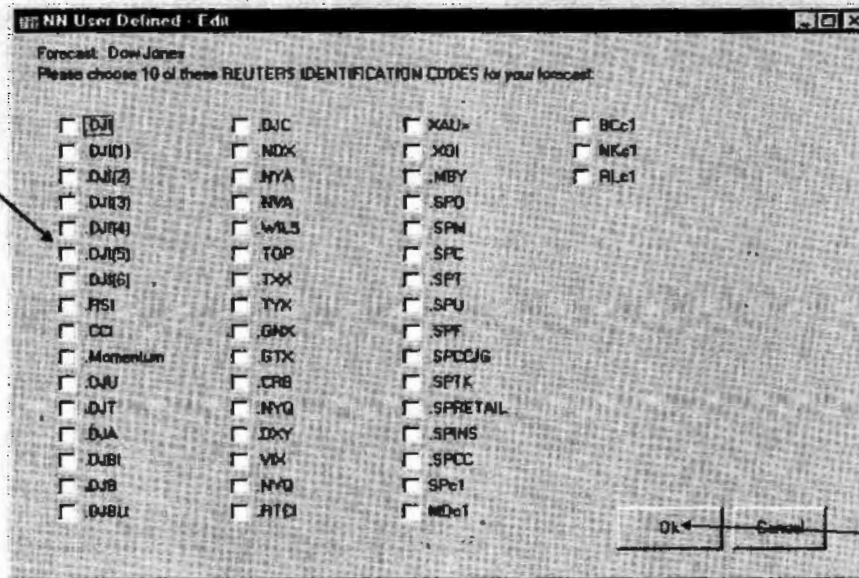
Press OK

Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kührer, PSE NL12,
Page 8

PowerProphetANN User

Select
Time series



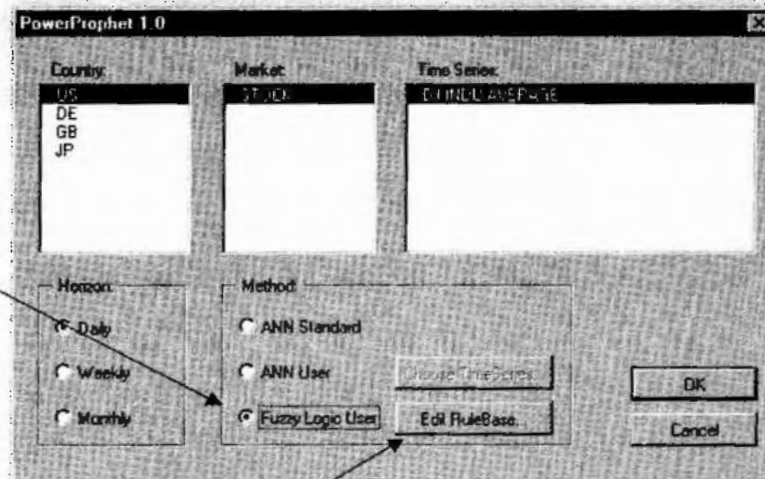
Press OK

Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kührer, PSE NLT2,
Page 9

PowerProphet Fuzzy Logic User

Select
Fuzzy Logic User

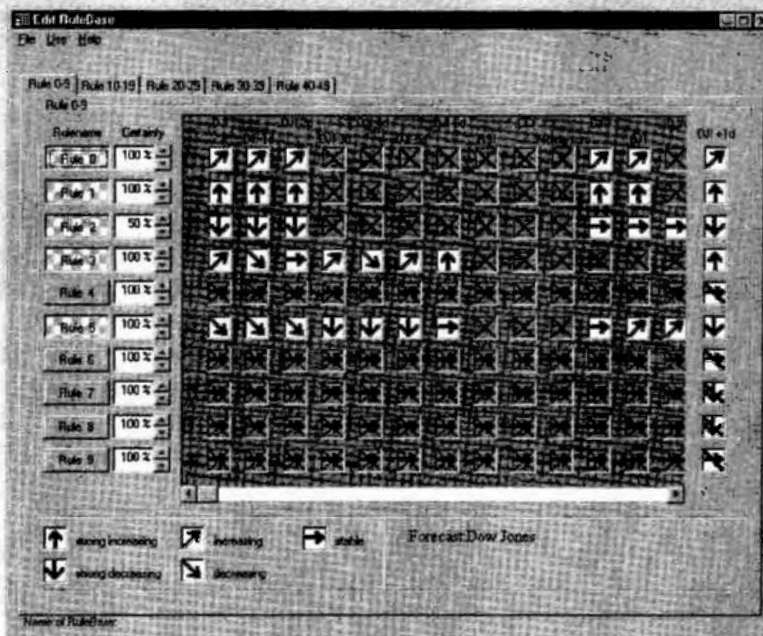


Press the Edit RuleBase button.

Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kührer, PSE NLT2,
Page 10

PowerProphet User Interface



Software Initiative

Insertion of
heuristics and
knowledge based
rules.

Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kührer, PSE NL12,
Page 11

Performance Hitrate

Software Initiative

	.DJI	.FTSE	.GDAX	.N225
Month	65-70%	65-70%	70-75%	55-60%
Week	65-70%	65-70%	70-75%	70-75%
Day	60-65%	65-70%	75-80%	65-70%

Input Parameters



German DAX

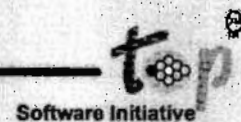
Input parameters:

Reuters Instrument Code	Time Lag	Description
aDEFXFW3M	2	3m Forward DM fixes
aDEMBOND	2	Deutsche Bonds
aJPPRIME	2	Prime rate in Japan
aDEFXFW3M	1	3m Forward DM fixes
XAU=	1	Gold
aDECBLR	0	Central Bank Lending Rate
aDECURAC	0	Current Account

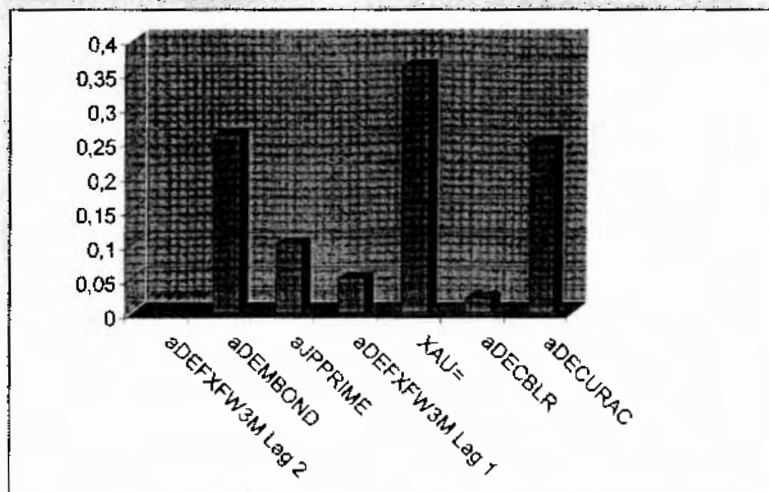
Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kühner, PSE NLT2,
Page 13

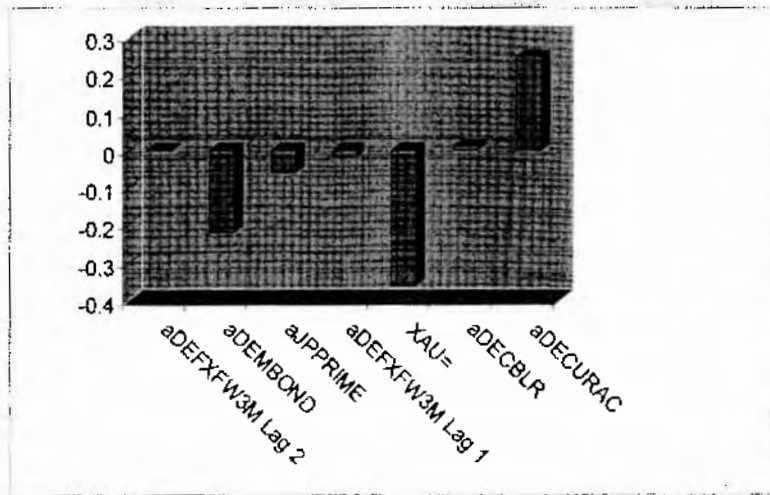
Input parameters



Mean sensitivity 1992 to 1998



Input parameters



Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kührer, PSE NL72,
Page 15

24 day simulation

Input

GBP / \$	
Day	
ANN Standard	

GBP/USD

actual position: SHORT

60	5.5%
40	3.5%

TJ	OR	NN	OR	NN	Trend Corr.	OR	NN	Index	60/40	NN	Correct OPTI	NN Index	OPTI Index
25 98-08-03	1.63					0.8%	0.0%	100.00	60	0.8%	100	0.00	0.00
24 98-08-04	1.64	1.63			✓	-0.1%	-0.2%	100.70	40	0.3%	101	0.08	0.06
23 98-08-05	1.64	1.64			✓	-0.2%	-0.2%	101.06	40	0.3%	101	0.07	0.07
22 98-08-06	1.63	1.63			✓	-0.3%	-0.1%	101.34	40	0.2%	102	0.06	0.08
21 98-08-07	1.63	1.63			✓	0.4%	0.0%	101.57	60	0.6%	102	0.14	0.14
20 98-08-10	1.64	1.63			✓	-0.3%	-0.1%	102.14	40	0.2%	103	0.15	0.15
19 98-08-11	1.63	1.63			✓	-0.2%	-0.1%	102.38	40	0.3%	103	0.16	0.16
18 98-08-12	1.63	1.63			✓	-0.1%	0.0%	102.65	40	0.3%	103	0.18	0.18
17 98-08-13	1.63	1.63			✓	-0.5%	-0.1%	102.96	40	0.2%	103	0.19	0.19
16 98-08-14	1.62	1.62			✓	-0.1%	-0.1%	103.06	40	0.3%	104	0.18	0.18
15 98-08-17	1.62	1.62			✓	0.3%	0.1%	103.44	60	0.6%	105	0.20	0.20
14 98-08-18	1.62	1.61			✓	0.4%	0.0%	103.88	40	0.5%	105	0.34	0.32
13 98-08-19	1.62	1.62			✓	0.5%	0.0%	104.46	60	0.7%	106	0.21	0.28
12 98-08-20	1.63	1.62			✓	0.2%	-0.2%	105.07	40	0.4%	106	0.17	0.42
11 98-08-21	1.64	1.63			✓	-0.1%	0.0%	105.72	40	0.3%	107	0.16	0.41
10 98-08-24	1.64	1.63			✓	1.0%	-0.2%	106.23	40	0.7%	108	0.15	0.40
9 98-08-25	1.64	1.64			✓	1.6%	-0.2%	106.57	60	1.0%	109	0.04	0.52
8 98-08-26	1.64	1.64			✓	0.1%	-0.3%	106.93	40	0.4%	109	-0.14	0.89
7 98-08-27	1.65	1.64			✓	-0.4%	0.1%	107.85	60	0.2%	110	-0.19	0.75
6 98-08-28	1.68	1.65			✓	-0.3%	-0.2%	109.13	40	0.2%	110	-0.18	0.77
5 98-08-31	1.68	1.68			✓	0.4%	-0.1%	109.60	40	0.5%	110	-0.23	0.82
4 #OutOfDer	1.68	1.68			✓	-0.3%	-0.1%	109.77	40	0.2%	111	-0.22	0.83
3 98-09-02	1.67	1.67			✓			110.00	40				
2 98-09-03	1.68	1.67			✓			110.61	40				
1 98-09-04	1.67	1.67			✓			110.86	40				
0 #OutOfDate	1.67	1.67			✓			110.86	40				

Hits: 67%

0.2

0

1.2

24 week simulation

Input

German DAX
Week
ANN Standard

Userparam:

GDAX
actual position:

LONG

Forecast

110
90
5.50%
0.11%

	TI	OR	NN	OR	NN	Trend. Corr.	OR	NN	Index	long	LONG	110/90	NN	Correct OPTI
25	98-01-09	4238.94	4241.18						100.0		100.0		100.0	100.0
24	98-01-16	4184.46	4241.18				-1.2%	0.1%	98.8	110	-1.37%	98.6	98.6	98.9
23	98-01-23	4237.31	4198.91				1.3%	0.3%	100.0	110	1.38%	100.0	100.0	100.3
22	98-01-30	4440.38	4269.68				4.8%	0.8%	104.8	110	5.26%	105.2	105.2	105.6
21	98-02-06	4496.33	4464.57				1.3%	0.5%	106.1	110	1.38%	106.7	106.7	107.0
20	98-02-13	4502.48	4519.17				0.1%	0.5%	106.3	110	0.14%	106.8	106.8	107.1
19	98-02-20	4602.65	4521.26				2.2%	0.4%	108.6	110	2.44%	109.4	109.4	109.7
18	98-02-27	4709.83	4632.33				2.3%	0.6%	111.2	110	2.55%	112.2	112.2	112.5
17	98-03-06	4715.95	4740.03				0.1%	0.6%	111.3	110	0.13%	112.4	112.4	112.7
16	98-03-13	4889.85	4749.50				3.7%	0.7%	115.4	110	4.04%	116.9	116.9	117.2
15	98-03-20	5001.55	4912.16				2.3%	0.5%	118.0	110	2.51%	119.9	119.9	120.2
14	98-03-27	5083.16	5027.83				1.6%	0.5%	120.0	110	1.78%	122.0	122.0	122.3
13	98-04-03	5223.52	5114.01				2.8%	0.6%	123.3	110	3.03%	125.7	125.7	126.0
12	98-04-10	5317.22	5259.77				1.8%	0.7%	125.5	110	1.96%	128.2	128.2	128.5
11	98-04-17	5268.75	5338.52				0.9%	0.4%	124.4	110	-1.01%	126.9	126.9	127.5
10	98-04-24	5144.28	5292.11				-2.4%	0.4%	121.4	110	-2.61%	123.6	123.6	124.8
9	98-05-01	5107.44	5161.74				-0.7%	0.3%	120.5	110	-0.80%	122.6	122.6	124.0
8	98-05-08	5257.58	5136.34				2.9%	0.6%	124.1	110	3.22%	126.5	126.5	128.0
7	98-05-15	5393.14	5274.48				2.6%	0.3%	127.3	110	2.83%	130.1	130.1	131.8
6	98-05-22	5564.21	5409.98				3.2%	0.3%	131.3	110	3.48%	134.6	134.6	136.2
5	98-05-29	5569.08	5592.86				0.1%	0.5%	131.4	110	0.09%	134.7	134.7	136.3
4	98-06-05	5688.50	5607.14				2.1%	0.7%	134.3	110	2.35%	137.9	137.9	139.5
3	98-06-12	5670.83	5728.94				-0.3%	0.7%	133.8	110	-0.35%	137.4	137.4	139.1
2	98-06-19	5702.61	5697.70				0.6%	0.5%	134.6	110	0.61%	138.2	138.2	140.0
1	98-06-26	5870.42	5745.04				2.8%	0.7%	138.6	110	3.23%	142.7	142.7	144.5
0	00-01-00		5907.05							110		110		

Hits: 79%

-4.1

0.0

0

1.8

SIEMENS



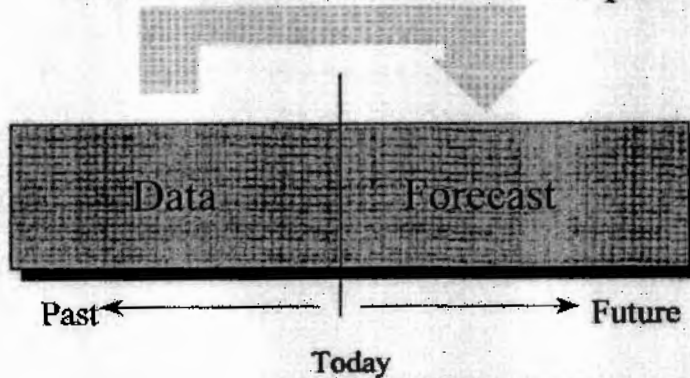
Technologies

Financial Market Forecast... ... Soft Computing Techniques



Software Initiative

What can be derived from the past ?



○ Information Theory

○ Fuzzy Logic

○ Neural Networks

○ Genetic Algorithms

Simulate and optimize strategies !

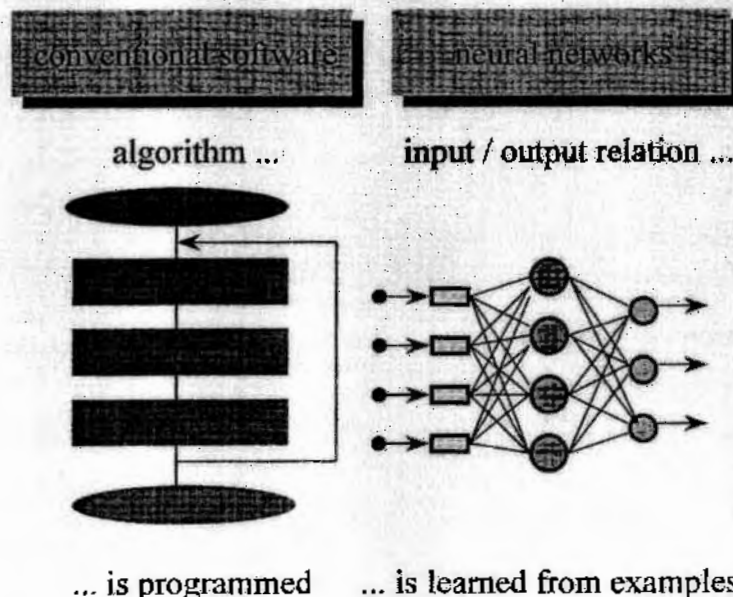
Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kührer, PSE NLT2,
Page 19

Soft Computing Techniques... ... Neural Networks



Software Initiative



○ theoretical background

□ learning by examples

□ generalisation

□ data-driven

○ application

□ nonlinearity

□ complex systems

○ advantages

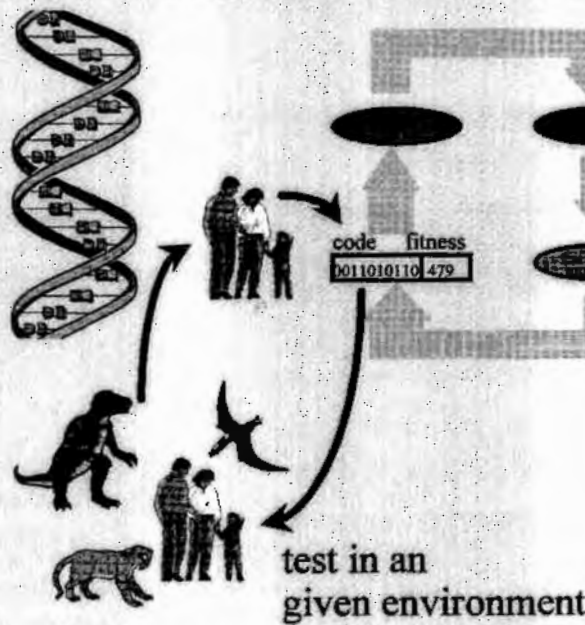
□ adaptive

□ fault tolerance

Competence in Software and Engineering
Market, Methods, Opportunities

© Siemens AG, 1998
Kührer, PSE NLT2,
Page 20

Soft Computing Techniques... ... Genetic Algorithms

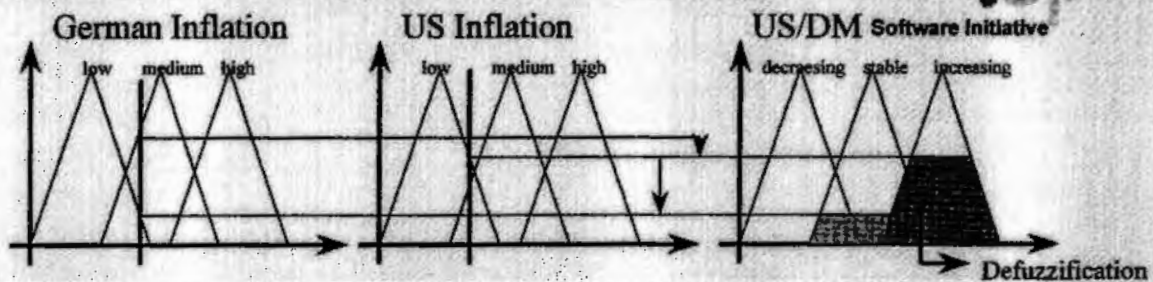


Software Initiative

- theoretical background
 - search procedure
 - based on the mechanism of natural selection, recombination and mutation
- application
 - optimization of complex systems
- advantage
 - outperforms conventional optimization procedures

Competence in Software and Engineering
Market, Methods, Opportunities© Siemens AG, 1998
Kührer, PSE NLT2,
Page 21

Soft Computing Techniques... ... Fuzzy Logic



IF German Inflation = low AND US Inflation = low THEN US/DM = stable
 IF German Inflation = medium AND US Inflation = low THEN US/DM = increasing

- theoretical background
 - heuristic knowledge, linguistic variables
- application
 - complex systems, which are difficult to model by analytic means
- advantages
 - processing of fuzzy information
 - simulation of heuristic knowledge
 - traceable

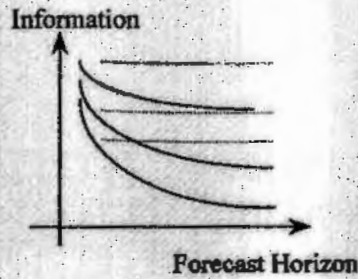
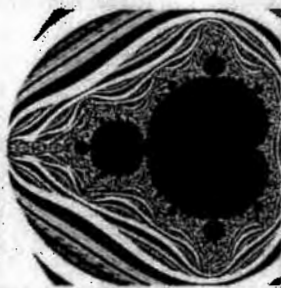
Competence in Software and Engineering
Market, Methods, Opportunities© Siemens AG, 1998
Kührer, PSE NLT2,
Page 22

Soft Computing Techniques...

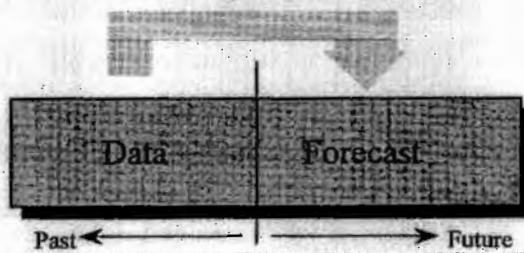
... Information Theory



Software Initiative



What can be derived from the past
in the optimal case ?



○ theoretical background

- ☐ mutual information
- ☐ marginal redundancy

○ application

- ☐ time series analysis
- ☐ non linear data analysis

○ advantages

- ☐ stochastic data processing

Lawinenprognose mit ECANSE

Karl Kleemayr
Stefan Vollsinger



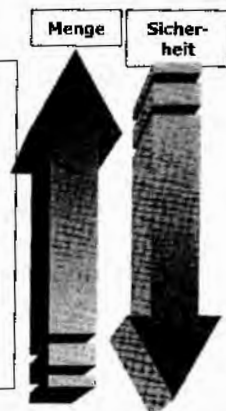
INSTITUT FÜR WILDBACH
UND LAWINENSCHUTZ

Problemstellung - Zielsetzung (NAFT)

- Erhöhung der Sicherheit für das Betriebspersonal
- Verbesserung der bestehenden Prognose
- Verwendung selbstlernender Verfahren
- Werkzeuge müssen zuverlässig sein

INFORMATIONSTYPEN:

- Meteorologische Daten
Niederschlag, Wind, Temperatur, Strahlung, etc.
- Schneedeckendaten
Kornform, Temperaturprofil, Feuchtigkeit, etc.
- Stabilitätsdaten
Rutschblock, Norwegermethode, Scherrahmen, etc.



INSTITUT FÜR WILDBACH
UND LAWINENSCHUTZ

Datengrundlagen - Rahmenbedingung

Dat	S_hohe0	S_hohe1	S_hohe2	S_hohe3	Nschl0	Nschl1	Nschl2	Nschl3	Lt0	Lt1	Lt2	Lt3	Wiri0	Wiri1	L_Code0	Lgrad
811203	98	95	90	145	11	5	7	4	-9	-7	-6	-11	27	27	0	200
811204	90	98	95	90	4	11	5	7	-8	-9	-7	-6	31	27	0	200
811205	110	90	98	95	27	4	11	5	-6	-8	-9	-7	27	31	1	302

- Verhältnis zwischen Lawinentagen-Nichtlawinentagen der Datengrundgesamtheit:

16% : 84% !!!

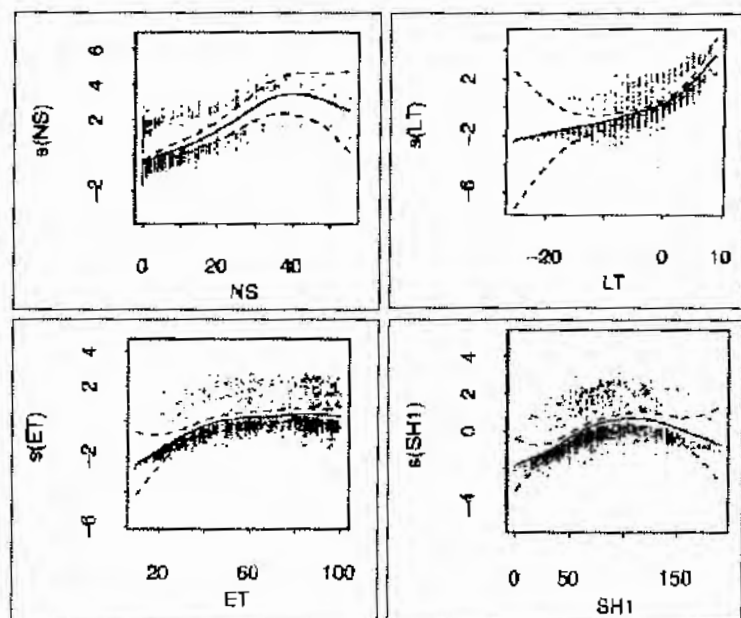
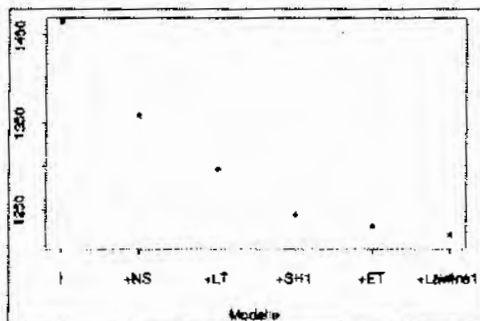
- Möglichst hohe Gesamttreffer: >70%
- Möglichst hohe Treffer an Lawinentagen: >80%
- Möglichst hohe Treffer Nicht-Lawinentagen: >75%

Keiner der Werte sollte von diesen Richtwerten deutlich abweichen.

Beobachtung	Prognose		
	Schne L.	Lawine	
	Schne L.	Lawine	
	T1 %	F1 %	100 % = Summe NLT
	F2 %	T2 %	100 % = Summe LT

Statistische Voruntersuchungen

Lineare und Nicht-Lineare
Regressionsuntersuchungen

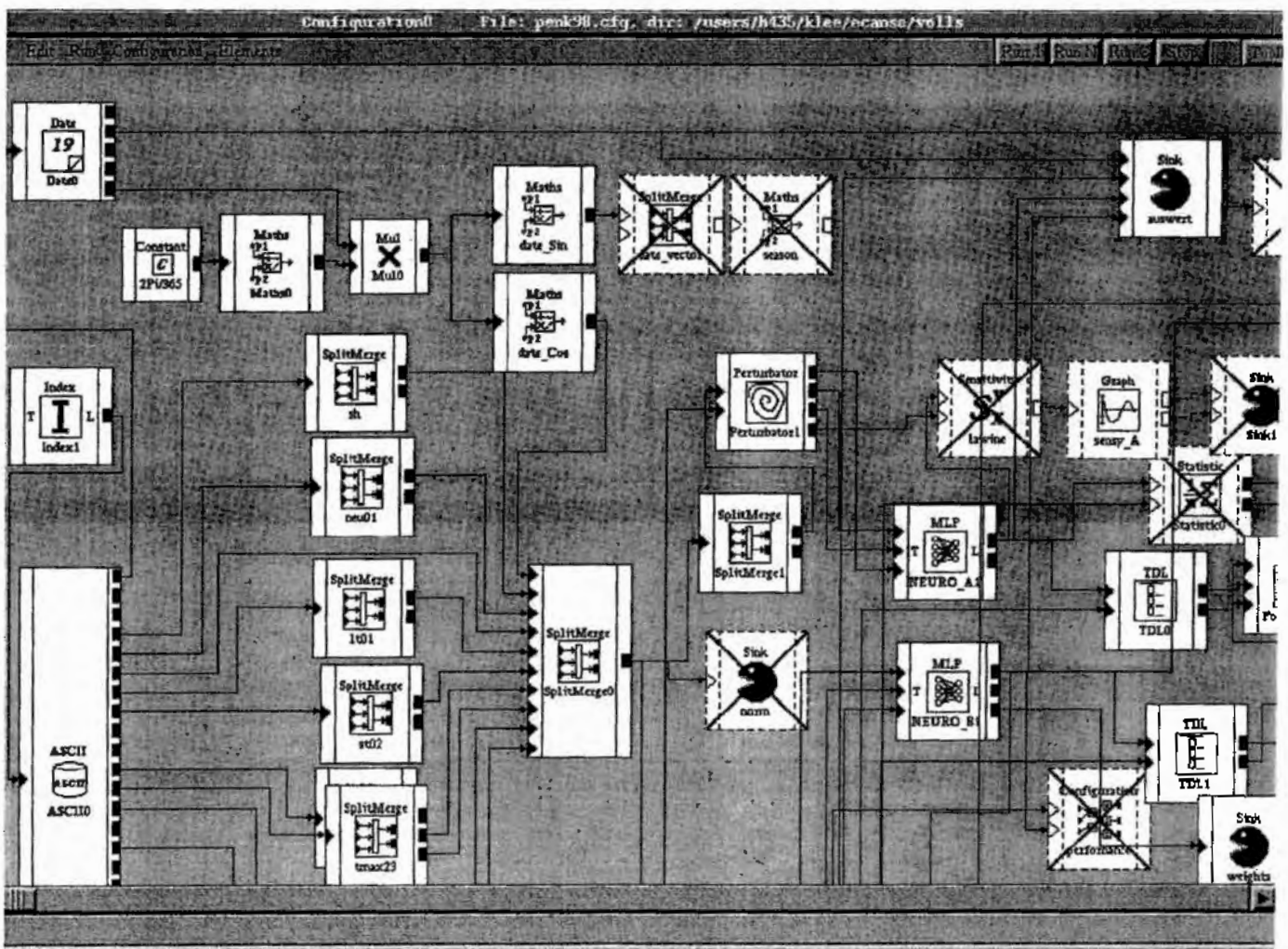


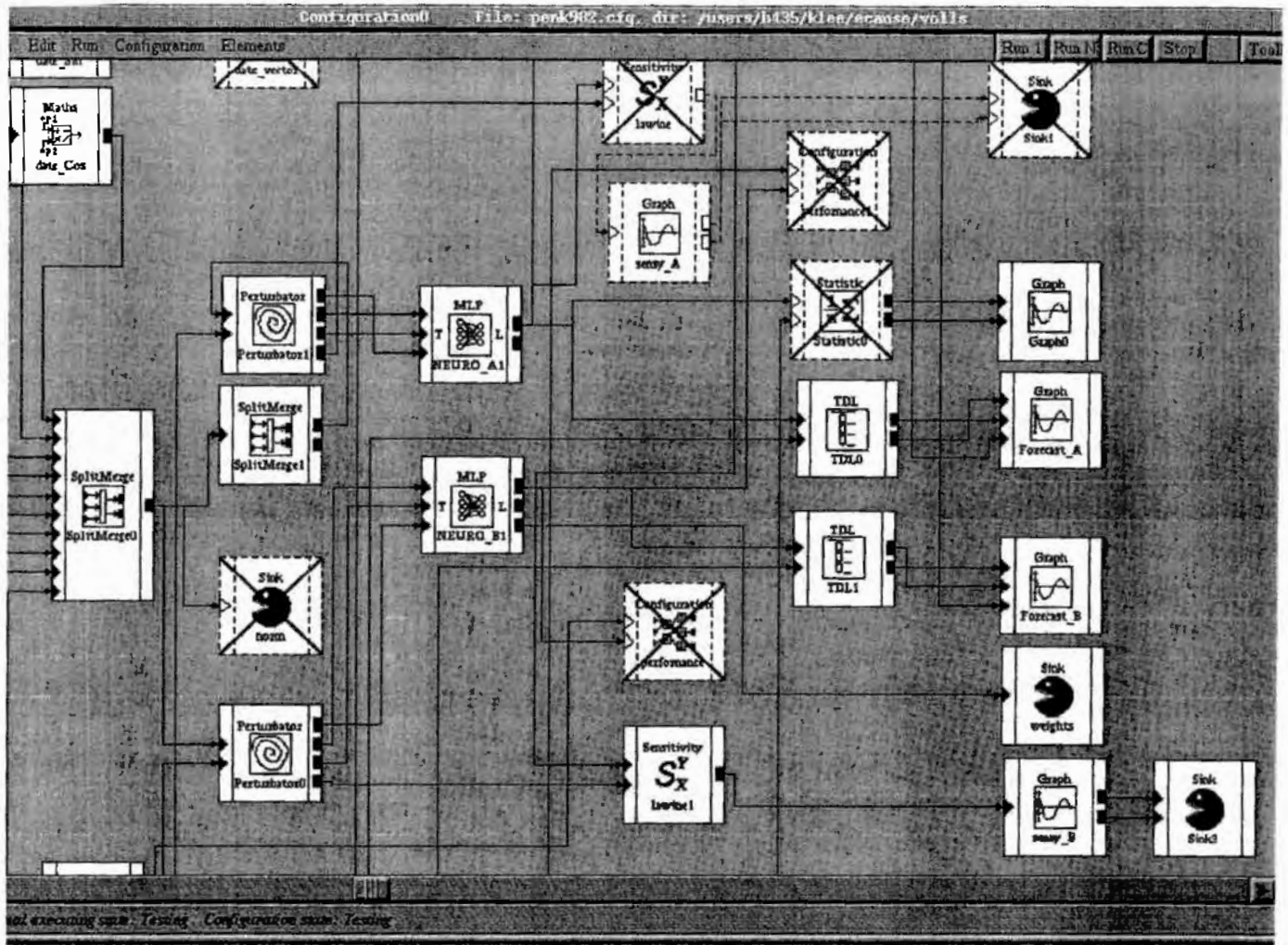
Auswahl des Netztypus

- Überwachtes Lernen: 75% Training; 25% Test
- **Multilayer Perzeptron**
- Variantenstudien:
 - + unterschiedliche Anzahl von Hidden Layern
 - + Unterschiedliche Anzahl der Lernschritte
 - + Unterschiedliche Länge des Datensatzes
 - + Unterschiedliche Variablenanzahl

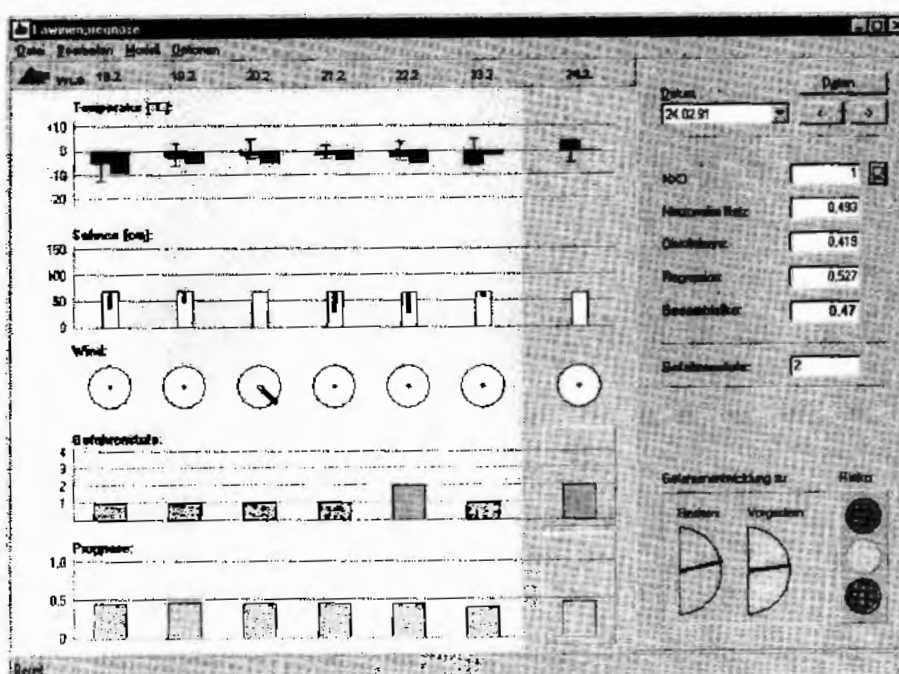


Interim für V. Adolph und J. Langenbacher





Integration der Ergebnisse in einen Windowsprogramm

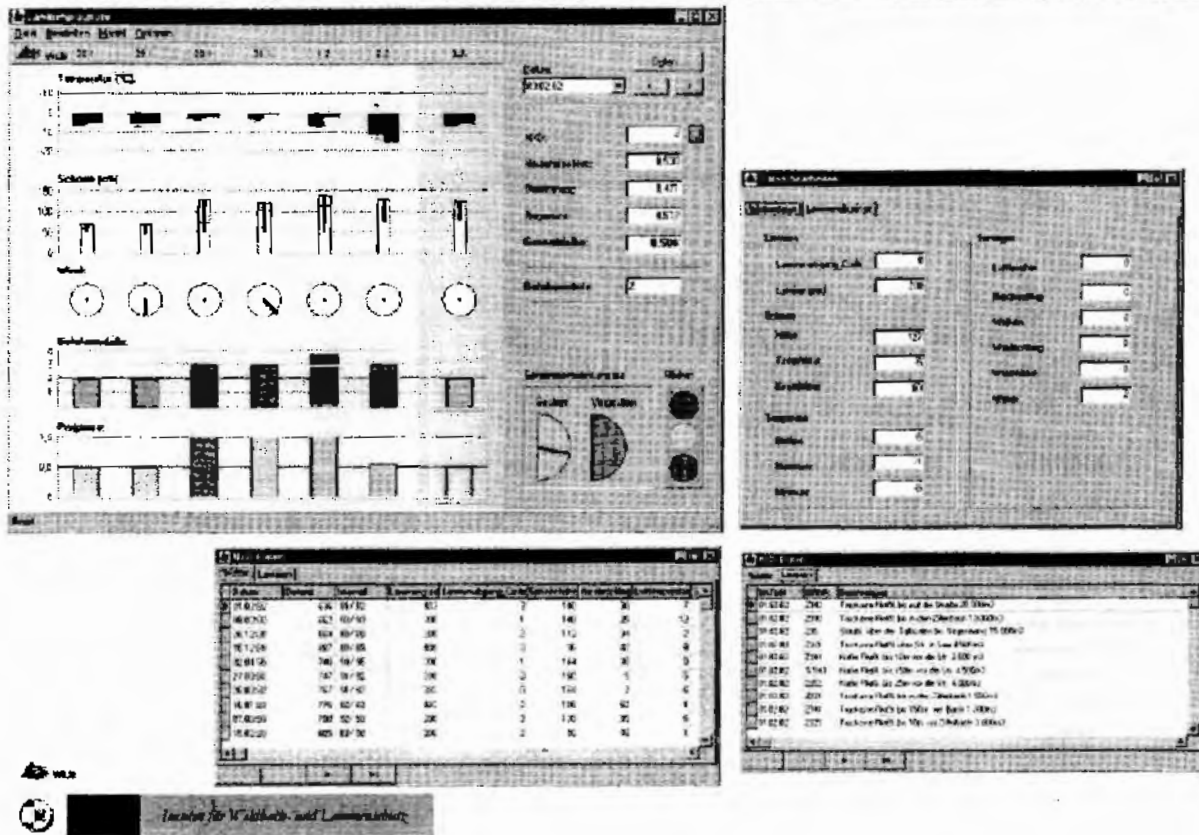


Integration von

- Wetter-
- Lawinen-
- Prognose-

Informationen
auf einer Oberfläche

Integration der Ergebnisse in einem Windowsprogramm



Interpretation der Ergebnisse - Perspektiven

- Die Erhöhung der Anzahl der Hiddenlayer brachte keine Steigerung der Prognosegenauigkeit - Alle maßgeblichen Informationen eingeflossen?
- Räumliche Differenzierung der Informationen notwendig!
- „Entzerrung“ der Einflußgrößen entsprechend der Nichtlinearen Regressionsuntersuchungen könnte weitere Steigerung bringen
- Hauptproblem ist der zu dünne Datenraum !!
- Kombination mit physikalischen Modellen oder Expertensystemen notwendig:

Danke für Ihre Aufmerksamkeit

Fuzzy Logik in diskreter Modellbildung und Simulation

F. Breiteneker, M. Lingl
Abt. Simulationstechnik, Technische Universität Wien
Email: Felix.Breiteneker@tuwien.ac.at
mlingl@osiris.tuwien.ac.at

Theorie und Anwendungen der Fuzzy Mengen bzw. der Fuzzy Logik haben in vielen Bereichen zu neuen Erkenntnissen und Entwicklungen geführt, z.B. im Bereich der Regelungstechnik. Auch in der Simulationstechnik spielt Fuzzy Logik damit eine immer bedeutendere Rolle, allerdings bisher nur im Bereich der kontinuierlichen Modellbildung und Simulation.

Dieser Beitrag führt nach einer kurzen Standortbestimmung die Methoden der Fuzzy Logik in die diskrete Simulation ein, wobei folgende Klassifizierung getroffen wird: Verwendung von Fuzzy Parametern (Attributen) nach regelungstechnischen Vorbild, Fuzzy Logik als Entscheidungswerkzeug (stochastische Defuzzifizierung) und Fuzzy Logik in Markov-Ketten. Abschließend werden Implementierungsaspekte und der Zusammenhang von Fuzzy Methoden und Wahrscheinlichkeitstheorie diskutiert.

1 Einleitung

Die klassische Einteilung der Simulationstechnik in die Bereiche kontinuierliche und diskrete Modellbildung und Simulation mit Berücksichtigung ihrer Hauptanalysemethoden (numerische Lösung von Differentialgleichungen bzw. stochastisches Event Handling) ist heute einerseits noch sichtbar, andererseits werden mehr und mehr gemeinsame Wurzeln analysiert, wenn sich die Betrachtung auf die Analyse im Zeitbereich beschränkt, u. a. im Bereich der Ereignisbehandlung ([PLAN97]). In letzter Zeit werden aber auch vermehrt die Methoden des sogenannten Soft Computings in die Simulationstechnik eingeführt, die die klassischen Analysemethoden wesentlich ergänzen können und die diese Einteilung teilweise auch obsolet machen.

Zu Soft Computing werden gerechnet

- Fuzzy Systeme und Fuzzy Logik
- Genetische Algorithmen
- Neur(on)ale Netze
- Schließverfahren

Zusätzlich können auch

- Symbolische Verfahren (analytische Verfahren)

in diese Klasse neuer Methoden gerechnet werden, die nun wiederentdeckt werden und dank fortgeschrittener Computereffizienz und Software auch handhabbar sind.

Die Methoden des Soft Computing gehorchen keiner einheitlichen Klassifizierung, denn es sind „bunt gemischt“ Modellbildungsverfahren, Optimierungsverfahren, qualitative Methoden, logische Verfahren etc., die in unterschiedlichen Ebenen der Simulationstechnik eingesetzt werden können, z.B. in der Modellbeschreibung, in der Modellsynthese (Reglersynthese), in der Modellableitung, in der Experimentierumgebung, auf der algorithmischen Ebene, etc. Zudem sei auf den direkten Einsatz der Methoden des Soft Computing verwiesen (Ersatz der Zeitbereichsanalyse, [MÖLL95]), z. B. auch der Einsatz genetischer Algorithmen zur Optimierung diskreter Systeme ([SALZ96], [SALZ96]).

Tabelle 1-1 zeigt beispielhaft den möglichen Einsatz des Soft Computing in den klassischen Domänen der Simulationstechnik.

Tabelle 1-1 Einsatzgebiete des Soft Computing in der Simulationstechnik

	Kontinuierliche Modellbildung und Simulation	Diskrete Modellbildung und Simulation
Fuzzy Systeme und Fuzzy Logik	Fuzzy Regelung, qualitative Modelle	
Neur(on)ale Netze	Neuronale Modelle	Neuronale Modelle
Genetische Algorithmen	Optimierung, Adaption	Optimierung, Adaption
Schließverfahren	AI-Methoden für Algorithmenauswahl	Entscheidungsfindung (statisch, dynamisch)
Symbolische Verfahren	Analytische Integration und Differentiation	Warteschlangenmodelle, Markov-Ketten

Die Tabelle 1-1 zeigt einen „weißen Fleck“ bei der Verwendung von Fuzzy Methoden in der diskreten Simulation. Sieht man von der direkten Übertragung der Parameterfuzzifizierung (siehe Kap. 3) und vom direkten Einsatz ab, so finden sich in der Literatur und auch im Internet nur wenige Angaben (Tab. 1-2, tatsächlich verringert noch durch die Mehrdeutigkeit der Suche).

Tabelle 1-2 Informationen über „Fuzzy“ im Internet, Alta Vista - Suche

Suchkriterium	discrete	continuous, dynamic, control
+fuzzy	142900	
-"- +logic	45970	
-"- +simulation	6200	
-"-	931	7460
-"- +modeling	802	6034
-"- -ai -neural net	160	1253

Es stellt sich nun die Frage, wie Fuzzy Methoden sinnvoll in diskrete Modellbildung und Simulation eingebunden werden können. Nach Betrachtung verschiedener Aspekte kristallisieren sich drei sinnvolle und anwendungsorientierte Möglichkeiten dazu heraus:

- Verwendung von Fuzzy Parametern (Attributen) nach regelungstechnischem Vorbild („Klassische“ Parameterfuzzifizierung und Defuzzifizierung) in zeitabhängigen Eventmodellen (Kap. 3)

- Fuzzy Logik als Entscheidungswerkzeug (stochastische Defuzzifizierung für nichtgeordnete Parameterräume (Auswahlpfade)) in zeitabhängigen Eventmodellen (Kap. 4)
- Fuzzy Logik in Markov-Ketten (Fuzzifizierung und stochastische Defuzzifizierung für die Übergangswahrscheinlichkeiten in Markov-Ketten) in Modellen mit Markov-Ketten (Kap. 5)

Derzeit stellen diskrete Simulatoren kaum Werkzeuge (Funktionen etc.) zur Implementierung von Fuzzy Methoden zur Verfügung - sie müssen selbst programmiert werden und sind im Prinzip komplexe Table Lookup- Verfahren. Kapitel 6 beschäftigt sich kurz mit der Frage einer effizienten Implementierung.

Im Prinzip sind die Fuzzy Methoden nur eine vereinfachte Beschreibung komplexer Zusammenhänge, wobei die Komplexität in den Hintergrund eines variabel-mehrwertigen logischen Kalküls verlagert wird und im Vordergrund eine komfortable Beschreibung mit umgangssprachlichen Kenngrößen (linguistischen Variablen; [ZADE84]) angeboten wird. Jede Regel ist auch klassisch mit IF-THEN-ELSE-Konstruktionen beschreibbar - aber nur viel mühsamer.

Eine neue Betrachtungsweise allerdings entsteht bei Einbindung stochastischer Elemente, die die Frage nach dem Zusammenhang zwischen „Fuzzy Kalkül“ und Wahrscheinlichkeitstheorie aufwerfen - dieser Zusammenhang (diskutiert in Kap. 7) hebt die Fuzzy Methoden über eine bequeme Beschreibungsform wesentlich hinaus.

2 Fuzzy Logik - Motivation und Einsatz in der Simulation

Seit ihrer Entwicklung 1965 ([ZADE65]) wurde die Theorie der Fuzzy Mengen und der Fuzzy Logik zu einer anerkannten Methode der Ingenieurwissenschaften ([ZADE68]), [BALD81], [BAND93], [COX94], [HOFE94], besonders der Gebrauch von Fuzzy Reglern ist weit verbreitet ([TILL92]).

Es gibt mehrere Gründe, Fuzzy Logik auch in der diskreten Simulation zu verwenden, und sie decken sich zum Großteil mit den Gründen, warum Fuzzy Methoden in der Regelungstechnik und damit auch in der kontinuierlichen Modellbildung und Simulation ([HOFE94], [MÖLL95]) verwendet werden:

- **Unscharfe Daten:** Das Sammeln von Daten stellt einen schwierigen, aber auch sehr wichtigen Teil einer diskreten Simulation dar. Leider sind diese Daten oft ungenau oder schwierig zu bekommen. Fuzzy Logik ist eine Möglichkeit, diese Ungenauigkeiten zu beschreiben.
- **Unscharfe Regeln:** Die Information über die Funktionsweise, die von anderen Personen meist in verbaler Form zur Verfügung gestellt wird, ist oft ungenau und subjektiv. In Fuzzy Logik können die Regeln verbal formuliert werden und sind doch für den Computer verständlich.
- **Verständliche Parameter:** Auch Menschen, die nichts mit Mathematik, Informatik oder Simulationstechnik zu tun haben, können mit etwas Interesse die Bedeutung von Fuzzy Mengen und von Fuzzy Regeln verstehen und so gemeinsam mit dem Simulationsexperten an der Anpassung der Parameter arbeiten (linguistische Variablen; [ZADE84]).

- **Standardisierung der Unschärfe:** Es mag mehrere Arten geben, Unschärfen zu beschreiben. Aber eine Standardisierung erhöht die Übertragbarkeit und die Lesbarkeit der Beschreibung wesentlich, zudem erfolgt eine Abgrenzung zum wahrscheinlichkeitstheoretischen Ansatz.

Fuzzy Logik in der Regelungstechnik

Fuzzy Logik fand sehr rasch Eingang in die Regelungstechnik, und damit auch in die Modellbildung und Simulation von Regelungssystemen.

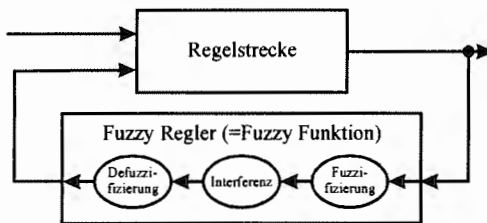


Bild 2-1 Arbeitsweise einer klassischen Fuzzy Funktion (Regler)

Ein Regler kann als m -dimensionale Funktion eines n -dimensionalen Parameters aufgefaßt werden, das heißt der Regler hat n Eingänge und m Ausgänge. Wenn wir den Zusammenhang zwischen den Ein- und Ausgängen mit den Mitteln der Fuzzy Logik beschreiben, erhalten wir eine Fuzzy Funktion (Bild 2-1).

Da solche Fuzzy Funktionen hauptsächlich in der Regelungstechnik verwendet wurden und werden, spricht man üblicherweise von Fuzzy Reglern. Für die in diesem Beitrag dargestellte Verallgemeinerung ist dieser Begriff aber nicht zutreffend.

Die Auswertung einer Fuzzy Funktion erfolgt in drei Schritten gemäß den Zugehörigkeitsfunktionen der linguistischen Variablen (Fuzzy Mengen) und der auf ihnen definierten Operationen: Fuzzifizierung, Interferenz, Defuzzifizierung (Bild 2-1, Bild 2-1)

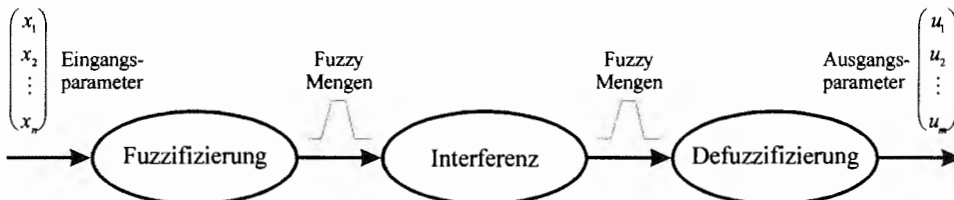


Bild 2-2 Fuzzy Funktion (Regler); Eingang: Zustände x_i , Ausgang: Regelgrößen u_k

Das *Fuzzifizieren* bedeutet die Auswertung von Tabellenfunktionen (Implementierung der Zugehörigkeitsfunktionen), die die Fuzzy Mengen beschreiben. Die erhaltenen Fuzzy Werte werden Fuzzy Mengen zugeordnet.

Die *Interferenz* verbindet die erhaltenen Fuzzy Mengen zu einer Fuzzy Menge für jeden Ausgangsparameter nach dem vorgegebenen Regelsystem.

Um einen reellen Parameterwert für jeden Ausgang zu erhalten, muß die errechnete Fuzzy Menge noch einer *Defuzzifizierung* unterzogen. Dafür werden in der Regelungstechnik üblicherweise zwei Methoden verwendet:

- **Maximum:** Der Abszissenwert des Maximums wird als Wert verwendet. Probleme können entstehen, da es nicht immer ein eindeutiges Maximum geben muß.
- **Schwerpunkt:** Der Abszissenwert des Schwerpunkts ist der defuzzifizierte Wert der Fuzzy Menge. Diese Methode funktioniert nur bei endlichen Fuzzy Mengen.

3 Fuzzy Parameter und Attribute in diskreter Simulation

Die einfachste Art der Anwendung der Fuzzy Logik in der diskreten Simulation besteht im direkten Übertragen der Vorgangsweise aus der Regelungstechnik. Aus mehreren Eingangsparametern (z.B. Attribute von Entities, Systemgrößen) werden durch Fuzzifizierung, Interferenz und (klassische) Defuzzifizierung die Ausgangsparameter (neue Attributwerte) errechnet. Bild 3-1 zeigt diese Übertragung der klassischen Vorgangsweise.

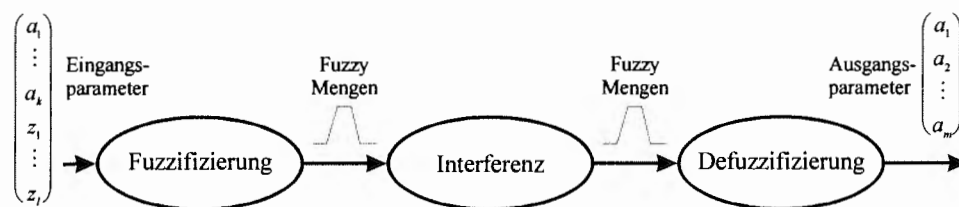


Bild 3-1 Fuzzy Funktion; Eingang: Attribute a_i , Zuständen z_i , Ausgang: Attribute a_k

Das klassische Jobshop Problem kann diese Vorgangsweise an Hand der Ermittlung der aktuellen Prioritäten in einer Warteschlange mit einer Fuzzy Funktion erläutern und mit einem reinen FIFO - Ansatz und einem die Fuzzy Funktion approximierenden klassischen logischen (binären) Ansatz vergleichen.

Fallstudie Jobshop Problem

Das hier untersuchte Jobshop Problem besteht aus sechs verschiedenen Maschinengruppen mit verschiedener Anzahl von identen Maschinen (4-16). Drei verschiedene Typen von Arbeitsaufträgen sind zu erledigen, die sich in der Anzahl und Reihenfolge der Fertigungsschritte und in der Bearbeitungszeit unterscheiden. Vor jeder Maschinengruppe warten die Aufträge in einer Warteschlange auf das Freiwerden einer Maschine, bzw. auf die Auswahl zur Bearbeitung.

Drei Methoden, die Auswahl eines Auftrags aus einer Warteschlange nach Prioritäten der Aufträge vorzunehmen, werden in dieser Fallstudie ([ACKE97]) betrachtet, und zwar: FIFO-Strategie, Fuzzy Strategie, binär-logische Strategie.

FIFO - Strategie

Das klassische FIFO-Prinzip: First In, First Out - Die Auswahl der Aufträge zur Bearbeitung ergibt sich automatisch aus der Reihenfolge des Eintreffens.

Fuzzy Strategie

Mithilfe einer Fuzzy Funktion wird aus der Warteschlange ein Auftrag ausgesucht, der nach derzeitigem Stand am dringendsten bearbeitet werden muß (charakterisiert über die

Priorität des Auftrages, einem reellwertigen Parameter). Die aktuelle Dringlichkeit (neue Priorität) ergibt sich aus der ursprünglichen Priorität des Auftrages (von der letzten Auswertung) und aus seiner bisherigen Wartezeit.

Tabelle 3-1 Regeln der Fuzzy Strategie

ursprüngliche Priorität	bisherige Wartezeit		
	kurz	mittel	lang
gering	gering	gering	mittel
mittel	gering	mittel	hoch
hoch	mittel	hoch	hoch

Der defuzzifizierte Wert (wieder ein reellwertiger Parameter, ermittelt nach der Schwerpunktmethode) wird als neue Priorität der Aufträge verwendet, nach der dann der „dringendste“ Auftrag ausgewählt wird. Die Berechnung der neuen Prioritäten aller in der Warteschlange befindlicher Aufträge findet jeweils vor der Auswahl eines Auftrages (bei Freiwerden einer Maschine) statt.

Die Beschreibung der Fuzzy Funktion erfolgt in übersichtlicher Form unter Angabe der Zugehörigkeitsfunktionen, der Regeln (Tab. 3-1) und der Defuzzifizierungsform, im „Hintergrund“ werden die Auswertung dieser Fuzzy Funktion durch (allgemeine) Module zur Tabellenverarbeitung durchgeführt. Der dahinterstehende Algorithmus braucht nicht beschrieben zu werden.

Binär-logische Strategie

Ähnlich der Fuzzy Strategie wird hier eine Strategie mit aktueller Priorität und bisheriger Wartezeit durchgeführt - mit herkömmlicher binärer Logik. Die Parameterbereiche werden dabei ebenfalls segmentiert, allerdings mit scharfen Rändern (Tab. 3-2).

Die Beschreibung und Abarbeitung dieser Regeln ist im Gegensatz zu einer Fuzzy Beschreibung sehr aufwendig, denn sie ist individuell zu erstellen, mit direkter Angabe des Algorithmus (Programmteil in C++, Tab. 3-3).

Tabelle 3-2 Segmentierung der Parameterbereiche bei der binär-logischen Strategie

Wartezeit	Bereich
kurz	[0, 3)
mittel	[3, 8)
lang	[8,)

Priorität	Bereich
niedrig	[0, 3]
mittel	[4, 6]
hoch	[7, 10]

Tabelle 3-3 Programmteil in C++; direkte Programmierung der binär-logischen Strategie

```
-- .fzy_warteschlange.regeln_auswert_crisp
.....
if( @.prioritaet < 3.5) then -- geringe prioritaet
  if( verweilzeit < str_to_time( "3:00:00" )) then
    @.neue_prioritaet := 3;
  elseif( verweilzeit < str_to_time( "8:00:00" ))
then
  @.neue_prioritaet := 3;
  else
    @.neue_prioritaet := 6;
  end;
elseif(@.prioritaet<7.5) then--mittlere prioritaet
  if( verweilzeit < str_to_time( "3:00:00" )) then
    @.neue_prioritaet := 3;
  elseif( verweilzeit < str_to_time( "8:00:00" ))
then
  @.neue_prioritaet := 6;
end;
```

I

In Experimenten mit verschiedenen Auftragsklassen zeigt sich, daß die Prioritätssteuerung auf Basis von Fuzzy Methoden gegenüber der Prioritätssteuerung mit klassischen logischen Methoden die Durchlaufzeiten der Aufträge hoher Priorität wesentlich und die der Aufträge mittlerer Priorität geringfügig verbessert. Gleiches gilt in beiden Fällen für die Abweichungen - es gibt weniger Ausreißer. Dies geht natürlich spürbar auf Kosten der Aufträge niedriger Priorität. Es verhält sich aber keine der Warteschlangen im Fuzzy Modus „schlechter“ als bei der binär-logischen Strategie, sondern es erfolgt eine Art „Ausgleich“. Die FIFO- Strategie weist generell längere Wartezeiten auf.

4 Fuzzy Logic - Entscheidungswerkzeug in diskreter Simulation

Welche Art der Modellbeschreibung auch gewählt wird (z.B. Ereignislisten, prozeßorientiert, objektorientiert), immer werden an einzelnen Punkten Entscheidungen über den weiteren Ablauf fallen müssen. Es gibt also Verzweigungen, an denen der Programmablauf mehrere verschiedene Fortsetzungen nehmen kann.

Traditionell gibt es in der diskreten Simulation drei Möglichkeiten, solche Entscheidungen zu fällen, Fuzzy Methoden eröffnen einen vierten Weg:

- deterministisch
- algorithmisch
- stochastisch
- *fuzzy*

4.1 Allgemeine Fuzzifizierung einer Entscheidungsfindung

Eine Fuzzy Funktion kann nun verwendet werden, um diese „vierte“ Art der Entscheidungsfindung zu realisieren. Diese kann entweder als Erweiterung der algorithmischen Methode oder als Mischung der algorithmischen und der stochastischen Methode gesehen werden ([LING96], [LING97]).

Die Fuzzy Funktion hat als Eingangsparameter (reellwertige) Attribute oder/und Werte von Systemgrößen, als Ausgangsparameter muß jedoch eine Entscheidung für eine bestimmte Auswahl (Pfad) stehen. Eine heuristische Betrachtung zeigt zunächst folgenden Unterschied zu einer Fuzzy Funktion mit klassischem regelungstechnischen Prinzip:

- Bei der Regelauswertung und Interferenz werden (auch in der Praxis) beliebige Operatoren auf Fuzzy Mengen verwendet.
- Ausgangsgrößen sind diskrete Werte, keine Fuzzy Mengen.
- Es gibt üblicherweise keine natürliche Ordnung auf den Ausgangsgrößen, auch wenn sie im Modell meistens durch natürliche Zahlen repräsentiert werden.

Aufgrund dieser Unterschiede müssen neue Formen der Defuzzifizierung verwendet werden. Zur besseren Veranschaulichung verwenden wir als Beispiel eine Verzweigung in einem diskreten System, wo n Pfade ($P_1, P_2, \dots, P_n$) wegführen. Es soll möglich sein, genau einen Pfad zu wählen, mehr als einen oder gar keinen. Die Regeln, nach denen die Pfade ausgewählt werden, müssen in der folgenden Form vorhanden sein:

$$\text{IF condition } c(i) \text{ THEN } P(i), \quad (i = 1, \dots, n) \quad (4.1)$$

Jede Bedingung liefert einen Fuzzy Wert in $[0,1]$. Es gibt nun mehrere Möglichkeiten, mit Hilfe dieser Fuzzy Werte Pfade auszuwählen (Defuzzifizierung):

1. **Schwerpunkt-Methode** (regelungstechnisch-klassisch):
Nur wenn es eine natürliche Ordnung auf den Pfaden gibt, und sie nicht willkürlich sortiert sind, kann es sinnvoll sein, ein diskretes Analogon zur Schwerpunkt-methode zu verwenden, nämlich den gewichteten Mittelwert. Es ist zu beachten, daß dabei die Pfade in der Mitte tendentiell öfter gewählt werden als die Pfade am Rand.
2. **Maximum-Methode** (klassisch):
Der Pfad mit dem größten Fuzzy Wert wird gewählt (mindestens ein Pfad).
3. **Mindestwert-Methode** (speziell):
Jeder Wert wird mit einer vorgegebenen Grenze verglichen. Der entsprechende Pfad wird gewählt, wenn der Fuzzy Wert größer oder gleich der Grenze ist (0 bis n Pfade).
4. **Stochastische Methode**:
Der Pfad wird zufällig ermittelt, wobei die Wahrscheinlichkeiten der einzelnen Pfade durch die Fuzzy Werte vorgegeben sind. Diese Methode liefert genau einen Pfad.

4.2 Stochastische Defuzzifizierung

Die stochastische Methode ist die allgemeinste und erlaubt eine Formalisierung und weitergehende Betrachtung.

Aus den Fuzzy Bedingungen $c(i)$ nach (4.1) definieren wir Werte $w(i)$:

$$s := \sum_{i=1}^n c(i), \quad w(i) := \frac{c(i)}{s}, \quad (i = 1, \dots, n) \quad (4.2)$$

Für die Werte $w(i)$ ergibt sich durch die Normalisierung mit s nach (4.2) die Eigenschaft

$\sum_{i=1}^n w(i) = 1$, sodaß wir die erhaltenen Werte $w(i)$ als Wahrscheinlichkeiten für die Pfade $P(i)$ verwenden können, indem wir jedem Pfad $P(i)$ entsprechend seiner Wahrscheinlichkeit ein Intervall $I(i) \subseteq [0,1]$ zuweisen:

$$I: \{1, \dots, n\} \rightarrow \{[a, b] \mid 0 \leq a < b < 1\}: \quad (4.3)$$

$$i \mapsto I(i) := [a, b], \text{ mit: } a = \sum_{k=1}^{i-1} w(k), \quad b = \sum_{k=1}^i w(k)$$

Mit Hilfe einer in $[0,1]$ gleichverteilten Zufallsvariablen ξ wählen wir nun einen Pfad, indem wir jenes i_0 bestimmen, für das $\xi \in I(i_0)$. Das ist klarerweise möglich, da gilt:

$$\forall \xi \in [0,1] \quad \exists! i_0 \in \{1, \dots, n\}: \xi \in I(i_0), \text{ weil} \quad (4.4)$$

$$\bigcup_{i=1}^n I(i) = [0,1], \text{ und } I(i) \cap I(j) = \emptyset \quad \forall i, j \in \{1, \dots, n\}, i \neq j$$

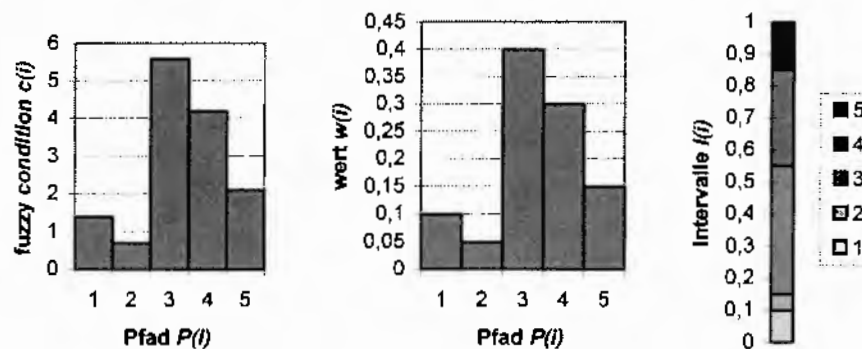


Bild 4-1 Veranschaulichung der Verfahrensweise bei der stochastischen Defuzzifizierung

Bild 4-1 veranschaulicht diese formale Beschreibung nach (4.1) bis (4.4) für einen Parameter. Mit dieser Art der Defuzzifizierung entsteht ein weiterer Berührungspunkt von Fuzzy Kalkül und Wahrscheinlichkeitstheorie (der erste Berührungspunkt besteht wohl in einem Vergleich der Zugehörigkeitsfunktion; siehe Kap. 7).

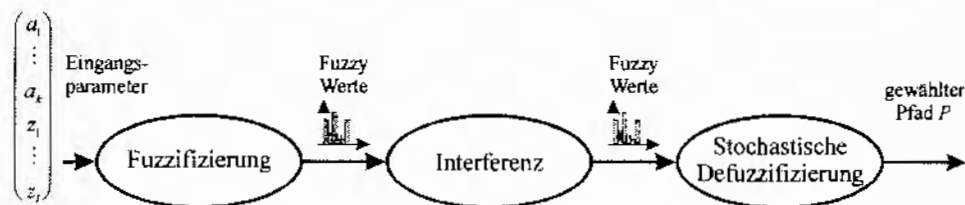


Bild 4-2 Fuzzy Funktion; Eingang: Attribute a_i , Zuständen z_i ; Ausgang: Pfade P_k

Damit ist eine Fuzzy Funktion konstruiert, die geordnete reelle Parameter (Attribute, Zustandsgrößen) auf nichtgeordnete Ausgangsparameter abbildet. Die Fuzzifizierung erfolgt in klassischer Weise, die Interferenz arbeitet nur mit Fuzzy Werten, nicht mit Fuzzy Mengen, was auch den Rechenaufwand enorm vermindert. Die Defuzzifizierung ist wie oben beschrieben „stochastisch“ (Bild 4-2).

4.3 Fallstudie

Eine Fallstudie untermauert die Aussage, daß die Fuzzy Entscheidung in der vorgestellten Form in ihrer „Effizienz“ zwischen algorithmischer und rein stochastischer Entscheidung liegt ([LING96]).

Fallstudie: Kunden verteilen sich auf 10 verschiedenen schnelle Bankschalter mit verschiedenen langen Warteschlangen (Bild 4-3). Die Kunden machen ihre Entscheidung von der Geschwindigkeit der Bankschalter und der Länge der Warteschlangen abhängig, wobei die Entscheidungsstrategie auf mehrere Arten formuliert wurde, unter anderem mittels Fuzzy Logik.

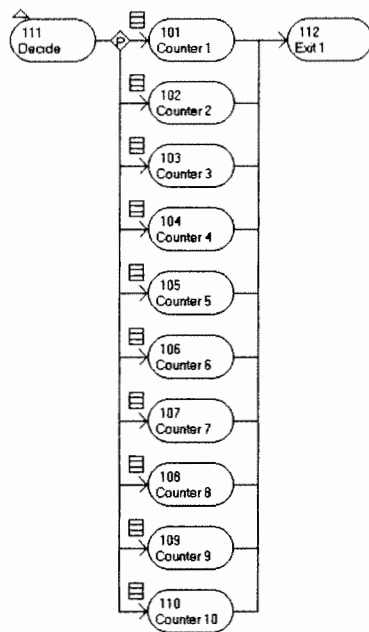


Bild 4-3 Fallstudie zum Einsatz von Fuzzy Logik als Entscheidungsstrategie

Tabelle 4-1 Ergebnisse mit verschiedenen Entscheidungsstrategien

zufällige Verzweigung			
	Länge der Schlange	Wartezeit	Leerzeit
Schnitt	334,33	388,97	0,57
Std.-Abw.	163,18	442,87	3,43
Maximum	550,00	1959,38	69,80
„beste“ Strategie (laut Erwartungswert)			
	Länge der Schlange	Wartezeit	Leerzeit
Schnitt	28,30	27,80	0,29
Std.-Abw.	21,10	20,95	1,66
Maximum	87,00	80,38	38,56
fuzzy			
	Länge der Schlange	Wartezeit	Leerzeit
Schnitt	50,48	55,53	0,35
Std.-Abw.	31,32	66,45	1,35
Maximum	124,00	407,13	23,31

Zwar liefert die (optimale) algorithmische Entscheidung bei den Verzweigungen natürlich die besten Lösungen (Tab. 4-1), aber für diese Methode stehen in der Praxis fast niemals ausreichend viele Daten zur Verfügung (im Prinzip müsste jede individuelle Bedienzeit a priori bekannt sein). Die Ergebnisse mit der wirklichkeitsnäheren Fuzzy Strategie sind besser als eine rein stochastische Strategie.

5 Fuzzy Logic in Modellen mit Markov-Ketten

Markov-Ketten sind ein klassisches Mittel zur Beschreibung endlicher stochastischer Automaten. Markov-Ketten können Systeme beschreiben, die nur endlich viele Zustände annehmen, und deren Verhalten durch Abfolgewahrscheinlichkeiten dieser Zustände beschrieben wird.

Der Systemzustand zur Zeit t wird durch die *Zustandsvariable* $z(t)$ ausgedrückt, die Übergangswahrscheinlichkeiten für den Zustand im nächsten Schritt stehen in der *Übergangsmatrix* $A=(a_{ij})$. Dabei gilt:

$$p(z(t+1) = j \mid z(t) = i) = a_{ij} \quad (5.1)$$

5.1 Markov-Ketten in diskreter Modellbildung und Simulation

Damit Markov-Ketten in der diskreten Modellbildung und Simulation eingesetzt werden können, müssen drei Bedingungen erfüllt sein:

- Das System muß getaktet sein (diskrete Zeit).
- Das System darf nur endlich viele Zustände besitzen.
- Für jeden Zustand müssen die Wahrscheinlichkeiten aller möglichen Nachfolgezustände erfaßbar sein.

Die oben erwähnten Bedingungen sind bei näherer Betrachtung allerdings keine ernsthaften Einschränkungen:

- Eine diskrete Zeit ist auch bei konventionellen Simulatoren de facto gegeben, da Digitalrechner eben nur eine gewisse Rechengenauigkeit haben. Darüber hinaus ist es gar nicht sinnvoll, diese Kapazität voll auszunutzen, da die Genauigkeit der Ergebnisse durch die ungenauen (fehlerbehafteten) Eingangsdaten ohnedies begrenzt ist.
- Die Einschränkung auf endlich viele Zustände kann durch die Einführung von parametrisierten Markov-Ketten umgangen werden. Der momentane Systemzustand wird dabei nicht nur durch den Zustand der Markov-Kette, sondern auch von Parametern bestimmt. Die Übergangsmatrix hängt dann von diesen Parametern ab.
- Die Übergangswahrscheinlichkeiten müssen aus der Beschreibung des Systems berechenbar sein, andernfalls wäre die Beschreibung auch für eine herkömmliche Simulation zu ungenau.

Die Modellbildung und Simulation mit und die Analyse von Markov-Ketten fällt teilweise in den Bereich der symbolischen Verfahren. Weiters ist zu beachten, daß die Matrix der Übergangswahrscheinlichkeiten sehr groß werden kann (und in der Praxis auch wird). Daher wird die Beschreibung mit Markov-Ketten nun wieder zu einer echten Alternative, da hinreichend große Rechnerleistung immer leichter verfügbar wird und entsprechende Software zur Verfügung steht.

Es ist oft ein erheblicher Aufwand, die (wie schon erwähnt, meist sehr große) Übergangsmatrix aufzustellen. Es ist daher ratsam, über Strategien zur Vermeidung von unnötigem Aufwand nachzudenken. Das bedeutet vor allem, daß die Anzahl der Zustände gering gehalten werden sollte, da der Aufwand quadratisch mit dieser Anzahl steigt. Dazu gibt es mehrere Wege:

- Das System sollte nur so detailgetreu wie unbedingt notwendig abgebildet werden. Ähnliche Zustände, deren Unterscheidung sich nicht lohnt, können entweder durch Analyse und Überlegung gefunden werden, oder numerisch, indem die Wahrscheinlichkeiten ihrer Nachfolgezustände verglichen werden.
- Die Ausnützung von Symmetrien kann die Anzahl der Zustände auf einen Bruchteil reduzieren. Außerdem gibt es oft Zustände, die zwar theoretisch möglich sind, aber in der Praxis unter bestimmten Rahmenbedingungen nicht eintreten.

Diese beiden Punkte entsprechen der Bildung von Äquivalenzklassen auf der Menge der Zustände. Die entstehenden vereinfachten Markov-Ketten sind immer noch klassische Markov-Ketten und können mit klassischen, analytischen Methoden bearbeitet werden.

5.1 Parametrisierte Markov-Ketten in diskreter Simulation

Weitere Möglichkeiten zur Verminderung der Anzahl der Zustände sind:

- Ähnliche Zustände können zu einem einzigen zusammengefaßt werden, wenn sie durch Einführung eines Parameters immer noch unterscheidbar sind. Die Matrix wird dadurch zwar kleiner, ist dann aber parametrisiert.
- Für parallel ablaufende Prozesse müssen in einer Markov-Kette alle möglichen Zustandskombinationen berücksichtigt werden. Statt dessen können aber auch mehrere Markov-Ketten betrachtet werden, wobei der Zustand der jeweils anderen Ketten als Parameter eingeht.

Bei diesen beiden Methoden ist zu beachten, daß die dabei entstehenden Konstruktionen über die klassische Theorie der Markov-Ketten hinausgehen, wodurch sie kaum mehr analytisch zu behandeln sind. Es bleibt praktisch nur noch, sie als Simulation im Zeitbereich ablaufen zu lassen.

Zusätzlich kommt bei diesen parametrisierten Markov-Ketten noch die Schwierigkeit hinzu, daß sämtliche Übergangswahrscheinlichkeiten parameterabhängig formuliert werden müssen. Bei unzureichender Information oder einfach nur, um den Aufwand zu reduzieren, liegt nun der Gedanke nahe, diese Abhängigkeit mit den Mitteln der Fuzzy Logik zu formulieren, indem die in Kap. 4 vorgestellte Fuzzy Funktion modifiziert wird.

5.3 Fallstudie „Dining Philosophers’ Problem“

Vor einer Formalisierung soll die Ermittlung von Fuzzy Übergangswahrscheinlichkeiten am „Dining Philosophers’ Problem“ exemplarisch betrachtet werden. Dieses an sich einfach zu formulierende System wirft erstaunlich komplexe Fragen auf, wie Gleichzeitigkeit, Deadlock, etc. ([BREI96]). Es kann mit Hilfe klassischer Markov-Ketten sehr gut analysiert werden ([KISS97]). Aber die Betrachtung mittels paralleler Markov-Ketten, mit oder ohne zusätzlichem Parameter, führt zu einer genaueren Analyse.

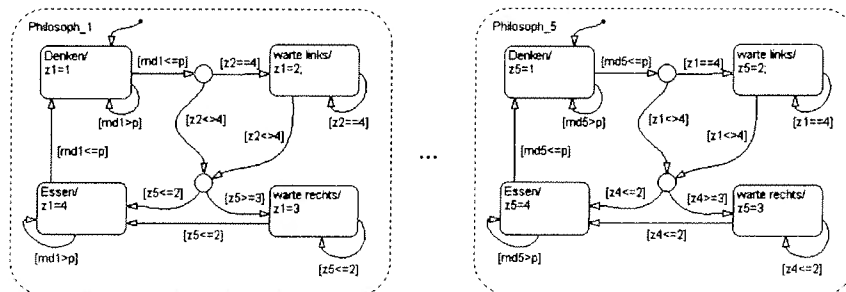


Bild 5-1 Modell des Dining Philosophers’ Problem mit parallelen Markov-Ketten (Stateflow- Darstellung in SIMULINK 2.1)

Das Stateflow - Modell (Bild 5-1) bildet das Verhalten jedes einzelnen Philosophen durch eine eigene Markov-Kette ab. Der Zustand des Denkens bzw. des Essens wird in jedem Schritt mit einer gewissen (konstanten) Wahrscheinlichkeit verlassen. Diese stellen allerdings eine Vereinfachung gegenüber dem ursprünglichen Modell dar.

Um diese Vereinfachung zu vermeiden, müßte die Zeit, die der Zustand bereits andauert, in die Wahrscheinlichkeit eingehen, was zu einer Parametrisierung und damit zu sehr komplexen Algorithmen für die Übergangswahrscheinlichkeit führt.

Dieser Zusammenhang kann allerdings relativ einfach mittels Fuzzy Logik verbal formuliert werden („Je länger der Zustand dauert, desto größer ist die Wahrscheinlichkeit Verlassen“). Das kann zunächst mit einer Zugehörigkeitsfunktion (Bild 5-2) formuliert werden, die zusammen mit einem (hier einfachen) Satz von Regeln für jeden möglichen Nachfolgezustand die Übergangswahrscheinlichkeit liefert (Tab. 5-1).

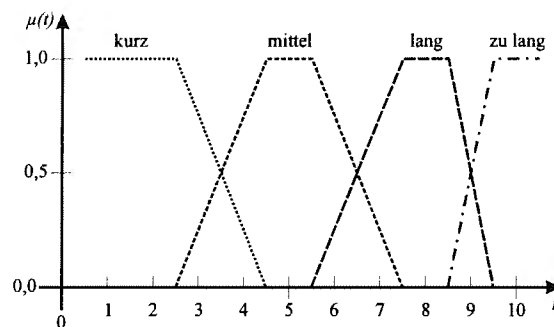


Bild 5-2 Dining Philosophers Problem: Zugehörigkeitsfunktionen für Verweilzeiten in den Zuständen

Tabelle 5-1 Fuzzy-Regelsystem für Zustand „Denken“

Nachfolgezustand	verstrichene Zeit	Gewicht	Nachfolgezustand	verstrichene Zeit	Gewicht
Denken	kurz	0,9	warten oder Essen	kurz	0,1
Denken	mittel	0,6	warten oder Essen	mittel	0,4
Denken	lang	0,3	warten oder Essen	lang	0,7
Denken	zu lang	0,0	warten oder Essen	zu lang	1,0

Die in Tab. 5-1 zugeordneten Gewichte können nun nach Normalisierung als Übergangswahrscheinlichkeiten interpretiert werden. Die Auswertung kann entweder während des Modellablaufs geschehen (langsam) oder vorher für alle möglichen Eingangswerte, was die Markov-Kette mit Fuzzy Logik wieder in eine gewöhnliche parametrisierte Markov-Kette übersetzen würde.

5.4 Formalisierung von Fuzzy Übergangswahrscheinlichkeiten

Die zuvor vorgestellte Fuzzy Berechnung von Übergangswahrscheinlichkeiten kann nun ähnlich zu der Fuzzy Methode in Kap. 4 zu einer Fuzzy Funktion formalisiert werden.

Als Eingangsgrößen werden nun der augenblickliche Zustand z_i und Parameter (die selbst andere Zustände z_j sein können) verwendet und klassisch fuzzifiziert. Die Interferenz und die Defuzzifizierung arbeiten mit diskreten Fuzzy Werten. Wesentlich sind die Gewichte in den Regeln (Tab. 5-1), die im Prinzip die nichtnormalisierten Wahrscheinlichkeiten darstellen.

Auf diese Gewichte baut die Defuzzifizierung auf, die nun in Abänderung zur Fuzzy Methode in Kap. 4 nur mehr eine Normalisierung durchzuführen hat, wodurch als Ausgangs-

parameter der Fuzzy Funktion die Zeile (a_{ik}), $k=1, \dots, n$ der Übergangswahrscheinlichkeiten ermittelt wird (Bild 5-3).

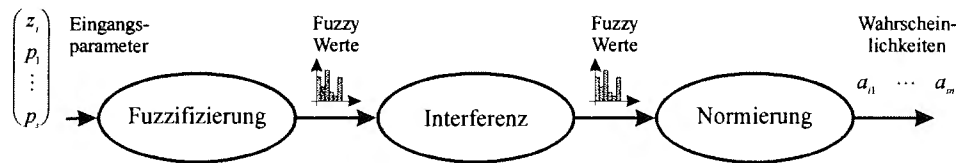


Bild 5-3 Fuzzy Funktion: Eingang: Zustand z_i , Parameter p_k ; Ausgang: Zeile der Übergangswahrscheinlichkeiten (a_{ij})

6 Implementation von Fuzzy Methoden in Simulatoren

Im Prinzip sind die Fuzzy Methoden nur eine vereinfachte Beschreibung komplexer Zusammenhänge, wobei die Komplexität in den Hintergrund eines variabel-mehrwertigen logischen Kalküls verlagert wird und im Vordergrund eine komfortable Beschreibung mit umgangssprachlichen Kenngrößen angeboten wird. Jede Fuzzy Regel ist auch „klassisch“ individuell mit IF-THEN-ELSE-Konstruktionen beschreibbar - aber nur viel mühsamer.

6.1 Verfügbarkeit von Implementierungen

Kontinuierliche Simulatoren stellen mehrheitlich bereits Module für die Beschreibung (Fuzzy Rule Editor) und implementierte allgemeine Algorithmen für die Fuzzifizierung, Interferenz und Defuzzifizierung zur Verfügung. Diese Algorithmen sind im Prinzip komplexe mehrdimensionale Table Lookup-Verfahren, die effizient implementiert sein sollten. Diskrete Simulatoren lassen aber derartige Werkzeuge nahezu vollständig vermissen - sie müssen noch selbst programmiert und implementiert werden. Dabei geht der Anwender einen schmalen Grat zwischen zu individueller, aber rascher und zu allgemeiner und daher langer Implementierung.

Da die Programmiersprachen, die von Simulationssystemen zur Implementierung von Zusatzmodulen zur Verfügung gestellt werden, fast ausschließlich interpretativ abgearbeitet werden, ist die Ausführung der Fuzzy Methoden extrem langsam, was vor allem an ebendiesen Tabellenzugriffen liegt.

6.2 Fallstudien: Effizienz der Implementierungen

Die Fallstudien in Kap. 3 und Kap. 4 zeigen dieses Problem auf. Die Fallstudie „Jobshop Problem“ wurde in SIMPLE++ implementiert. Ein Simulationslauf mit der Fuzzy Strategie dauert etwa 25 mal so lang wie einer mit FIFO Strategie, während die binär-logische und direkt programmierte Strategie die Rechenzeit „nur“ verdoppelt.

Ein Test zeigt, daß die Rechenzeit offensichtlich bei den Tabellenzugriffen verbraucht wird. So benötigt SIMPLE++, um eine 100x100-Matrix mit Integerwerten zu füllen (keine großartige Berechnung - nur die Summe aus Spalten- und Zeilennummer), einige Sekunden - dieselbe Aufgabenstellung kann von einer übersetzbaren Programmiersprache in Millisekunden, wenn nicht sogar weniger, bewältigt werden. Ein C - Interface, das nur zusätzlich erhältlich ist und für die Fallstudie nicht verfügbar war, könnte hier teilweise

Abhilfe schaffen, allerdings wird dennoch jeder Aufruf der Fuzzy Funktion zunächst interpretativ behandelt, bevor ein übersetzter Modul die Tabellenauswertung bearbeitet.

Die Fallstudie „Bankschalter“ wurde in MicroSaint und GPSS/H implementiert. MicroSaint arbeitet ebenfalls interpretativ, was zu einer etwa zehnfachen Rechenzeit bei einer Fuzzy Strategie im Vergleich zur rein stochastischen Strategie führte (wie im Falle von SIMPLE++ eine allgemeine Implementierung zur Beschreibung und Auswertung von Fuzzy Funktionen).

GPSS/H ist eine der wenigen übersetzend arbeitenden diskreten Simulatoren. In GPSS/H wird daher ein Modul zur Verarbeitung einer Fuzzy Funktion mitübersetzt. Damit ergab sich bei der Fuzzy Strategie eine nur geringfügig längere Rechenzeit als bei der stochastischen Strategie. Allerdings ist zuzugeben, daß in GPSS/H bereits wie in einer Programmier- und nicht wie in einer Simulationssprache zu arbeiten ist.

6.3 Implementierungsanforderungen

Für eine vernünftige Anwendung einer Implementierung einer Fuzzy Methode, die über ein bloßes Austesten der Anwendbarkeit hinausgeht, führt daher kein Weg an einer übersetzbaren Implementierung in einer Programmiersprache, etwa C oder FORTRAN, vorbei.

Bei einer vernünftigen Implementierung, die auch auf die speziellen Bedürfnisse der diskreten Simulation wie stochastische Defuzzifizierung eingeht, ergeben sich allerdings Vorteile für komplexe Studien, Experimente, etc. Aus „philosophischer“ Sicht ist auch festzustellen, daß eine Anwendung der Fuzzy Methoden über die klassische, der regelungstechnischen Vorgangsweise angelehnten Form, hinaus eine Erweiterung des Experimenthorizonts mit sich bringt.

7 Fuzzy Methoden und Wahrscheinlichkeitstheorie

Bei Betrachtung einer Zugehörigkeitsfunktion stellt sich dem Fuzzy Methoden - Novizen bald die Frage nach dem Zusammenhang zur Wahrscheinlichkeitstheorie. Insbesondere die Aussage „a gehört mit 0.4 zur Menge A“ wird leicht als „a gehört mit 40%iger Wahrscheinlichkeit zur Menge A“ falsch interpretiert.

7.1 Überschneidung der Kalküle

Je nach Sichtweise kann man in der Tat Fuzzy Kalkül und Wahrscheinlichkeitskalkül einander untergeordnet bzw. verzahnt betrachten:

- Aus theoretischer Sicht sei zunächst auf die Maßtheorie, die Grundlage der Wahrscheinlichkeitstheorie, verwiesen. Die Maßtheorie erfuhr Erweiterungen um Fuzzy Maße (z.B. [KLEM82a]), und statistische Grenzwertsätze können für Fuzzy Zufallsvariable angegeben werden (z.B. [KLEM86]).
- Betrachtet man spezielle normierte Zugehörigkeitsfunktionen, so können äquivalente Beschreibungen mit bedingten Wahrscheinlichkeiten angegeben werden.

- Interessant ist auch der Ansatz „Möglichkeit“ (*possibility* $p_o(a)$) als Maß für einen Fuzzy Wert a im Gegensatz zur Wahrscheinlichkeit (*probability* $p_r(a)$), siehe [BAND93].
- Der in den Kapiteln 4 und 5 vorgestellte Weg zum Aufbau von Fuzzy Funktionen „arbeitet“ mit einer Verzahnung der beiden Kalküle.

Der Grund für den scheinbaren Gegensatz, für die scheinbare Vereinnahmung etc. beider Kalküle liegt allerdings im „Phänomen“ der Wahrscheinlichkeitstheorie, zwar alle „Unschärfen“ erfassen zu wollen, aber doch nicht zu können.

7.2 Unzulänglichkeit der Wahrscheinlichkeitstheorie

Um nähere Einsicht in die Problematik zu gewinnen, seien hier ausschnittsweise die Ausführungen in [BAND93] zusammenfassend zitiert und dem Interessierten zum weiteren Quellenstudium empfohlen.

Seit Kolmogorov die Wahrscheinlichkeitstheorie durch seine berühmten Axiome für Wahrscheinlichkeitsmaße in den Kanon der mathematischen Wissenschaften aufgenommen hat, ist es eine weitverbreitete Meinung in der Mathematik genauso wie in ihren Anwendungen, daß diese Theorie *jede* Art von Unschärfe gut zu modellieren vermag.

Nach [DUBO85] muß dieser Meinung widersprochen werden: Das fundamentale Axiom ist das der Additivität von Wahrscheinlichkeiten disjunkter Ereignisse. Nach der klassischen Definition der Wahrscheinlichkeit, basierend auf dem Abzählen von Möglichkeiten, (Quotient der günstigen durch die möglichen Fälle), und auch nach dem frequentistischen Wahrscheinlichkeitsbegriff ([MISE19]), ist diese Additivität plausibel.

Allerdings postulieren diese Ansätze u. a. die unendliche Wiederholbarkeit eines Experiments unter den selben Bedingungen. Um die Schwierigkeiten mit diesem Postulat bei der Behandlung von realen Problemen zu vermeiden, interpretieren Verfechter der Theorie der subjektiven Wahrscheinlichkeit die Wahrscheinlichkeit als *Maß des Gefühls der Unsicherheit*, zurückgreifend auf sehr frühe Ideen der Wahrscheinlichkeitsrechnung ([LEIB], [COUR43]). „Um dieses Gefühl der Unsicherheit in Zahlen auszudrücken, wird die Wahrscheinlichkeit eines Ereignisses als jene Quote definiert, die ein Spieler in Kauf zu nehmen bereit ist, wenn er wettet, daß dieses Ereignis nicht stattfindet. Wenn diese Summe existiert, kann gezeigt werden, daß die so erhaltene Wahrscheinlichkeit den Axiomen der Wahrscheinlichkeitstheorie genügt, ein „rationales“ Verhalten vorausgesetzt ([SAVA72])“.

Darauf aufbauend gelang es den Subjektivisten, zu zeigen, daß die Axiome von Kolmogorov die einzig vernünftige Basis für die mathematische Beschreibung von subjektiver Unsicherheit wäre.

Dieser Standpunkt ist von einem philosophischen und von einem praktischen Standpunkt aus anfechtbar:

„So ist es etwa gewagt zu behaupten, jedes unscharfe Schließen gehorche den Regeln des Wettens. Das nötige monetäre Engagement, das ein essentieller Teil des Modells darstellt, könnte den Spieler aus Angst vor finanziellen Verlusten davon abhalten, seinen wahren Wissensstand bekanntzugeben.“

Ein professioneller Spieler wird seine Einsätze gleichmäßig verteilen, wenn er weiß, daß alle Wetten gleich gut sind. Bei Fehlen jeglicher Information wird sich der Neuling genauso verhalten, da es die vernünftigste Strategie ist. Die Theorie der subjektiven Wahrscheinlichkeit erlaubt keine Unterscheidung zwischen diesen beiden Informationszuständen und ist auf Fälle, wo der Informationsstand sehr gering ist, nur schwer anwendbar“.

Besonders der Fall totaler Unwissenheit wird vom wahrscheinlichkeitstheoretischen Modell schlecht erklärt, da es die Identifikation einer Menge gegenseitig disjunkter Ereignisse voraussetzt. Diesen wird ja im endlichen Fall allen die gleiche Wahrscheinlichkeit zugeordnet. Im Fall totaler Unwissenheit ist aber anzunehmen, daß schon die Bestimmung der Anzahl der möglichen Ereignisse unmöglich ist. Daher ist es auch fragwürdig, das Maß der Unsicherheit von der Anzahl der Alternativen abhängig zu machen, wie das bei Wahrscheinlichkeiten der Fall ist.

Vom praktischen Standpunkt aus gesehen ist es klar, daß die Zahlen, die von Menschen zur Beschreibung ihres Wissensstandes in Form von Wahrscheinlichkeiten benutzt werden, als das gesehen werden müssen, was sie sind, nämlich ungefähre Hinweise. Die Theorie der subjektiven Wahrscheinlichkeit scheint sich mit dieser Art der Ungenauigkeit nicht weiter aufzuhalten, sondern davon auszugehen, daß rational denkende Menschen eben in der Lage sein müßten, genaue Zahlen zu liefern, wenn man nur die richtigen Methoden verwendet, um sie ihnen herauszulocken.

„Daher scheint die Wahrscheinlichkeitstheorie zu normativ, um alle Arten des unscharfen Schließens gut zu beschreiben“.

Soweit die Argumentation nach [DUBO85] und [BAND93], der wir hier folgen und die zeigt, daß die Unschärfe (Fuzzy Phänomene) nicht mit Methoden der Wahrscheinlichkeit erfaßt werden kann - vor allem auch aus praktischer Sicht. Ein Vergleich mit der Modellbildung kontinuierlicher Systeme drängt sich auf, wo die lineare Theorie eben nicht für die Beschreibung aller Phänomene ausreicht (vgl. Additivitätsforderung für Wahrscheinlichkeitsmaße).

Allerdings macht es durchaus Sinn, sich bei Anwendung von Fuzzy Methoden der Wahrscheinlichkeitstheorie zu bedienen, was z. B. beim Einsatz in der diskreten Modellbildung und Simulation zu einer „Horizontenerweiterung“ führt.

Literatur

- [ACKE97] Ackerl W.: Praktikumsarbeit. Abt. Simulationstechnik, Technische Univ. Wien, 1997.
- [BAND93] Bandemer H., Gottwald S.: Einführung in die Fuzzy Methoden. Akademie Verlag GmbH, Berlin, 1993.
- [BREI96] Breitenecker F., Schmidt B.: Comparison 10: Dining Philosophers II. SNE EUROSIM Simulation News Europe 18, 1996, 32-33.
- [COUR43] Cournot A.A.: Exposition de la théorie des chances et des probabilités. Paris, 1843.
- [COX94] Cox E.: The Fuzzy Systems Handbook. Academic Press, Cambridge, Massachusetts, 1994.

- [DUBO85] Dubois D., Prade H.: *Théorie des Possibilités: Application à la Représentation des Connaissances en Informatique*. Masson, Paris 1985
(Engl. Übersetzung: *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. Plenum Press, New York, 1988).
- [HOFE94] Hofer J.: *Fuzzy-Logik simulieren und erfolgreich anwenden*. Franzis-Verlag GmbH, München, 1994.
- [KISS97] Kiss C., Lingl M.: A Markov Chain Approach to the Dining Philosophers' Problem. SNE EUROSIM Simulation News Europe 21, Nov. 1997, im Erscheinen.
- [KLEM82] Klement E.P., Schwyhla W.: Correspondence between fuzzy measures and classical measures. *Fuzzy Sets and Systems* 7, 1982, 57-70.
- [KLEM86] Klement E.P., Puri M.L., Ralescu D.A.: Limit theorems for fuzzy random variables. *Proc. R. Soc. Lond. A* 407, 1986, 171-182.
- [LEIB] Leibniz G.W.: Brief an Bernoulli vom 3.12.1703. In: Gebhardt (Hrsg.): *Mathematische Schriften III/1*, Halle, 1855.
- [LING96] Lingl M.: *Fuzzy-Logic in Discrete Simulation, Theoretical Considerations - Implementation and Case Study in Micro Saint*. Diplomarbeit TU Wien, 1996.
- [LING97] Lingl M., Breiteneker F.: Fuzzy Logic - A New Decision Strategy in Discrete Simulation. In: Troch I., Breiteneker F. (Hrsg.): *Proc. Second MATHMOD Vienna*. February 1997.
- [MISE19] von Mises R.: *Grundlagen der Wahrscheinlichkeitsrechnung*. *Math. Zeitschr.* 5, 1919, 52-99.
- [MÖLL95] Möller D.P.F.: Fuzzy Systems in Modelling and Simulation. In: Breiteneker F., Husinsky I. (Hrsg.): *Proc. EUROSIM'95*, Elsevier Science, Amsterdam 1995, 65-74.
- [PLAN97] Plank J., Breiteneker F.: Beschreibung von State Events in kontinuierlicher Modellbildung und Simulation - Klassifikation und neuer methodologischer Zugang. In diesem Tagungsband.
- [SALZ95] Salzmann M., Breiteneker F.: Genetic Algorithms in Discrete Event Simulation. In Breiteneker F., Husinsky I. (Hrsg.): *Proc. EUROSIM '95*, Elsevier Science, Amsterdam 1995, 213 - 218.
- [SALZ96] Salzmann M.: Genetische Algorithmen in diskreter Simulation. *ASIM Fortschrittsberichte Simulation Bd. 2*, Wien, 1996.
- [SAVA72] Savage L.J.: *The Foundations of Statistics*. Dover Publications, New York, 1972.
- [TILL92] Tilly T.: *Fuzzy-Logik: Grundlagen, Anwendungen, Hard- und Software*. Franzis-Verlag GmbH, München, 1993.
- [ZADE65] Zadeh L.A.: Fuzzy Sets. *Information and Control* 8, 1965, 338-353.
- [ZADE68] Zadeh L.A.: Fuzzy Algorithms. *Information and Control* 12, 1968, 94-102.
- [ZADE84] Zadeh L.A.: Making computers think like people. *IEEE. Spectrum* 8/1984, 26-32.

Datenanalyse und Data Mining bei Business Intelligence Anwendungen



Datenanalyse und Data Mining bei Business Intelligence Anwendungen

- Vorstellung der MIT GmbH
- Anwendungen im Business-Bereich
- Datenauswertung im Business-Bereich - Ist-Situation und Potentiale
- Vorführung von DataEngine
- Anwendungen von Data Warehouse und Data Mining
- Competence Center Data Mining



Vorstellung der MIT GmbH

Firmenprofil der MIT GmbH



MIT - Management
Intelligenter
Technologien GmbH

- gegründet 1991
- 35 interdisziplinär ausgebildete Mitarbeiter
- Sitz in Aachen
- Systemhaus für intelligente Technologien (Fuzzy Technologien, Neuronale Netze, Machine Learning)
- Internationale Kooperationen mit führenden Anbietern und Nutzern aus unterschiedlichen Bereichen

Präsentation CRM 6

09.11.98

© MIT-Management
Intelligenter Technologien GmbH
Aachen, Germany



Vorstellung der MIT GmbH

Das MIT - Leistungsspektrum



Professional Services

- Projektrealisierung
- Softwarelösungen
- Machbarkeitsuntersuchungen

Softwarewerkzeuge

- DataEngine
- DataEngine ADL
- DataEngine V.i
- WINROSA
- Plugins



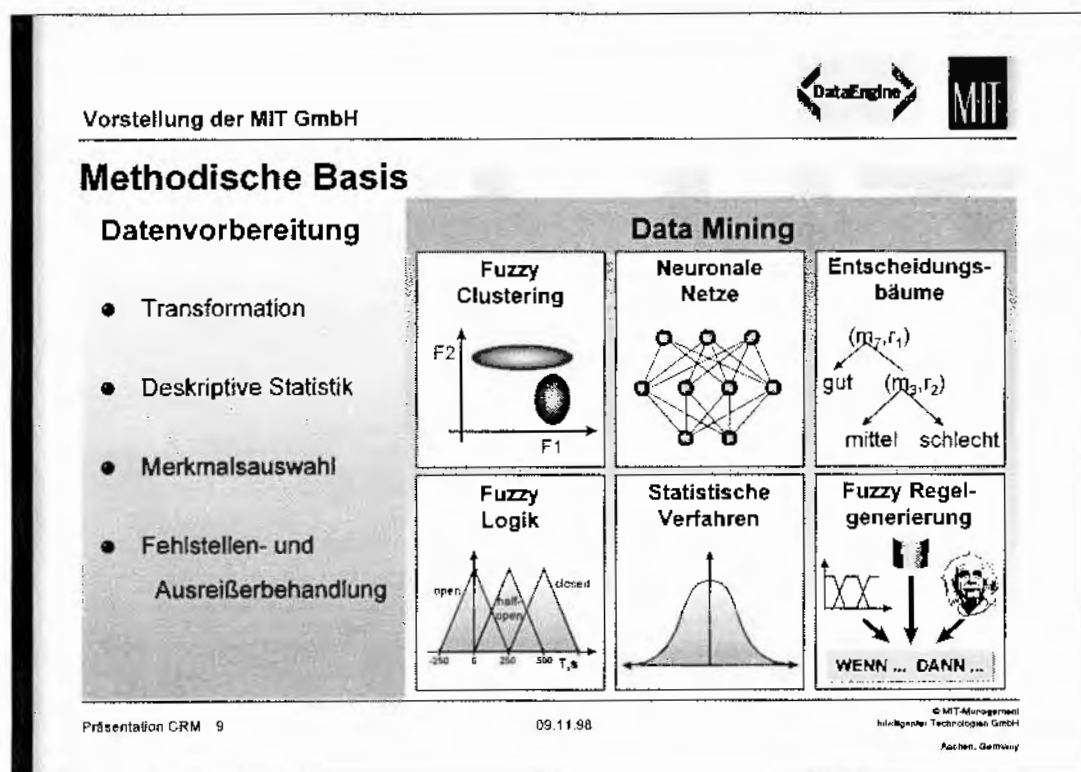
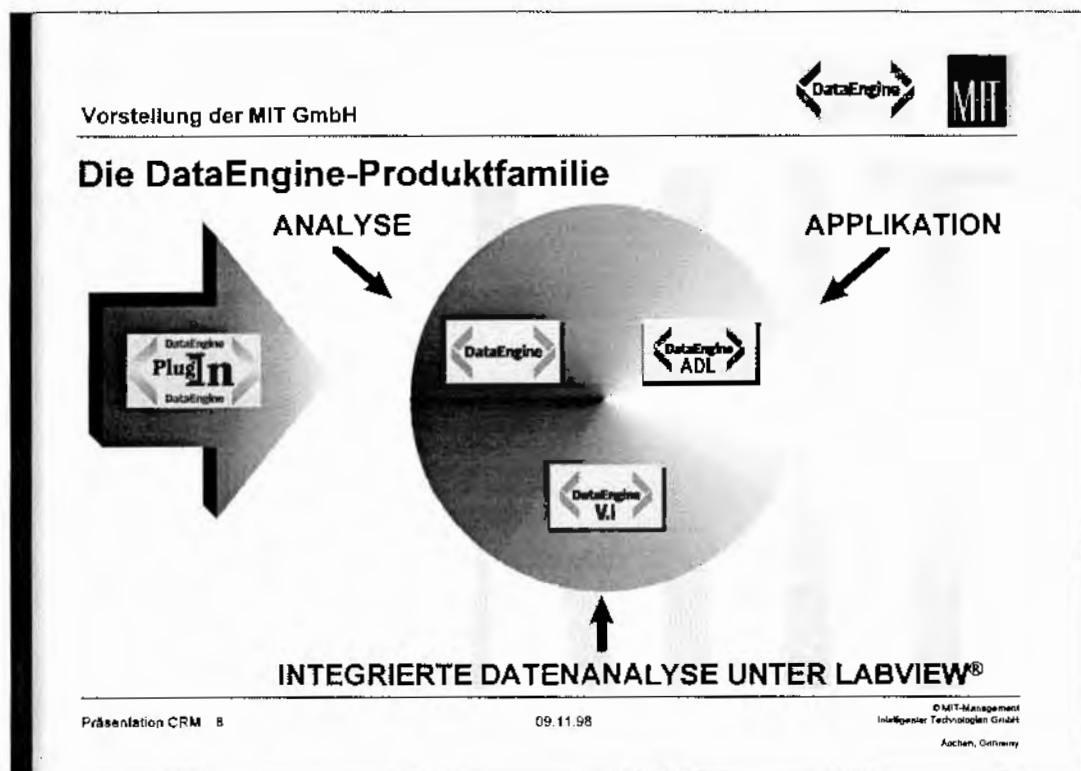
Transfer

- Konferenzen
- Symposien
- Schulungen

Präsentation CRM 7

09.11.98

© MIT-Management
Intelligenter Technologien GmbH
Aachen, Germany






Anwendungen im Business-Bereich

Controlling

- Analyse und Reporting  Deutsche Post
- Kosten / Nutzen -Relation
 - Vergleich von Filialen / Regionen
- Planung und Vorgabe
 - Absatzprognose  Bayer
 - Vorhersage Kapazitätsbedarf (Anlagen, Standorte) zur Ermittlung von Zeitpunkt und Volumen der Wiederbeschaffung

hals Infracor

Präsentation CRM 10 09.11.98 © MIT-Management
Intelligent Technologies GmbH
Aachen, Germany




Anwendungen im Business-Bereich

Customer Relationship Management

- Kundensegmentierung und -profilierung  Sparkasse
- Abwanderungsanalyse  Sparkasse
- Warenkorbanalyse  Werk In uitvoering
- Responseanalyse  oppermann
- Steuerung des Außendienstes  bofrost 

Präsentation CRM 11 09.11.98 © MIT-Management
Intelligent Technologies GmbH
Aachen, Germany




Datenauswertung im Business-Bereich

Nutzen von Data Mining

- **Effizienzsteigerung**
 - z.B. durch höhere Responseraten
- **Kostenreduktion**
 - z.B. durch Identifikation unrentabler Produkte
- **Höhere Kundenzufriedenheit**
 - z.B. durch individuell abgestimmte Information

Präsentation CRM 12

09.11.98

© MIT-Management
Intelligenter Technologien GmbH
Aachen, Germany




Datenauswertung im Business-Bereich

Ist-Situation

- Verteilte Datenbanken enthalten (fast) alle relevanten Daten
- Abfragen werden manuell konfiguriert,
aber: Wissen über die Zusammenhänge fehlt
- Datenanalyse $\cong$ deskriptive Statistik / OLAP

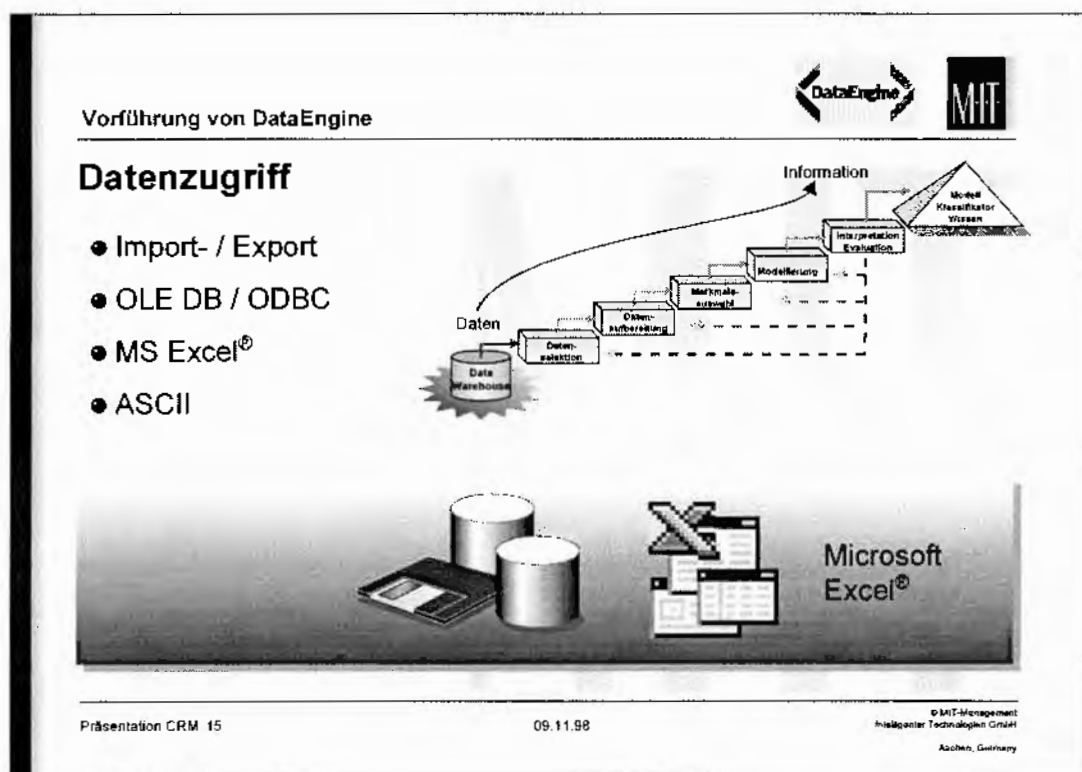
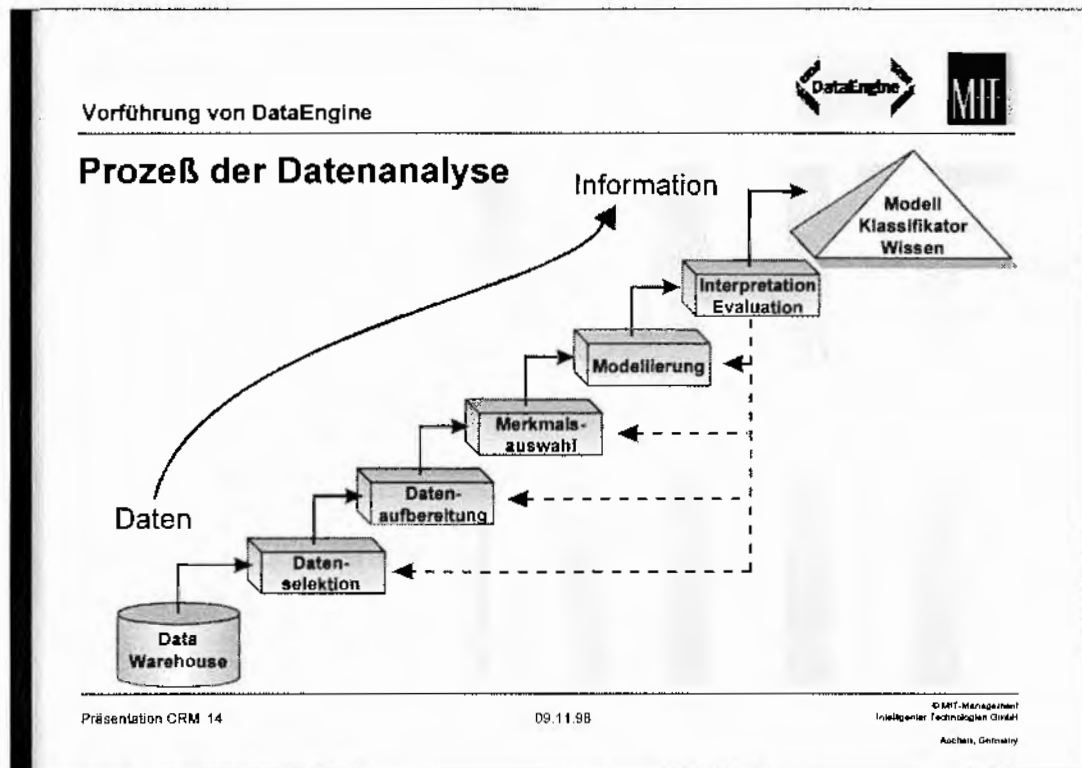
Potentiale

- automatische Extraktion von Wissen
- Data Mining findet Zusammenhänge
- Auswertung von Feedback-Information

Präsentation CRM 13

09.11.98

© MIT-Management
Intelligenter Technologien GmbH
Aachen, Germany



DataEngine MIT

Vorführung von DataEngine

Datenbearbeitung

- Statistik
- Fehlstellenbehandlung
- Signalverarbeitung
- Mathematik

Präsentation CRM 16 09.11.98 © MIT-Management Intelligente Technologien GmbH Aachen, Germany

DataEngine MIT

Vorführung von DataEngine

(Automatisierte) Merkmalsauswahl

- FeatureSelector
- Pruning
- Entscheidungsbaum
- Analyse

Präsentation CRM 17 09.11.98 © MIT-Management Intelligente Technologien GmbH Aachen, Germany

Vorführung von DataEngine

Datenanalyse & Data Mining

- Neuronale Netze
- Regelbasis
- Fuzzy Clustering
- Entscheidungsbaum
- Regelgenerierung
- Plugins

Präsentation CRM 18

09.11.98

© MIT-Management
Intelligent Technology GmbH
Aachen, Germany

Vorführung von DataEngine

Interpretation

- Visualisierung
- Karten
- Automatisierung
- Stapelverarbeitung

Präsentation CRM 19

09.11.98

© MIT-Management
Intelligent Technology GmbH
Aachen, Germany

Vorführung von DataEngine

Modellanalyse

- What-if-Analyse / Simulation
- Transferfunktionen
- Sensitivitätsanalysen
- Gütemaße

The diagram illustrates the DataEngine workflow: **Daten** (Data) from a **Data Warehouse** undergo **Daten-selektion** (Data selection), **Daten-Substitution** (Data substitution), and **Modellbildung** (Model building) to produce **Information**. This information is then used for **Modell-Evaluation** (Model evaluation) and **Modell-Konstruktion** (Model construction). A screenshot of the DataEngine software shows a table with columns: Name, Min, Max, Wert, and Ergebnis. The table lists various features and their values. To the right, a 3D surface plot is displayed.

Präsentation CRM 20

09.11.98

© MIT-Management
Intelligence-Technologien GmbH
Aachen, Germany

Vorführung von DataEngine

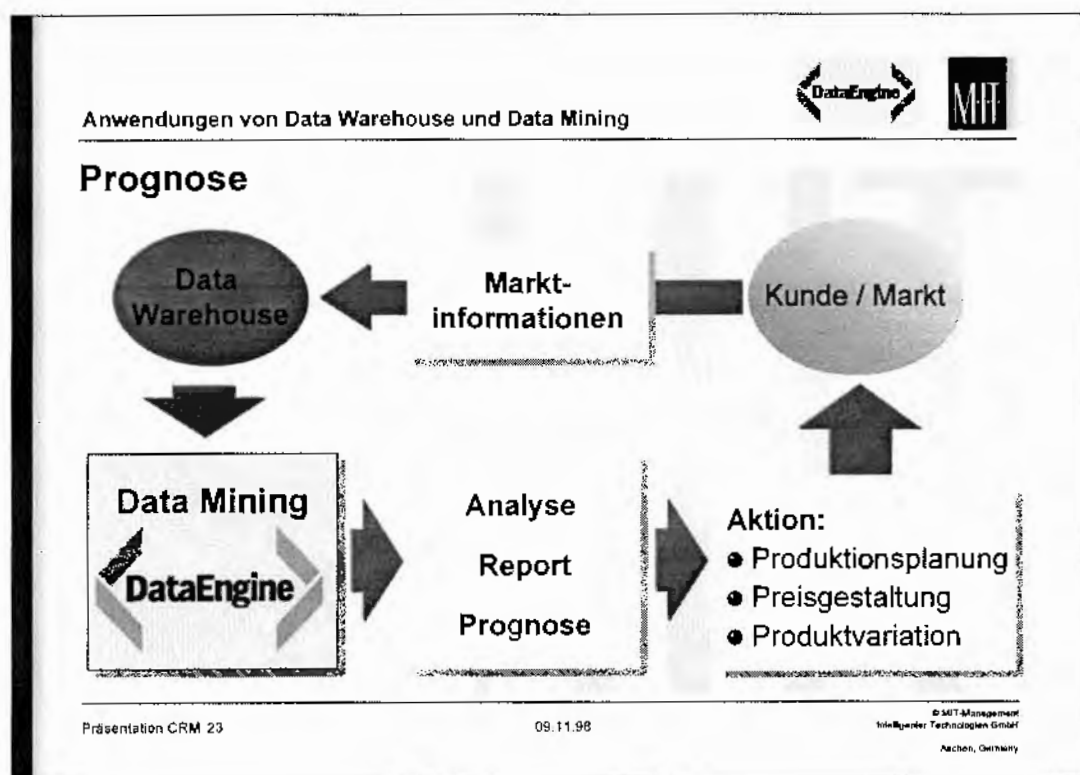
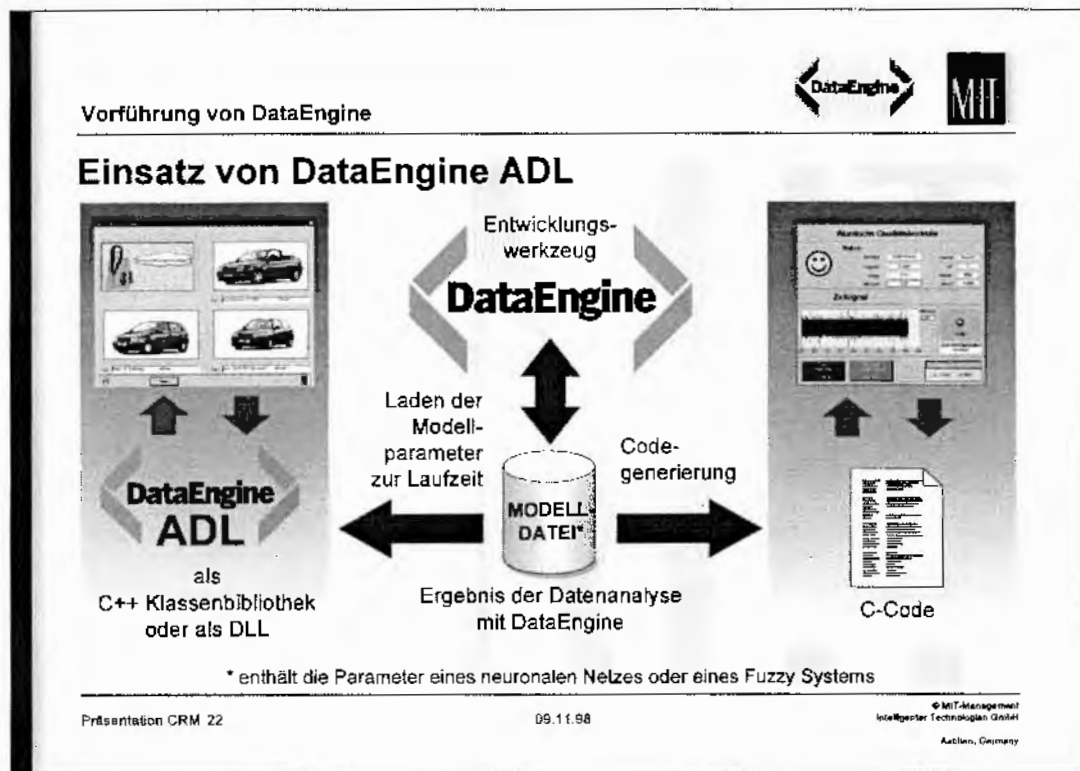
Die DataEngine-Produktfamilie

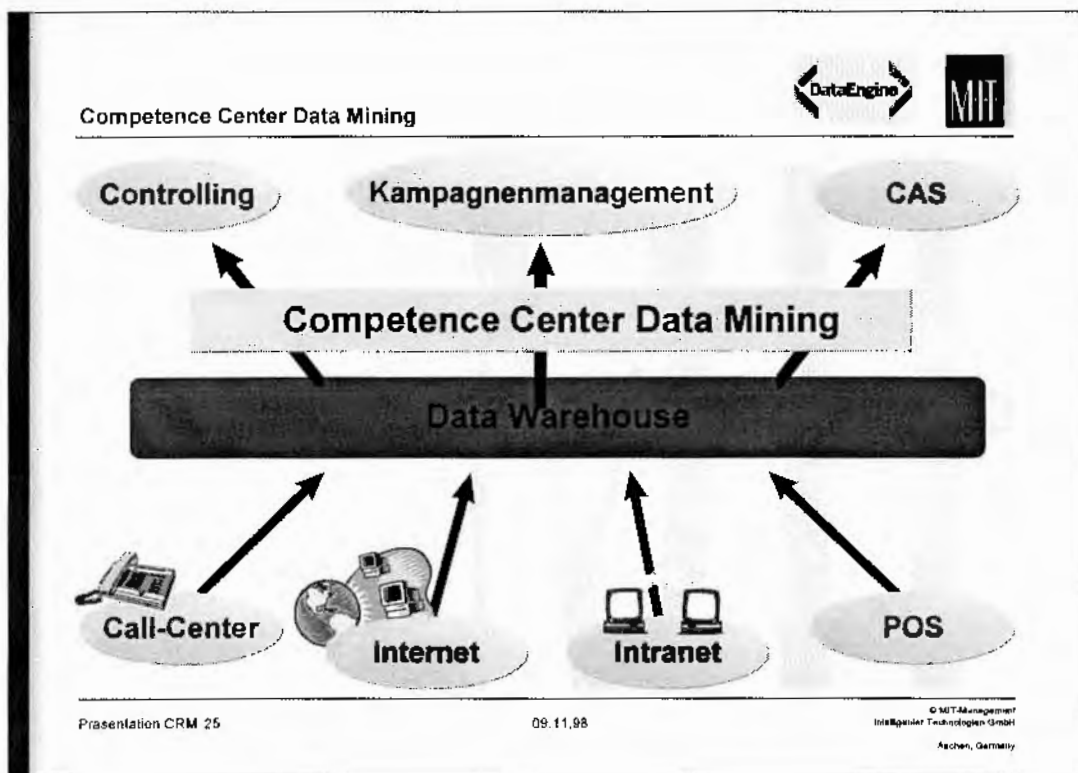
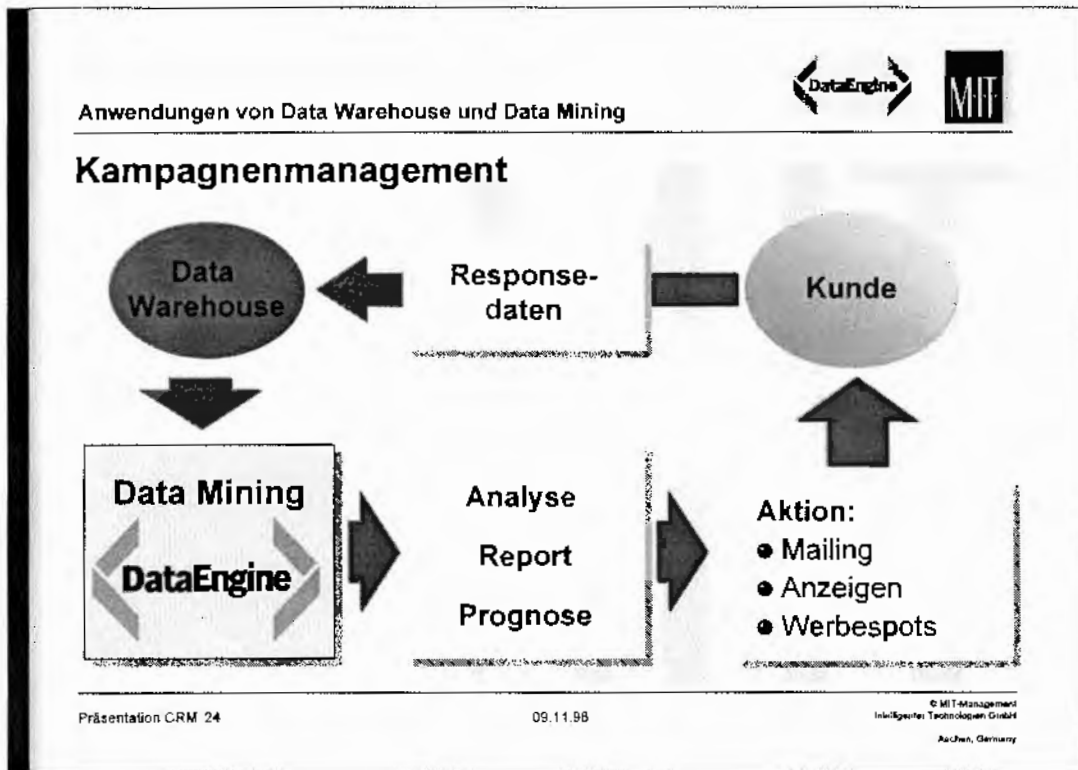
The diagram shows the DataEngine product family. A large arrow labeled **ANALYSE** points to a central circle containing three DataEngine logos: **PlugIn**, **ADL**, and **VI**. A large arrow labeled **APPLIKATION** points from the circle to the right. Below the circle, the text **INTEGRIERTE DATENANALYSE UNTER LABVIEW®** is displayed.

Präsentation CRM 21

09.11.98

© MIT-Management
Intelligence-Technologien GmbH
Aachen, Germany








Competence Center Data Mining

Motivation

	Qualifikation	Werkzeug
Modellierung: "Datenanalyst"	Kenntnis der Daten Kenntnis der Problemstellung Methodenkenntnisse	Datenanalyse / Data Mining Software
Anwendung: "Sachbearbeiter"	Kenntnis der Zielsetzung Wirtschaftlicher Hintergrund	auf Endanwender zugeschnittene Speziallösung

Präsentation CRM 26
 09.11.98

 © MIT-Management
 Intelligenten Technologien GmbH
 Aachen, Germany




Competence Center Data Mining

Positionierung der MIT GmbH

- Methodenkompetenz im Bereich Data Mining
- Professionelle Software-Entwicklung
- Weltweiter Vertrieb der Software-Werkzeuge
- Training zu Data Mining im Data Warehouse Umfeld
- Projektarbeit im Bereich Business Intelligence

Präsentation CRM 27
 09.11.98

 © MIT-Management
 Intelligenten Technologien GmbH
 Aachen, Germany

Data Warehouse Support to Data Mining: a Database Marketing Perspective

Joachim Angstenberger¹, Marco Poloni², Richard Weber¹

1) MIT GmbH

Promenade 9, 52076 Aachen, Germany

Phone: +49 - 24 08 - 9 45 80, FAX: +49 - 24 08 - 9 45 82

2) SEQUENT Computer Systems GmbH

Balanstr. 49, 81541 Munich, Germany

Phone: +49 - 89 - 45 868-314, FAX: +49 - 89 - 45 868-100

ABSTRACT: The paper describes the application of advanced data mining technologies to database marketing. In particular the support that a data warehouse architecture can give to data mining is considered. Many companies have built or are building data warehouses. The availability of large amounts of data increases the request for this data being mined in the expectation to find useful patterns in it. Although a data warehouse is not necessary for data mining, it is easier to mine already organized and checked data. This paper explains how a combination of database and data mining techniques can be used to support the great variety of analysis tasks in database marketing in the financial sector. The tasks of customer profiling and response analysis will be investigated in more detail, showing an application implemented using the DataEngine software package.

1. Introduction

According to [Link, Hildebrand 1994] about 60% of all German insurance companies and banks already had some kind of database marketing system by 1994. The biggest challenge all companies are facing was to get good-quality information about their customers. This is not surprising because the way the data is stored in the transactional systems is unsuited for data analysis for several reasons:

- (1) Transactional databases are built for quick access to small portions of data while running a program in the production environment. They were not designed to access lots of data for analysis under different business aspects.
- (2) The data is spread over various application systems.
- (3) The production systems do not store historical data.
- (4) There are serious data quality problems such as:
 - incorrect data (incorrect attribute values, inconsistent data, integrity violations)
 - missing data
 - data heterogeneity (synonymous names, application specific formats and attribute values)

One reason for building a data warehouse is to overcome exactly this kind of problems. Usually this is accomplished during the first project phases.

Let's look at an example: A bank decided as a first step in customer behavior analysis to cluster their private customer database, see e.g. [Weber 1996]. Among the data elements included in the segmentation task were age, income, total savings, total loan usage and profit contribution. The following three problems arise during the data gathering process:

1. Customer View: The bank's focus is still on the account, not on the individual (customer). The marketing perspective that would pay more attention to the individual has come to the bank after the production systems and procedures were installed. As a result, there is relatively poor quality assurance on the names and addresses of the account holders. A typical bank is doing well if it can find more than 80% of the actual instances where the same individual has multiple accounts.
2. A private customer usually uses several offerings of a bank's portfolio such as current and savings account, mortgage, investment or personal loans, credit card, securities or stocks. Thus several data sources have to be looked up to determine a customer's total savings or loan usage. These amounts have to be assigned to a common name, the same format, the same range of attribute values and the same time. All of these adjustments, but es-

pecially the latter can be difficult because the transactional systems usually do not store historical information and the archived information is unsuitable for extraction.

3. The distinct departments of the bank speak their own language and use different software for data processing. Thus, the same data element can have different names in different applications with various ranges and formats. At this point, caused by the need of a customer view and other kinds of summarized information for data analysis, these types of difficulties show up. Work would be easier by first creating an integrated data base

- reflecting the customer view instead of the account view
- containing only well defined data elements with proven and cleaned data
- containing historical information in the form of (monthly) snapshots of all important basic data elements
- allowing the analysis of data under every interesting business aspect.

By definition, a database with the properties mentioned above is called a **data warehouse**. It is a separate non-transactional database built up from all interesting data sources, updated at pre-defined points of time with clean and adjusted data. The underlying logical data model (**star schema**) supports the analysis of **business facts** like amount, cost or contribution along several **business dimensions** such as customer, time, product and so on. Such a star schema is the basic structure for on-line multidimensional data analysis (OLAP). See e.g. [Schöpke 1997] for more details. Figure 1 shows a simple multidimensional data model for a bank at the account level with some sample attributes.

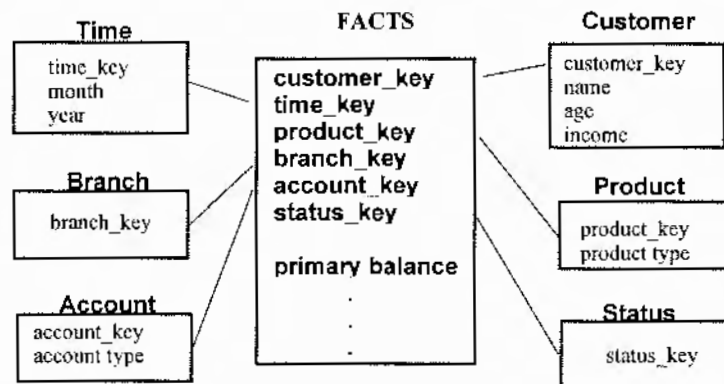


Figure 1: Sample multidimensional data model for a bank

2. Mining Summarized as well as Detailed Data

A data warehouse for database marketing should at least contain all customer related data for existing customers as well as for potential customers. In addition, often several external data sources - like socio-demographical data - are included. Containing all data at the level of highest available granularity the data warehouse becomes the single information source for all customer related departments like operational and strategic marketing, risk management, risk controlling etc. Based on the quality of the warehouse data, improved decision support will be possible.

Often the warehouse dimensions are structured hierarchically, reflecting the hierarchy of business terms or the hierarchical structure of an enterprise. Common examples are "account, account type", "coverage, market segment, line of business" or "month, quarter, year".

Data structured in this way can easily be summarized along one or more of the dimensions, for instance "total loan by customer in December 1996" or "contribution by all customers by product type in 1996".

Beside this, by fixing one or more attribute values in the dimensions every useful business view can be reflected by a star model. Examples are "Comparison of contribution generated by middle-age male persons versus contribution generated by middle-age female persons" or "Contribution in December 1996 versus contribution in December 1995". The difference between both numbers can be calculated and stored in the database as a derived data element. The calculation of derived business facts is the third possibility of the reflection of a business view. Other examples of derived business facts are "customer value" or a customer's "profit contribution". The data belonging to a specific concept or business view can easily be extracted by formulating the appropriate SQL-statement or by using one of the numerous OLAP-frontend-tools.

The warehouse data is seldomly mined directly, but on a subset of data, extracted in a flat structure and transformed for a special analysis (see Figure 2). The multidimensional structure of a data warehouse with all basic information at the most detailed level provides the flexibility to extract the appropriate data for a variety of analyses in a short time frame and in an "ordered" way. This is of great advantage if you follow the primary

underlying hypothesis that on different aggregation levels of data and along different dimensions different patterns can be discovered.

Mining the data only along one of the dimensions or mining only detailed data or only summarized data - all this can just be a rough approximation and may give misleading results. Equipped with a scaleable computer system supporting parallel work and on the basis of a multidimensional warehouse model it is easy to do a lot of analyses in parallel along all dimensions and aggregate levels just obtaining a universal picture of what's hidden in the data.

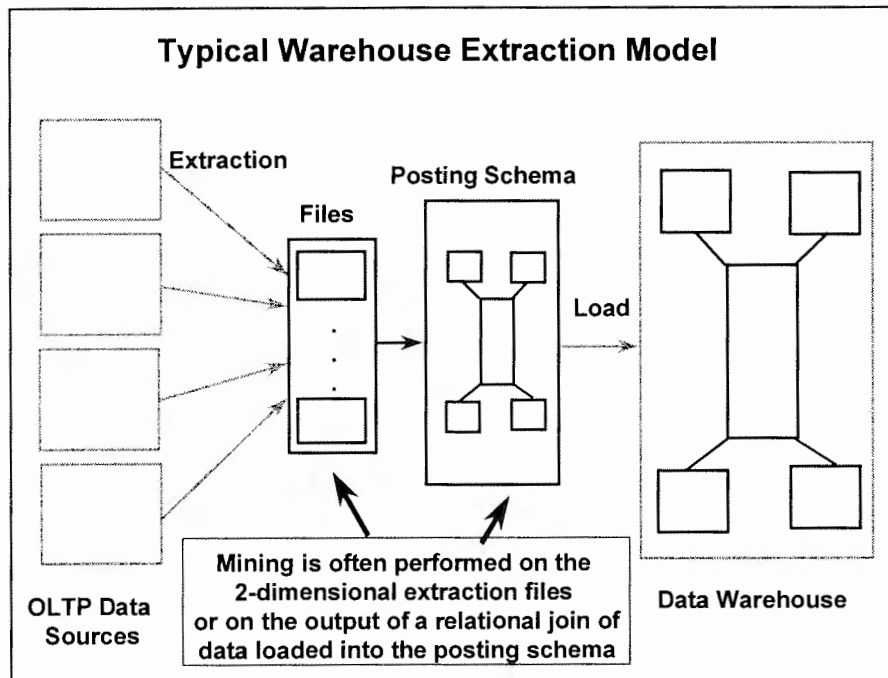


Figure 2: Data Mining Integrated with Extraction and Load (from [Sweiger 1997])

Let's take a look at the sample customer and product dimensions of the above given star schema, the following data being given:

Customer	Product type	Amount
Mueller	saving account	300
Sneeder	saving account	3,500
Sneeder	mortgage loan	-200,000
Rich	saving account	1,000
Rich	bonds	50,000
Rich	funds	80,000
Rich	personal loan	-300,000
Major	saving account	40,000
Major	bonds	30,000

Customer	Age	Income
Mueller	20	650
Sneeder	33	3,400
Rich	42	15,000
Major	70	2,500

From these tables an analyst can generate several aggregates like

Product type	Age	Amount
saving account	<30	300
	30 - 45	4,500
	>65	40,000
bonds	<30	-
	30 - 45	50,000
	>65	30,000
funds	<30	-
	30 - 45	80,000
	>65	-
loans	<30	-
	30 - 45	-500,000
	>65	-

Age	Amount
<30	300
30 - 45	-361,400
>65	70,000

Product type	Amount
saving account	44,800
bonds	80,000
funds	80,000
personal loans	-300,000
mortgage loans	-200,000

The analyst will possibly find some patterns like „middle age people tend to invest their money in funds or bonds“ or „middle age people invest a lot of money, but also have high liabilities to the bank“.

This is basically a manual query driven search. Although patterns **can** be found this way - it is rather by accident. There are so many patterns in large databases that the user can rarely think of the right question to ask to find the real assets.

As well known, the "real" data mining and querying of a relational database differ mainly in that the first one is a discovery driven, to some degree automated process of finding new patterns in the data while the latter tries to find the confirmation of an hypothesis stated before the search process.

With respect to our example such a hypothesis could be: „The overall amount invested by a customer in some kind of product is determined by his life-cycle phase“.

3. Combination of OLAP-type Questions and „Real“ Data Mining

Proceeding with the example, the bank did not want to restrict themselves to confirmatory data mining. They wanted to know more about their customers and understand their behavior.

A first step in this process was to find important customer groups by customer segmentation. Because it is not useful to include all primary data elements in the analysis - the result would be too noisy- the analysts had to make a choice. The specialists proposed to choose „Total savings“ and „Total profit contribution“ as important business facts. Both aren't primary data elements. But they could easily be derived on the basis of the data warehouse. For this purpose it was also necessary to include cost information in the warehouse which is not defined at the account level, but could be included in an aggregate fact table at customer level.

Customer	Total Profit Contribution (derived)	Total Savings (derived)
Mueller	-7	650
Sneider	197	-196.000
Rich	433	-155.000
Major	60	70.000

Beside these, further data elements like age, income and some socio-demographical data elements were included. Such data elements can either be found by the assumption of experienced analysts or by starting an influence analysis using a decision tree or regression analysis which automatically detect important attributes with respect to customer profit contribution.

Two things can be pointed out here: First, in order to automatically detect important attributes an objective function must be given. In this case: profit contribution. Thus, for an analysis of customer retention probably other attributes will be included than for an analysis of customer contribution.

And second, ideas for derived data elements which are either calculated or a combination of attribute values (for example age and sex alone might be of no importance, but young male can be an important combination) arise from people rather than from machines.

Thus, mining the database is not a fully automated process, but a combination of mind work and machine power. This underlines the iterative nature of the knowledge discovery process. While interpreting the analysis results, usually ideas for the inclusion of further data elements in the analysis or ideas for the investigation of another aspect of a problem arise. The multi-dimensionality of the warehouse data provides the flexibility needed for quick generation of the desired data sets.

Back to the example: In this case, using a fuzzy clustering algorithm seven customer segments (classes) were identified. The fuzzy c-means algorithm was applied for segmentation. For further details on fuzzy clustering see e.g. [Bezdek 1981], [Zimmermann 1996]. For each class a customer profile was derived which was used to determine each customer's class, see e.g. [Weber 1996].

At this point a marketing campaign addressing one of the identified customer classes could be started. In most cases further investigations at the chosen segment will help to identify the most valuable customers. This would make the marketing campaign more profitable.

On the basis of the warehouse data the customer value for every customer can be calculated and either stored in the warehouse or be combined in further analyses. The marketing campaign can then be directed only to those people for whom a high response probability and a high customer value are calculated. The corresponding customer keys can be used to query the warehouse for the customer addresses and start the marketing campaign.

As already mentioned the marketing warehouse besides atomic data at the account level will also contain consolidated data at customer level. The customer level is also the right level for storing information about all marketing campaigns a customer is involved in and his response to these campaigns. In the fact table of this campaign star schema information about cost and contribution of the campaign can be stored as shown in figure 3. This information can be used to further improve the efficiency of marketing campaigns and to improve the customer-bank relationship.

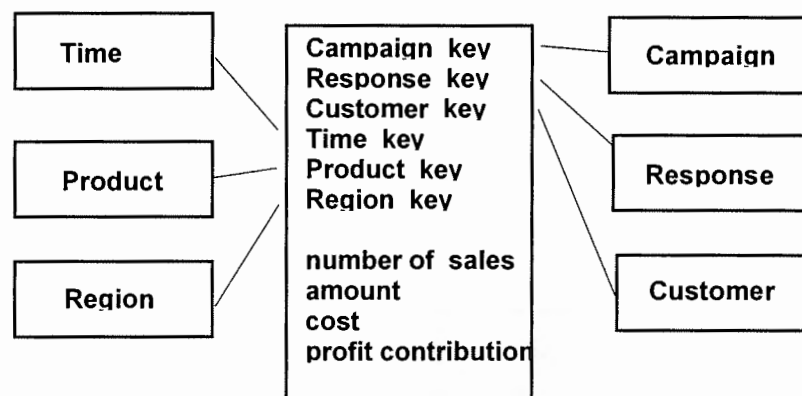


Figure 3: Star schema for controlling of marketing campaigns

The application described above has been performed using DataEngine, the software tool for intelligent data analysis [MIT 1997a]. This software runs on a PC under Windows 95 / NT. An application program for use in the marketing department has been implemented by applying DataEngine ADL - Application Development Library [MIT 1997b]. This C++ class library is available for different hardware platforms including Sequent's NUMA Q 2000 platform.

4. An application for customer profiling and response analysis

A well-organized customer database contains information about the customer's personal and demographic background as well as the specifications of the products the customer already uses. The objective of database marketing is the exploitation of this inhomogeneous information in order to select as exactly as possible those persons from a database who show the greatest potential to actually buy an advertised product.

The application presented here was undertaken for a German bank, its products being for example checking accounts, credit cards and investment schemes. Depending on the type of product the target group may differ considerably. Offering useful and affordable products to its customers is not only a question of the bank's marketing expenses but also of its credibility.

ANALYSIS PROCEDURE

State of the art

So far, the selection of persons to include in a mailing has been done by manually configuring a prototypical customer profile and comparing each database record to this prototype during a database query persons matching the pre-specified requirements being included in the mailing. The manual composition of the profile to match introduces an arbitrariness into this process. Considering the amount of information available for each customer, it is clearly non-trivial to set up a query like "if age < 25 and job = worker and credit amount < 5.000,- then include customer in mailing". It is practically impossible to manually specify the non-linear interaction of the individual data fields describing a typical person that is probably going to buy the advertised product.

It would be extremely useful if the prototypical customer profile could be established automatically. In the below we show that a neural network can be applied to accomplish this task. The development of a neural classifier used to extract those people from a database who are likely to buy a credit card in response to a mailing is described.

Target Group Selection by Response Analysis Using Neural Networks

The analysis was based on approximately 9000 records of customers from a special sub-segment called "young adults". The data records included personal data, such as age, gender and job as well as detailed information about the customers' product utilization. 18 different products were considered, ranging from checking and savings accounts to investment plans and securities. Additionally, a classification of their place of residence was available for each person in the database. Altogether about 180 data fields were available for each customer in the database.

From a previous mailing the response (in terms of buying the product or not) of about 2500 people was known. The response rate of this previous mailing, calculated as a ratio between the number of actual buyers responding to the mailing and the mailing volume, was 8.2%.

First statistical examinations of the data showed that the distributions of the individual data fields of buyers were very similar to those of non-buyers. Thus, from the statistical distributions the profile of a typical buyer could not be established. This means that it was not enough to look at isolated customer features. The customer profile obviously had to consider the combination of different pieces of information about the customer. The nature of this combination of information was unknown, as well as what information actually had to be included.

For this reason it was decided to establish the relationship between the customer profile and his tendency to buy a product by using a neural network. Neural networks are able to represent the kind of non-linear and inhomogeneous relationships required here. However, in order to use only the most important features the next step of the analysis procedure was the reduction of the considered feature space. This should also give an indication of the features which are of actual relevance.

Fig. 4 shows an outline of the overall analysis procedure described above. The most relevant features are selected using a modification of the method described in [Behr, Strackeljan 1995], [Behr, Kocher, Strackeljan 1997] and [MIT 1997a]. This procedure is capable of considering an initially very high-dimensional feature space and of selecting those combinations of features resulting in the highest possible reclassification rates. It returns a list of feature combinations together with an estimation of the reclassification rate for each combination.

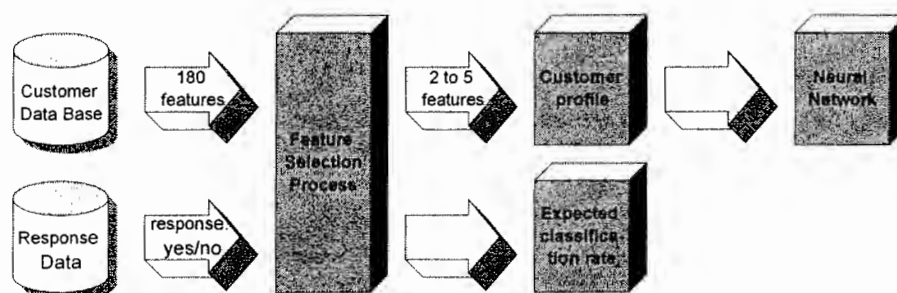


Figure 4: Outline of the global analysis procedure

The design and development of a classification system is frequently achieved by separating the feature selection from the classification step. The feature selection concept in [Behr, Strackeljan 1995] combines these two steps. This method has the advantage that the selected features are well suited for the given classifier, because the applied optimizing criteria are directly related to the classifier. Here the concept is modified to work with

different algorithms in the selection and the classification step, because the training time, especially for neural networks, is not acceptable for a selection process where frequently more than a thousand different combinations of features have to be considered. The results of our investigation show that the best feature combination determined by the fast algorithm in the selection step will also produce a high reclassification rate when using a neural network. This transfer of results keeps computing times within acceptable limits.

A second problem is the very different sample sizes for the classes 'buyer' and 'non-buyer' in the available learning and validation data sets. The scoring of the classifier capacity via the reclassification rate over all classes has to be replaced by a more complex cost function for each feature combination, because the costs are linked to a classification decision. As a technical example one can mention quality testing for the mass production of car tire valves which are without a doubt a safety-relevant component. Considering the low production costs of a valve, a misclassification of a good valve into the reject class is tolerable, whereas a defective valve must not pass the quality check under any circumstances. However, if the costs of misclassification of a defective valve are very high, then a classifier with the objective of cost minimization will find almost all valves defective. The integration of the costs into a classifier therefore requires a problem-specific analysis. For the mailing problem the feature selection process can be optimized by an analysis of the profitability. As an example we assume that the costs for the mailing are DM 2,- per person and the cover of contribution for the product is DM 20,- per year. The results in table 1 show that profit could be a criterion for the feature selection process.

Table 1: Example of a cost function for the feature selection process

Number of mails	Cost of mailing	Number 'Buyers'	Sales	Profit (Cover of Contribution)
100000	200.000,-	10000	200.000,-	0
80000	160.000,-	9000	180.000,-	+20.000,-
60000	120.000,-	8000	160.000,-	+40.000,-
30000	60.000,-	6500	130.000,-	+70.000,-
10000	20.000,-	4000	80.000,-	+60.000,-

The feature selection step resulted in combinations of two to five features that were further examined. Neural networks of the Kohonen type (self-organizing feature map [Kohonen 1989]) were trained on a subset of the response data and evaluated using a different subset for each of the selected feature combinations.

Results

The results were compared with the actual results of the mailing and the corresponding response rates were calculated. These are shown in table 2.

Table 2: Response rates of neural classifiers

Number of features	Response rate	Size of target group selected
2	20.0 %	350
3	18.3 %	460
4	17.4 %	530
5	17.6 %	500

It may seem a bit surprising that the highest response rate can be achieved with a classifier that uses only two customer features to decide whether a person should receive the mailing or not. However, this is related to the fact that this classifier selects a relatively small number of persons ("size of target group selected" in table 2). The nature of the response rate being a ratio means it is optimal if all the actual buyers (people buying the product in response to the mailing) are selected for the mailing in spite of a small mailing volume.

Unfortunately the real world behaves differently. If all subsequent buyers are to be selected from the database, the total mailing volume has to be increased more than proportionally. This leads directly to smaller response rates. Those classifiers listed in table 1 using more than two features achieve lower response rates, but are more capable of selecting the actual buyers¹. A trade-off has to be decided depending on financial and marketing considerations.

If the analysis results shown in table 2 are compared to the previously obtained response rate of 8.2 %, it is obvious that the rate can easily be doubled if the selection process is carried out in the way described. This leads

¹ This statement is clearly only possible due to the analysis of the existing response data.

to a dramatic reduction of mailing costs and at the same time the mailing volume chosen by the neural network is considerably smaller than the volume selected manually. Thus, an increase of the overall efficiency of the mailings is achieved.

At present, the knowledge acquired during this analysis is transferred to match the requirements of products other than the credit card considered so far. This concerns mainly the feature selection process that obviously has to be done separately for each product.

A SOFTWARE SYSTEM FOR TARGET GROUP SELECTION

Pre-processing and neural network design was carried out exclusively using the data analysis software package DataEngine® [Angstenberger 1996], [MIT 1997b]. The feature selection functionality is integrated in DataEngine® as described in [MIT 1997a]. Neural network development is completely separate from the user module that was integrated into the company database as an additional module. The system architecture is shown in figure 5. All the information needed is extracted directly from the customer database. As a result a flag is set at a specific database field. This flag is then used to collect all the customer data records for the mailing.

For the marketing employees, using the system does not involve any direct contact with neural networks. The only thing they have to specify via the user interface is the target group (selected only by age) they want to address the mailing to and the product that is to be advertised.

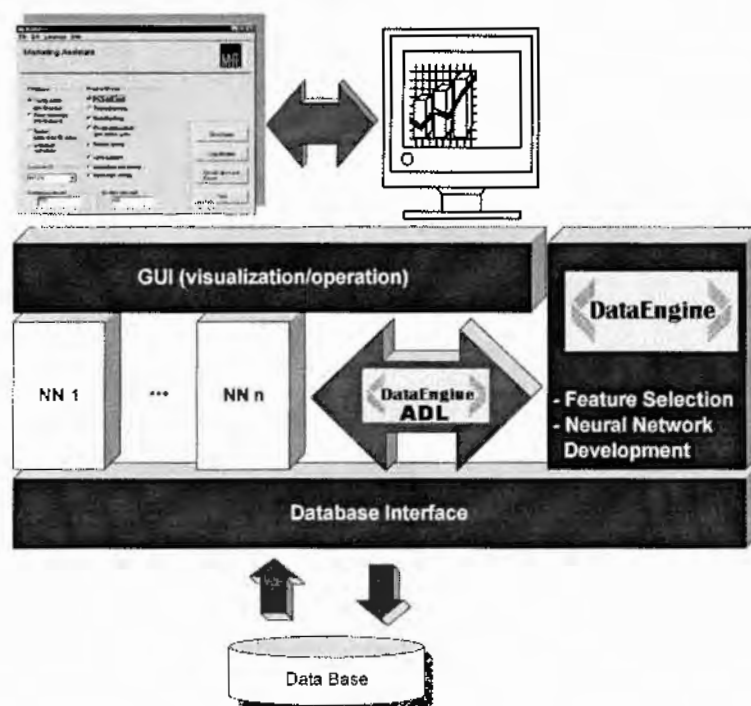


Figure 5: System architecture of neural network based customer classification module

Above the design of a neural network classifier for a single target group/product combination was shown. However, as can be seen from figure 5, the system contains multiple neural networks that have been designed to cover all the combinations of products and target groups. The appropriate neural network is selected automatically. All the people belonging to the target group are then classified. If the user wants a specific number of people to be included in the mailing, this number can be pre-set.

In the course of time, new response data is made available. This additional information can be easily integrated into the system. Whenever a system update is to be performed, the complete response data is read in and passed on to DataEngine® as an integral part of the system. The feature selection and neural network development steps are then performed and the system is updated by simply exchanging the underlying neural networks.

The system accesses the neural networks via parameter files that are interpreted by DataEngine® ADL, the runtime library for the DataEngine® modules. This also means that there is no need for a system shutdown during the updating procedure.

Conclusions and Future Perspectives

The paper has presented in its first part an overview of how data warehousing can support data mining and a general example has been given using a database marketing application. In the second part a detailed application

description has been proposed, where the design of a classification system for the selection of potential customers from a database has been shown and the structure of the corresponding software system presented. This system is currently employed at a German bank producing good results.

The structure of the presented solution can easily be transferred to similar problems at mail-order or insurance companies. Moreover, the basic system design has also been applied in [Angstenberger, Heinrich, Kleiner 1996] for the design of a sales forecasting system at Bayer AG.

5. References

- Angstenberger, J. (1996): DataEngine: A software framework for intelligent data analysis. Proc. EUFIT 96 - Fourth European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, pp. 2255-2258.
- Angstenberger, J., Heinrich, J., Kleiner, F. G. (1996): Sales forecasting in industry using neural networks. Proceedings EUFIT '96, Aachen, pp. 2140-2144.
- Behr, D., Kocher, T., Strackeljan, J. (1997): Fuzzy pattern recognition for automatic detection of different teeth substances. Fuzzy Sets and Systems 85, pp. 275-286.
- Behr, D., Strackeljan, J. (1995): A Feature selection method for acoustic analysis problems. Proc. EUFIT 95 - Third European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, pp. 1175-1180.
- Bezdek, J. C. (1981): Pattern Recognition with Fuzzy Objective Function Algorithms. New York London
- Kohonen, T. (1989): Self-organization and associative memory, 3rd ed., Springer-Verlag.
- Link, J., Hildebrand, V. (1994): Verbreitung und Einsatz des Database Marketing und CAS, Frankfurt 1994 (in German)
- MIT (1997a): DataEngine 2.1 - Manual. Aachen
- MIT (1997b): DataEngine ADL - Application Development Library 2.1 - Manual. Aachen
- Schöpke, G. (1997): Data Warehouse als Grundlage für effizientes Database Marketing. In: Proceedings MIT-Anwendersymposium für Finanzdienstleister, Frankfurt 1997
- Sweiger, M. (1997): Data Mining Overview, Presentation at the DWI Conference, Boston, Massachusetts
- Weber, R. (1996): Customer Segmentation for Banks and Insurance Groups with Fuzzy Clustering Techniques. In: J. F. Baldwin (ed.): Fuzzy Logic, John Wiley and Sons, Chichester, 187 - 196
- Zimmermann, H.-J. (1996): Fuzzy Set Theory - and Its Applications. Third Edition. Kluwer Academic Publishers, Boston Dordrecht London

OPTIMIERUNG EINES ZEMENTMAHLPROZESSES MIT NEURONALEN NETZEN UND GENETISCHEN ALGORITHMEN

JOACHIM ANGSTENBERGER, MARTIN NELKE, TORSTEN SCHRÖTTER

MIT GMBH, AACHEN

TELEFON: (02408) 9458-0

FAX: (02408) 9458-2

EMAIL: JA, MN, TOS@MITGMBH.DE

URL: [HTTP://WWW.MITGMBH.DE](http://www.mitgmbh.de)

Zusammenfassung: Fuzzy-Technologien und neuronale Netze werden in zunehmendem Maße in der Grundstoffindustrie eingesetzt. Viele Prozesse können wegen ihrer Komplexität und Nichtlinearität nicht oder nur mit hohem Aufwand exakt mathematisch modelliert werden. Zu dieser Klasse von Prozessen zählt auch das Zementmahlen. Hier eignen sich neuronale Netze, um Zusammenhänge in Prozeßdaten zu erlernen. Aufbauend darauf kann mit genetischen Algorithmen die Prozeßführung optimiert werden. Der Beitrag beschreibt ein Projekt, in dem diese Vorgehensweise für den Zementmahlprozeß einer Kugelmühle umgesetzt wurde.

Stichwörter: Neuronale Netze, Genetische Algorithmen, Prozeßoptimierung, Zementmühle

1 EINLEITUNG

Bei der Zementherstellung werden Mühlen zur Zerkleinerung des Rohmaterials und zum Grob- und Feinmahlen eingesetzt. Mahlen ist einer der energieaufwendigsten Verfahrensschritte in der Grundstoffindustrie. Um Kosten hierbei senken zu können, ist es unablässig, alle Möglichkeiten zur Verringerung des Energieverbrauchs zu beschreiten.

Durch die Optimierung der Prozeßführung des Zementmahlkreislaufes mit Hilfe intelligenter Technologien kann die Anlageneffizienz noch gesteigert und ein größerer Gewinn realisiert werden. Denn konventionelle, meist lineare Techniken versagen, wenn

- viele Prozeßvariablen Einfluß auf die Arbeitsweise eines Prozesses haben,
- die Prozeßdynamik stark verkoppelt und nichtlinear ist
- oder der Prozeß nach (teilweise) unbekannten Gesetzmäßigkeiten abläuft.

Zur Auswahl der dominanten Einflußgrößen eines Prozesses oder zur Beschreibung des Wirkungszusammenhangs zwischen den Prozeßgrößen können diese Verfahren also nicht herangezogen werden. In diesen Fällen ist der Einsatz moderner Methoden wie neuronaler Netze oder Fuzzy Systeme zur Prozeßmodellierung und -optimierung zu empfehlen.

Einen Überblick hinsichtlich der Eignung verschiedener moderner Methoden zur Zustandserkennung und Zustandsbewertung eines Zementmahlkreislaufes gibt [Drews, Schmidt 1996]. Automatische Methoden können für den Wissenserwerb in der Prozeßoptimierung verwendet werden, um relevante Zusammenhänge zwischen Daten in Bezug auf eine konkrete Fragestellung aufzudecken. In [Locher 1996] ist dies am Beispiel einer Rollenmühle durchgeführt worden. Fuzzy-Methoden, neuronale Netze und Expertensysteme werden auch in anderen Verfahrensschritten der Zementherstellung eingesetzt, wie beispielsweise bei der Klinkerproduktion zur übergeordneten Prozeßführung des Drehrohrofens mit angekoppelten Subsystemen [Erens, Saint-Paul 1995] [Østergaard 1996].

2 ANWENDUNG NEURONALER NETZE UND GENETISCHER ALGORITHMEN ZUR PROZEOPTIMIERUNG

In diesem Beitrag werden neuronale Netze in Verbindung mit genetischen Algorithmen angewendet, um kostenoptimale Parametereinstellungen für den Zementmahlprozeß zu ermitteln, so daß sich vorgegebene Zementqualitäten bei höchster Wirtschaftlichkeit herstellen lassen.

Der Abschnitt beschreibt zunächst den untersuchten Prozeß. Anschließend wird die Vorgehensweise bei der Modellierung und Optimierung der Zementmahanlage vorgestellt.

2.1 HERSTELLEN VON ZEMENTPULVER

Kugelmühlen werden in einem Zementwerk zum Feinmahlen von Klinker, Gips und Kalkstein zu Zementpulver eingesetzt. Die Mühlenbauform besteht aus einem horizontal gelagerten, drehbaren Zylinder und setzt als Mahlkörper Stahlkugeln verschiedener Größe zum Zerkleinern des eingebrachten Materials ein. Die Mühle wird neben den Grundstoffen mit weiteren Einsatzstoffen in verschiedener Zusammensetzung befüllt.

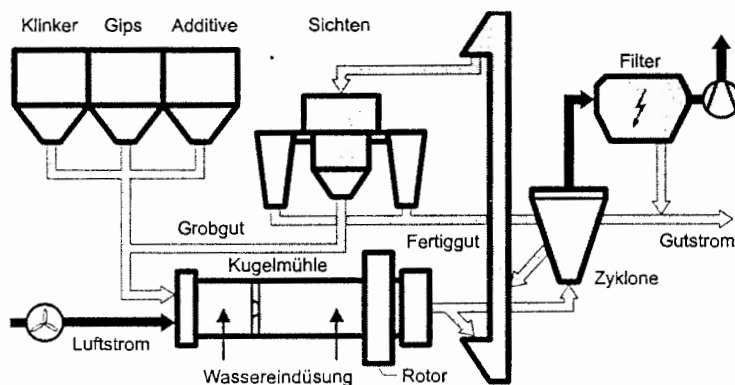


Abbildung 1: Prinzipielles Fließbild einer Zementmahanlage

Das Mahlen des Zements ist ein kontinuierlich ablaufender Prozeß. Die Mühle ist mit Sichern so angeordnet, daß der anfallende Gieß zum Teil wieder in die Mühle zurückgeführt wird (Abbildung 1). Der Gutstrom fließt dem Speichersilo zu. Zum Abtransport der mit dem Mahlgut und durch die Mahlung zugeführten Wärme wird die Mahlkammer von einem Luftstrom durchflossen. Zusätzlich kann durch Wassereindüsung in die Mahlkammer die Wärmeabfuhr verbessert werden.

Der Mahlprozeß läßt sich über die Separatordrehzahl, die Ventilator-drehzahl und die Wassereindüsung steuern. Während des Mahlens können die Quantitäten und die Zusammensetzung des Materialflusses gesteuert bzw. geregelt werden.

Das Mahlen als letzter Verarbeitungsschritt bei der Zementherstellung dient der Erzeugung größerer spezifischer Oberflächen und bestimmt somit die Feinheit des Zementpulvers. Um die spezifische Oberfläche des Zements zu ermitteln, wird in einem physikalischen Testverfahren die Blaine-Feinheit bestimmt. Sie gibt für körnige Materialien die gesamte Oberfläche der Körner pro Masseneinheit an Material an.

Mit der Blaine-Methode wird die Luftdurchlässigkeit einer standardisierten Probe von verdichtetem Zementpulver gemessen. Diese Durchlässigkeit ist ein Maß für die Feinheit des Zements. Allerdings kann mit der Blaine-Methode nur ein Mittelwert für die spezifische Oberfläche des Zementmaterials gemessen werden, sie gibt keinen Aufschluß über die Verteilung der Partikelgröße.

Die Pulverfeinheit ist ein wichtiges Qualitätsmerkmal. Sie beeinflusst die Leistungsfähigkeit des Zements. Je feiner das Zementpulver ist, desto schneller läuft die Reaktion mit Wasser ab. Daraus resultiert dann schnelleres Abbindeverhalten und höhere Festigkeit.

2.2 PROZEßMODELLIERUNG MIT NEURONALEN NETZEN

Ziel der Prozeßmodellierung ist es, die Transparenz des Prozeßzustands zu verbessern, indem die Zementfeinheit (Blaine-Wert) als qualitätsrelevante Größe über ein neuronales Netz aus Prozeß- und Steuergrößen abgebildet wird.

Zunächst trainiert man das neuronale Netz auf den Zusammenhang zwischen dem Momentanzustand der Kugelmühle und der zukünftigen Zementfeinheit. Eingesetzt wurde dazu ein Multi-Layer Perceptron (MLP) mit Backpropagation Lernregel und einer verdeckten Schicht. Als Eingangsgrößen wurden unter anderem die Separatordrehzahl und Separatorventilation verwendet. Abbildung 3 (links) zeigt die prinzipielle Struktur des MLP-Netzes. Das Netz verhält sich nach erfolgreichem Training wie ein Modell des Mahlprozesses (Abbildung 2) Mit diesem Prozeßmodell kann man vorhersagen, zu welcher Produktqualität eine bestimmte Einstellung der Kugelmühle führt.

Umgekehrt läßt sich in einem zweiten Schritt durch Invertierung des Modells die wichtige Information gewinnen, wie die Stellgrößen einzurichten sind, damit eine vorgegebene Zementfeinheit hergestellt werden kann (Abbildung 3).

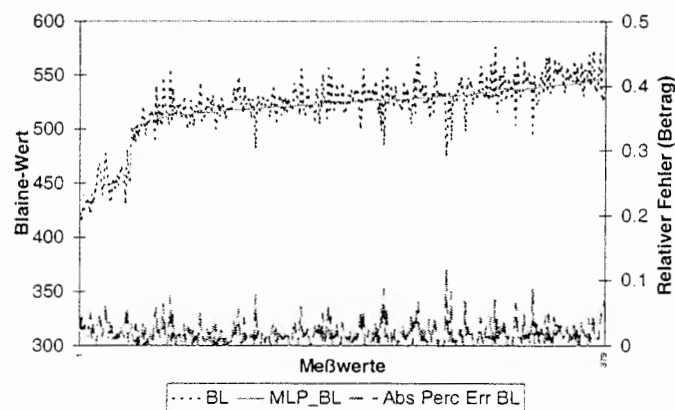


Abbildung 2: Approximation des Blaine-Wertes mit Hilfe eines neuronalen Netzes.

Abschließend ist in einem dritten Schritt derjenige Betriebspunkt zu suchen, für den die gewünschte Zementfeinheit zu minimalen Kosten erzielt werden kann. Diese beiden Schritte wurden mit Hilfe eines genetischen Algorithmus ausgeführt.

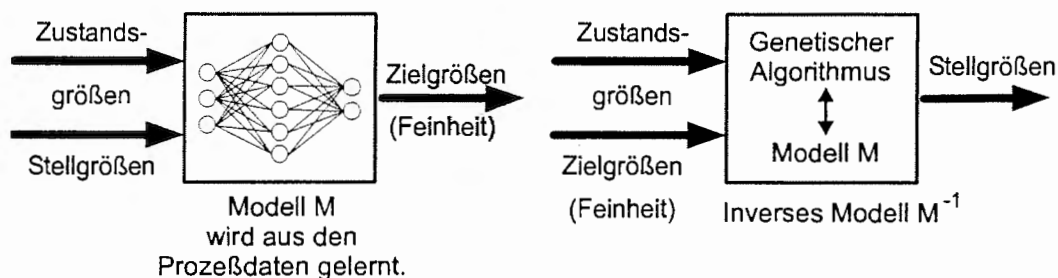


Abbildung 3: Prinzipielle Vorgehensweise zur Invertierung des Prozeßmodells.

Ein genetischer Algorithmus benutzt das erstellte Prozeßmodell zur Bestimmung der Stellgrößen, die zu einer gewünschten Zementfeinheit führen [Sprave 1995]. Dabei werden im Laufe der Berechnung die Stellgrößen so variiert, daß der euklidische Abstand der aktuellen Zielgrößenausprägung von einem vorgegebenen Blaine-Zielwert minimiert wird. Das Verfahren schließt ab, wenn

eine Anlageneinstellung gefunden ist, die zu dem betrachteten Vorgabewert führt oder diesem nach einer begrenzten Suchzeit am nächsten kommt.

2.3 KOSTENOPTIMALE ANLAGENEINSTELLUNG

Ein Vorteil des Einsatzes genetischer Algorithmen besteht darin, daß durch Definition von Kostenfunktionen für die Einsatzmengen der Stellgrößen die Betriebspunktermittlung ohne weiteres zu einer Vorgehensweise für die Ermittlung der kostenoptimalen Anlageneinstellung erweitert werden kann. Wegen ihrer Robustheit gegenüber lokalen Optima der Zielfunktion sind genetische Algorithmen in Verbindung mit einem Prozeßmodell universell zur Optimierung einsetzbar.

Dazu werden die beeinflussbaren Prozeßvariablen mit einer Kostenfunktion bewertet und die übrigen Größen und Nebenbedingungen entsprechend dem untersuchten Prozeßzustand vorgegeben:

- Struktur der Kostenfunktion, z.B. $K_x = \sum_{i=1}^n k_i x_i$, mit
 x_i Prozeßvariable
 k_i spezifische Kosten
- Restriktionen für Stellgrößen x , z.B. $b_{\min,i} \leq x_i \leq b_{\max,i}$, mit
 $b_{\max,i}$ maximal zulässige Einsatzmenge der Prozeßvariablen x_i
 $b_{\min,i}$ minimal zulässige Einsatzmenge der Prozeßvariablen x_i

Auf der Outputseite legt man den gewünschten Wert der Zielgröße fest. Der genetische Algorithmus sucht dann den kostengünstigsten Betriebspunkt für den vorgegebenen Blaine-Wert. Kostenfunktionen und Stellgrößenbeschränkungen können auch nichtlinear strukturiert sein.

Die folgende Abbildung zeigt exemplarisch einen Verlauf der minimalen Kosten (normiert) über der Zielgröße Zementfeinheit bei gegebener Kostenstruktur.

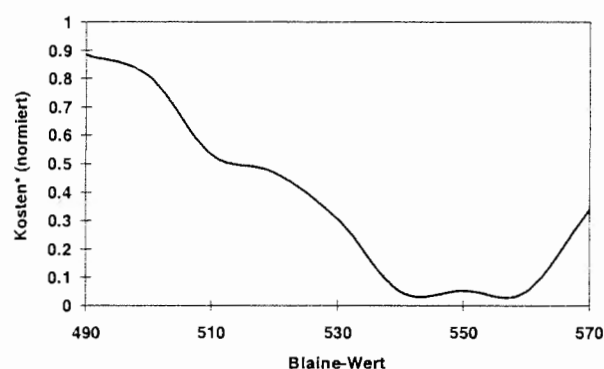


Abbildung 4: Verlauf der minimalen Kosten über der Zielgröße Zementfeinheit

3 VERWENDETE SOFTWAREWERKZEUGE ZUR MODELLIERUNG UND -OPTIMIERUNG DES MAHLPROZESSES

Die vorstehend dargestellte Modellierung und Optimierung wurden mit dem Software Werkzeug DataEngine als Kernbestandteil realisiert. Diese Software zur intelligenten Datenanalyse kombiniert konventionelle statistische Verfahren und Instrumente der digitalen Signalverarbeitung mit intelli-

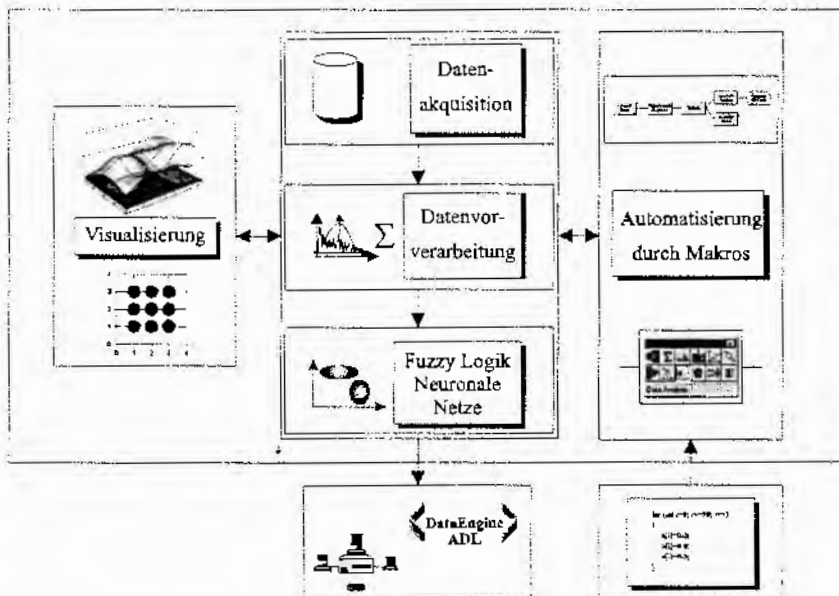


Abbildung 5: DataEngine

genten Technologien, wie Fuzzy Logik und neuronalen Netzen [MIT GmbH 1997a]. Die Bibliothek DataEngine ADL integriert das erstellte Modell zur Laufzeit in eine Systemumgebung mit Prozeßankopplung. Diese Kombination führt zu einer leistungsfähigen Softwarearchitektur, die in einem sehr weiten Anwendungsbereich zur Optimierung verfahrenstechnischer Prozesse eingesetzt werden kann (Abbildung 5).

Neben anderen Methoden beinhaltet DataEngine den Backpropagation-Algorithmus, der auch für die Optimierung des Zementmahlprozesses verwendet wurde. DataEngine ist in einem objektorientierten Konzept in C++ implementiert und steht für die Betriebssysteme Windows NT/95 zur Verfügung. Interaktive und automatische Bedienung ergänzt durch eine effiziente und komfortable graphische Benutzerschnittstelle erleichtern die Nutzung dieses Werkzeuges (Abbildung 6).

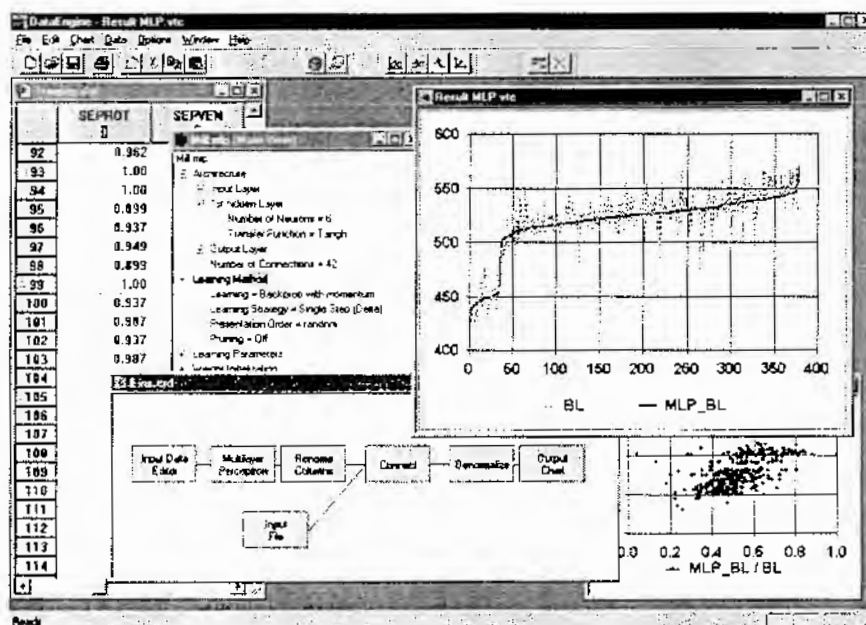


Abbildung 6: Prozeßmodellierung mit DataEngine

Die DataEngine ADL Funktionsbibliothek ist eine Softwarekomponente, die als C++ Klassenbibliothek für eine Reihe von Compilern und Plattformen verfügbar ist [MIT GmbH 1997b]. Mit DataEngine ADL wurden in dem oben beschriebenen Projekt die zuvor über DataEngine modellierten neuronalen Netze in ein Anwendungsprogramm integriert und mit der darauf aufsetzenden Optimierungsfunktion verbunden (Abbildung 7).

Das Anwendungsprogramm besteht aus den Modulen Optimierung, What-If Analyse und Sensitivitätsanalyse. Die Funktion Optimierung unter Nebenbedingungen mit parametrierbarer Kostenfunktion bietet Zugriff auf den genetischen Algorithmus zur Umsetzung einer zu produzierenden Zementqualität in eine Anlageneinstellung. Die What-If Analyse informiert den Anwender für vorgegebene Stellgrößen und Prozeßzustände über die daraus resultierende Produktqualität. Für ergänzende Untersuchungen ist eine Sensitivitätsanalyse integriert, die zu einem gewählten Arbeitspunkt den Einfluß von Variationen einzelner Prozeßgrößen um diesen Arbeitspunkt auf die Produktqualität visualisiert.

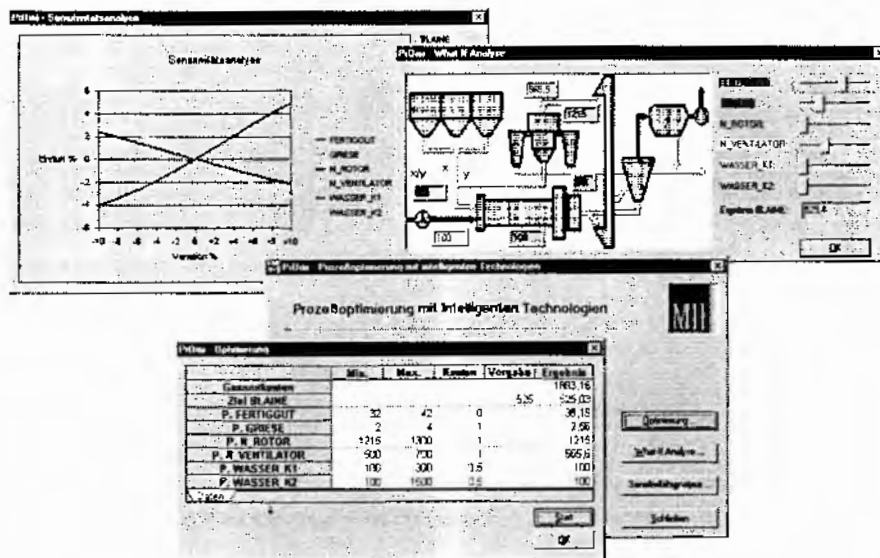


Abbildung 7: Anwendungsprogramm zur Optimierung des Verfahrensschritts Zementmahlen

Diese Software kann zur Entscheidungsunterstützung bei der Wahl der Einstellungen der Zementmühle eingesetzt werden.

4 NUTZEN DER PROZESSOPTIMIERUNG

Neuronale Netze stellen ein zusätzliches Hilfsmittel dar, um intelligente Automatisierungssysteme höherer Leistungsfähigkeit zu realisieren. Dabei sind im wesentlichen zwei Ansatzpunkte von Interesse:

- Erstellen eines Modells des zu automatisierenden Prozesses („What-If-Analyse“)
- Optimieren der Anlageneinstellung je nach gewünschtem Zielgrößenniveau

Eine What-If Analyse simuliert die Auswirkungen von Parameteränderungen auf die Zielgröße Zementfeinheit. Sie ermöglicht dem Bediener, das mit der Prozeßmodellierung erlernte und im neuronalen Netz gespeicherte Wissen über das Verhalten des Verfahrensschritts Zementmahlen systematisch abzurufen.

Die Optimierung der Anlageneinstellung setzt einen Produktionsplan mit Vorgaben für Produkteigenschaften, die über eine bestimmte Zementfeinheit erzielt werden können, in operationale Prozeßführungsstrategien mit prozeßnahen Sollwerteinstellungen um.

Bei einer verbesserten Prozeßführung sind erhebliche wirtschaftliche Vorteile durch schnellere Einregelvorgänge, optimierten Ressourceneinsatz und geringere Qualitätstoleranzen zu erwarten. Im vorliegenden Anwendungsfall unterstützt das neuronale Netz den Bediener der Kugelmühle hierbei durch folgende Leistungen:

- Vorschlag von Steuerhandlungen für die kostenoptimale Fahrweise der Zementmühle im Normalbetrieb
- Vorhersage des Verfahrensablaufs zur Bestätigung der eigenen Fahrweise - das Prozeßwissen steht jederzeit reproduzierbar zur Verfügung

- Früherkennung von Prozeßstörungen in der Mühle und sichere Einflußnahme auf den Prozeßablauf
- Verbesserte Qualität in der Zementproduktion

Sensitivitätsanalysen für aktuelle Anlageneinstellungen ermöglichen zustandsbezogene Rückschlüsse auf dominante Prozeßgrößen. Hieraus kann abgeleitet werden, für welche Prozeßgrößen die genaue Einhaltung vorgegebener Sollwerte besonders wichtig ist, um Qualitätsschwankungen des Endprodukts niedrig halten zu können.

Die Produktfamilie DataEngine ermöglicht die effiziente Nutzung der Prozeßdaten zur Verbesserung der Prozeßführung. Durch den Einsatz intelligenter Technologien können zusätzliche Informationen über das Prozeßverhalten gewonnen werden. Methoden der Fuzzy-Technologie und der neuronalen Netze stehen als Softwarebausteine zur Verfügung, um relevante Einflußgrößen für die Modellierung von chemisch-technischen Verfahren zu finden, Prozeßmodelle zu identifizieren und die Anlageneinstellung oder die Prozeßsteuerung zu optimieren.

Ein Überblick über verfahrenstechnische Anwendungsgebiete dieser intelligenten Technologien und deren vielfältige Einsatzmöglichkeiten mit den jeweils erzielten Verfahrensverbesserungen vermittelt [MIT GmbH 1997c]. Die Verbesserungen lassen sich verglichen mit konventionellen Modellierungs- und Optimierungsmethoden durch einen geringeren Aufwand im Rahmen einer Projektdurchführung realisieren. Das White Paper skizziert die dazu erforderlichen Arbeitsschritte.

5 LITERATUR

- Drews, F., Schmidt, H-W. (1996): Analyse von Prozeßdaten eines Zementmahlkreislaufts mit DataEngine, in MIT GmbH (Hrsg.): Anwendungssymposium Berlin, S.41-47, Aachen.
- Erens, F., Saint-Paul, O. (1995): Fuzzy Logical, in FLS Automation (Hrsg.): Review No. 15, Kopenhagen.
- Locher, G. (1996): Automatischer Wissenserwerb bei technischen Prozessen am Beispiel einer Rollenmühle, Zement - Kalk - Gips International, 49. Jg, Heft 8, S. 464-476.
- MIT GmbH (1997a): DataEngine Version 2.1 - Handbuch. Aachen.
- MIT GmbH (1997b): DataEngine ADL - Application Development Library Version 2.1 - Handbuch. Aachen.
- MIT GmbH (1997c): Intelligente Prozeßoptimierung in der Verfahrenstechnik, White Paper, Aachen.
- Østergaard, J.-J. (1996): Graphical Configuration of High Level Control Strategies, EUFIT '96, S.1905-1908.
- Sprave, J. (1995): Evolutionäre Algorithmen zur Parameteroptimierung, at - Automatisierungstechnik, Jg. 43, Heft 3, S. 110-117.
- Zimmermann, H.-J. (1996): Fuzzy Set Theory - and Its Applications. Third Edition. Kluwer Academic Publishers, Boston Dordrecht London.

Akustisches Prüfsystem auf Basis neuronaler Netze zur Qualitätskontrolle von Kassettenlaufwerken

Dipl.-Ing. M. Fochem¹, Dipl.-Phys. P. Wischniewski¹, Dipl.-Ing. R. Hofmeier²

¹ *MIT - Management Intelligenter
Technologien GmbH,
Promenade 9, 52076 Aachen,
Email: products@mitgmbh.de*

² *Philips Components
Automotive Playback Modules,
Philipsstrasse 1, 35576 Wetzlar,
Email: Ralf.Hofmeier@km-
apm.comp.philips.com*

Abstract

Der zuverlässigen Prüfung von Produkten und Erzeugnissen kommt aufgrund der erhöhten Anforderungen im Bereich der Qualitätssicherung und den gestiegenen Kundenanforderungen eine zunehmende Bedeutung zu. Dabei werden immer geringere Fehlerraten gefordert, um Imageschäden und Folgekosten aufgrund von Kundenreklamationen zu vermeiden. Dies führt dazu, daß die subjektive Geräuschprüfung durch das menschliche Ohr in zunehmendem Maße durch die Integration von Geräuschprüfständen in der Fertigungsendkontrolle ersetzt wird. Insbesondere im Bereich der Automobilindustrie, der Kfz-Zulieferindustrie sowie im Bereich der Haus- und Elektrogeräteindustrie wird die Kaufentscheidung der Kunden sehr stark vom Geräuschbild der Produkte beeinflusst. Im folgenden wird am Beispiel von Kassettenlaufwerken beschrieben, wie ein akustisches Prüfsystem zur Qualitätskontrolle auf Basis der Körperschallanalyse und anschließender Beurteilung mittels neuronaler Netze realisiert wurde. Das Prüfsystem wurde nach den Richtlinien des amerikanischen MSA (Measurement Systems Analysis) Standards zertifiziert, welcher in der QS-9000 Norm enthalten ist.

Einleitung

Philips Components APM produziert weltweit an verschiedenen Standorten Kassettenlaufwerke für den Einbau in Pkws (Abb.1). Diese müssen einen firmeninternen und standortübergreifenden hohen Qualitätsstandard erfüllen. Dabei sind neben der Hifi-Qualität insbesondere auch das Geräuschbild und die Laufruhe wichtige Kriterien, um den gestiegenen Kundenansprüchen zu entsprechen. Erschwert wird die Einhaltung der hohen Qualitätsstandards unter anderem durch die weltweite Verteilung der Produktionsstandorte, das in den einzelnen Standorten vorhandene Schichtbetriebssystem sowie auftretende Umgebungsgeräusche in den Produktionshallen.

Die Prüfung der Laufwerke erfolgte dabei bisher durch erfahrene Mitarbeiter, die auf Basis einer kurzen Hörprobe die Qualität der Kassettelaufwerke beurteilten, einhergehend mit all den bekannten Problemen, die bei derartigen Prüfmethoden auftreten können. Die unterschiedlichen Empfindungen und Beurteilungskriterien bei der Prüfung schwanken hierbei nicht nur von Mitarbeiter zu Mitarbeiter, sondern sind auch von dessen Tageskonstitution abhängig. Darüber hinaus machen störende Umgebungsgeräusche eine objektive Qualitätsbeurteilung des Laufgeräusches sehr schwierig. Neben den wirtschaftlichen Nachteilen bereitete somit vor allem diese subjektive Prüfung durch den Prüfspezialisten und die Reproduzierbarkeit seiner Entscheidung den Verantwortlichen in der Qualitätssicherung Kopfzerbrechen.

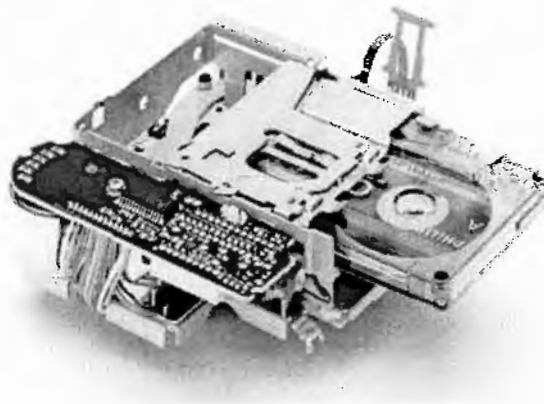


Abb. 1: Chassis eines Kassettelaufwerks

Ziel war es, ein akustisches Prüfsystem zur Online Qualitätskontrolle der Kassettelaufwerke zu entwickeln, um an allen Produktionsstätten mit dem selben Prüfsystem arbeiten zu können und einen einheitlichen Qualitätsstandard zu gewährleisten. Neben der Verbesserung der Prüfsicherheit sollen mit der automatischen Geräuschprüfung nicht nur Erkenntnisse über das akustische Verhalten der Laufwerke erzielt werden, sondern auch über Montage-, Fertigungs- und Funktionsfehler, um diese umgehend beseitigen zu können.

Die Philips APM übernahm hierbei die Konzeption und Konstruktion des Prüfstandes einschließlich der kompletten Sensorik und Meßwertaufnahme. Auf Seiten der MIT GmbH wurden die Schritte Signalanalyse, Merkmalsauswahl, Erstellung des Klassifikators und Implementierung der Software durchgeführt. Die Software sollte dabei unter LabVIEW® entwickelt werden, da LabVIEW® eine benutzerfreundliche Gestaltung der Bedienoberfläche und einfache Integration in den Fertigungsprozeß unterstützt. Darüber hinaus kam DataEngine Vi zum Einsatz, eine Zusatzbibliothek,

die Verfahren zur intelligenten Signal- und Datenanalyse enthält. Im folgenden soll auf die einzelnen Schritte und Komponenten detaillierter eingegangen werden.

Sensorik und Meßwertaufnahme

Die Schwingungen der beweglichen Bauteile des Kassettenlaufwerks wie Zahnräder, Lager, Motoren, Wellen usw. breiten sich von ihrem Entstehungsort über angrenzende Bauteile weiter aus und werden an den Grenzflächen an die Umgebungsluft abgegeben. Zur Bewertung der Kassettenlaufwerke ist der Körperschall, wie die Schwingungen innerhalb des Laufwerks bezeichnet werden, besser geeignet als der Luftschall, da er weniger äußere Störgrößen wie z.B. den Lärm der umgebenden Produktionsmaschinen enthält. Der Körperschall wird an der Laufwerksoberfläche mit Hilfe eines Körperschallsensors (Piezosensor) aufgenommen.

Hierzu wurde ein spezieller Prüfstand konzipiert und aufgebaut, in den die Kassettenlaufwerke eingespannt werden, so daß sie auf dem Sensor fest zur Auflage kommen (vgl. auch Abb. 5). Der Sensor formt die mechanischen Schwingungen in ein elektrisches Signal um und stellt es zur weiteren Verarbeitung einem PC mit Meßwerterfassungskarte zur Verfügung. Zum Einsatz kommt eine Soundkarte, die als Vorverstärker dient, sowie eine Signalprozessorkarte des Typs AT-DSP2200 von National Instruments, welche die Schwingungen mit einer Abtastrate von 8 kHz aufnimmt und digitalisiert. Die Abtastzeit beträgt rund 4 Sekunden, so daß insgesamt 32768 Abtastwerte je Messung für die anschließende Signalanalyse zur Verfügung stehen.

Signalanalyse

Die Analyse der gemessenen Signale hat die Aufgabe, Merkmale zu gewinnen, mit denen sich die Prüflingeigenschaften beschreiben lassen. Auf die Körperschallsignale bezogen heißt dies, daß die mechanischen Fehler erkannt werden sollen, die zu einem erhöhten oder charakteristischen Schwingungsverhalten führen.

Typische Merkmale können die Varianz, die Schiefe, die Wölbung oder der Crestfaktor des gemessenen Signals sein. Aber auch das Vorhandensein einzelner Frequenzanteile im Signal kann charakteristisch sein für die Unterscheidung der verschiedenen Klassen, die im Vorfeld der Untersuchungen von den Prüfexperten definiert wurden. Dabei gilt es die fehlerfreien Laufwerke von insgesamt 4 unterschiedliche Fehlerklassen zu trennen. Typische Zeitsignale für jede der einzelnen Fehlerklassen sowie für die Klasse "in Ordnung" sind in Abb. 2 dargestellt.

Aus den Zeitsignalen sind bereits einige fehlerspezifische Eigenschaften deutlich erkennbar wie z.B. die Peaks in Abb. 2④. Weitere relevante Merkmale ergeben sich aus dem Amplitudenspektrum nach Durchführung einer FFT und aus dem

Leistungscepstrum. Abb. 2 ① bis ④ zeigen Beispiele der vier Fehlerklassen, Abb. 2 ⑤ ist die Darstellung des Signals eines intakten Kassettenslaufwerks.

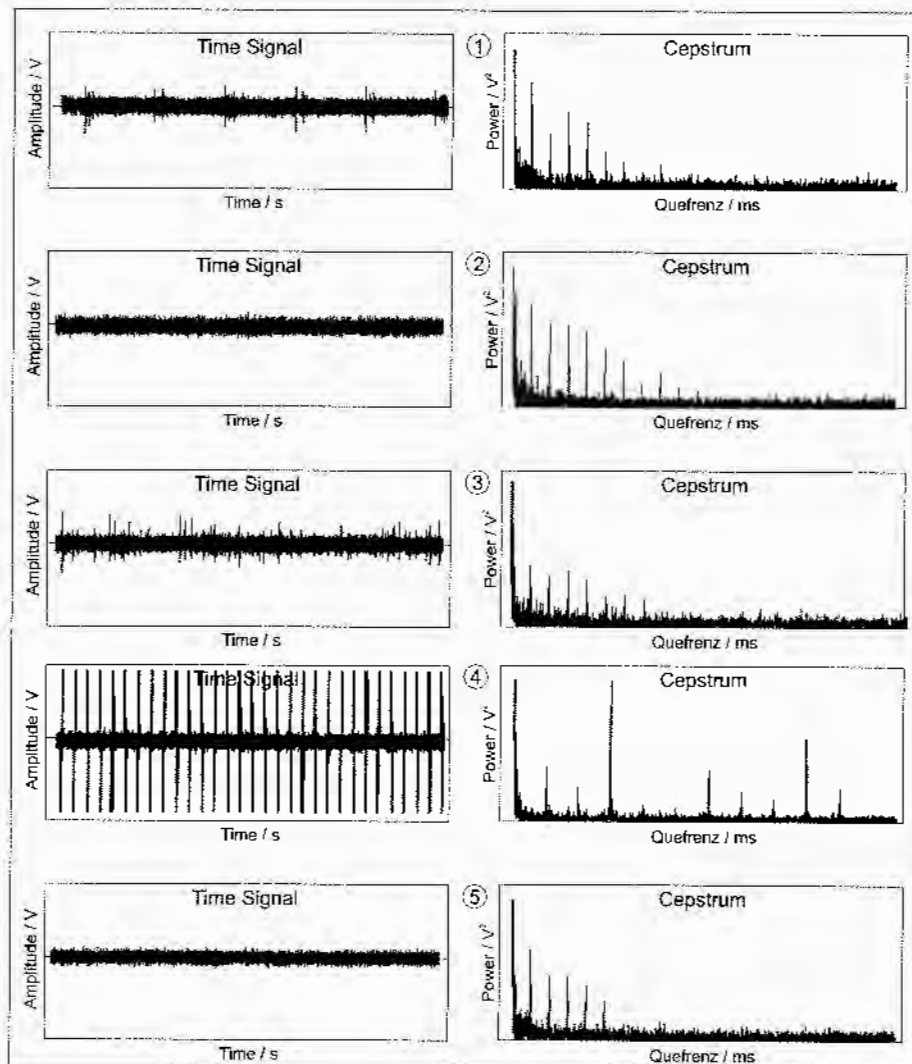


Abb. 2: ① bis ④ zeigen Beispiele von Zeitsignalen und Cepstren der vier Fehlerklassen, ⑤ ist die Darstellung des Signals eines intakten Kassettenslaufwerks.

Der Wahl der richtigen Merkmale kommt eine entscheidende Bedeutung zu, da die spätere Klassifikation im Online Betrieb nur so gut sein kann, wie die ihr zugrunde liegenden Merkmale /7/. Die Merkmale müssen das Kassettenlaufwerk möglichst vollständig hinsichtlich seiner akustischen Eigenschaften beschreiben. Daher wird zusätzlich auf ein Verfahren der Signalanalyse, das Cepstrum, zurückgegriffen, welches sich insbesondere bei rotierenden Bauteilen zur Fehlerdiagnose bewährt hat. Dieses ist zwar rechenaufwendig, liefert aber aussagekräftige Merkmale für die spätere Klassifikation.

Das Cepstrum

Die Cepstrumanalyse ist im Bereich der akustischen Qualitätskontrolle ein relativ neues Verfahren. Es erlaubt unter anderem die Detektion regelmäßiger Strukturen in einem Frequenzspektrum. Solchen regelmäßigen Strukturen wie Harmonischen, Seitenbändern, Echos und Reflexionen begegnet man häufig im Zusammenhang mit Fragen der Fehlerdiagnose bei rotierenden Maschinen und Bauteilen. Verstärktes Auftreten von Harmonischen und Seitenbandfamilien kann ein Anzeichen für bestimmte Fehler sein. Das Cepstrum ist dabei allgemein definiert als inverse Fouriertransformation einer logarithmierten Frequenzbereichsfunktion /5/. In der Praxis wird häufig das Leistungsscepstrum verwendet, welches sich aus dem Leistungsspektrum ableitet.

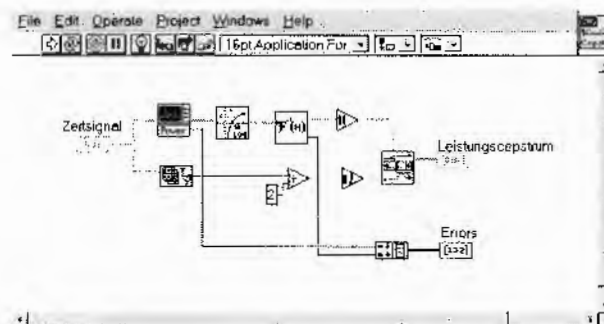


Abb. 3: Umsetzung des Cepstrums unter LabVIEW®

Mit den in LabVIEW® vorhandenen Routinen zur Signalverarbeitung wurden insgesamt 32 Merkmale aus dem Zeit-, Frequenz- und Queffrenzbereich des Laufgeräusches berechnet und für ein neuronales Netz nach Kohonen aufbereitet, welches im Prüfsystem zum Einsatz kam.

Gerade neuronale Netze weisen bei geringen Datenmengen und verrauschten Signalen Vorteile gegenüber klassischen Verfahren auf. Bei der Auswertung und Beurteilung kommt die Robustheit neuronaler Netze gegenüber gestörten Eingangssignalen voll zum Tragen. Darüber hinaus verfügen neuronale Netze über die Fähigkeit der Abstraktion und Generalisierung /1/. Dadurch können auch Signale, die nicht exakt den Lernbeispielen entsprechen, zuverlässig klassifiziert werden.

Selbstorganisierende Merkmalskarten von Kohonen

Die selbstorganisierenden Merkmalskarten von Kohonen - im folgenden kurz als Kohonen Netze bezeichnet - gehören zu den Netzwerkmodellen, die unüberwacht lernen und in der Lage sind, Signale selbsttätig in unterschiedliche Klassen einzuteilen /6/. Das Netzmodell selbst besteht dabei im allgemeinen aus einer zweidimensionalen Neuronenanordnung (vgl. Abb. 4) mit Verbindungsgewichten zur Eingabeschicht - dem Gewichtsvektor des Neurons. Die Kohonen Netze haben die Fähigkeit zur Selbstorganisation, da die Neuronen in einer festen Nachbarschaftsbeziehung zueinander stehen. Dies ermöglicht es, Unterschiede in den Eingabesignalen durch räumliche Entfernungen im Netz widerzuspiegeln. Der einfache Aufbau und die simple Lernregel prädestinieren sie zu Lösungsmöglichkeiten für eine Vielzahl von Steuerungs- und Klassifikationsaufgaben.

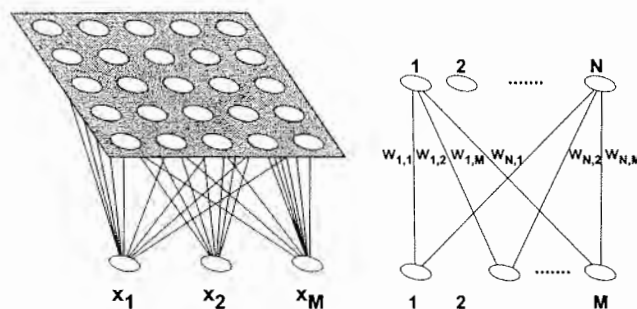


Abb. 4: Selbstorganisierende Merkmalskarte nach Kohonen

Ein Eingangssignal wird durch einen mehrdimensionalen Ortsvektor repräsentiert. Während der Trainingsphase wird in jedem Lernschritt ein Eingangssignal zufällig ausgewählt, das Neuron im Netz bestimmt, welches den geringsten euklidischen Abstand zwischen Ortsvektor des Eingangssignals und Gewichtsvektor des Neurons aufweist, und dessen Gewichtsvektor in Richtung des Ortsvektors verschoben. Der Lernalgorithmus bewirkt weiterhin, daß diese Anpassung der Neuronen, die auch als Adaption bezeichnet wird, nicht einzeln sondern im Verbund mit weiteren, auf der Karte benachbarten Neuronen geschieht. In folgenden Lernschritten mit weiteren

Eingangsvektoren werden mit der Zeit sowohl die umgebende Nachbarschaft als auch die Lernschrittweite, also die Größe der Verschiebung, allmählich reduziert. Dies hat zur Folge, daß das Kohonen Netz sich selbsttätig so organisiert, daß signalähnliche Eingangsmuster durch benachbarte Neuronen auf der Karte repräsentiert werden.

Trainingsprozeß

Im Überblick stellt sich der Trainingsalgorithmus folgendermaßen dar:

1. Alle Gewichtsvektoren werden zufällig vorbesetzt
2. Ein Eingangssignal wird zufällig ausgewählt, entsprechend der Verteilungsdichte der Eingangssignale, und dem Netz präsentiert
3. Das Neuron („Winner-Neuron“), welches die größte Ähnlichkeit mit dem Eingangsvektor aufweist, wird bestimmt. Dies entspricht dem Neuron, welches den geringsten euklidischen Abstand zwischen seinem Gewichtsvektor und dem Ortsvektor des Eingangssignals aufweist.
4. Das „Winner-Neuron“ und alle seine benachbarten Neuronen werden gemäß der Lernvorschrift modifiziert (adaptiert)
5. Verringern der Lernrate und dem Nachbarschaftsradius
6. Fahre fort bei Schritt 2 solange nicht alle Lernschritte durchgeführt wurden. Ansonsten beende das Training

Labeling

Nach dem Training des Netzes ist es möglich, ein unbekanntes Eingangssignal dem Netz zu präsentieren und eine Ausgabe zu erhalten.

Dabei wird wiederum verglichen, welcher Gewichtsvektor der Neuronen den geringsten euklidischen Abstand zum Ortsvektor des Eingangssignals hat, und dieses „Winner-Neuron“ wird an den Anwender weitergegeben. Diese Information ist für den Anwender jedoch nicht von praktischer Bedeutung, vor allem nicht, wenn er das Netz als Klassifikator nutzen will. Daher ist es notwendig, in einem überwachten Schritt jedem Neuron eine der vorher definierten Bedeutungsklassen zuzuweisen. Dieser Schritt wird als „Labeling“ bezeichnet. Hierzu wurde die manuelle Einschätzung der Experten herangezogen, die vorab 250 ausgewählte Laufwerke auf Basis ihrer akustischen Geräuschsignale bewertet hatten. Abbildung 5 zeigt ein gelabeltes Kohonen Netz, wie es zur Klassifikation eingesetzt werden kann. Deutlich ist eine Clusterbildung von Neuronen auf der Karte zu erkennen, die ähnliche Eingangssignale repräsentieren.

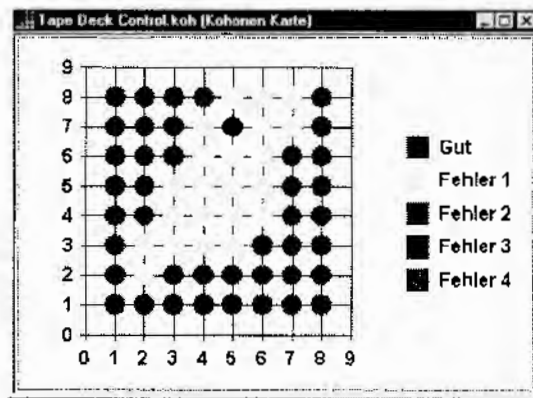


Abb. 5: Gelabeltes Kohonen Netz.

Online Klassifizierung

Nach Fertigstellung des Prüfstands, Kalibrierung der MeßgröÙe und Integration in den Fertigungsablauf kann somit eine 100% Endkontrolle der Kassettenlaufwerke erfolgen. Abbildung 6 zeigt den schematischen Aufbau des Qualitätssicherungssystems. Dabei ist die Signalkette von der Meßwertaufnahme über die Merkmalsberechnung bis hin zur Klassifikation durch das Kohonen Netz dargestellt.

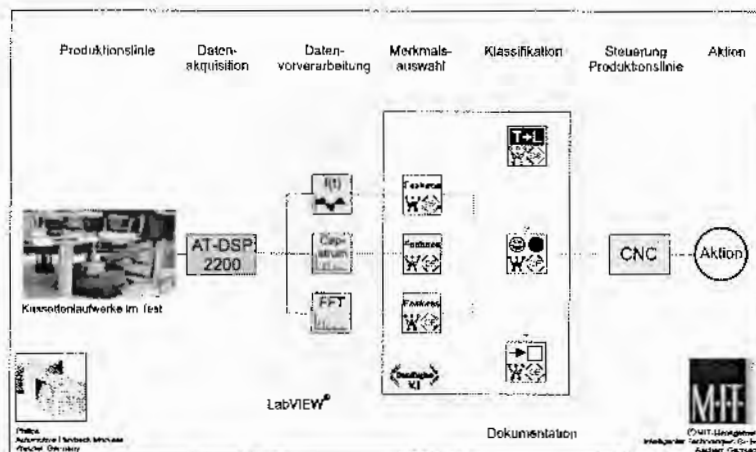


Abb. 6: schematischer Aufbau des Qualitätssicherungssystems

Die Klassifikationsergebnisse (gut/schlecht) einschließlich der entsprechenden Fehlerklasse werden auf der Bedienoberfläche als grün/rote Symbole dargestellt. Dies ermöglicht den Einsatz des Prüfstandes im internationalen industriellen Umfeld unabhängig von den Sprachkenntnissen des Bedieners. Für den Bediener nicht sichtbar werden die Klassifikationsergebnisse auf einer Festplatte protokolliert (Abb. 7) und über ein Steuersignal werden fehlerhafte Laufwerke automatisch aussortiert. Produktänderungen können ohne zeitraubende Programmier- und Anpassungsschritte durchgeführt werden. Lediglich ein anderer Klassifikator muß für das neue Produkt, z.B. CD-Laufwerke, in die bestehende Anwendung geladen werden.

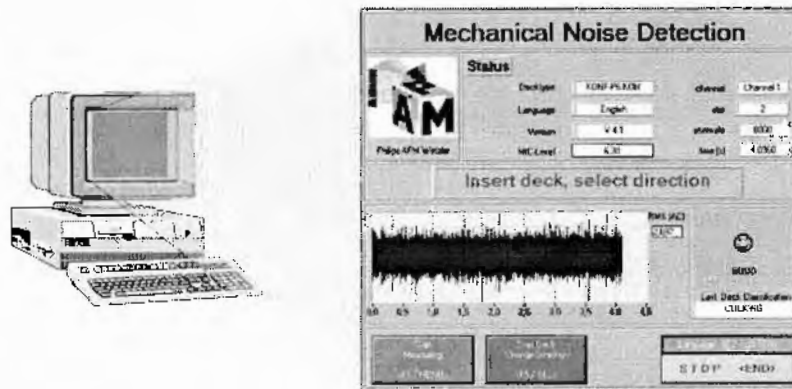


Abb. 7: Oberfläche des Prüfsystems

Zusammenfassung

Dieser Anwendungsbericht beschreibt wie neuronale Netze erfolgreich zur Qualitätskontrolle in industriellen Produktionslinien eingesetzt werden können. Auf die beschriebene Art und Weise lassen sich fehlerhafte Laufwerke erfolgreich und bedeutend zuverlässiger als durch manuelle Beurteilung aussortieren. Darüber hinaus ermöglicht die Zuordnung in Fehlerklassen bei gehäuftem Auftreten eines Fehlers eine schnelle und zielgerichtete Korrektur innerhalb des laufenden Produktionsprozesses.

Erste Ergebnisse an den Produktionsstandorten in Wetzlar und in Ungarn bestätigen den Erfolg des entwickelten Prüfsystems zur akustischen Qualitätskontrolle. Neben der Verbesserung der Prüfsicherheit und Reproduzierbarkeit der Prüfentscheidung liefert nun eine statistische Auswertung der Prüfprotokolle Hinweise und Erkenntnisse über Montage-, Fertigungs- und Funktionsfehler. Das Prüfsystem wurde in der Zwischenzeit

nach den Richtlinien des amerikanischen MSA (Measurement Systems Analysis) Standards zertifiziert, welcher in der QS-9000 Norm enthalten ist.

Die graphische Benutzeroberfläche von LabVIEW® sowie die bedienerfreundlichen Routinen in DataEngine V.i ermöglichen die online Anwendung in einer rauen industriellen Umgebung ohne zeitraubende Analyse in der Entwicklungsphase. Dies bestätigte sich auch bei weiteren Anwendungen wie z.B. der Füllstandskontrolle /2/, der Prüfung von Waschmaschinen /4/ sowie in der akustischen Qualitätskontrolle von Elektromotoren /3/. Die eingesetzten Verfahren können ohne weiteres auch auf andere Applikationen übertragen und angewendet werden /8/.

Referenzen

- /1/ MIT 1996. DataEngine V.i 1.5 User Manual. Aachen, Germany
- /2/ Mikenina L., Weber R.: Fuzzy Liquid Level Control in LabVIEW. NI-week 1996
- /3/ Fogliardi R., Dr. Eng., Wischnewski P., Fochem M.: Online Fuzzy Identification of Noisy Electric Motors on the Production Line with LabVIEW® and DataEngine V.i, EUFIT 1997.
- /4/ Fochem M., Weber R., Wischnewski P. Classifier Design Using Intelligent Technologies and Integration into Production Line of Household appliances. Data Analysis Symposium, 1997
- /5/ Kolerus J., Zustandsüberwachung von Maschinen, Expert Verlag, 1986
- /6/ Kohonen T.: Self-Organization and Associative Memory, 3rd edition, Springer Verlag, Berlin, Heidelberg, New York, 1989
- /7/ Niemann H., Klassifikation von Mustern, Springer Verlag, Berlin 1983
- /8/ Zimmermann H.-J. (Hrsg.), Datenanalyse: Anwendungen von DataEngine mit Fuzzy Technologien und Neuronalen Netzen, VDI-Verlag, Düsseldorf 1995

Vorstellung eines Fuzzy Entscheidungssystems zur Bonitätsbeurteilung von Unternehmen

T. Barczewski¹, H. J. Rust¹, R. Weber², H. Zygan²

1) Allgemeine Kreditversicherung AG, Mainz

Tel.: 06131/323-446, Fax: 06131/37 27 66

2) MIT-Management Intelligenter Technologien GmbH, Promenade 9, 52076 Aachen

Fax: 24 08 - 9 45 82; <http://www.mitgmbh.de>; e-mail: rw@mitgmbh.de

ABSTRAKT: Aufgrund der ständig steigenden Zahl in- und ausländischer Geschäftsbeziehungen einerseits und dem Anstieg der Unternehmensinsolvenzen andererseits nimmt die Kreditversicherung eine immer wichtiger werdende Rolle im Geschäftsleben wahr. Da die Anzahl der von Kreditversicherern zu beurteilenden Unternehmen ebenfalls ständig zunimmt, wird die automatische Bonitätsbeurteilung von Unternehmen zu einem Faktor, der auf Erfolg oder Mißerfolg von Kreditversicherern einen wesentlichen Einfluß hat. Bereits in diesem Bereich installierte automatische Systeme arbeiten gewöhnlich auf der Basis sequentieller Entscheidungsregeln. Fehlende Kompensationseffekte können in Grenzbereichen zu automatischen Entscheidungen führen, die von denen der Kreditexperten abweichen. Dieser Beitrag beschreibt ein System, das mit der Intention entwickelt wurde, auf der Grundlage von Fuzzy Methoden die automatischen Bonitätsbeurteilungen von Unternehmen zu verbessern.

1. Einführung

Die Bonitätsbeurteilung von Unternehmen ist ein Bereich, in dem bereits zahlreiche Techniken zur automatischen Entscheidung untersucht und eingesetzt wurden. Dazu gehören u. a. sowohl konventionelle statistische Verfahren, wie beispielsweise lineare Diskriminanzanalyse als auch Ansätze aus dem Gebiet der Neuronalen Netze [Beuter et. al. 1994]. Während das Verhalten Neuronaler Netze nur schwer nachzuvollziehen ist, bilden in Grenzbereichen die auf zweiwertiger Logik basierende Entscheidungssysteme das Verhalten der Kreditexperten nur unzulänglich ab.

Fuzzy Technologien stellen besonders in Situationen, in denen menschliches Erfahrungswissen abgebildet wird, ein sehr gut geeignetes Instrumentarium zur Verfügung, um das entsprechende Wissen angemessen in einem Rechner zu erfassen und zur Problemlösung einzusetzen. Um diese Vorteile für die automatische Bonitätsbeurteilung von Unternehmen und damit auch für automatische Kreditentscheidungen zu nutzen, wurde bei der Allgemeinen Kreditversicherung AG in Zusammenarbeit mit dem MIT ein System für automatische Kreditentscheidungen entworfen, das auf den Methoden der Fuzzy Logik beruht. Dieses Fuzzy System wird in diesem Beitrag beschrieben.

Kapitel 2 stellt die Allgemeine Kreditversicherung, das Aufgabengebiet eines Kreditversicherers und die speziell betrachtete Aufgabenstellung vor. In Kapitel 3 wird das entwickelte Fuzzy System vorgestellt und in Kapitel vier ein Ausblick auf das weitere Vorgehen gegeben.

2. Die Allgemeine Kreditversicherung

Die Allgemeine Kreditversicherung AG ist mit einem Prämienvolumen von mehr als 400 Mio. DM in 1996 einer der führenden Kreditversicherer der Bundesrepublik Deutschland. Das Versicherungsangebot der Allgemeinen Kreditversicherung AG umfaßt folgenden Sparten:

- Warenkreditversicherung (WKV)
- Ausfuhrkreditversicherung (AKV)

- Investitionsgüterversicherung (IKV)
- Kautionsversicherung (KTV)
- Vertrauensschaden-/Computermißbrauchversicherung (VSV/CMV)

Mit einem Prämienvolumen von mehr als 247 Mio. DM in 1996 ist der Bereich Warenkreditversicherung (WKV) der größte Geschäftszweig der Allgemeinen Kreditversicherung AG. Für diesen Geschäftszweig wurde das in diesem Beitrag beschriebene Fuzzy System entwickelt und soll nach erfolgreicher Implementierung auch auf den Bereich der Ausfuhrkreditversicherung (AKV) ausgeweitet werden.

Im Rahmen der WKV versichert die Allgemeine Kreditversicherung AG Lieferungen und Sonstige Leistungen (Dienstleistungen etc.) inländischer Firmen an ihre inländischen Kunden. Da Lieferungen/Leistungen i.A. auf Ziel erbracht werden, läuft der Lieferant in Gefahr, daß sein Kunde nach erbrachter Lieferung/Leistung (Rechnungsschreibung) aber noch nicht gezahlter Rechnung insolvent wird. Dies führt nicht nur zu einem Forderungsausfall, sondern kann sogar eine Insolvenzgefährdung des Lieferanten zur Folge haben.

Zur Absicherung gegen Forderungsausfälle bietet die Allgemeine Kreditversicherung AG dem Lieferanten (=Versicherungsnehmer) die Leistungen Schadenverhütung im Vorfeld einer Lieferung/Leistung an bzw. Schadenvergütung im Falle der Zahlungsunfähigkeit eines Kunden.

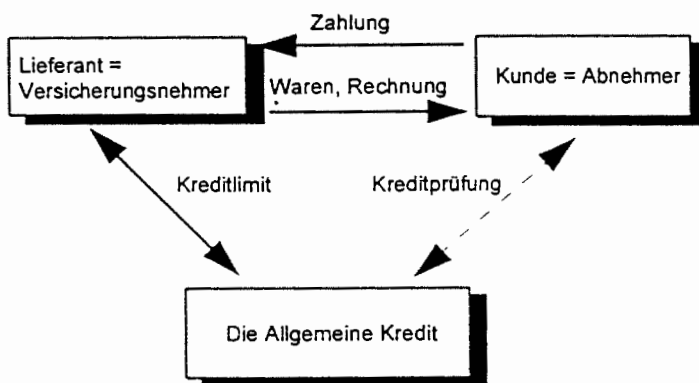


Abbildung 1: Ablauf der Kreditprüfung

Zum Zwecke der Schadenverhütung muß der Versicherungsnehmer im Vorfeld seiner Lieferungen den Versicherungsschutz für die zu beliefernden Kunden beantragen (siehe Abbildung 1), insofern der Wert der Lieferungen einen im Vertrag festgelegten Betrag überschreitet (Unterhalb dieses Betrages ist der Lieferant, falls vertraglich vereinbart, für die Bonitätsprüfung seines Kunden selbst verantwortlich, aber auch in diesem sogenannten Pauschaldeckungsbereich unterstützt die Allgemeine Kreditversicherung AG ihre Versicherungsnehmer über ihr Produkt Pauschaldeckung-Online). Die Beantragung der Versicherungssumme erfolgt über einen Kreditantrag bei der Allgemeinen Kreditversicherung AG. Nach einer Kreditwürdigkeitsprüfung

des Kunden durch die Allgemeine Kreditversicherung AG, wird der Versicherungsnehmer über eine Kreditentscheidung benachrichtigt, ob der Versicherungsschutz voll (die beantragte Versicherungssumme wird voll gewährt), teilweise (die gewährte Versicherungssumme liegt unter der beantragten Versicherungssumme) oder gar nicht gewährt wird. Mindestens einmal jährlich erfolgt im Rahmen einer sogenannten Folgeprüfung eine turnusmäßige Überwachung aller Kunden (Risiken).

Den monatlich bei der Allgemeinen Kreditversicherung AG anfallenden 60.000 Kreditentscheidungen stehen bei der

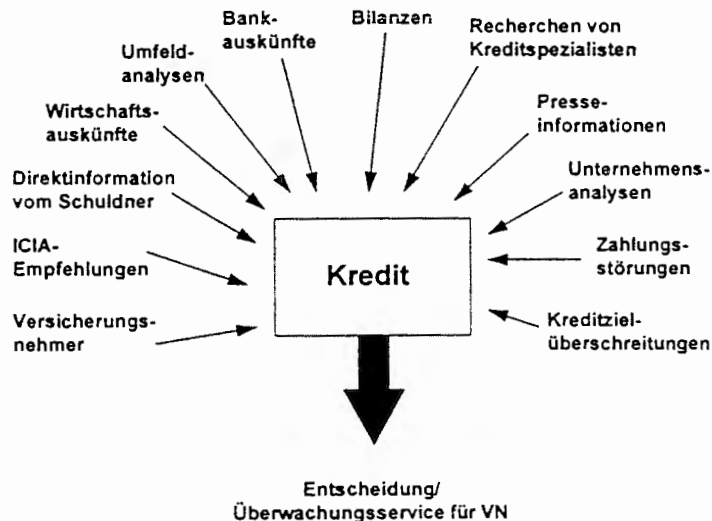


Abbildung 2: Informationsquellen

Allgemeinen Kreditversicherung AG weniger als 100 Kreditprüfer zur Bearbeitung gegenüber. Dieser hohe Umfang erfordert eine Unterstützung der Kreditprüfer durch die EDV. Unterschiedlichste Verfahren bilden das Entscheidungsverhalten der Kreditprüfer in der EDV ab und entlasten somit den Kreditprüfer vom Massengeschäft. Grundlagen für die Kreditentscheidungen bilden diverse Informationsquellen, die in Abbildung 2 aufgezeigt werden. Das in diesem Artikel beschriebene Fuzzy System führt seine Entscheidungen auf der Basis von Wirtschaftsauskünften aus, die in codierter Form vorliegen.

Kreditanträge können bei der Allgemeinen Kreditversicherung AG über die unterschiedlichsten Medien (Telefax, Teletex, Brief, T-Online, ...) gestellt werden. Seit dem Jahr 1990

stellt die Allgemeine Kreditversicherung AG ihren Versicherungsnehmern ein Online-System zur Verfügung (Allgemeine Kredit Online) über das sie u.a. interaktiv bei der Allgemeinen Kreditversicherung AG Kreditanträge stellen können. Liegt die Information über den angefragten Kunden bei der Allgemeinen Kreditversicherung AG vor, so werden die Anträge gemäß den in der EDV implementierten Verfahren für den größten Teil der Fälle in Sekundenschnelle entschieden. Kann ein Antrag nicht automatisch entschieden werden, so wird er einem Kreditprüfer zur Entscheidung vorgelegt.

Das bestehende System basiert auf sequentiell ausgeführten Entscheidungsregeln, die einfachen Abfragen entsprechen. Werden alle Entscheidungsregeln positiv durchlaufen, wird eine positive Kreditentscheidung getroffen. Die Verletzung einer einzigen Regel kann zu der Ablehnung des Kreditantrages führen, wodurch für die geplante Warenlieferung kein Versicherungsschutz gewährt wird.

Ablehnungen erfolgen, wenn die Bonität des Unternehmens als zweifelhaft oder gering eingestuft wird. So sind junge Unternehmen insolvenzgefährdeter als alte Firmen. Eine geschäftspolitische Entscheidung der Allgemeinen Kreditversicherung AG führt deshalb zu der Ablehnung von Unternehmen, deren Gründung am Tag der Entscheidung noch keine zwei Jahre zurückliegt. In Extremfall wird somit ein Unternehmen mit ansonsten guten Bonitätsmerkmalen nur deshalb abgelehnt, weil es zum Zeitpunkt der Entscheidung noch keine 24 Monate alt ist. Die Nachteile dieses Systems sind offensichtlich

- scharfe Grenzen
- fehlende Kompensationseffekte

Ein Kreditprüfer entscheidet aufgrund seines Gesamteindrucks über die vorliegende Information. Um das automatische Kreditentscheidungssystem zu verbessern und die automatischen Kreditentscheidungen besonders in den Grenzbereichen an das Entscheidungsverhalten der Kreditprüfer anzupassen, fiel die Wahl für eine Umsetzung des Entscheidungsverhalten in der EDV auf das Verfahren der Fuzzy Logic. Der Entschluß für die Fuzzy Logic resultierte aus der folgenden Zielsetzung:

- Nachbildung der Arbeit der Kreditprüfer bei der Bonitätsprüfung (keine scharfen Grenzen)
- Verarbeitung von Wissen der Kreditprüfer, das in umgangssprachlicher Form vorliegt.
- Simultane Betrachtung mehrerer Merkmale einer Firma. (Keine sequentielle Prüfung einzelner Merkmale einer Firma.)
- Eine gegenseitige *Kompensation* einzelner Merkmale soll möglich sein.
- Als Ergebnis der Bonitätsprüfung soll ein Bonitätsgrad berechnet werden.

Das Fuzzy System soll gegenüber dem herkömmlichen sequentiellen Verfahren folgende Vorteile bringen:

- qualitativ bessere Kreditentscheidungen durch Aufweichen starrer Grenzen und Berücksichtigung von Kompensationseffekten
- Höhere Kundenzufriedenheit durch qualitativ bessere Kreditentscheidungen
- Entlastung der Kreditabteilungen durch höhere Kundenzufriedenheit (weniger Revisionsentscheidungen).

Der folgende Abschnitt beschreibt die Umsetzung dieser Ziele.

3. Beschreibung des Fuzzy Systems zur Bonitätsbeurteilung

In der Literatur findet man zahlreiche unterschiedliche Vorgehensweisen zur Kreditwürdigkeitsbeurteilung, die auch auf den vorliegenden Fall übertragen werden können. Dazu gehören beispielsweise [Romaniuk, Hall 1992], [Rommelfanger 1993] und [Zimmermann, Zysno 1983]. Diese Aufzählung erhebt nicht den Anspruch auf Vollständigkeit, spiegelt aber dennoch die gebräuchlichsten Fuzzy Ansätze zur Kreditwürdigkeitsbeurteilung wider. Dies sind konnektionistische Ansätze, regelbasierte Fuzzy Systeme und hierarchische Aggregation der vorliegenden Eingangsinformation. Nachfolgend wird das für die Allgemeine Kreditversicherung AG erstellte regelbasierte System und die Ergebnisse aus dessen Anwendung präsentiert.

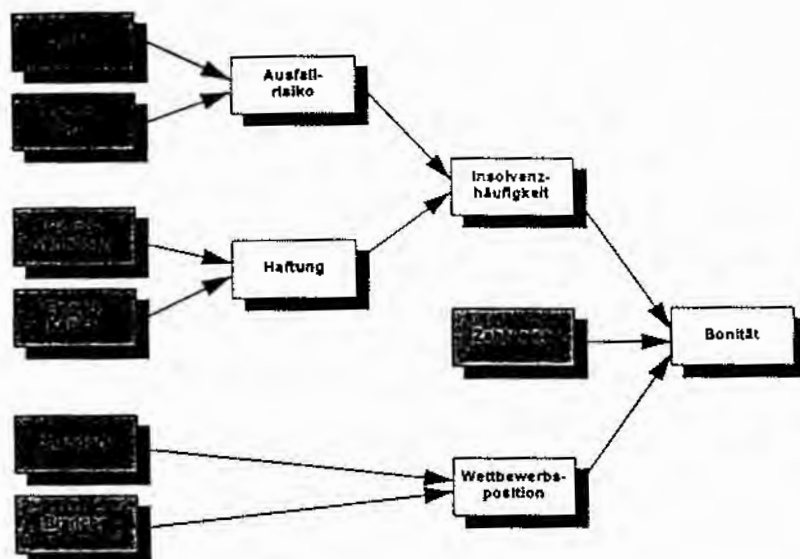


Abbildung 3: Das Modell für WKV

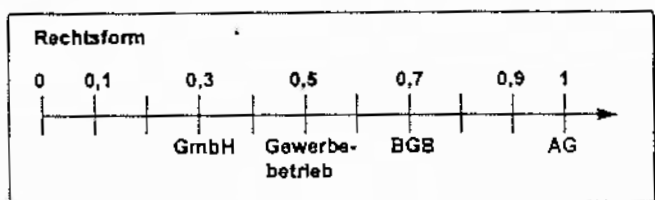


Abbildung 4: Rechtsformbewertung

Die im Modell definierten Größen wurden als linguistische Variablen definiert. So wird z.B. die Größe Alter einer Firma durch die Terme jung, mittel und alt beschrieben. Das Alter wird im Bereich von 12-24 Monaten „unscharf“.

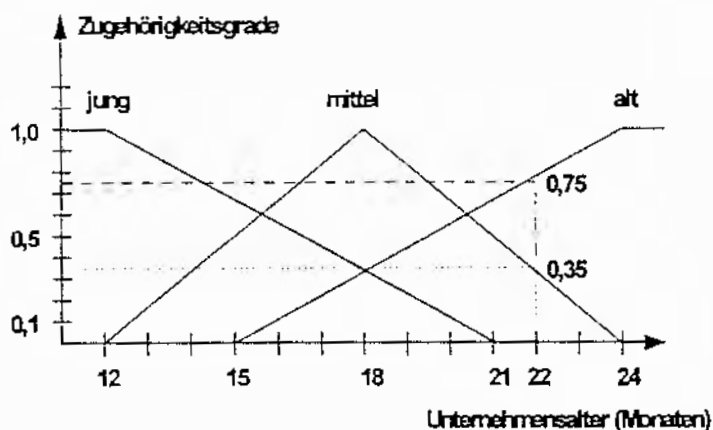


Abbildung 5: Linguistische Variable Alter

Die im Modell definierten Größen wurden als linguistische Variablen definiert. So wird z.B. die Größe Alter einer Firma durch die Terme jung, mittel und alt beschrieben. Das Alter wird im Bereich von 12-24 Monaten „unscharf“.

Das Fuzzy System wurde für den Geschäftsbereich Warenkreditversicherung entwickelt, um die automatischen Kreditentscheidungen qualitativ zu verbessern. Eine Ausweitung auf den Bereich Auslandskreditversicherung ist geplant. Die Entscheidungen des Systems basieren auf den Inhalten von Wirtschaftsauskünften. Die erste Aufgabe für die Experten bestand in der Auswahl der bonitätsrelevanten Merkmale der Wirtschaftsauskunft und der Verknüpfung dieser Merkmale zu einer Bonitätsgröße. Dies resultierte in dem in der Abbildung 3 angegebenen Modell.

Im Gegensatz zum Alter einer Firma ist z.B. die Rechtsform von Firmen a priori nicht meßbar. Deshalb wurden im zweiten Schritt die nicht direkt aus der Wirtschaftsauskunft zu quantifizierenden Größen durch die Kreditprüfer bewertet. Ein Beispiel stellt die Bewertung von Rechtsformen auf einer Skala von 0 bis 1 dar. Eine AG ist mit 1 als sehr gute Rechtsform bewertet, eine GmbH erhält mit 0,3 eine unterdurchschnittliche Bewertung. Die nicht meßbaren Größen Branche und Standort wurden von den Experten in analoger Art und Weise bewertet.

Eine Firma, die noch kein Jahr alt ist, soll auf jeden Fall abgelehnt werden. Ein Firma, die bereits mehr als zwei Jahre existiert, soll „vollständig als alt“ und damit zumindest im Bezug auf ihr Alter als voll kreditwürdig gelten. In dem Bereich zwischen ein und zwei Jahren verbessert sich die Kreditwürdigkeit der Firma aufgrund der Definition der Zugehörigkeitsfunktionen (siehe Abbildung 4) und der aufgestellten Regeln praktisch täglich. Ob ein Antrag auf eine mehr als ein Jahr alte Firma gezeichnet wird, hängt von den individuellen Ausprägungen der weiteren Bonitätsmerkmale ab.

Die Bonitätsmerkmale einer Wirtschaftsauskunft werden über Regelbasen zu weiteren Größen aggregiert, die schließlich in der Bonität der betrachteten Firma resultieren. Die Regelbasis, die die Bonitätsmerkmale Rechtsform und Alter zu der Größe Insolvenzhäufigkeit aggregieren, finden Sie in der folgenden Tabelle:

Insolvenzhäufigkeit:

Alter→ Rechtsform↓	zu jung	<i>jung</i>	mittel	alt
<i>gut</i>	sehr hoch	<i>hoch</i>	mittel	sehr niedrig
mittel	sehr hoch	sehr hoch	hoch	niedrig
schlecht	sehr hoch	sehr hoch	hoch	mittel

Die anderen Regeln sind nach einem ähnlichen Schema aufgebaut

Das vorgestellte Fuzzy System wurde mit DataEngine erstellt und getestet [MIT 1996]. Der Test des Systems erfolgte anhand von 46 Wirtschaftsauskünften aus der Datenbank der Allgemeinen Kreditversicherung. Um ein Maß für die Güte des Systems zu erhalten, wurde das Ergebnis des Systems mit den Entscheidungen von sieben Kreditprüfern und der Entscheidung des konventionellen Systems verglichen:

- Das konventionelle System entscheidet positiv über einen Kreditantrag, falls alle Kriterien unabhängig voneinander erfüllt sind.
- Das Fuzzy System berechnet dagegen einen Bonitätsgrad für das betrachtete Unternehmen. Dem Bonitätsgrad, der in einem Intervall zwischen null und eins liegt, wird über eine Umrechnungstabelle ein Entscheidungsbetrag zugeordnet.

Um die Entscheidung der sieben Experten zu einer Entscheidung zu verdichten, wurde folgende Konvention getroffen. Die Experten lehnen einen Antrag ab, falls die Mehrheit der Experten den Kreditantrag ablehnt. In einem solchen Fall ist die Entscheidung beider Systeme richtig, falls diese ebenfalls eine Ablehnung vorschlagen. Falls die Mehrheit der Experten eine Versicherungssumme gewährt, gilt der Kreditantrag als positiv entschieden. In diesem Fall wird eine Entscheidung der automatischen Systeme als richtig akzeptiert, falls das vom System vorgeschlagene Kreditlimit zwischen der minimalen und maximalen Kreditentscheidung der Experten liegt. Die untere Tabelle zeigt die Ergebnisse der 46 untersuchten Testfälle.

Gruppe der 7 Experten	Fuzzy System	Konventionelles System
32 Risiken wurden akzeptiert	Von diesen 32 Risiken wurden... 31 Risiken akzeptiert (96,9%) 1 Risiko abgelehnt (3,1%)	Von diesen 32 Risiken wurden... 23 Risiken akzeptiert (71,9%) 8 Risiken abgelehnt (25%) 1 Risiko konnte nicht entschieden werden (3,1%)
14 Risiken wurden abgelehnt	Von diesen 14 Risiken wurden... 12 Risiken abgelehnt (85,7%) 2 Risiken akzeptiert; dabei lag die vorgeschlagene Versicherungssumme zwischen Minimum und Maximum der von den Experten vorgeschlagenen Werten (14,3%)	Von diesen 14 Risiken wurden... 13 Risiken abgelehnt (92,9%) 1 Risiko konnte nicht entschieden werden (7,1%)

Das Fuzzy System ist in guter Übereinstimmung mit den Entscheidungen der Kreditprüfer. Selbst in den wenigen Fällen, in denen die Entscheidungen des Fuzzy Systems nicht mit Entscheidungen der Experten übereinstimmen, sind diese Abweichungen jedoch für die Experten akzeptabel und stellen keine krassen Fehlentscheidungen des Systems dar.

Die unteren beiden Diagramme verdeutlichen die Arbeitsweise des Fuzzy Systems. In der Abbildung 6 wurden alle Bonitätsmerkmale bis auf die Branche fixiert und für zwei Firmen mit unterschiedlichen Ausprägungen die Variation der Kreditwürdigkeit in Abhängigkeit von der Branchenbewertung aufgetragen. In der Abbildung 7 wurde die Branche fixiert und dafür das Alter eines Unternehmens variiert.

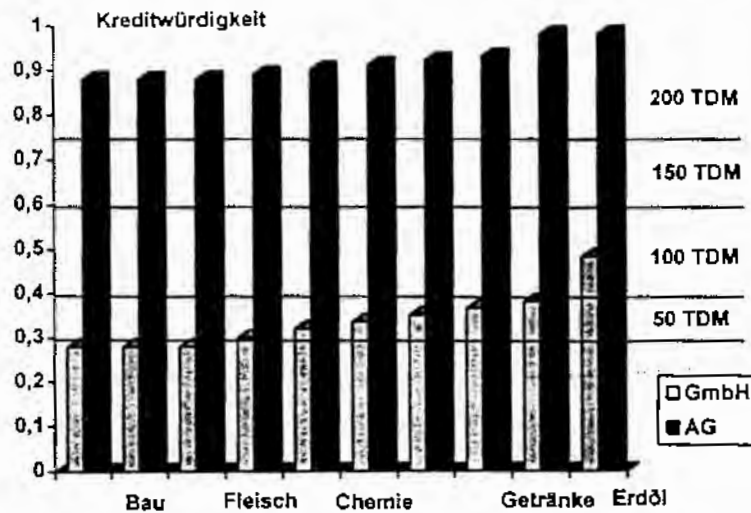


Abbildung 6: Kreditwürdigkeit vs. Branche

GmbH	AG
18 Monate	30 Monate
50 TDM	500 TDM
Kein Grundvermögen	Grundvermögen
Skontoausnutzung	Skontoausnutzung
Alte Bundesländer	Alte Bundesländer

In der zweiten Abbildung ist die Kreditwürdigkeit gegen das Unternehmensalter aufgetragen. Die Kreditwürdigkeit ist für eine GmbH mit den angegebenen Ausprägungen berechnet worden. Sie steigt mit zunehmenden Alter an. Ab dem 21. Monat wird auf einen Antrag für

eine GmbH mit diesen Ausprägungen eine Versicherungssumme von 50 TDM gewährt. Für jede Firma existiert praktisch ein individuelles Grenzalter, ab dem zum erstenmal eine Versicherungssumme gewährt wird.

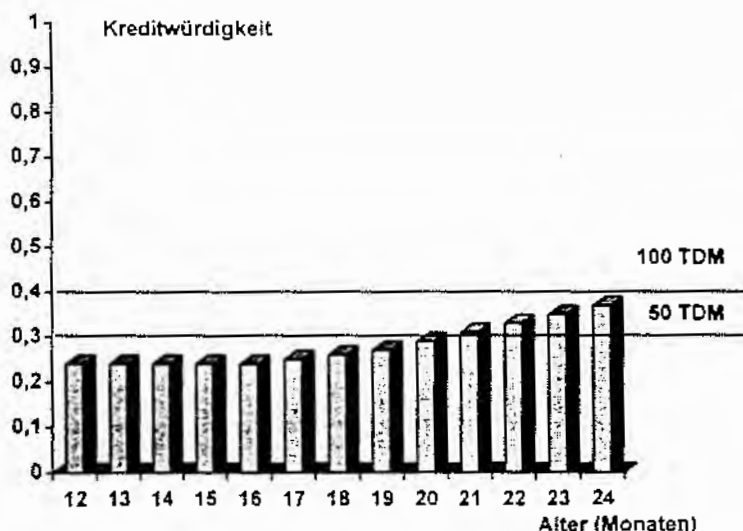


Abbildung 7: Kreditwürdigkeit vs. Alter

GmbH
50 TDM
Kein Grundvermögen
Bauhauptgewerbe
innerhalb vereinbarter Ziele
Alte Bundesländer

4. Zusammenfassung und Ausblick

Mit dem Fuzzy System wurden die gesteckten Ziele, qualitativ bessere Kreditentscheidungen durch Aufweichen starrer Grenzen und Berücksichtigung von Kompensationseffekten, erreicht. Bei der Entwicklung des Systems hat sich das

Verfahren der Fuzzy Logic als ein sehr gutes Instrument für die Abbildung des Entscheidungsverhalten der Kreditprüfer erwiesen. Die Verbesserungen mußten allerdings mit einem bedeutend höheren Entwicklungsaufwand, als bei Systemen mit sequentiellen Regeln, erkauft werden.

Der Praxiseinsatz dieses Systems und eine Ausweitung auf den Bereich Auslandskreditversicherung ist geplant.

Literaturangaben

Allgemeine Kreditversicherung AG (1995): Jahresabschlußbericht

Beuter, H. B. Reiss, I., Rust H. J. (1994): Erfahrungen mit formalisierten Verfahren bei der Kreditwürdigkeitsprüfung. In: Schiemenz, B. (Hrsg.): Interaktion - Modellierung, Kommunikation und Lenkung in komplexen Organisationen. Duncker & Humblot, Berlin, 55 - 73

MIT (1996): DataEngine 2.0. Handbuch, Aachen

Romaniuk, S.G., Hall, L. O. (1992): Decision Making on Creditworthiness, using a Fuzzy Connectionist Model. Fuzzy Sets and Systems 48, 15 - 22

Rommelfanger, H. J. (1993): Fuzzy Logic-Based Processing of Expert Rules Used for Checking the Creditability of Small Business Firms. In: Klement, E.P., Slany W. (Eds.): Fuzzy Logic in Artificial Intelligence. Springer-Verlag, 103 - 113

Zimmermann, H. - J. (Hrsg.) (1995): Datenanalyse - Anwendung von DataEngine mit Fuzzy Technologien und Neuronalen Netzen. VDI-Verlag Düsseldorf

Zimmermann, H. - J. (1996): Fuzzy Set Theory - And Ist Applications. 3rd rev. ed., Kluwer, Boston

Zimmermann, H. - J., Zysno, P. (1983): Decisions and Evaluations by Hierarchical Aggregation of Information. Fuzzy Sets and Systems 10, 243 - 260

Zygan, H.-J. (1996): Erstellung eines regelbasierten Fuzzy-Entscheidungssystem zur Bonitätsbeurteilung von Unternehmen bei einer Kreditversicherung. Diplomarbeit, Aachen.

„Was die Gesellschaft vorwärt bringt ist die Wissenschaft, und die Sprache der Wissenschaft ist die Mathematik, und die Mathematik basiert auf der Logik, und das Urgestein der Logik ist Aristoteles, und hier beginnt Fuzzy“.

B. Kosko

SOFT COMPUTING IN LABVIEW

FUZZY-CONTROL UND GENETISCHEN ALGORITHMEN IN LABVIEW

EDMOND HAJRIZI
TU-Wien, Institut für Fertigungstechnik,
Abt. Intelligent manufacturing systems,
Karlsplatz 13/311, 1040 Wien

Inhaltsverzeichnis

1. Einführung
2. Grafische Programmentwicklung in LabVIEW
3. Fuzzy-Logik und Fuzzy-Control in LabVIEW
 - 3.1. Fuzzy-Toolkit für LabVIEW
4. Genetische Algorithmen in LabVIEW
5. Anwendungsbeispiele
 - 5.1. Mustererkennung
 - 5.2. Optimierung von Fuzzy-Controllern mittels genetischen Algorithmen
6. Zusammenfassung
7. Conclusion
8. Literatur

1. Einführung

Gegenstand der vorliegenden Arbeit ist die Implementierung von Fuzzy-Systemen und genetischen Algorithmen unter Verwendung der interaktiven Programmierungsumgebung LabVIEW.

2. Grafische Programmentwicklung in LabVIEW

Programme innerhalb der interaktiven grafischen Programmierungsumgebung LabVIEW [1,2,4] werden als *virtuelle Instrumente* (kurz VIs) bezeichnet. Sie können als Software-ICs angesehen und entsprechend als Funktionsblöcke eingesetzt werden. Ihre Strukturierung erfolgt in zwei Abstraktions-Ebenen: Dem *Front Panel*, das die interaktive grafische Benutzeroberfläche des Programms realisiert und dem sogenannten (*Block-*) *Diagram*, in dem die zur algorithmischen Lösung der Problemstellung benötigten Funktionsblöcke über entsprechende Datenflüsse miteinander verbunden werden (Bild 1).

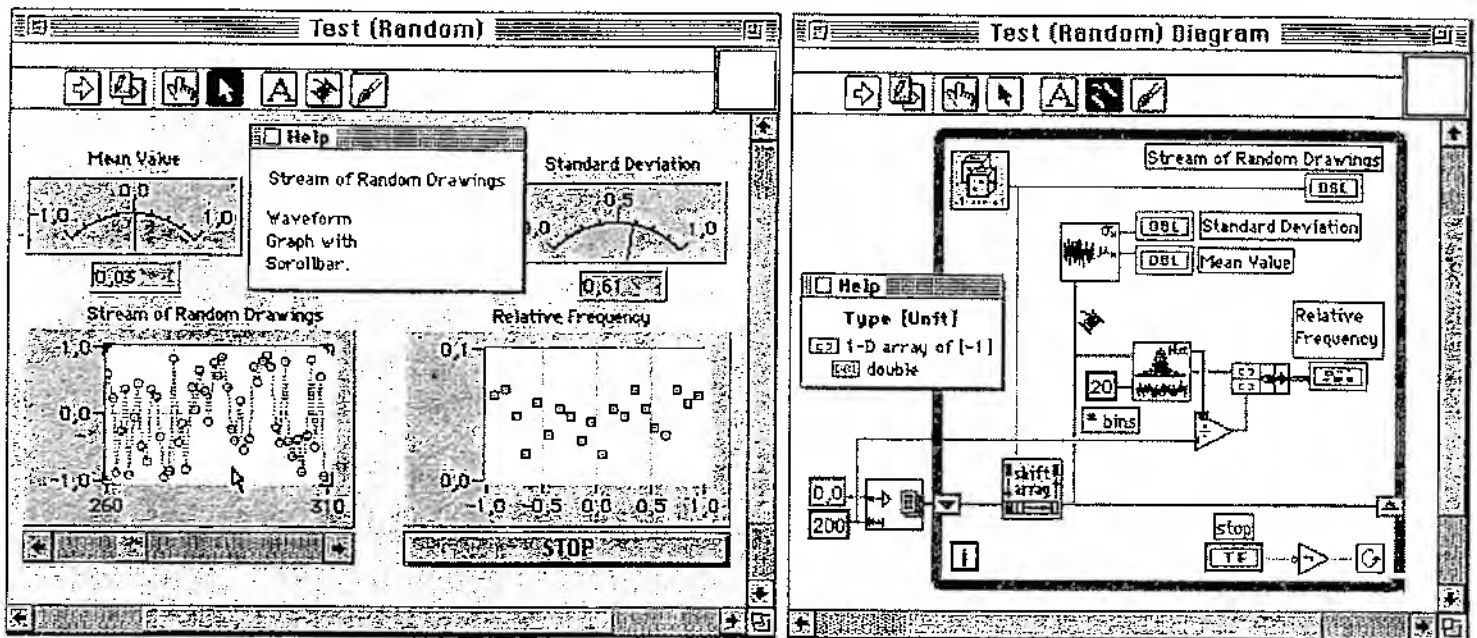


Bild 1. Frontpanel und (Block-) Diagram

Im Gegensatz zu konventionellen textbasierten Programmierungstechniken erfolgt die Programmierung in LabVIEW auf der (Block-) Diagramm-Ebene auf grafischen Wege in Form eines Datenflußdiagramms (Blockschaltbild), in dem die zur Lösung der Aufgabe verwendeten Funktionen (einfache mathematische Operationen, komplexe Funktionen wie FFT, digitale Filter usw. oder auch Benutzer-definierte Sub-VIs) über Daten- bzw. Signalflüsse miteinander verbunden, d.h. wie ICs „verdrahtet“ werden. Die einzelnen Daten- bzw. Signalflüsse können beliebig strukturiert, d.h. aus einfachen Datenstrukturen wie reelle, ganzzahlige und Boole'sche Variable, Zeichenketten, aber auch aus komplexen Strukturen wie ein- und mehrdimensionale Arrays, Tabellen, Graphen sowie Cluster aus diesen Datentypen aufgebaut sein (Bild 1).

Im Sinne der Modularisierung von Programmen kann ein Programm als sogenanntes Sub-VI wie ein IC im (Block-) Diagramm eines anderen Programms verwendet werden. Hierzu wird dem Sub-VI ein benutzerdefiniertes Programmsymbol (*Icon*) zugewiesen, das auch die auf dem Frontpanel angeordneten Ein- und Ausgabeelementen als Anschlüsse (*Terminals*) des Sub-VIs bereitstellt (Bild 2). Zur Erfüllung von Echtzeitanforderungen können VIs auch als *reentrant* spezifiziert werden.

Zur Realisierung von Programm- bzw. Ablaufstrukturen stehen die aus der strukturierten Programmierung bekannten Konstrukte wie Sequenz, Case, WHILE-Loop, FOR-Loop usw. zur Verfügung (Bild 2). Die Programmentwicklung in LabVIEW konzentriert sich somit auf die Realisierung des (Block-) Diagramms, das die Problemstruktur repräsentiert. Das generierte (kompilierte) Programm ist auf unterschiedlichen Rechnerplattformen problemlos wiederverwendbar.

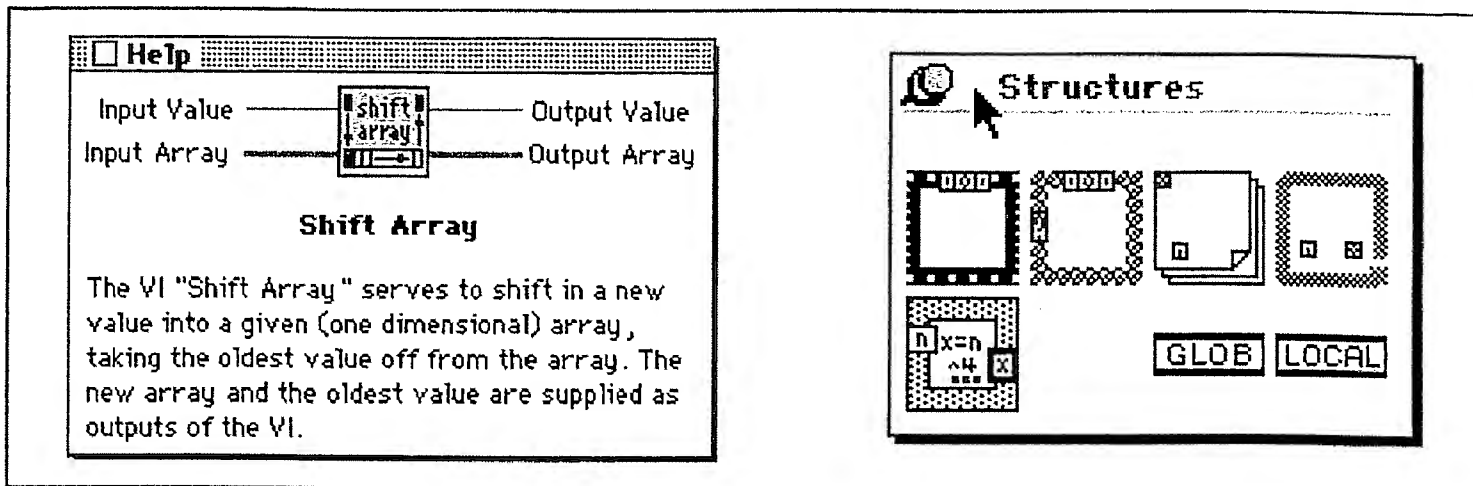


Bild 2: Programmsymbol (Icon) mit E/A-Terminals sowie Strukturelemente

LabVIEW verfügt über eine Vielzahl von vordefinierten Funktionen, die in verschiedenen VI-Bibliotheken bereitstehen und zur Lösung von Problemstellungen aus allen Bereichen technisch-wissenschaftlicher Anwendungen wie z.B. Bildverarbeitung, Datenbankmanagement, Prozessautomatisierung sowie Netzwerk und Feldbusanwendungen unmittelbar eingesetzt werden können. Der Software-Entwicklungsprozeß wird durch leistungsfähige Dokumentations- und Debug-Werkzeuge, eine ausgefeilte Online-Hilfe, eine große Anzahl von lauffähigen Anwendungsbeispielen sowie durch spezielle Module zur Laufzeitoptimierung und zur Handhabung umfangreicher Anwendungssysteme unterstützt.

In Verbindung mit multifunktionalen analogen und digitalen Ein-/Ausgabekarten (IEEE-488, VXI/VME können Test-, Meßwerterfassung- und Verarbeitungsaufgaben wie auch Automatisierungsanwendungen zur Online-Regelung von Prozessen realisiert werden.

3. Fuzzy-Logik und Fuzzy-Control in LabVIEW

Eines der zentralen Konzepte in der Fuzzy-Logik [4,5,6] ist die mathematische Beschreibung von sprachlich (linguistisch) unscharfen Begriffen und Sachverhalten mittels sogenannter Fuzzy-Sets, die üblicherweise durch normierte durch Zugehörigkeitsfunktionen mit beliebigen reelle Funktionswerten gemäß $0 \leq \mu(x) \leq 1$ modelliert werden. Elementarer Bestandteil eines Fuzzy-Systems ist dabei die sogenannte *linguistische Variable*, die unterschiedliche subjektive Ausprägungen eines unscharfen umgangssprachlichen Sachverhaltes in Form von *linguistischen Termen* zusammenfaßt.

Eine weiteres Konzept in der Fuzzy-Logik befaßt sich mit dem regelbasierten Schließen, auch Inferenz genannt. In der Praxis liegt oftmals ingenieurmäßiges Wissen zum optimalen „Fahren“ einer Anlage oder eines Prozesses vor, ohne die physikalisch-technischen Zusammenhänge in jedem Detail mathematisch korrekt und vollständig beschreiben zu können. Fuzzy-Logik gestattet nun die Implementierung von umgangssprachlich abgefaßten Regelstrategien in Form von *Wenn... –Dann... –Regeln*, die das zur Regelung einer Anlage notwendige Engineering-Wissen repräsentieren.

Hinsichtlich der regelungstechnischen Anwendung von Fuzzy-Logik (man spricht dann von Fuzzy-Control) ergibt sich die in Bild 3 dargestellte Grundstruktur.

Eine linguistische Variable läßt sich in LabVIEW recht einfach als Array von linguistischen Termen realisieren. Beschränkt man sich bei der Modellierung der linguistischen Terme auf nominierte trapez- und dreiecksförmige Zugehörigkeitsfunktionen (Fuzzy-Sets), so läßt sich jeder Term, wie in Bild 3 gezeigt, als Cluster bestehend aus einem String (der Name des Terms) und einem Array aus vier reellen Werten (die Argumentwerte, an denen die Zugehörigkeitsfunktion die Grenzwerte 0 und 1 annimmt) realisieren.

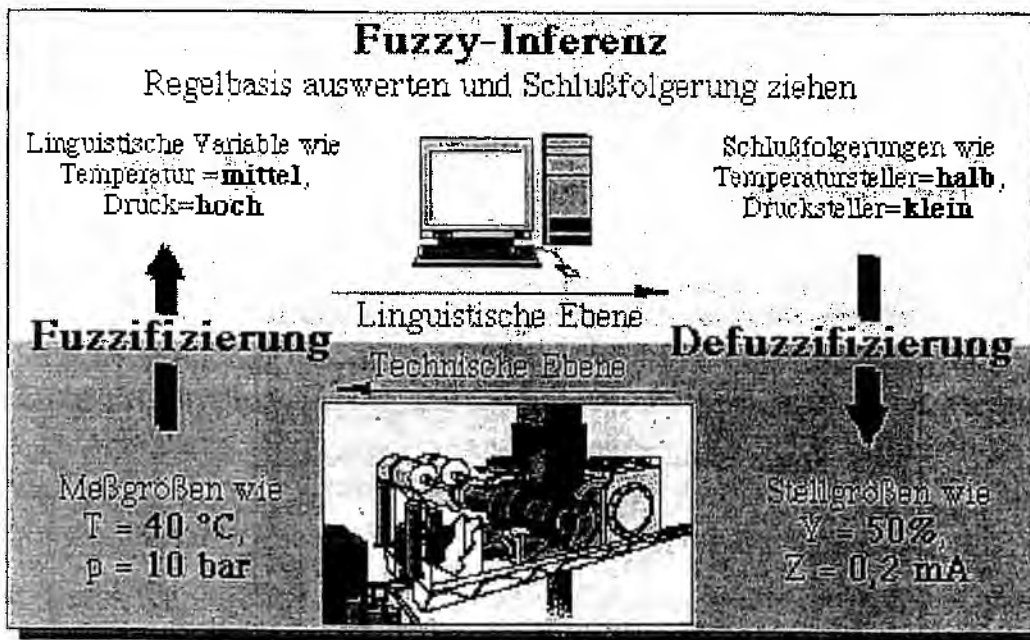


Bild 3: Grundstruktur von Fuzzy-Control

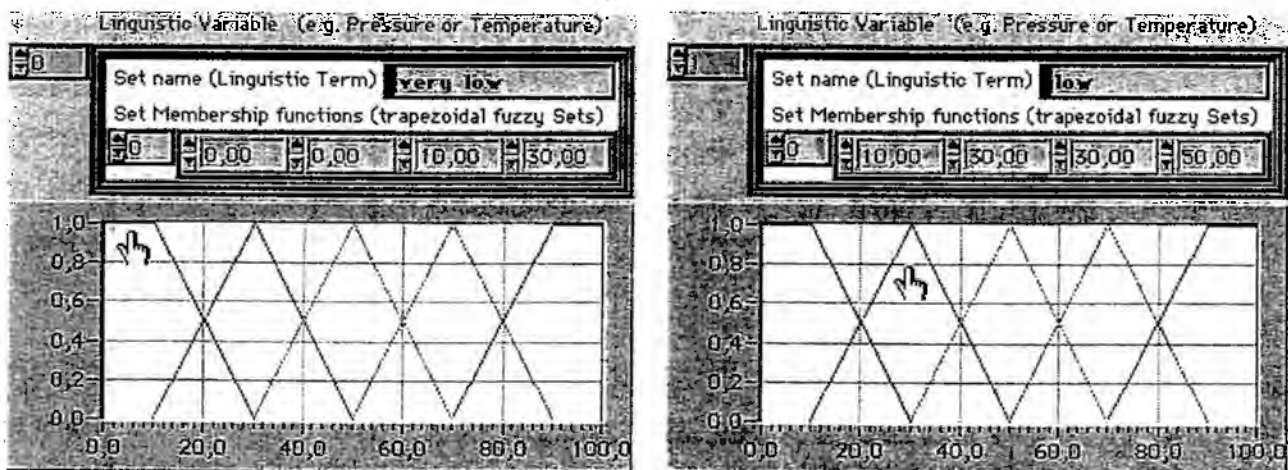


Bild 3.a: Realisierung einer linguistischen Variablen und ihrer Terme

Eine Regelbasis läßt sich ebenfalls relativ einfach als Cluster aus einem oder mehreren Prämissen-Arrays (sich enthalten die Prämissen-Terme) und einem Konklusions-Arrays (es enthält die Konklusions-Terme) realisieren, wobei der für alle Arrays (gemeinsame Index die Nummer der einzelnen Regel beschreibt).

Im Falle eines Fuzzy-Controllers für zwei Eingänge und eine Ausgang läßt sich die Regelbasis, wie in Bild 3.a gezeigt, recht übersichtlich in Matrixform (ähnlich einer Zustands-Ereignis-Matrix) realisieren. Die Zeilen- und Spaltenüberschriften enthalten die jeweiligen linguistischen Terme der beiden Eingangsvariablen. Die als Menüring realisierten Matricelemente gestatten dann, jeweils den für die durch Zeilen- und Spalten-Koinzidenz gegebene Termkombination (UND-Verknüpfung der beiden Prämissen-Terme) gewünschten Konklusionsterm auszuwählen.

Die in Bild 4 dargestellte Situation zeigt die Einstellung des Konklusionsterm $\gg stop \ll$ für die Regel $\gg \text{WENN Positionsfehler=ZR UND Padelwinkel=ZR DANN Ausgang=stop} \ll$ (ZR bedeutet Zero).

Input Variable:	Pendelwinkel		
Input Var:	NG	ZR	PG
Positionsfehler			
NG	forward	back	back
NS	forward	back half	back
ZR	forward	✓ stop	back
PS	forward half	forward half	back half
PG	forward	forward	back

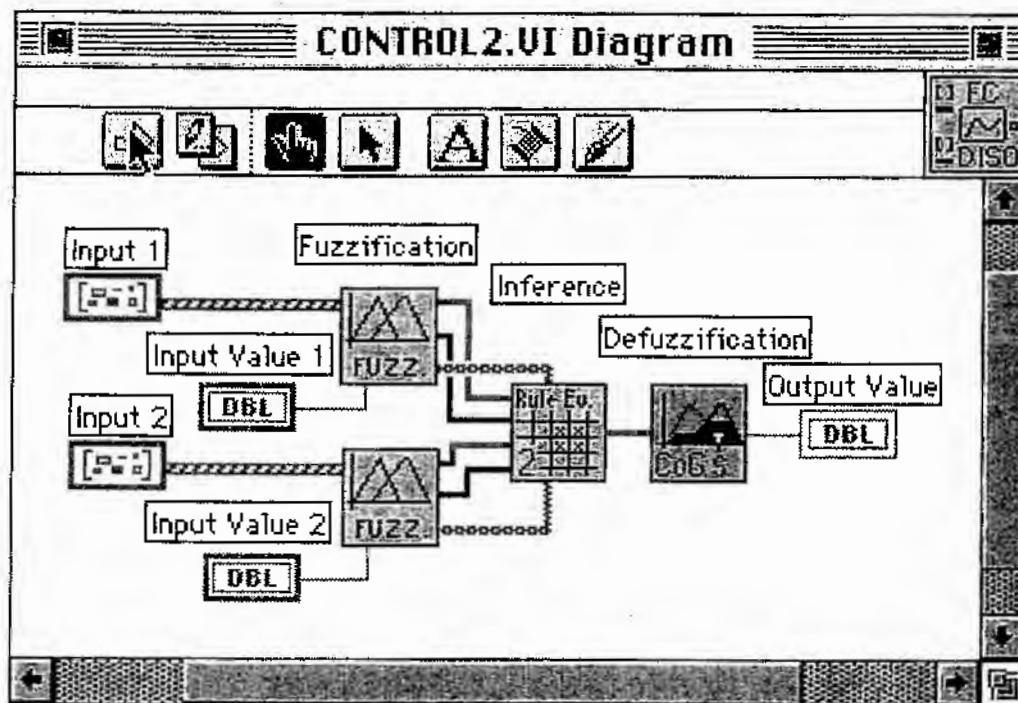


Bild 4: Regelbasis und Aufbau eines Fuzzy-Controllers für zwei Eingänge

Der Aufbau des in Bild 4 gezeigten Fuzzy-Controllers CONTROL2 entspricht dem im vorigen Abschnitt skizzierten dreistufigen Inferenzschema. Als Werte für die beiden Eingangsgrößen (*Input Value 1* und *Input Value 2* in Bild 4) können unmittelbar die über eine Ein-/Ausgabekarte erfaßten und A/D-gewandelten Meßgrößen eines realen Prozesses (hier z.B. Positionsabweichung und Auslenkung der Last eines Portalkrabs) dienen. Innerhalb einer Regelschleife aufgerufen, ermittelt so der Fuzzy-Controller den momentanen Prozeßzustand und liefert die zur Regelung benötigte Stellgröße (*Output Value* in Bild 4), ihrerseits wieder über eine entsprechende Ein-/Ausgabekarte an den Prozeß abgegeben werden kann.

3.1. Fuzzy-Toolkit für LabVIEW

Für die Implementierung von Fuzzy-Systemen steht in LabVIEW ein komplettes Entwicklungs-Werkzeug (Fuzzy-Logic-Toolkit) zur Verfügung. Das Werkzeug besteht aus den nachfolgenden Komponenten:

- Projekt-Editor (Project Manager) zur Bearbeitung von Fuzzy-Projekten
- Im Projekt-Editor stehen neben der Online-Hilfe Kommandos zum Laden und Speichern von Fuzzy-Projekten sowie zum Ausdruck der Projekt-Dokumentation zur Verfügung. Der Projekt-Editor verfügt ferner über ein Eingabefeld, in das die Projektbeschreibung eingegeben werden kann. Alle weiteren Funktionen des Werkzeugs werden von Projekt-Editor aus aufgerufen. In Bild 5 ist der Aufruf des Fuzzy-Set-Editors für ein geladenes Projekt gezeigt.

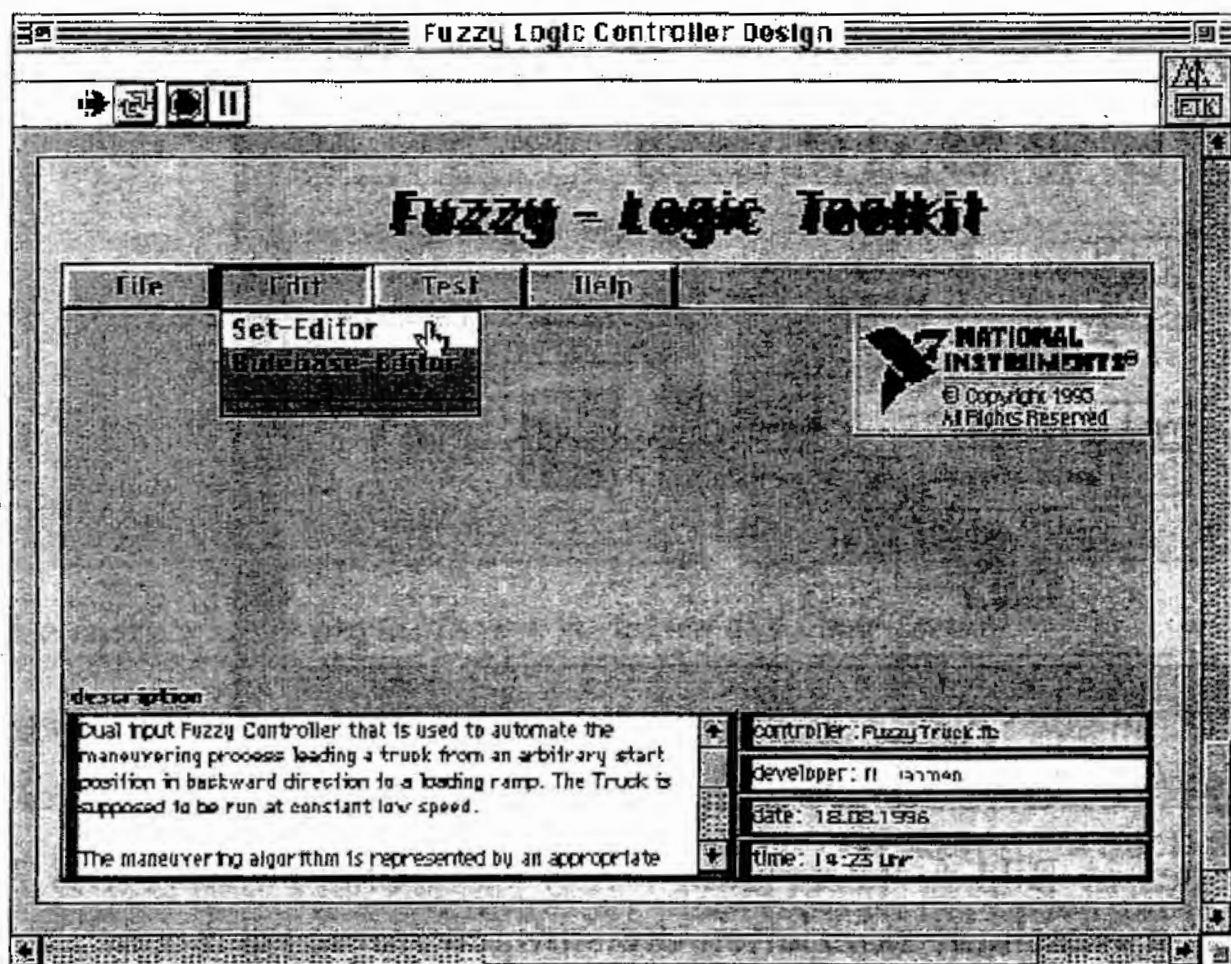


Bild 5: Projekt-Editor-Panel

- Fuzzy-Set-Editor (Fuzzy Set Editor) zur Definition und Bearbeitung der linguistischen Variablen und Terme.

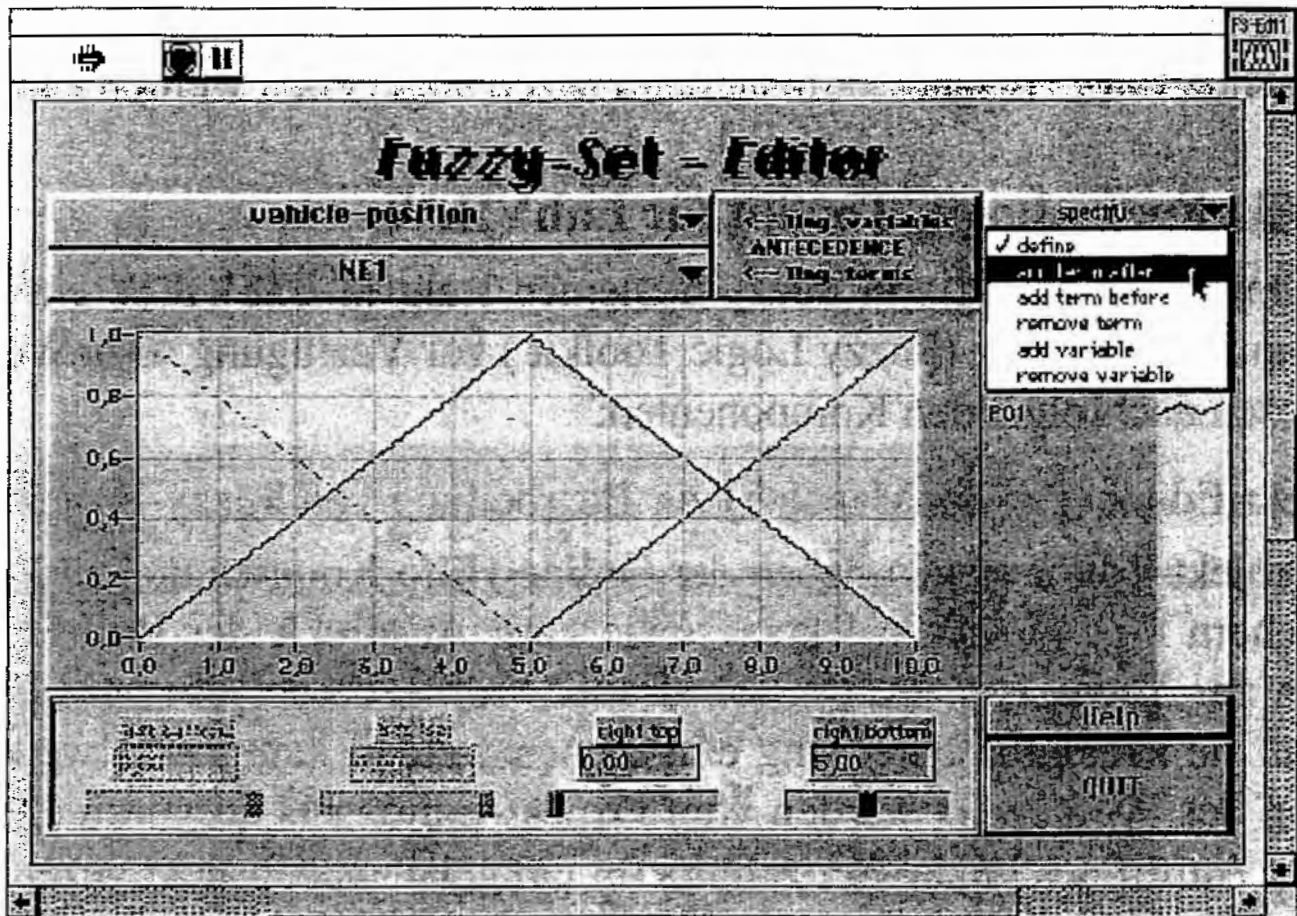


Bild 6: Fuzzy-Set-Editor-Panel (mit dem Kommando >>add term after<<)

Der Fuzzy-Set-Editor stellt neben der grafischen Anzeige der linguistischen Terme für die gerade ausgewählte linguistische variable eine Reihe von Funktionen zur Bearbeitung der Terme sowie des gesamten Term-Arragements zur Verfügung. Es können Variable und Terme hinzugefügt, entfernt und umbenannt werden. Die Gültigkeitsbereiche der Variablen können ebenso verändert werden wie die Formen der einzelnen Terme. Auch der Fuzzy-Set-Editor verfügt über eine Online-Hilfe. Sämtliche Aktionen werden einer Plausibilitätsprüfung unterzogen, so daß unplausible Term-Arragements vermieden werden. Bild 6 zeigt das Fuzzy-Set-Editor Panel mit dem ausgewählten Kommando >>add Term after<< aus dem Funktions-Menü >>define>>.

- Regelbasis-Editor (Rule Base Editor) zur Definition und Bearbeitung der Regelbasis.
Der Regelbasis-Editor stellt Funktionen zur Definition und Veränderung der einzelnen linguistischen Regeln bereit. Neben der Festlegung des Defuzzifizierungsverfahrens kann festgelegt werden, welcher Ausgangswert vom Fuzzy-Controller bereitgestellt werden soll, wenn im Falle einer unvollständigen Regelbasis keine der eingegeben Regeln zutrifft. Eine dem Regelbasis-Editor zugeordnete Online-Hilfe gibt Auskunft über alle verfügbaren Funktionen des Editors. Bild 7 zeigt das Panel des Regelbasis-Editors für den Fall eines Fuzzy-Controllers mit zwei Eingangsgrößen bei der Auswahl des Defuzzifizierungsverfahrens.

- Testfunktionen (Test Facility) zur Analyse des Übertragungsverhaltens.
Es steht eine Reihe von Testfunktionen zur Analyse des Übertragungsverhaltens zur Verfügung. So kann z.B. die Abhängigkeit des Übertragungsverhaltens von einer einzelnen Eingangsgröße bei festen Werten für die anderen Eingangsgrößen ermittelt und angezeigt werden. In Bild 8 ist das Panel für die einfache Controller-Charakteristik gezeigt. Mit einem Cursor kann die Kennlinie abgefahren werden, wobei sowohl die aktuelle Eingangssituation als auch nie abgefahren werden, wobei sowohl die aktuelle Eingangssituation als auch der zugehörige Ausgangswert wie auch die in dieser Situation gültigen Regeln angezeigt werden. Auch die Testfunktionen sind mit einer Online-Hilfe versehen.
- Online-Hilfe (online Help) mit Erläuterungen zu den einzelnen Funktionen des Werkzeugs.
- Dokumentationshilfen (Print Facilities)
- Vordefinierte Beispiele zur unmittelbaren Ausführung.

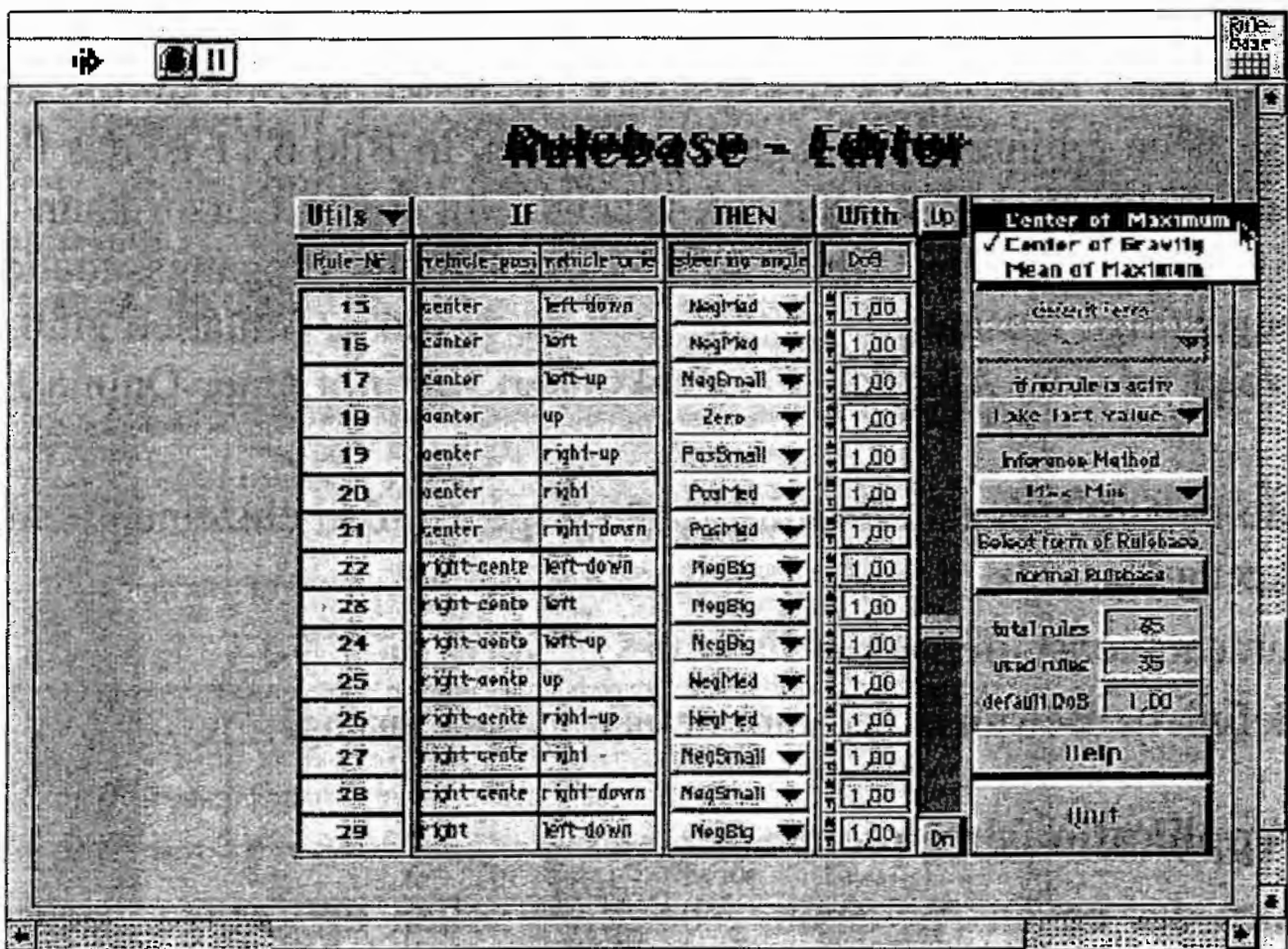


Bild 7: Projekt-Editor-Panel (Aufruf des Fuzzy-Set-Editors)

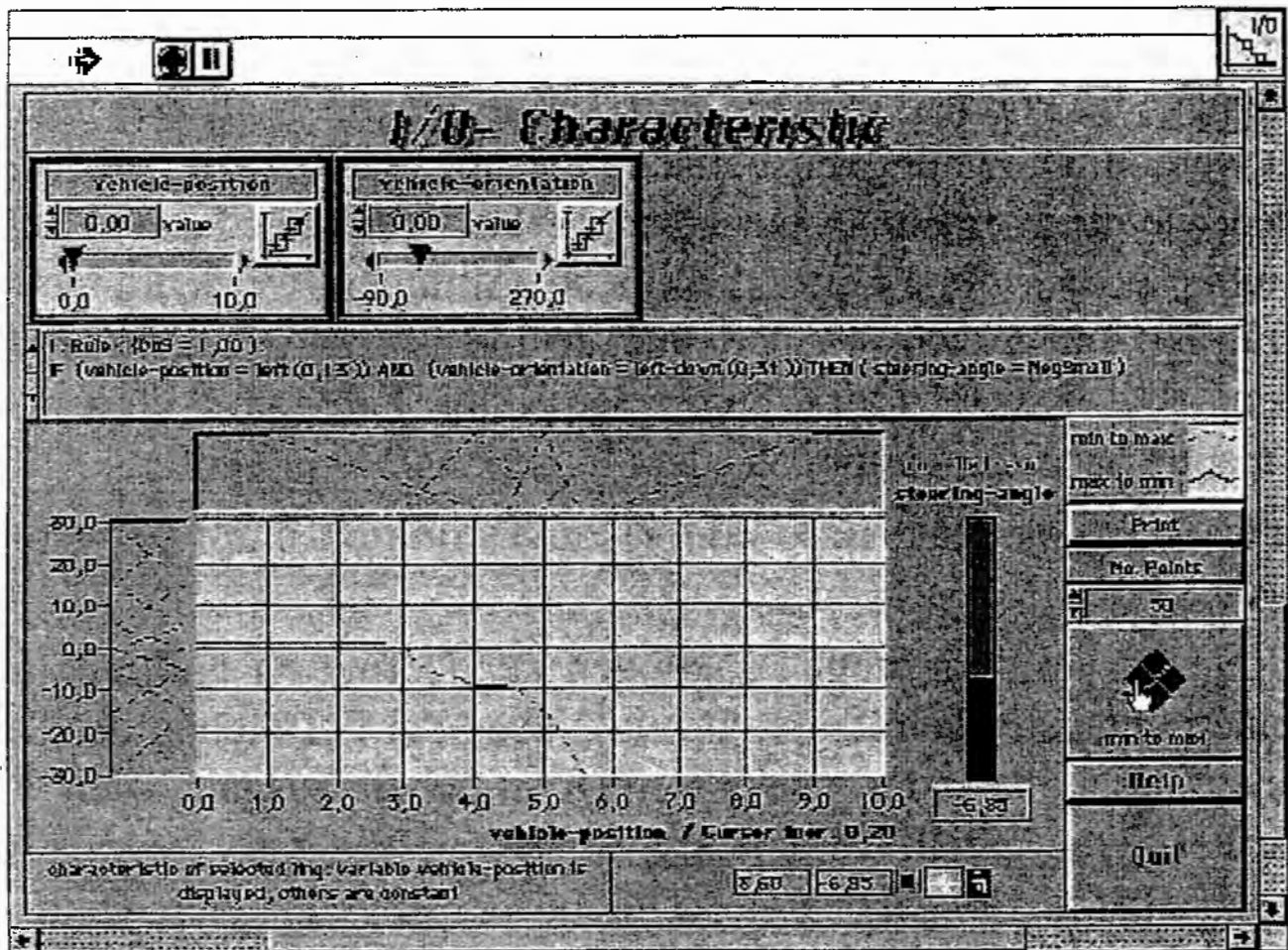


Bild 8: Funktion zur Ermittlung des Übertragungsverhalten

4. Genetischen Algorithmen in LabVIEW

Der universellen Einsatzbreite von LabVIEW entsprechend liegt es nahe, derartige Optimierungskonzepte in LabVIEW bereitzustellen. Das Bild 9 zeigt das Panel einer LabVIEW implementierten Test-Applikation mit der für genetischen Algorithmus gewählten Parametrierung und dem nach Ablauf von 15 Generationen vorliegenden Ergebnis. Die gegebene zweiartige multimodale Gütefunktion (Bild 10) ist auf dem Panel als Intensity Plot gezeigt, wobei die Maxima (Berge) weiß und Minima (hier die optimalen Täler) schwarz dargestellt sind. Anzumerken ist, daß der implementierte genetische Algorithmus auch dann noch zu dem oben gezeigten Ergebnis führt, wenn man beispielsweise die gesamte Startpopulation in eines der Minima plazierte.

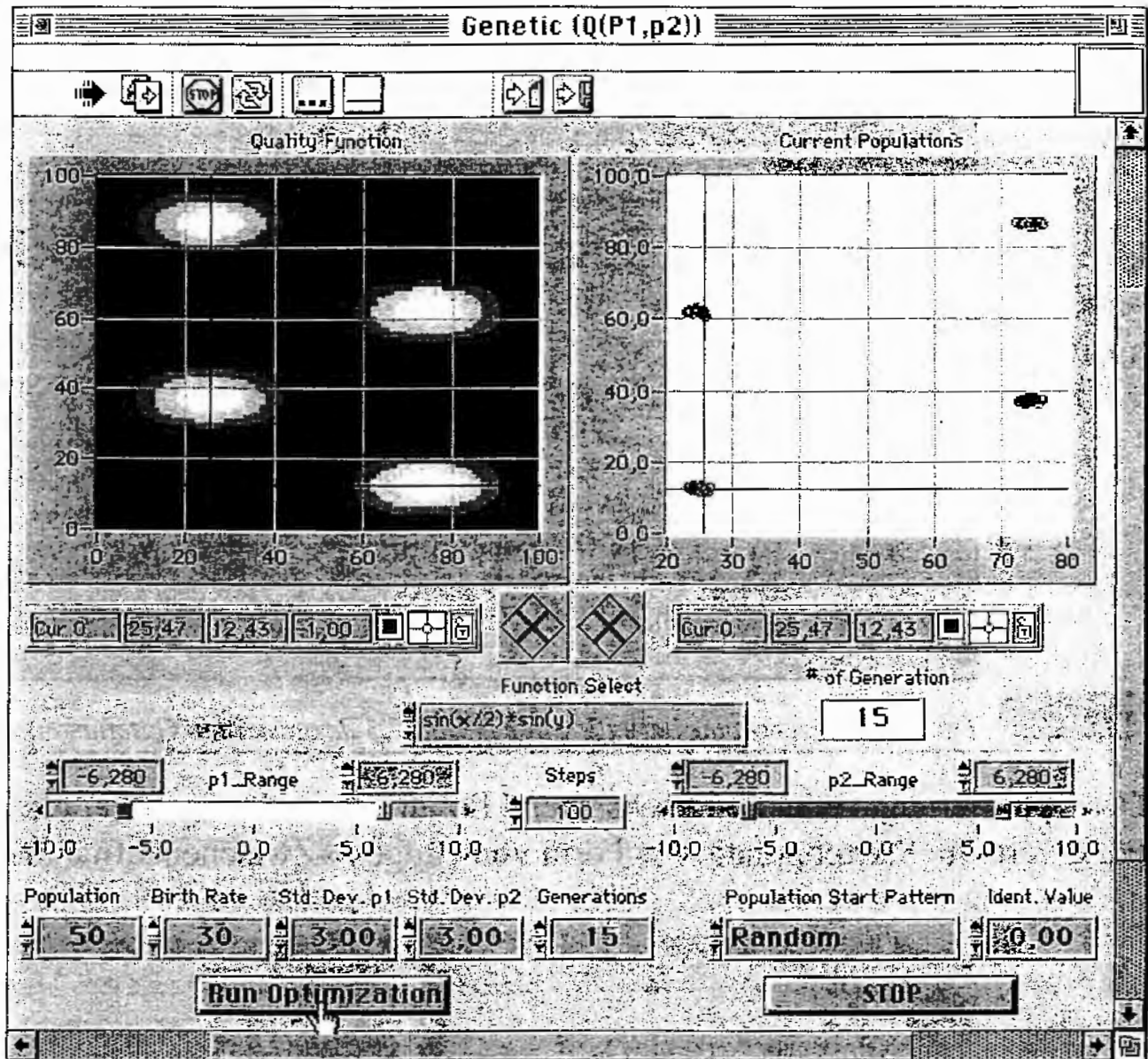


Bild 9: Optimierungsergebnisse bezüglich der in Bild 9 dargestellten Gütenfunktion

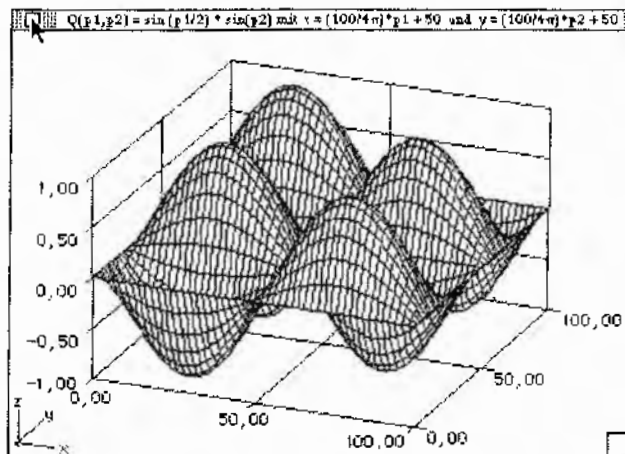


Bild 10: Multimodale Gütefunktion mit vier gleichwertigen Optima im gegebenen Parameterraum

Bild 11 zeigt wiederum das Panel der in LabVIEW implementierten Test-Applikation mit der für den genetischen Algorithmus gewählten Parametrierung und dem nach Ablauf von 15 Generationen vorliegenden Ergebnis bezüglich der in Bild 9 dargestellten Gütenfunktion.

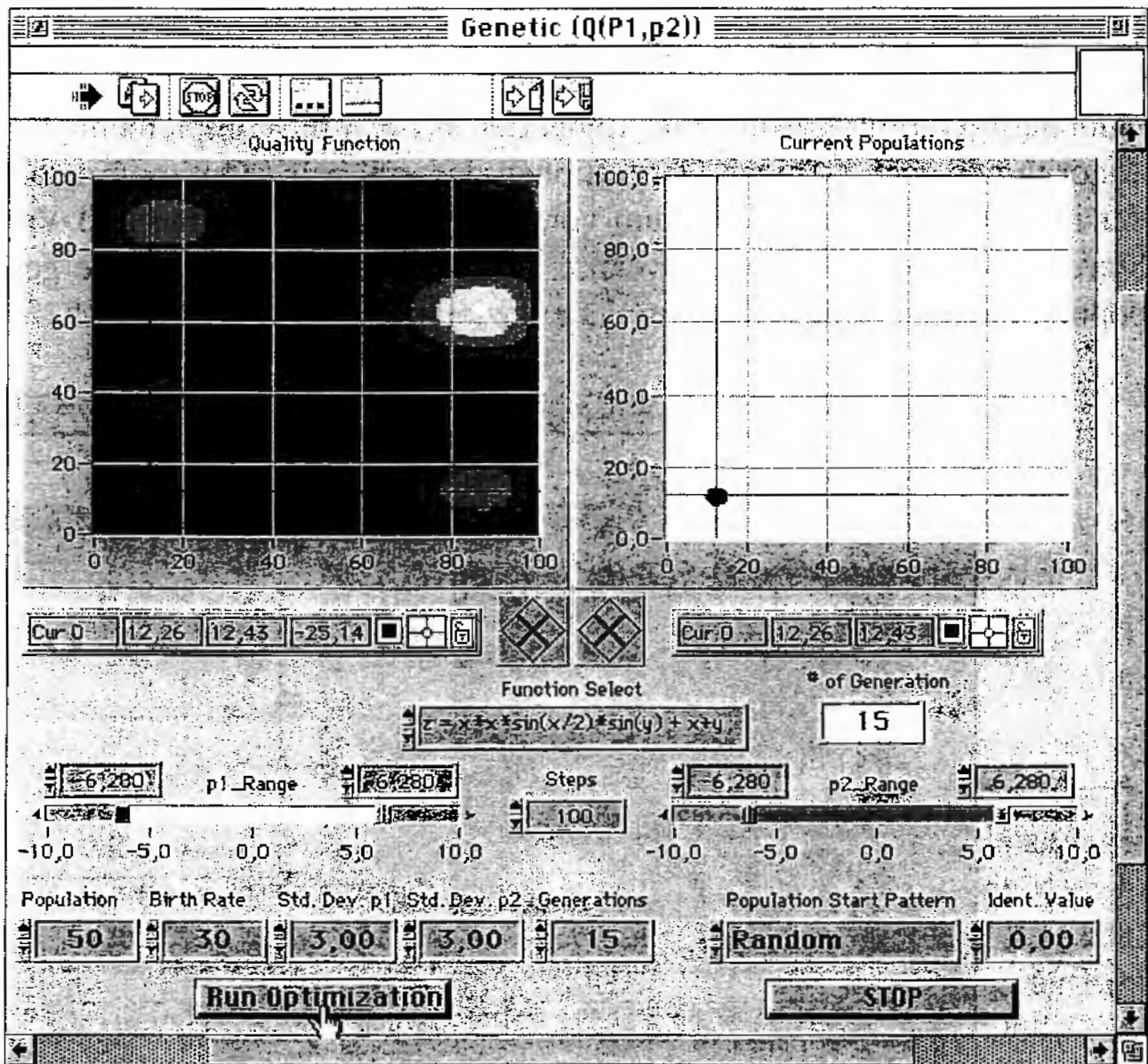


Bild 11: Optimierungsergebnisse bezüglich der in Bild 9 dargestellten Gütenfunktion

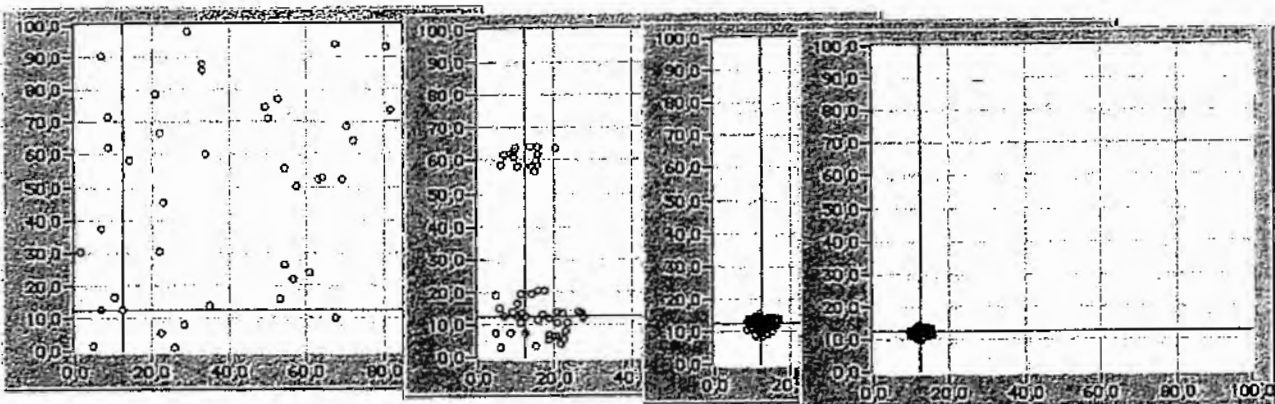


Bild 12: Optimierungsverlauf bezüglich der in Bild 9 dargestellten Gütenfunktion

Der Optimierungsverlauf ist nachstehend für eine im zulässigen Parameterraum zufällig verteilte Startpopulation in Form von einigen >>Zwischenaufnahmen<< gezeigt.

5. Anwendungsbeispiele

5.1. Anwendungsbeispiel Mustererkennung

Im Rahmen dieses Anwendungsbeispiels sollen flache dreieckige, sechseckige und rechteckig viereckige Kleinteile erkannt und sortiert werden, wobei Größe und Proportion der Kleinteile in einem relativ weiten Bereich variieren dürfen. Die auf einem Transportband aufgelegten Kleinteile werden hierzu durch ein recht einfache, lediglich aus einer Reihe von LDR-Widerständen aufgebaute Reflexionslichtschranke geführt (Bild 13). Der für die unterschiedlichen Formteile charakteristische jedoch stark variierende Spannungsverlauf (Schattenwurf) an der Reflexionslichtschranke wird von einem Fuzzy-Controller erfaßt und zur Erkennung ausgewertet. Signalaufnahme, Merkmalsextraktion und Erkennung erfolgen On-Line mit dem Teile-transport.

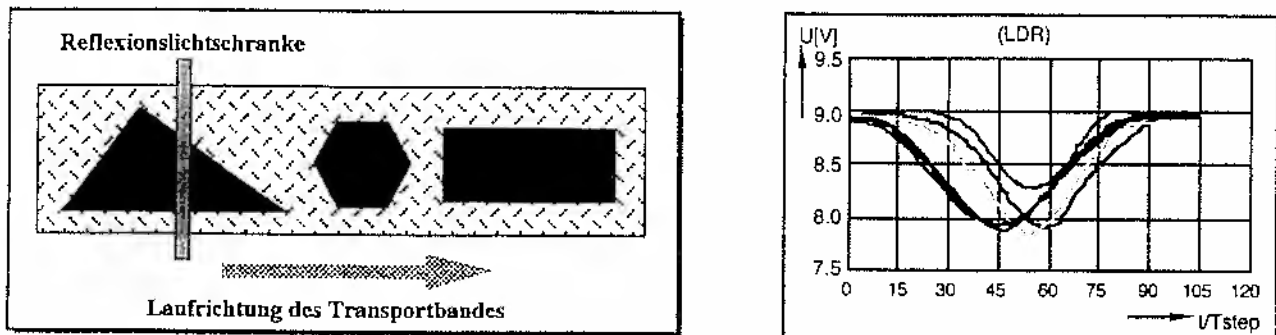
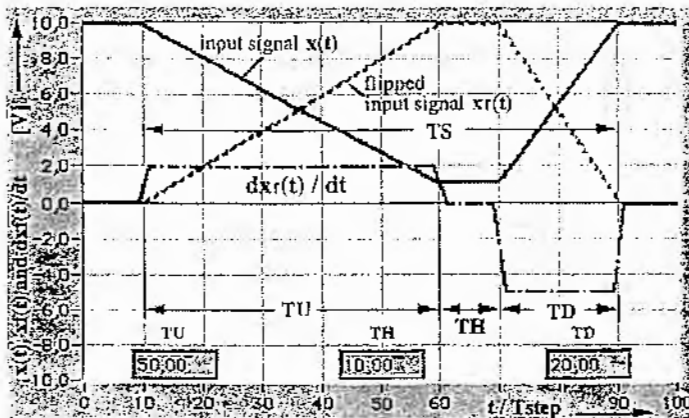


Bild 13: Sortierstecken-Modell und reale Signalverläufe bei einem linksschiefen Dreieck

Trotz der großen Varianz der Signalverläufe (erlebte Variation der Bauteilproportionen sowie störende Fremdlichtabhängigkeit) lassen sich für alle Bauteilsorten drei charakteristische Merkmale aus den Signalverläufen extrahieren: die Zeit bis zur maximalen Abdeckung, TU genannt, die Zeit maximaler Abdeckung, TH genannt und die Zeit bis zur Freigabe, TD genannt. Diese Zeiten lassen sich relativ leicht über eine entsprechende Signalvorverarbeitung ermitteln. Dies ist in Bild 8 für einen idealisierten Signalverlauf dargestellt. Bezieht man die genannten Zeitabschnitte auf die Gesamtzeit TS, so ergeben sich die in Bild 14 angegebenen Größen für eine von den Bauteilproportionen weitgehend unabhängige Merkmalsextraktion.

Aus den für die unterschiedlichen Kleinteile gemessenen typischen Signalverläufe lassen sich die in Bild 15 gezeigten linguistischen Terme für die beiden linguistischen Variablen (Eingangsgrößen) „TH/TS“ und „(TU-TD)TS“ ableiten. Als Ausgangsgröße dient die linguistische Variable „Object“. Die einzelnen Objektbezeichnungen (Terme) sind, wie in Bild 16 gezeigt, als sogenannte Singletons modelliert. Der zur Mustererkennung eingesetzte Fuzzy-Controller arbeitet mit der ebenfalls in Bild 16 dargestellten recht einfachen Regelbasis. Als Defuzzifizierungsverfahren wird MoM (Mean of Maximum) eingesetzt. Dieses Verfahren dient zur Ermittlung der plausibelsten Lösung.



$T_H / T_S \approx 0 \Rightarrow$ Dreieck ,
 $0 < T_H / T_S < 1 \Rightarrow$ Sechseck ,
 $T_H / T_S \approx 1 \Rightarrow$ Rechteck ,
 $(T_U - T_D) / T_S > 0 \Rightarrow$ linksschief ,
 $(T_U - T_D) / T_S \approx 0 \Rightarrow$ symmetrisch ,
 $(T_U - T_D) / T_S < 0 \Rightarrow$ rechtsschief ,

Bild 14: Signal-Vorverarbeitung (idealisierte Signale) und Größen zur Merkmalsextraktion

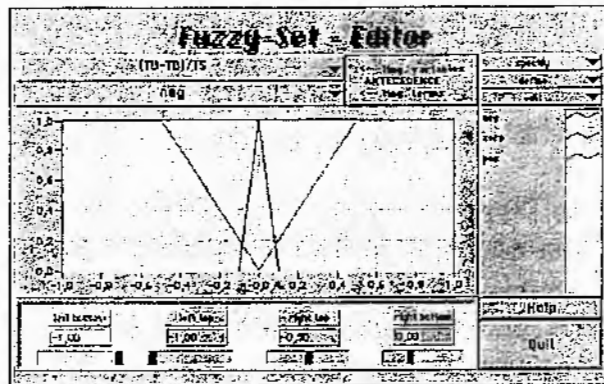
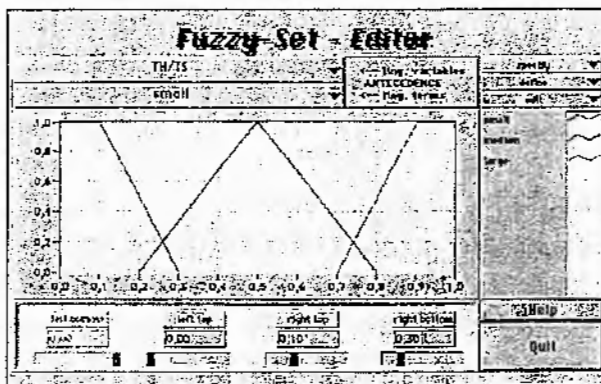


Bild 15: Term-Arrangement für die linguistische Variable Object und Regelbasis

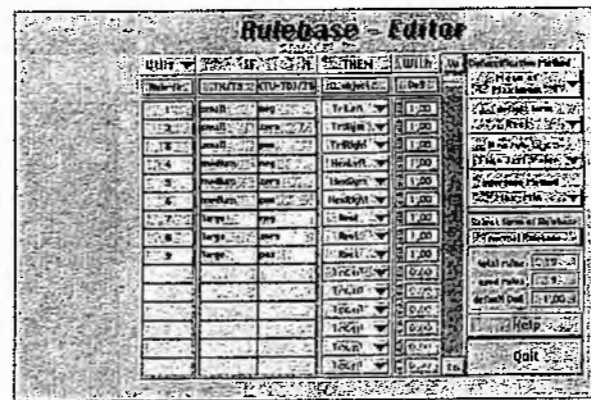
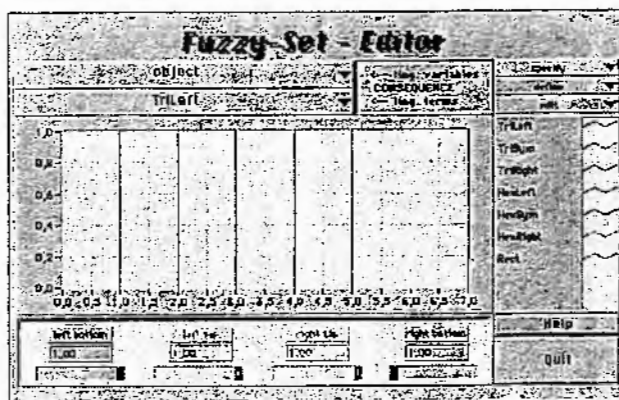


Bild 16: Term-Arrangement für die linguistische Variable Object und Regelbasis

Zum Experimentieren mit unterschiedlich parametrisierten Fuzzy-Controllern wird in der Übung neben der realen Meßanordnung auch ein Simulations-VI eingesetzt, das die von der Lichtschrankenordnung stammenden Signale simuliert (idealisierte Signale) und beliebig zu verändern gestattet. In Bild 17 sind sowohl Frontpanel und Diagramm dieses Vis gezeigt.

In beiden Fällen wurde lediglich die Regelbasis optimiert, wobei die linguistischen Terme beider Eingangsvariablen symmetrisch bei vollständiger Überlappung im jeweiligen Datenbereich verteilt waren. Als Eingangsgrößen des eingesetzten Fuzzy-PI-Controllers

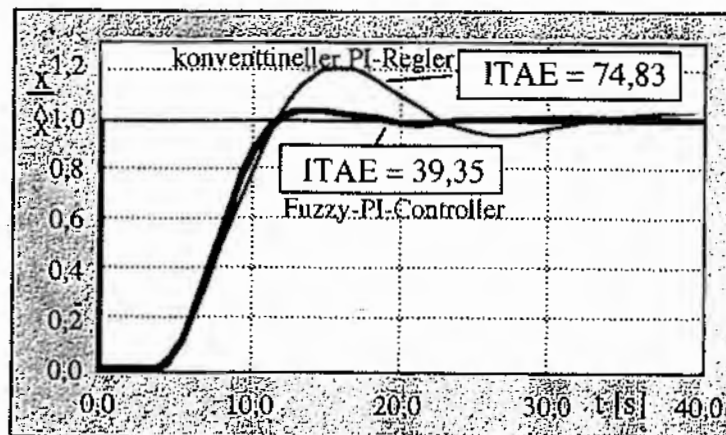
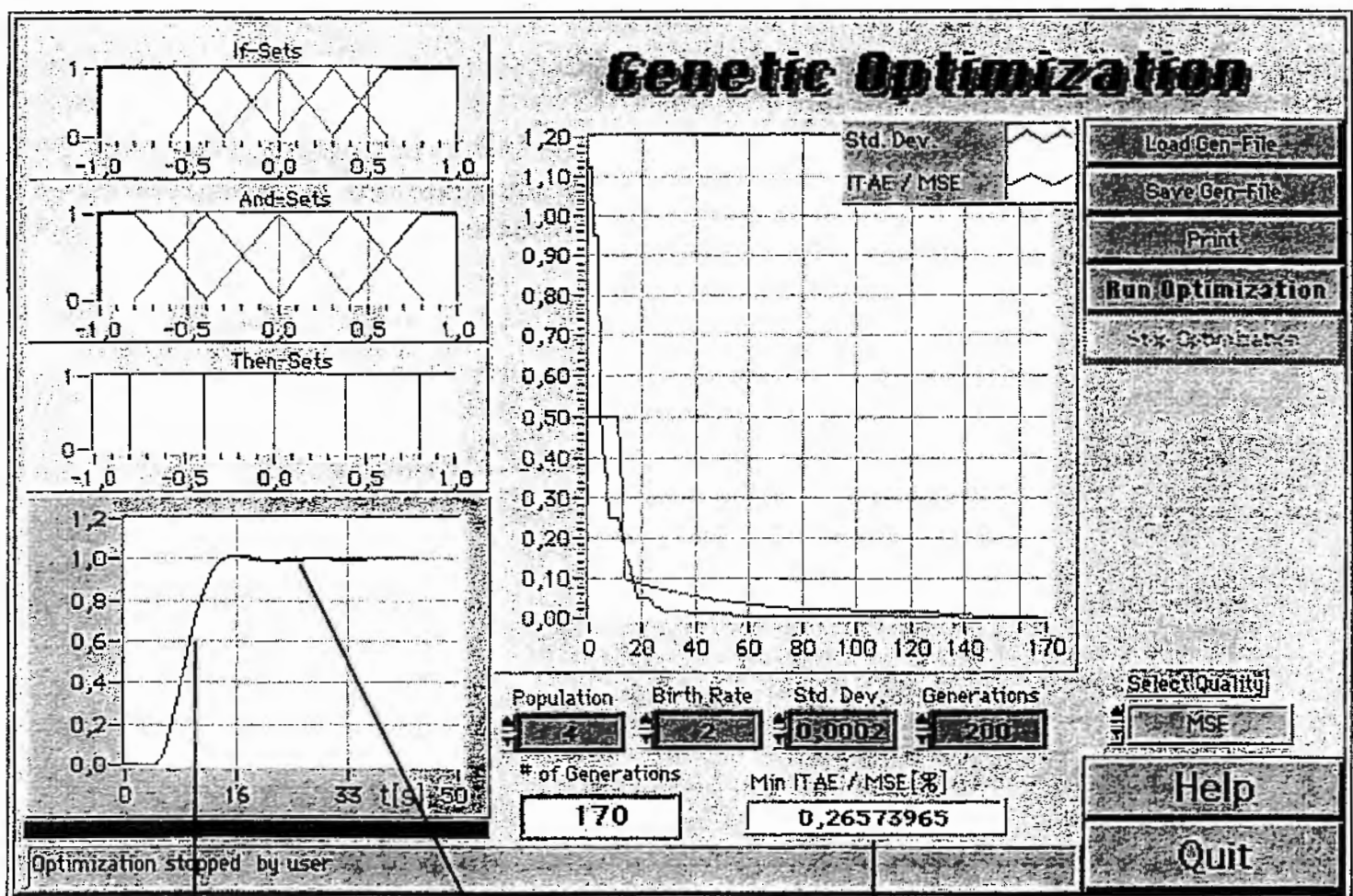


Bild 19: Optimierte Fuzzy-PI-Controller im Vergleich zu einem optimierten konventionellen PI-Regler



user-defined step response error [%]

step response obtained from the optimization process

Bild 20: Optimierter Fuzzy-PI-Controller zur Einstellung eines vorgegebenen Führungsverhaltens

Die Ergebnisse zeigen, daß Evolutionstrategien (oder genetische Algorithmen) zum „Trainieren“ eines Fuzzy-Controllers zur Einstellung eines vorgegebenen Führungsverhaltens des Gesamtsystems erfolgreich eingesetzt werden können.

6. Zusammenfassung

Die Nutzung von Fuzzy-Methoden in LabVIEW bedeutet für den Anwender, daß er von der Datenerfassung über die Vorverarbeitung und den Einsatz von Fuzzy-Methoden bis hin zur Ergebnispräsentationen in einem einzigen Entwicklungssystem arbeitet. Damit erhöhen sich die Produktivität und Effizienz für den Anwender, da die bekannte LabVIEW-Programmierungsmethodik bestehen bleibt und dennoch die Möglichkeiten der Fuzzy-Technologie genutzt werden können. Hinzu kommt die Tatsache, daß das Fuzzy-Toolkit komplett in G geschrieben und somit vollständig plattformunabhängig ist.

Online-Optimierung

Evolutionsstrategien (oder genetische Algorithmen) sind, wie die obigen Ausführungen zeigen, vielfältig einsetzbar. Insbesondere bei Fuzzy-systemen mit ihrer Vielzahl von zueinander in Wechselbeziehung stehenden Parametern sind sie konventionellen Verfahren weit überlegen. Der universellen Einsatzbreite von LabVIEW entsprechend, liegt es nahe, derartige Optimierungskonzepte in LabVIEW bereitzustellen. Aufgrund der grafischen Programmierbarkeit von Applikationen, unter direkter Einbeziehung analoger und digitaler Prozeßgrößen über entsprechende Datenerfassungskarten, ergeben sich sogar Möglichkeiten zur Online-Optimierung von Prozessen.

7. Conclusion

With the Fuzzy Logic for G Toolkit, you can design a fuzzy logic controller (or expert system for decision making) and implement the controller in your G applications. Fuzzy membership functions and the controller rule base are defined with the Fuzzy Logic Controller Design VI. The Controller Design VI is a standalone VI with a graphical user interface for completely defining all controller/expert system components. All parameters of the defined controller are saved into a controller data save.

Two additional VIs are used to implement the Fuzzy controller in your G application. The Load Fuzzy Controller VI is used to load all parameters of the fuzzy controller saved in a data file by the Controller Design VI. This data is then wired to the Fuzzy Controller VI, which implements the fuzzy logic inference engine. Process parameters (controller inputs) are wired to the inputs of the Fuzzy Controller VI, and controller outputs are output by the VI. By wiring data acquired by your data acquisition hardware to the fuzzy controller, you can implement real-time decision making or control of your physical system. Additionally, outputs of the fuzzy controller can be used by your data acquisition (DAQ) analog output hardware to implement real-time process control.

Offline Optimization

Within this design step the prototype controller is tested and simulated with either real process data previously recorded from the process or simulation data obtained from a mathematical process model. Transfer characteristics analysis and time response analysis can be performed to observe the system behavior and optimize the controller. LabVIEW support both types of analysis. In this step, NeuroFuzzy techniques, as well as Genetic or Evolutionary Algorithms, can also be used for system optimization.

8. Literatur

1. **Jamal, R. & Pichlik, H.:** *LabVIEW-Programmiersprache der vierten Generation*, Prentice Hall Verlag, 1997
2. **Johnson, G.W.:** *LabVIEW Graphical Programming*, McGraw-Hill, McGraw-Hill Series on Visual Technology, 1994
3. **Jamal/Jaschinski (Hrg.):** *Virtuelle Instrumente in der Praxis*, Begleitband zum Kongreß VIP'97, München, 1997
4. **Kahlert, J. & Frannk, H.:** *Fuzzy-Logik und Fuzzy-Control*, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 1993
5. **Von Altrock, C.:** *Mustererkennung mit Fuzzy-Logik*, Franzis-Verlag GmbH, München, 1993
6. **Kinnebrock, W.:** *Optimierung mit genetischen und selektiven Algorithmen*, R. Oldenberg Verlag GmbH, München, 1994.

SOFT COMPUTING IN LABVIEW

FUZZY -CONTROL UND GENETISCHEN ALGORITHMEN IN LABVIEW

Edmond HAJRIZI

TU-Wien, Institut für Fertigungstechnik, Abt. Intelligent manufacturing systems,
Karlsplatz 13/311, 1040 Wien

Zusammenfassung

Die Nutzung von Fuzzy-Methoden in LabVIEW bedeutet für den Anwender, daß er von der Datenerfassung über die Vorverarbeitung und den Einsatz von Fuzzy-Methoden bis hin zur Ergebnispräsentationen in einem einzigen Entwicklungssystem arbeitet. Damit erhöhen sich die Produktivität und Effizienz für den Anwender, da die bekannte LabVIEW-Programmierungsmethodik bestehenbleibt und dennoch die Möglichkeiten der Fuzzy-Technologie genutzt werden können. Hinzu kommt die Tatsache, daß das Fuzzy-Toolkit komplett in G geschrieben und somit vollständig plattformunabhängig ist.

Online-Optimierung

Evolutionsstrategien (oder genetische Algorithmen) sind, wie die obigen Ausführungen zeigen, vielfältig einsetzbar. Insbesondere bei Fuzzy-systemen mit ihrer Vielzahl von zueinander in Wechselbeziehung stehenden Parametern sind sie konventionellen Verfahren weit überlegen. Der universellen Einsatzbreite von LabVIEW entsprechend, liegt es nahe, derartige Optimierungskonzepte in LabVIEW bereitzustellen. Aufgrund der grafischen Programmierbarkeit von Applikationen, unter direkter Einbeziehung analoger und digitaler Prozeßgrößen über entsprechende Datenerfassungskarten, ergeben sich sogar Möglichkeiten zur Online-Optimierung von Prozessen.

Conclusion

With the Fuzzy Logic for G Toolkit, you can design a fuzzy logic controller (or expert system for decision making) and implement the controller in your G applications. Fuzzy membership functions and the controller rule base are defined with the Fuzzy Logic Controller Design VI. The Controller Design VI is a standalone VI with a graphical user interface for completely defining all controller/expert system components. All parameters of the defined controller are saved into a controller data save.

Two additional VIs are used to implement the Fuzzy controller in your G application. The Load Fuzzy Controller VI is used to load all parameters of the fuzzy controller saved in a data file by the Controller Design VI. This data is then wired to the Fuzzy Controller VI, which implements the fuzzy logic inference engine. Process parameters (controller inputs) are wired to the inputs of the Fuzzy Controller VI, and controller outputs are output by the VI. By wiring data acquired by your data acquisition hardware to the fuzzy controller, you can implement real-time decision making or control of your physical system. Additionally, outputs of the fuzzy controller can be used by your data acquisition (DAQ) analog output hardware to implement real-time process control.

Offline Optimization

Within this design step the prototype controller is tested and simulated with either real process data previously recorded from the process or simulation data obtained from a mathematical process model. Transfer characteristics analysis and time response analysis can be performed to observe the system behavior and optimize the controller. LabVIEW support both types of analysis. In this step, NeuroFuzzy techniques, as well as Genetic or Evolutionary Algorithms, can also be used for system optimization.

Literatur

1. **Jamal, R. & Pichlik, H.:** *LabVIEW-Programmiersprache der vierten Generation*, Prentice Hall Verlag, 1997
2. **Johnson, G.W.:** *LabVIEW Graphical Programming*, McGraw-Hill, McGraw-Hill Series on Visual Technology, 1994
3. **Jamal/Jaschinski (Hrg.):** *Virtuelle Instrumente in der Praxis*, Begleitband zum Kongreß VIP'97, München, 1997
4. **Kahlert, J. & Frannk, H.:** *Fuzzy-Logik und Fuzzy-Control*, Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig/Wiesbaden, 1993
5. **Von Altrock, C.:** *Mustererkennung mit Fuzzy-Logik*, Franzis-Verlag GmbH, München, 1993
6. **Kinnebrock, W.:** *Optimierung mit genetischen und selektiven Algorithmen*, R. Oldenbourg Verlag GmbH, München, 1994.

Neural Net Toolbox

Anwendungen in der Bildanalyse

H. Bischof

Pattern Recognition and Image Processing Group
Vienna University of Technology, Austria

- Anforderungen – Warum Matlab
- Neural Network Design
 - Function Approximation
 - Satellite Data
 - View Interpolation
- Design of Vector Quantizers
- Robot-Vision

Why Matlab?

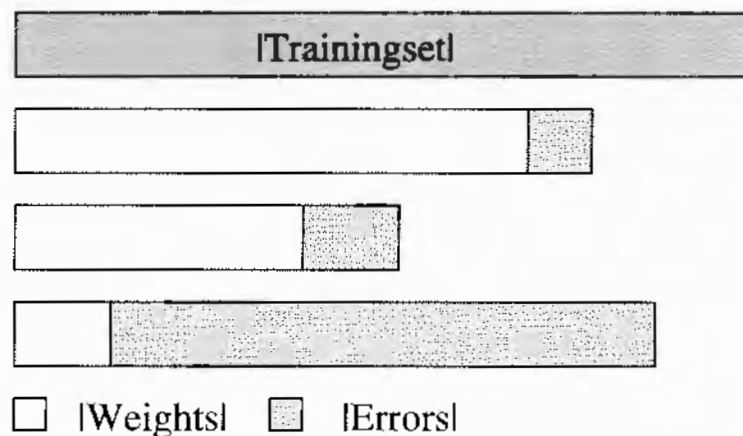
- Research:
 - New Algorithms
 - Improvement of Algorithms
 - Prototypes
- Combination of Image Processing & Neural Networks
 - Image Processing Functionality (e.g. Feature Extraction)
 - Classification (Neural Networks, Pattern Recognition)
- Toolboxes used:
 - Neural Net Toolbox
 - Image Processing Toolbox
 - Fuzzy Logic Toolbox
 - Netlab
<http://neural-server.aston.ac.uk/netlab/>
 - Fast ICA
<http://www.cis.hut.fi/projects/ica/fastica/>
 - SOM Toolbox
<http://www.cis.hut.fi/projects/somtoolbox/>

Neural Network Design with MDL

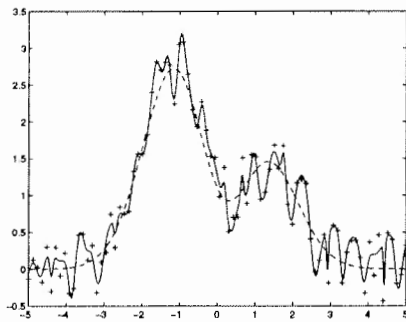
- Task of the network:

- Encode training set
- With minimal total encoding length

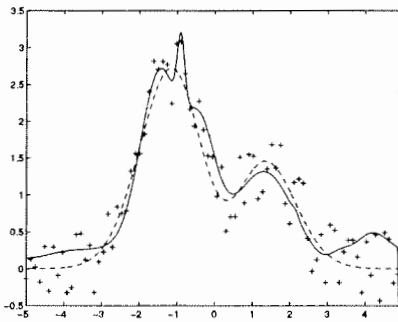
Minimum Description Length Principle (MDL)



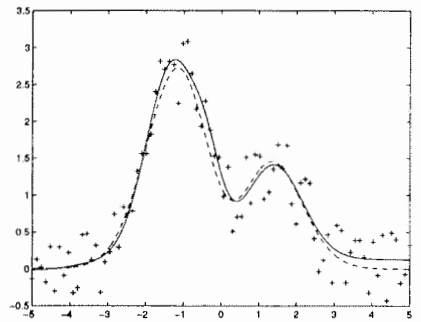
Function Approximation



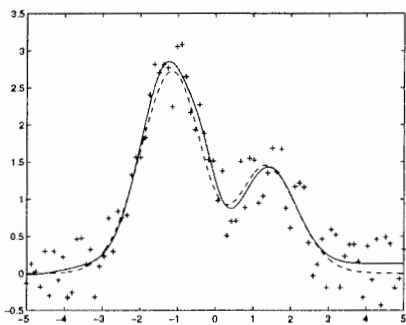
(a) Initialized network



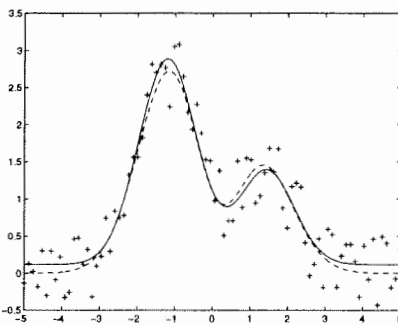
(b) 1st Selection



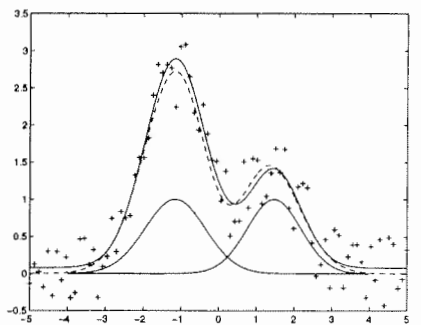
(c) 2nd Selection



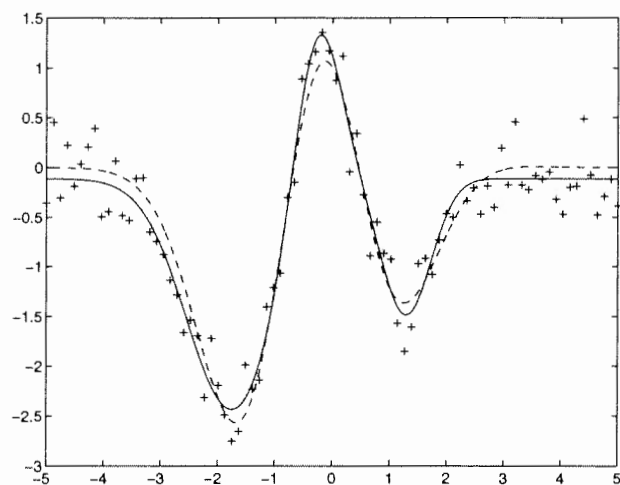
(d) 3rd Selection



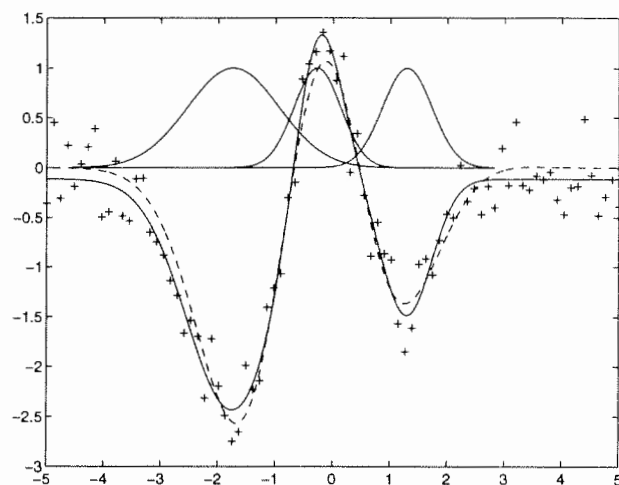
(e) 4th Selection



(f) Final result with superimposed basis functions



(g) Approximation Result



(h) Final result with superimposed basis functions

View Interpolation

- Stick figures 10 vertices under different viewing angles
- Output: Elevation and Azimuth

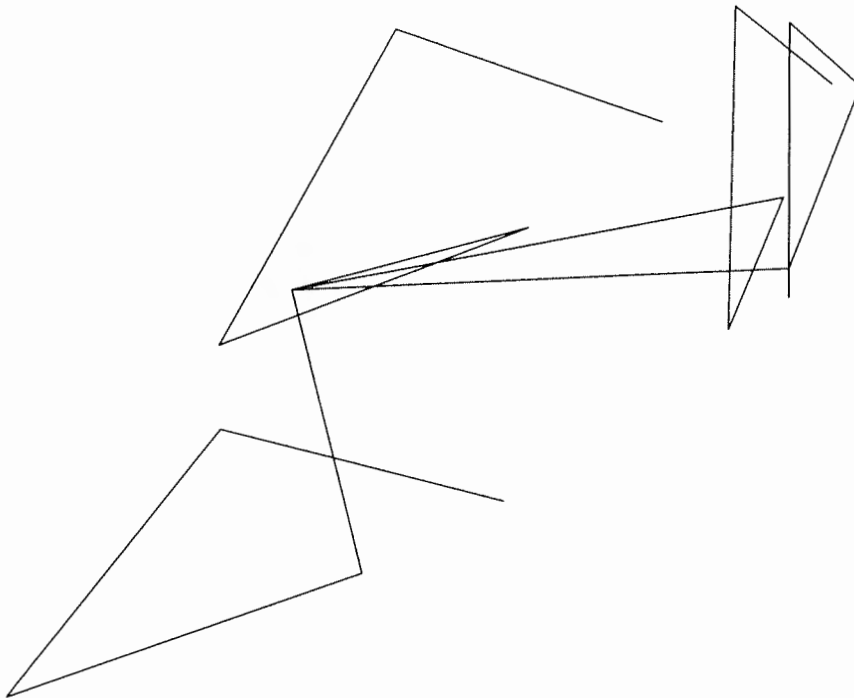
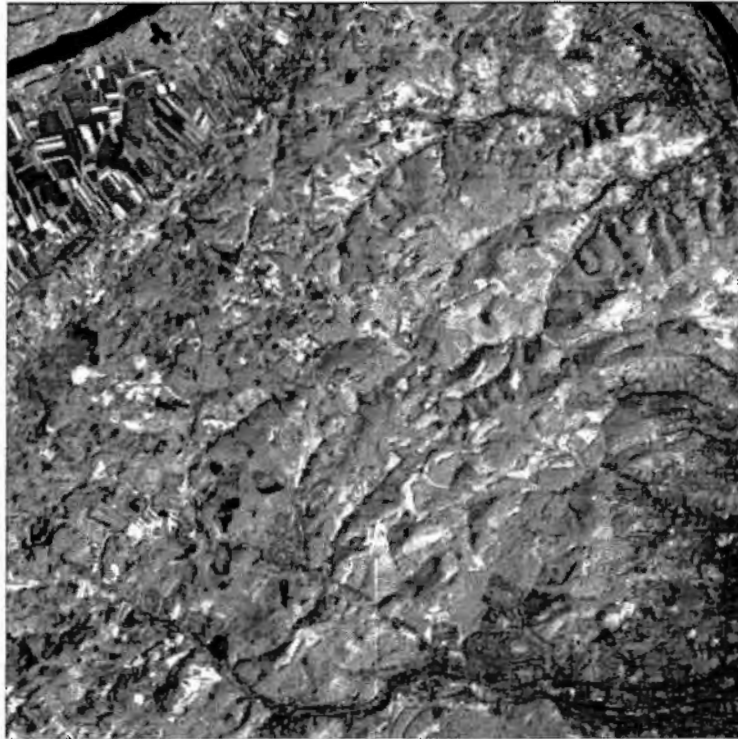


Figure 1: Four training views of Stick figure

- Training set: $[-45^\circ, 45^\circ]$ both in azimuth and elevation sampled at 10°
- Test set 2° sampling

Remote Sensing Data



1. built-up land
2. forest
3. water
4. agricultural area

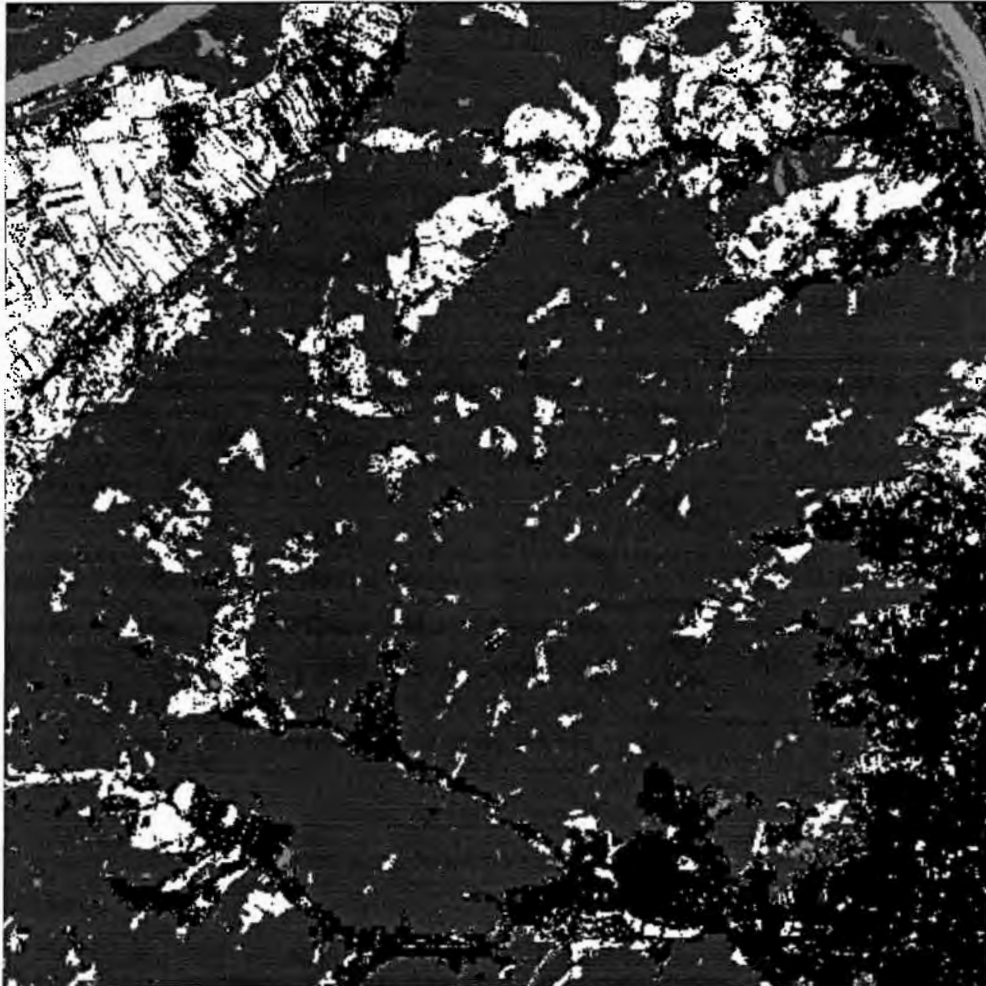
approx. 3000 pixels for training

Optimized Neural Network

- Started with 30 Hidden units
- After 12 selections 4 Hidden units remained
- Total 120 epochs of training

category	ML	NN	Opt. NN
built-up land	78.2%	87.5%	85.9%
forest	89.7%	89.9%	91.9%
water	84.7%	95.7%	90.4%
agricultural area	74.1%	70.6%	74.2%
average accuracy	84.7%	85.9%	86.2%

Optimized Network



Vector Quantization and MDL

The length of encoding S ($L(S(A))$)

1. The length of encoding $O = S - I$, denoted by $L(O)$
2. The length of encoding A , denoted by $L(A)$.
3. The length of encoding I using A , subdivided according to:
 - (a) The length of encoding the index of A , denoted by $L(I(A))$.
 - (b) The length of encoding the residual errors, $L(\epsilon(I_A))$.

$$L(S(A)) = L(A) + L(I(A)) + L(\epsilon(I_A)) + L(O).$$

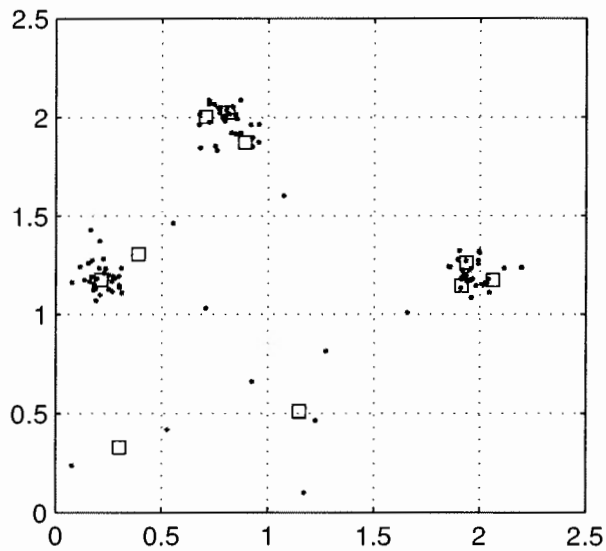
Main idea: Minimize $L(S(A))$

- Detect Outliers $L(O)$
- Check for each $\mathbf{c}_j$, $\Delta L_{\mathbf{c}_j}$ when it is removed.
- Remove vectors which decrease $L(S(A))$.

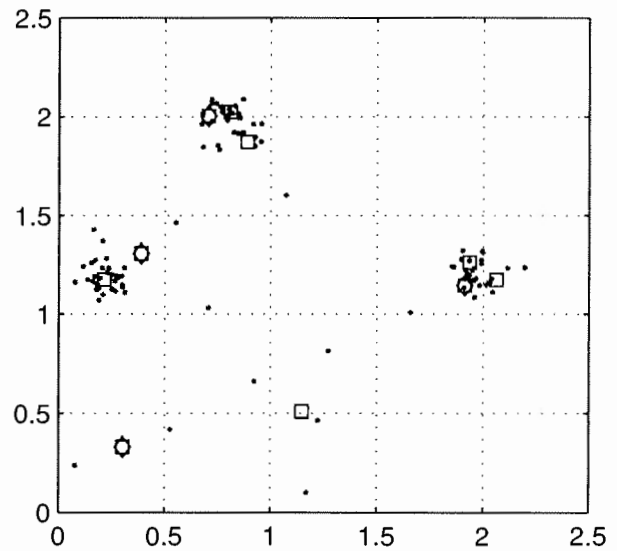
Features of the Algorithm

- Controlled and efficient way of removing reference vectors.
- Low additional cost, necessary terms are calculated during adaptation.
- Network need not be trained to convergence.
- Similarities to EM-type approaches:
 - In each step the description length is reduced.
 - We are guaranteed to find a local minimum of $L(S)$.

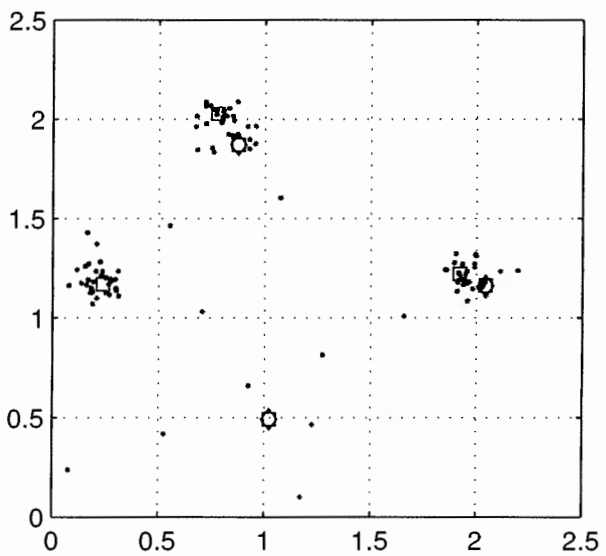
One run of the algorithm



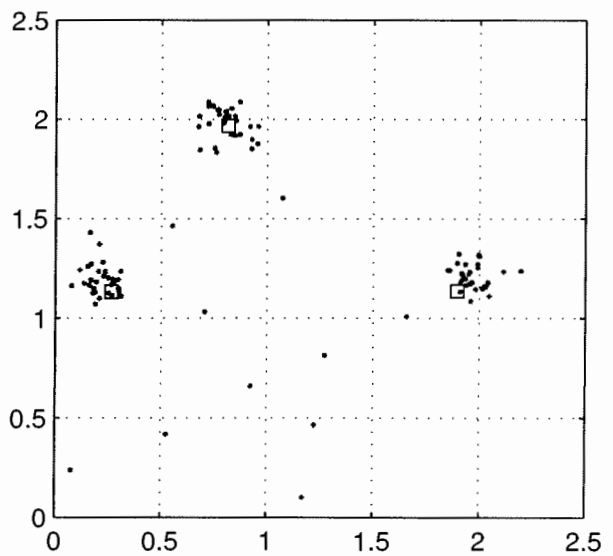
(a) Initialization



(b) After 1st selection

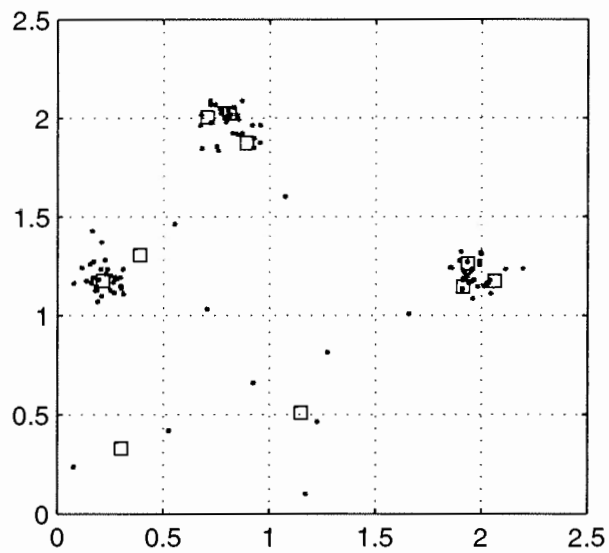


(c) After 2nd selection

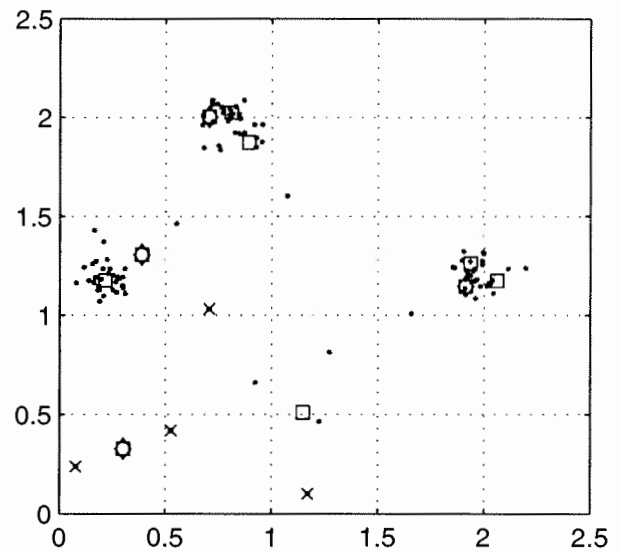


(d) Final result

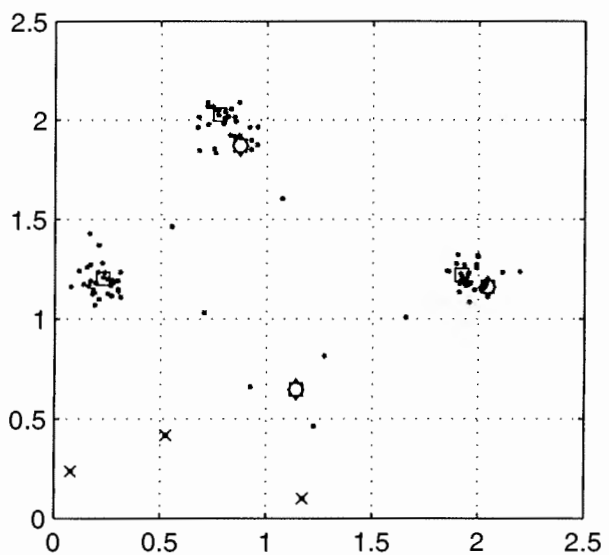
With outlier identification



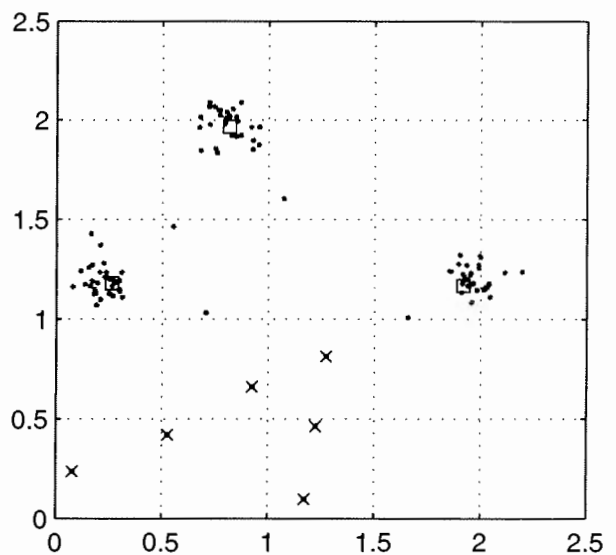
(a) Initialization



(b) After 1st selection

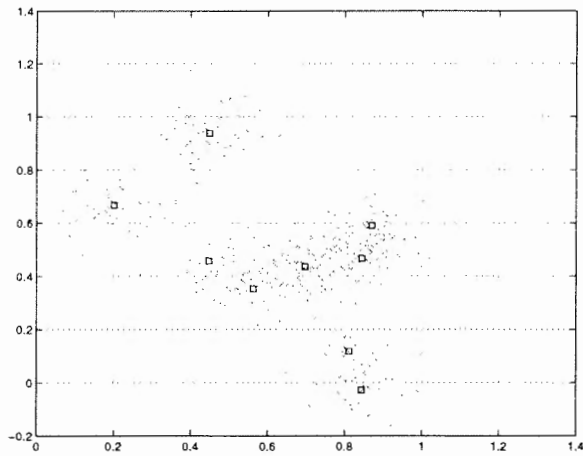


(c) After 2nd selection

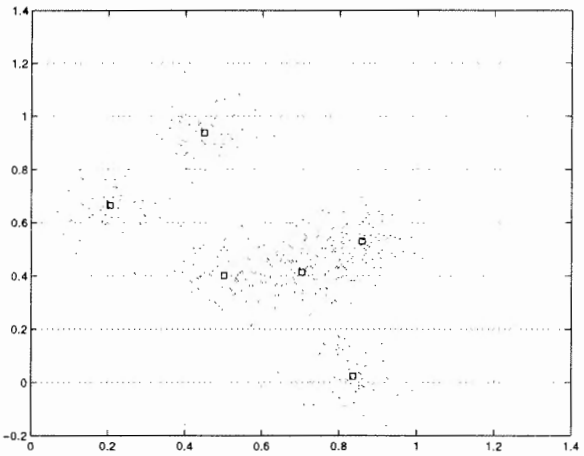


(d) Final result

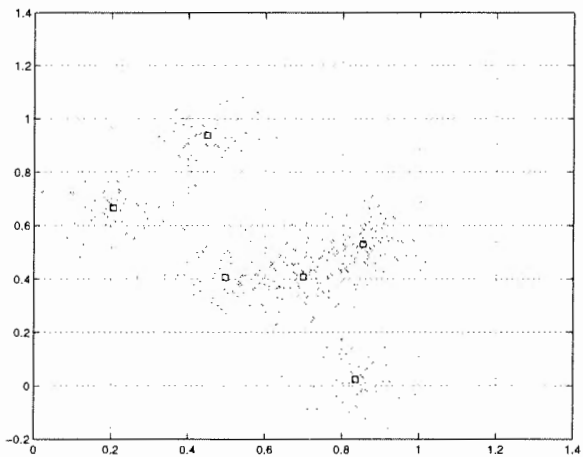
Hierarchical clustering



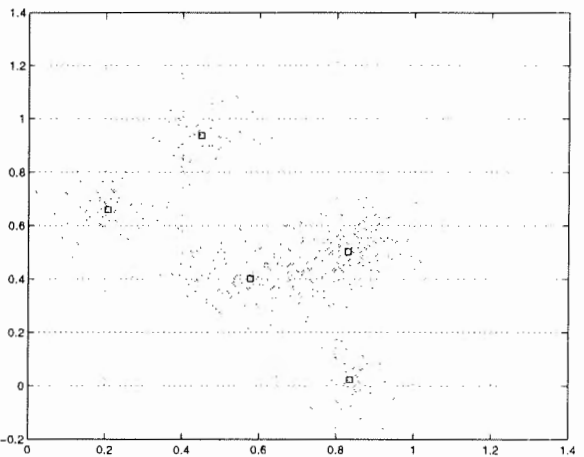
(a) $\sigma = 0.05$



(b) $\sigma = 0.07$

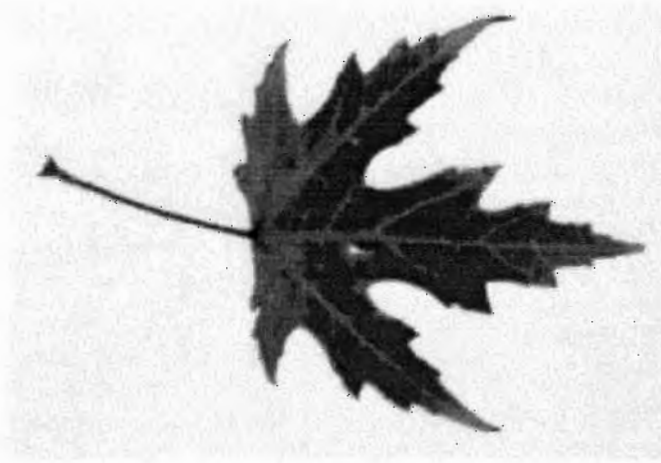


(c) $\sigma = 0.09$

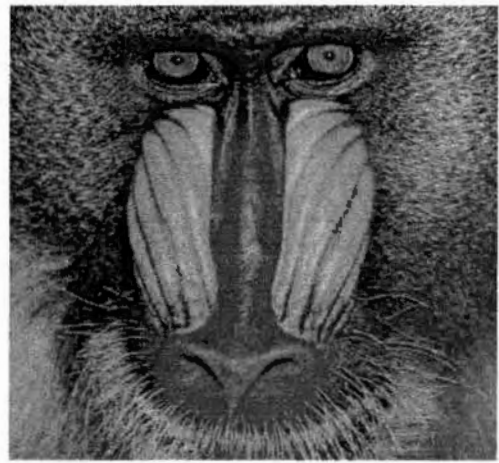


(d) $\sigma = 0.15$

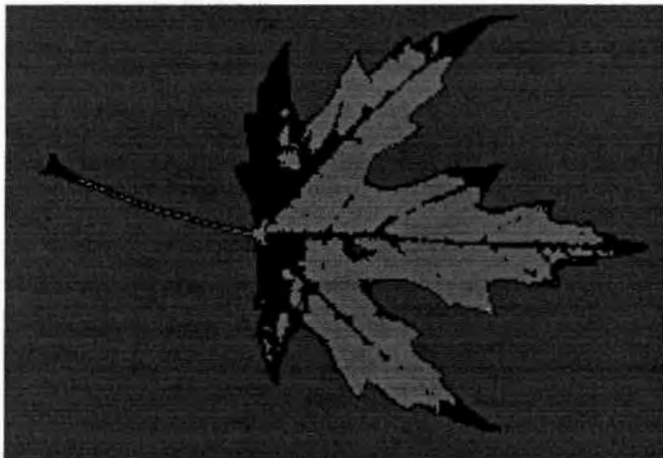
Color Segmentation



(a) Leave Image



(b) Mandrill image



(c) Segmented Leave (3 clusters)



(d) Segmented Mandrill (4 clusters)

Image Quantization

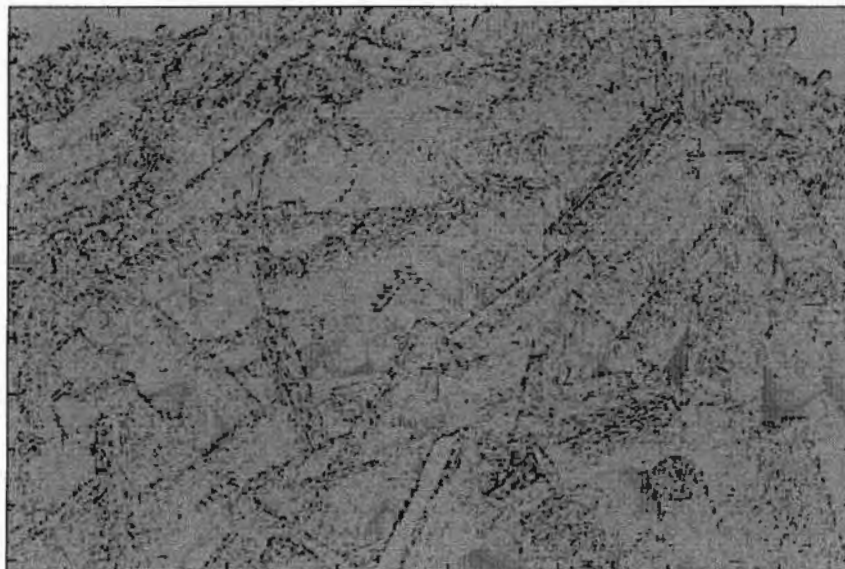
- Starting with 256 vectors.
- Half of 8×8 windows were used as training set.
- Converged after 7 selection steps at 32 vectors.
- Compression ratio 50



(a) Original Image



(b) Reconstructed Image



(c) Error Image

Robot-Vision

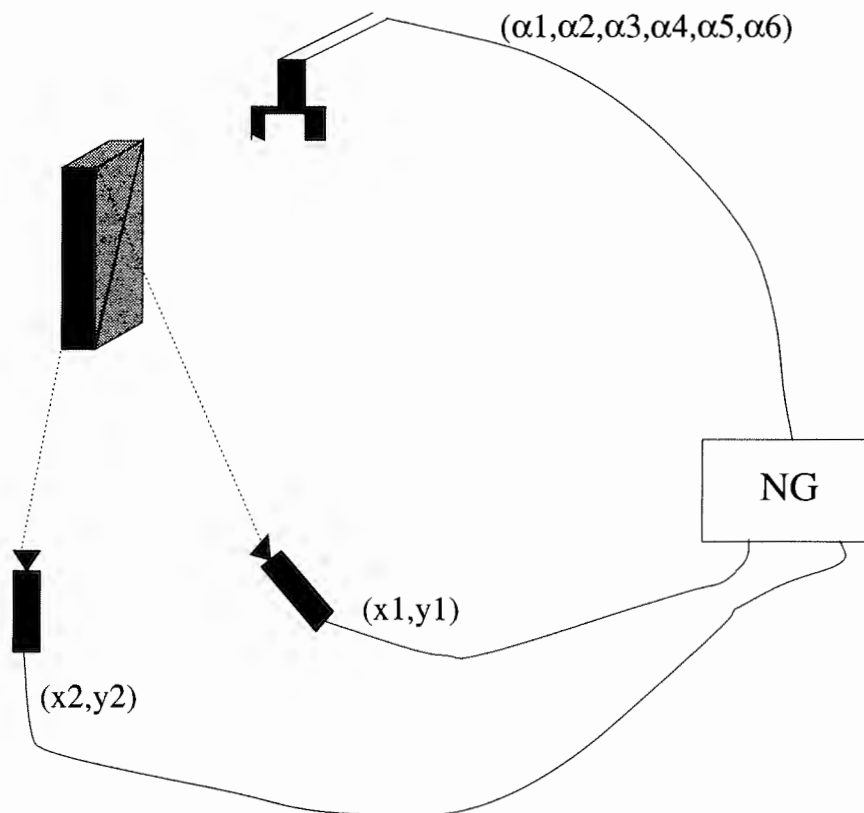


Horst Bischof

PRIP TU-Vienna

Hand-Eye Coordination with Neural Gas

- Learns a mapping between camera coordinates (4-D) and angles of robot (e.g. 6D).
- Tessellates the space with units
- Each unit learns output response



Regelung eines Drehstromlichtbogenofens

Ein Drehstromlichtbogenofen stellt ein elektrisch-mechanisches System dar, wie es in Abbildung 1 schematisch skizziert ist. Ein Lichtbogen, der zwischen den von den Tragarmen geführten Graphitelektroden und dem Einsatz (meist Schrott) brennt, liefert die zum Schmelzen des Einsatzes nötige Energie (ähnlich einem Elektroschweißgerät). Die Dynamik der Regelstrecke wird durch das Tragarmsystem bestimmt, die elektrischen Größen im System wirken als nichtlineare Verstärkung, die die 3 Stränge zusätzlich miteinander verkoppelt. Meist wird als Regelgröße die Impedanz der einzelnen Stränge verwendet, die einen festen Sollwert einhalten soll. Dieser Widerstand wird stark von der Länge des Lichtbogens bestimmt. Der Ohmsche Widerstand des Stranges steigt direkt proportional mit der Länge des Bogens, d.h. ein langer Bogen bedeutet hohe Spannung und niedrige Stromstärke, ein kurzer Bogen bedeutet hohe Ströme und niedrige Spannungen. Störungen des Systems werden meist durch sprungförmige Verkürzung der Lichtbögen (= Schrottsturz) verursacht, die in der Simulation als sprungförmige Veränderung der Badhöhe berücksichtigt werden können. Um eine möglichst effiziente Einschmelzung zu erreichen sind auch Sollwertänderungen nötig.

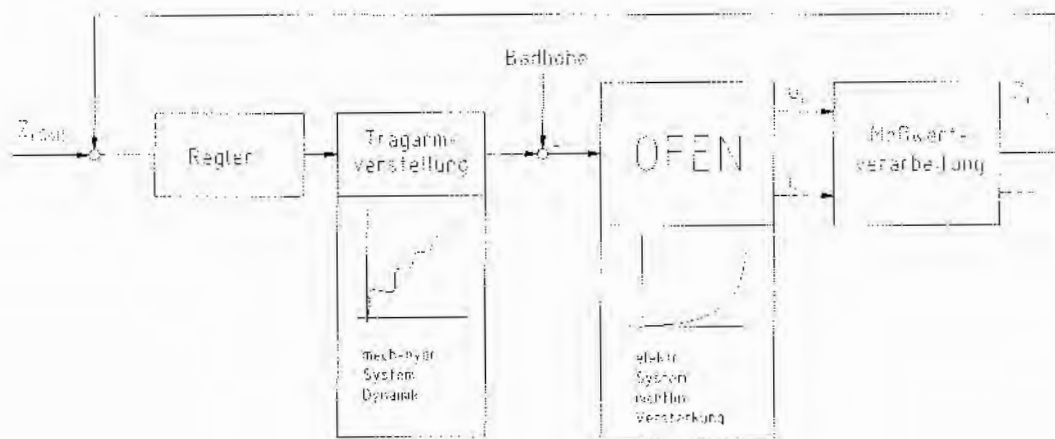


Abbildung 1: Schema eines Lichtbogenofenregelkreises

Zumeist wird jeder Strang extra von einem P-Regler geregelt, das heißt die durch das Drehstromsystem verursachten Kopplungen werden nicht berücksichtigt (siehe auch Simulationsausdrucke). Durch die nichtlineare Verstärkung der Regelstrecke sind für einen P-Regler für jeden Sollwert andere Verstärkungsfaktoren zu wählen. Durch eine Simulation soll untersucht werden, wie ein ähnlich aufgebauter Fuzzy-Regler auf Systemstörungen und Sollwertänderungen reagiert.

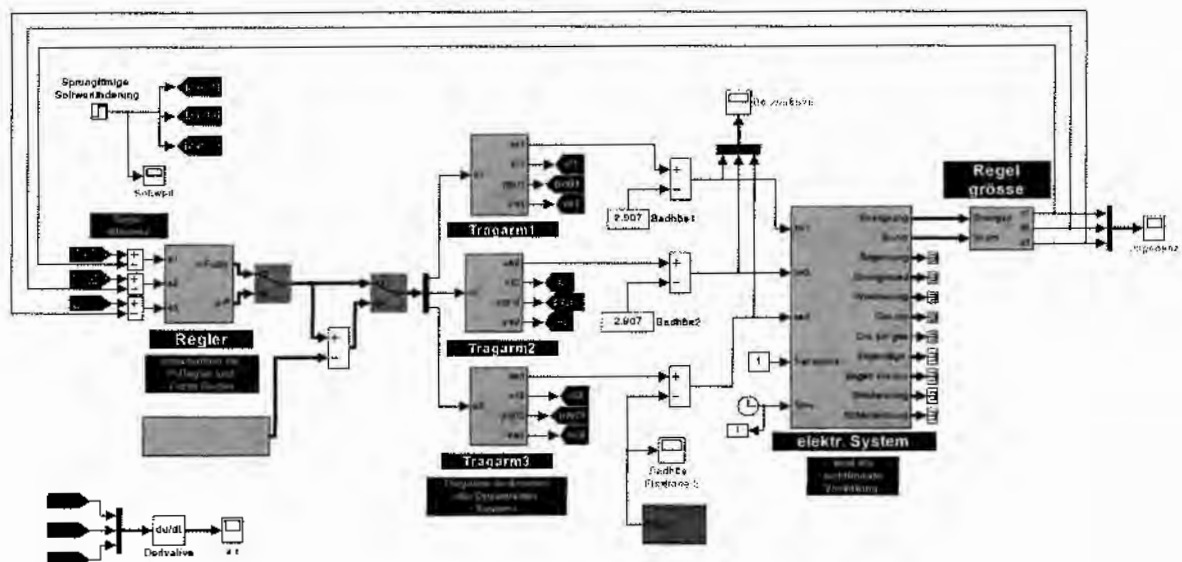


Abbildung 2: Simulink Modell des Lichtbogenofenregelkreises

Der Regler ist umschaltbar zwischen einem Fuzzy Regler und einem P-Regler mit und ohne Aufschaltung des Zylinderdruckes der Hydraulik, die die Tragarme verfährt. Eingänge für den Fuzzy Regler sind die Regelabweichung (Sollimpedanz – Istimpedanz) und der Hydraulikdruck im Zylinder. Mittels des MATLAB Fuzzy GUI Tools wurde ein Fuzzy Regler aufgebaut, der stark an den P-Regler mit Aufschaltung des Zylinderdruckes angelehnt ist.

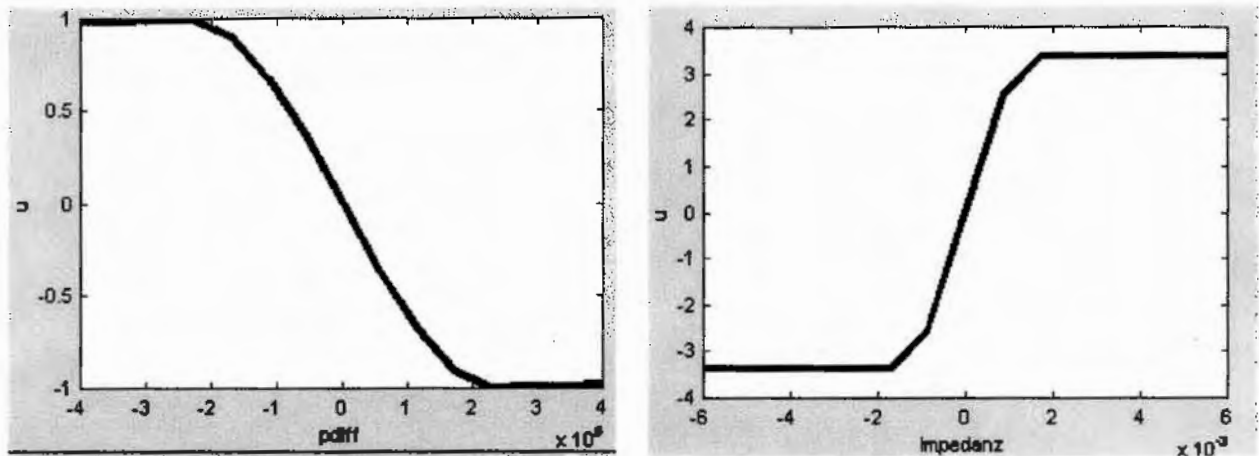


Abbildung 3

Abbildung 3 zeigt die Stellgröße u des Fuzzy Reglers, wenn nur eine Eingangsgröße betrachtet wird. Abbildung 4 zeigt das nichtlineare Reglerkennfeld des Fuzzy Reglers in Abhängigkeit beider Eingangsgrößen.

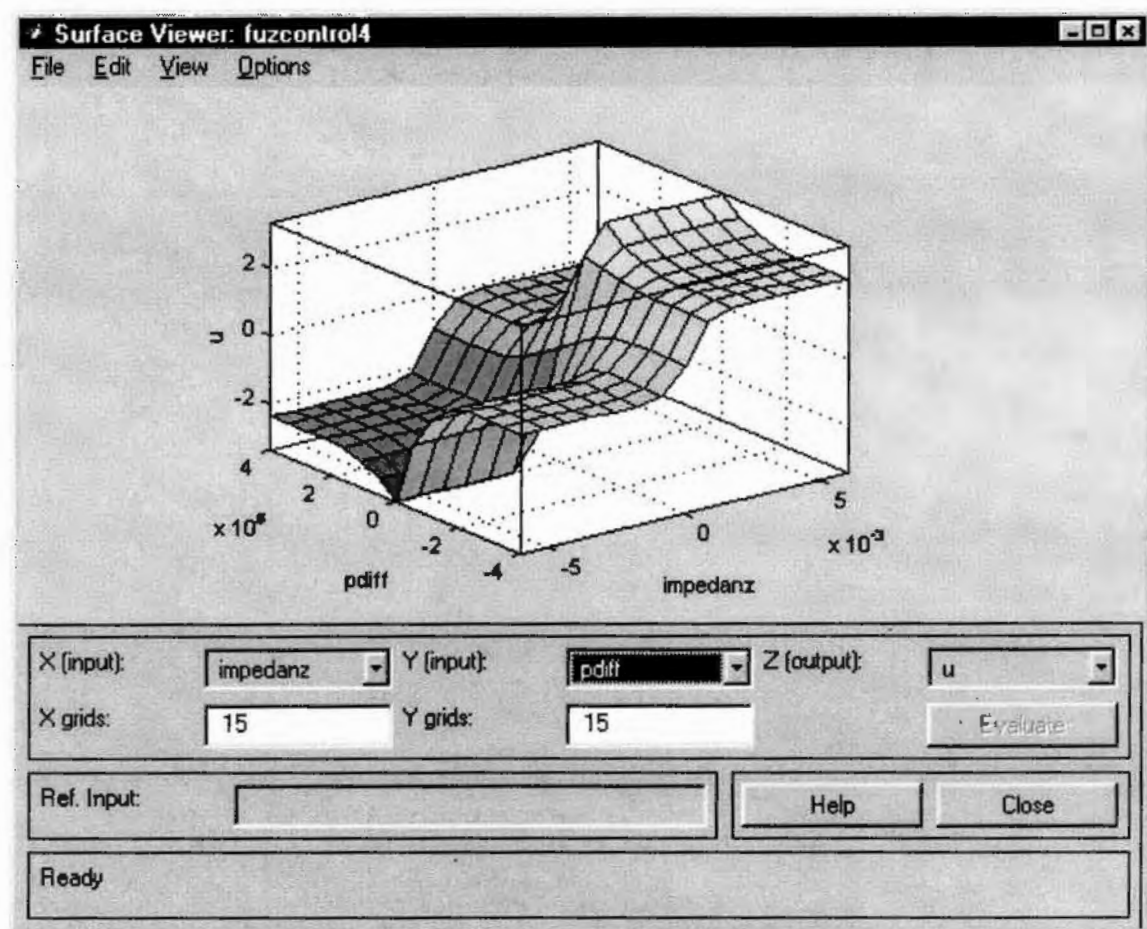


Abbildung 4: Nichtlineares Reglerkennlinienfeld

Das Verhalten des geschlossenen Regelkreises mit Fuzzy Regler und P-Regler mit Hydraulikdruckaufschaltung wurde mit folgenden Eingangsgrößen getestet:

Störung durch Badsprung im Strang 3
 Sprungförmige Sollwertänderung für alle 3 Stränge
 Störung durch Badsprung im Strang 3

bei $t = 5\text{sec}$
 bei $t = 15\text{sec}$
 bei $t = 20\text{sec}$

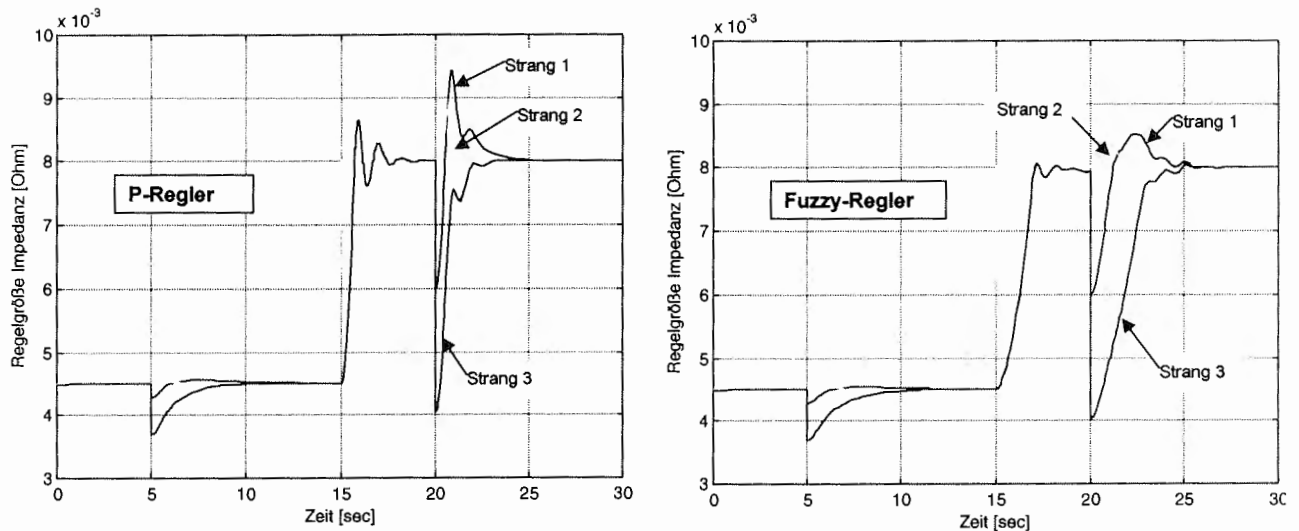


Abbildung 5: Verlauf der Regelgröße Impedanz

Abbildung 5 zeigt, daß das Störungsverhalten der Regler beim unteren Sollwert nahezu gleich ist. Beim Sollwertsprung zeigt die Regelgröße beim P-Regler ein signifikantes Überschwingen (bei schnellerer Reaktion), der Fuzzy-Regler kann dem Sollwertsprung ohne besonderes Überschwingen folgen. Bei der nachfolgenden Störung zeigt sich, daß der P-Regler durch die Nichtlinearität der Regelstrecke für diesen Sollwert eine zu hohe Verstärkung besitzt, der Fuzzy Regler zeigt bis kurz vor dem Ausregeln ein sehr schönes Störverhalten.

Da von der Hydraulik die Stellbefehle des Reglers in eine Hubbewegung der Elektroden umgesetzt werden, die in der Simulation praktisch die Bogenlänge darstellen wird in Abbildung 6 der Verlauf der Bogenlänge dargestellt. Durch Untersuchung der Bogenlänge und deren Ableitungen kann somit auf die Belastung des Tragarmsystems geschlossen werden, die durch die großen bewegten Massen von mehreren Tonnen ein wichtiges Beurteilungskriterium für den Regler darstellt.

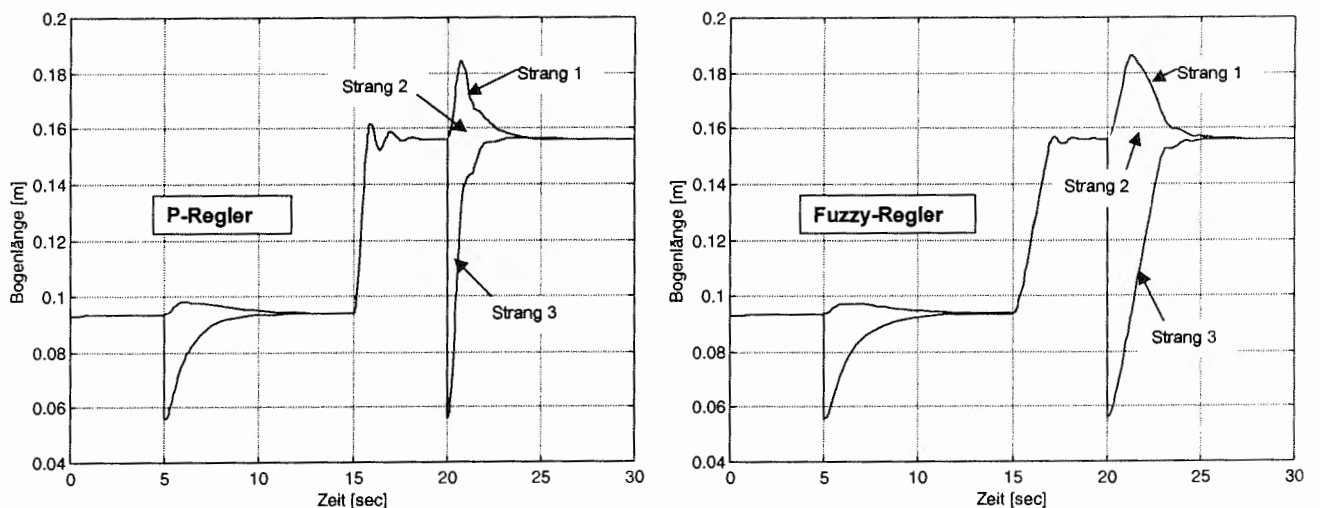


Abbildung 6: Verlauf der Bogenlängen

Abbildung 7 zeigt schließlich noch den Verlauf der Beschleunigung von Tragarm 3 beim Fuzzy-Regler und beim P-Regler. Der Fuzzy Regler zeigt bei allen Übergängen ähnliche Beschleunigungswerte, während der P-Regler das Tragarmsystem wesentlich stärker und nicht gleichmäßig belastet.

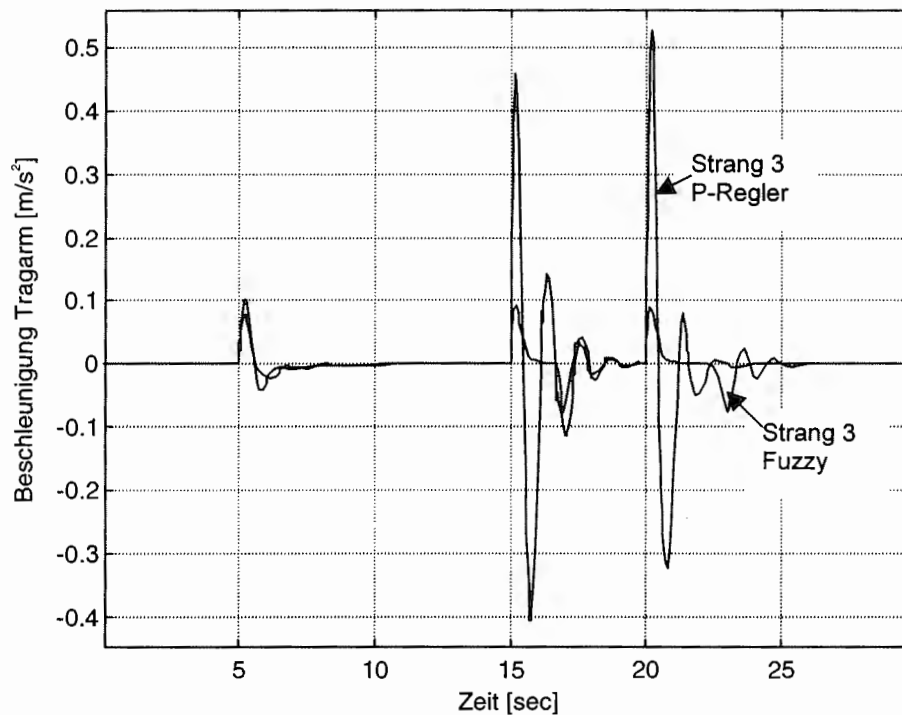


Abbildung 7: Verlauf der Tragarmbeschleunigung von Tragarm 3

Der Entwurf des Fuzzy Reglers mit Hilfe des MATLAB GUI's gestaltet sich sehr intuitiv und wird durch die grafische Oberfläche sehr gut unterstützt. Eine Ausprogrammierung des Fuzzy Reglers mittels MATLAB Befehlen ist wesentlich unübersichtlicher, bietet jedoch mehrere Möglichkeiten der Gestaltung des Reglers.

Comparison 9: Fuzzy Control of a Two Tank System

The number of applications containing fuzzy components is still increasing. Modern simulation systems provide enhancements to implement fuzzy components in a convenient way.

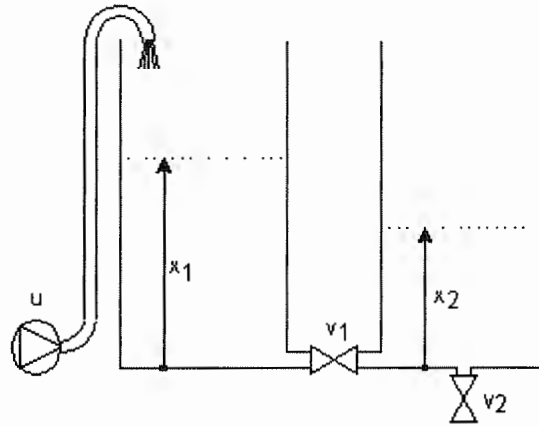


Figure 1: The two tank system

A two tank system (see figure 1) in a specific configuration is characterized by the following nonlinear ODE set

$$\begin{aligned} f &= 0.06624 v_1 \sqrt{|x_1 - x_2|} \operatorname{sign}(x_1 - x_2) \\ \dot{x}_1 &= 0.067 u - f \\ \dot{x}_2 &= f - 0.0605 r v_2 |x_2|^{0.43} \end{aligned}$$

where

$$r = \begin{cases} 1.2 & : x_2 < 16 \text{ cm} \\ 1 & : x_2 \geq 16 \text{ cm} \end{cases}$$

This model includes characteristics of the liquid (laminar, turbulent, friction). The valve positions are $v_1 = 0.4$, $v_2 = 0.3$. The task is to control the liquid level x_2 . A fuzzy controller is used according to figure 2.

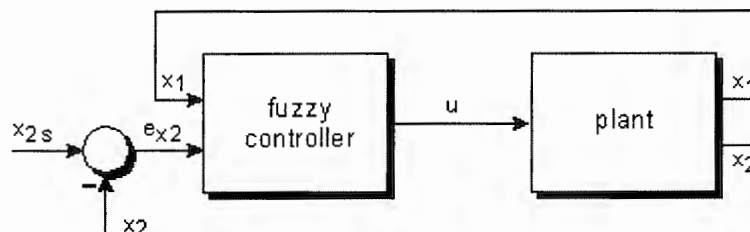


Figure 2: Fuzzy control of liquid level x_2

Two fuzzy controllers FC1 and FC2 will be defined. They should be implemented as discrete systems operating at one second sampling time. The membership functions for e_{x2} and x_1 are defined in figure 3. They are the same in the case of FC1 and FC2.

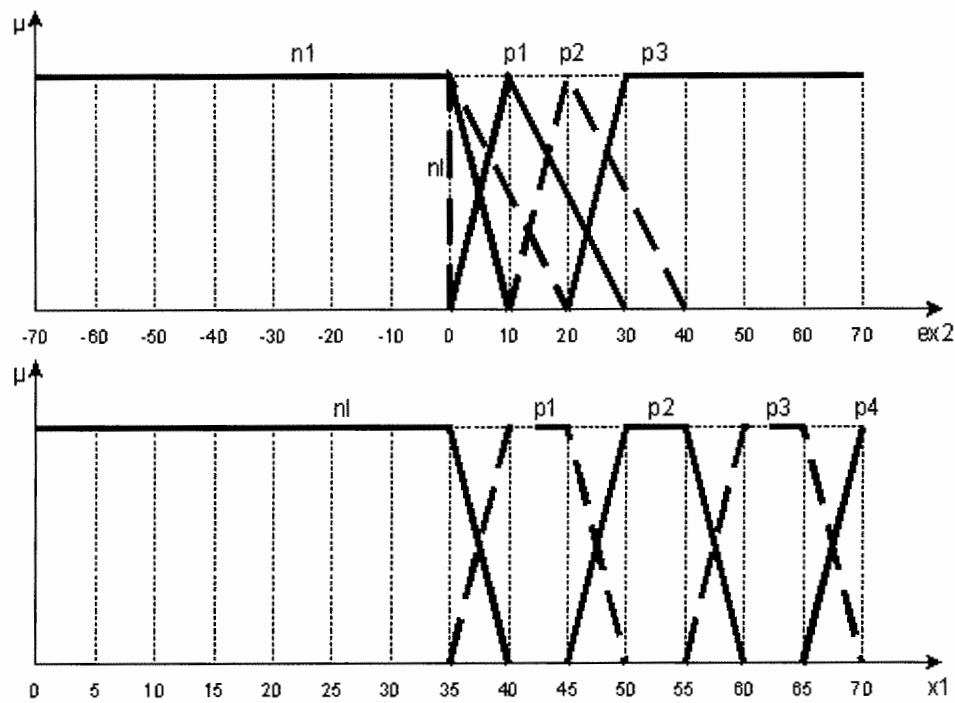


Figure 3: Membership functions of e_{x2} and x_I for FC1 and FC2

The membership functions for the linguistic output variable u in the case of FC1 are defined in figure 4 and in the case of FC2 singletons are used as shown in figure 5.

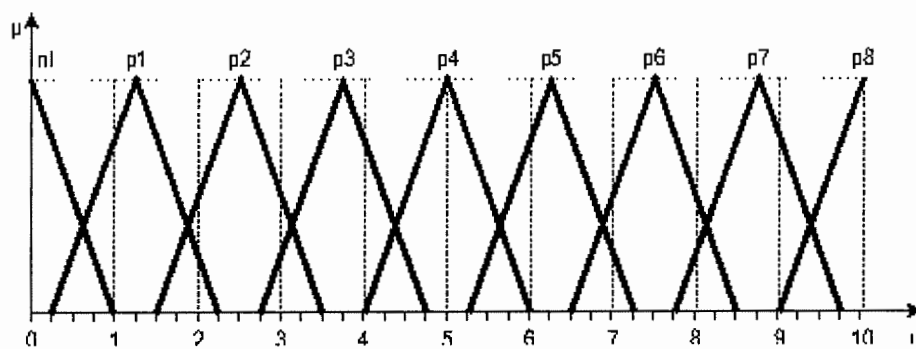


Figure 4: Membership functions of u for FC1

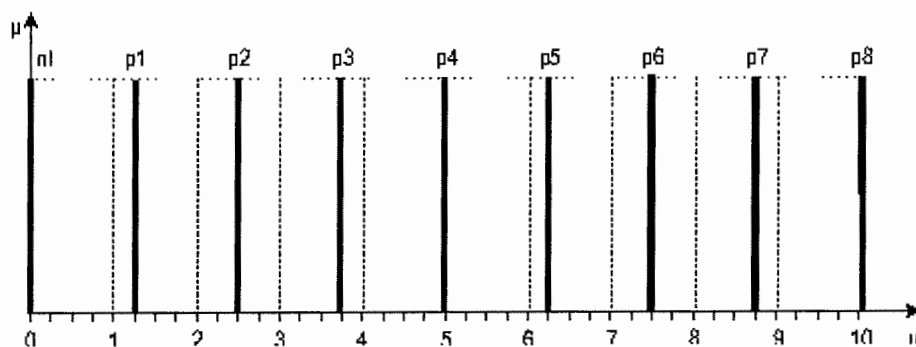


Figure 5: Singleton membership functions of u for FC2

The rulebase (figure 6) should be implemented using the operators *MIN* for AND and *MAX* for OR. For the inference algorithm use *max-prod* and for defuzzification *center of gravity*.

		x_1				
		nl	p1	p2	p3	p4
ex2	p3	p8	p7	p5	p3	nl
	p2	p7	p6	p4	p3	nl
	p1	p7	p5	p3	p2	nl
	nl	p4	p3	p2	p1	nl
	n1	nl	nl	nl	nl	nl
	n1	nl	nl	nl	nl	nl

Figure 6: Table of rules

The tasks to be performed are:

- **Task a**
 - (a1) Describe the features supporting fuzzy control in your simulator or the interface to an interfaced fuzzy tool. Model the controller by means of features of the simulator or an appropriate additional tool linked to the simulator. Give a rough model description of the controller and of the the overall model.
 - (a2) Compute and visualize the 3 dimensional characteristic (surface) of the fuzzy controller FC1. Place e_{x2} [-70 .. 70] on the x-axis, x_1 [0 .. 70] on the y-axis and u on the z-axis. Subdivide the x and y-axis 40 times (41 times 41 points). State the calculation time ta_{fc1} . Specify the machine used. Since the computation of fuzzy systems is a complex task, the calculation time for 1681 lookups documents the performance of the implementation.
 - (a3) Repeat (a2) using FC2. If your system does not support singletons directly, you may use any kind of *emulation*. State the calculation time ta_{fc2} and specify the ratio ta_{fc1} over ta_{fc2} .
- **Task b**
 - (b1) Simulate the whole system using FC1 for $x_{2s} = 25$ cm for 1000 seconds. Plot x_2, x_1 and u versus time. State the computation time tb_{fc1} .
 - (b2) Repeat (b1) using FC2. State the calculation time tb_{fc2} and specify the ratio tb_{fc1} over tb_{fc2} .
- **Task c**
 - (c1) FC3 is defined using FAM interference and is obtained by weighting the rules of FC2 according to the following table

		x_1				
		nl	p1	p2	p3	p4
ex2	p3	1	1	0.1	1	1
	p2	1	1	0.1	1	1
	p1	1	1	0.1	1	1
	nl	1	1	0.1	1	1
	n1	1	1	0.1	1	1
	n1	1	1	0.1	1	1

Figure 7: Table of weightings

Repeat (a2) using FC3. State the calculation time tc_{fc3} only (no surface plot). Describe how weighting can be implemented into your fuzzy description.

- (c2) Describe the outstanding features of the simulation system regarding to fuzzy modelling.

*Johannes W. Goldynia, Johann M. Marinits,
email: goldynia@iert.tuwien.ac.at, marinits@iert.tuwien.ac.at,
Vienna University of Technology, Institute for Electrical Control Engineering,
Gusshausstr. 27-29/375, A-1040 Vienna/Austria*

More Information about the criteria

The following tasks had to be performed:

- **Task a:** (a1) Compute and visualize the 3 dimensional characteristic (surface) of the fuzzy controller FC1. Place x_2 [-70 .. 70] on the x-axis, x_1 [0 .. 70] on the y-axis and u on the z-axis. Subdivide the x and y-axis 40 times (41 times 41 points). State the calculation time $tafc1$. Specify the machine used. Since the computation of fuzzy systems is a complex task, the calculation time for 1681 lookups documents the performance of the implementation. (a2) Repeat (a1) using FC2. If your system does not support singletons directly, you may use any kind of emulation. State the calculation time $tafc2$ and specify the ratio $tafc1$ over $tafc2$.
- **Task b:** (b1) Simulate the whole system using FC1 for $x_{2s} = 25$ for 1000 seconds (by integrating the fuzzy controller as discrete controller). Plot x_2 , x_1 and u versus time. State the computation time $tbfc1$. (b2) Repeat (b1) using FC2. State the calculation time $tbfc2$ and specify the ratio $tbfc1$ over $tbfc2$.
- **Task c:** (c1) FC3 is defined using FAM interference and is obtained by weighting the rules of FC2 according to a table. State the calculation time $tcfc3$ only (no surface plot). Describe how weighting can be implemented into your fuzzy description.

Criteria used:

- **General Criteria:**
 - Model description: textual - graphical - control-oriented textual - control-oriented graphical
 - Fuzzy control module: graphic available - textual available - not available
 - Membership functions: preprogrammed - parametrized - dynamical definition
- **Criteria - Task a:**
 - Generation of fuzzy control surface: very fast - fast - average - slow - very slow
 - Singletons: built in - emulated - programmed - not available
 - Ratio $tafc1/tafc2$ (time for control surfaces): $\sim 1x$ - $5x$ - more
- **Criteria - Task b:**
 - Description of discrete controller: integrated in fuzzy-controller - sample blocks - z-Trafo - discrete section - environment
 - Ratio $tafc1/tbfc1$ (surface/simulation run): $\sim 1x$ - $5x$ - more
 - $>$ Ratio simulation time $tbfc1/tbfc2$: $\sim 1x$ - $2x$ - $5x$ - more
- **Criteria - Task c:**
 - Weighting rules implementation: built in - rule extension - programmed - not available
 - Ratio comp. Speed with/without weighting: $\sim 1x$ - $2x$ - $5x$ - more

The evaluations of the Software-Comparisons are sortet by:

- Programs
- Criteria

Page automatically updated on November 27, 1998 by the ARGESIM-Database



Fuzzy Control of a Two Tank System

More information about criteria!

Why used this Methodology Classification	In principle, the system is a mixed continuous / discrete one. Investigating the eigenvalues shows that sampling time is less than the maximal stepsize required for smooth integration. "Continuous Approach": The classical (dynamic) approach sets up a continuous model, interrupted by sampling and calculation of fuzzy control (very near to a hardware-in-the-loop - situation with real plant and discrete controller at a DSP). It is to be noted that the continuous model is integrated with an appropriate algorithm giving discrete values. In a "Hybrid Approach" an environment or a state machine provides fuzzy controls and handles the simulation of the continuous model for each sampling period. Additionally, a (fully) "Discrete Approach" replaces the "continuous" model by a state update which is also done in an environment or driven by a state machine. This approach is equivalent to the hybrid-like approach with Euler integration using stepsize equal sampling interval. For this approach no simulator is necessary, a numerical desktop calculator is sufficient! Another big difference is whether the fuzzy control is evaluated at runtime ("Algorithmic Fuzzy Function") or whether the fuzzy control is calculated in advance and stored in a table function ("Fuzzy Table Function").
Why used these "general" criteria	The model description may be done in a "general way" (either graphical or textual), or due to the nature of the system using special components for control engineering (graphical or textual) - indicated in criterion 1. The number of simulators, which offer modules for fuzzy controllers is growing - criterion 2 tells, whether a graphical fuzzy module is available, or a textual module, or no special module for fuzzy control. A big difference is found in the definition of membership functions - checked in criterion 3: membership functions may be preprogrammed (only fixed number of functions), parametrized (fixed number of types), or defined dynamically. "Manually" means, that because of the lack of a fuzzy module not only fuzzification, interference and defuzzification, but also the structure of the membership functions have to be programmed manually. Note: A fuzzy module is very convenient, but fuzzy control can easily be programmed manually - by simple interpolation and table lookup. A specific manual implementation may be much more efficient than a fuzzy module, which has the more overhead the more convenience the module offers.
Why used these criteria for task a)	Since the computation of fuzzy systems is a complex task, the calculation time for the 1681 lookups documents the performance of the implementation. For comparison purposes the times are normalized with respect to a Pentium 100 processor and classified from very fast to very slow (criterion 1). Sometimes singletons are not supported as membership functions by fuzzy control modules. Consequently singletons may be "emulated" by the fuzzy module's features, or they must be programmed "manually" (criterion 2). The ratio of the computation time for the fuzzy control surface with or

	without singletons gives a hint for the quality of the singleton implementation / emulation / programming.
Why used these criteria for task b)	Description of sampled data systems may be done in a classical way by sample blocks (zero-order hold blocks), some simulators provide a generic use of the z-transformation (either graphically or textually). Discrete sections offer a more general approach, either by arbitrary or periodic scheduling. Criterion 1 indicates the method used - it does not necessarily indicate that the simulator can just use the given method, it indicates the method used by the person who solved the comparison. Calculation of a fuzzy control requires complex interpolation of a two-dimensional surface. The control is updated at least 1000 times (each sampling time instant). If integration stepsize is smaller than sampling time or/and if integration order is higher than 1, the number of updates is a multiple (RK4, $h = \text{sampling}/10 \rightarrow 40 \times 1000$ updates). Checking the ratio between simulation time and computation time for the fuzzy control surface (with 1681 lookups) - crit. 2 - gives an interesting relation between simulation speed and control update speed (depending on stepsize - and implementation of co-operation between discrete control and continuous plant). Criterion 3 checks the additional effort for singletons in the fuzzy control in conjunction with the simulation - by giving the classified ratio between simulation time with singletons and simulation time without singletons.
Why used these criteria for task c)	Fuzzy control may be supported by weighting rules during calculation of the interference. Here, simulators may offer built-in features for weighting rules, sometimes rules can be modified manually by weights (multiplication with a weighting matrix "rule extension"), and sometimes weighting must be programmed manually. This is checked in criterion 1. The efficiency of the implementation of the weighting factors can be measured by the ratio of simulation time with weighting / without weighting (criterion 3). It is to be noted that a manually programmed implementation may give better results !

Back

Page automatically updated on November 27, 1998 by the ARGESIM-Database

[[BACK to the ANA 2.x homepage](#)]

Comparison 9 - ANA 2.x

published in *EUROSIM - Simulation News Europe*, Number 17, 1996

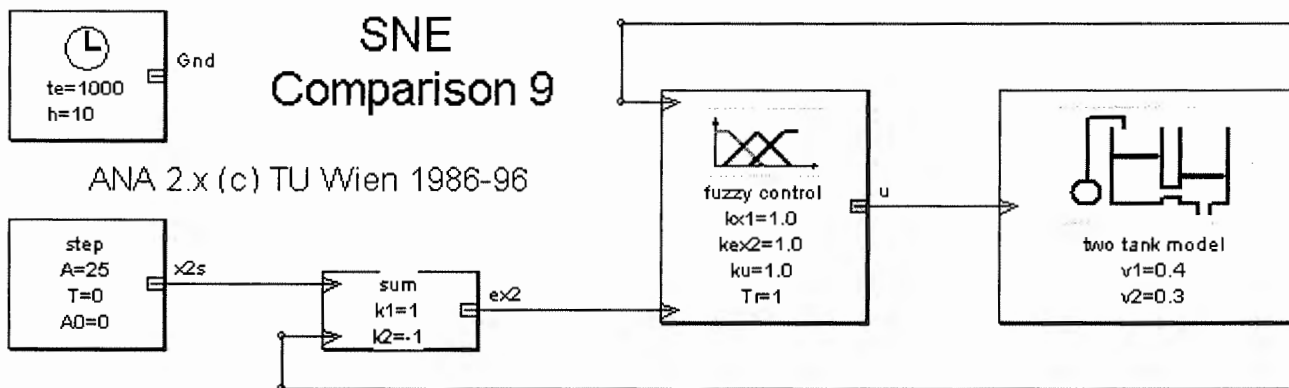
For the definition of this comparison read <http://eurosim.tuwien.ac.at/comparisons/c9/c9def.html>.

Short description

ANA 2.x is a comprehensive software tool for all control engineering tasks. A short description of *ANA 2.x* has been already given in [Goldynia96]. All tasks of this comparison were done using the interpreting *ANAmcl* compiler without the need of any additional computer language. The time for compiling the model is less than 4 seconds. Therefore a turnaround is extraordinarily fast.

Model description

ANA 2.x offers seamless integration of a development environment for fuzzy control named *ANAfuzzy 2.5*. The user interface is intuitive and simple but yet very powerful. *ANAfuzzy 2.5* comprises graphical editing of membership functions, specification of rules using a flexible syntax, several selectable methods for operator and defuzzification semantics, generation of surface plots and automatic creation of interface blocks to the fuzzy subsystem to become part of a block circuit diagram generated by *ANA 2.x*. The plant is specified using *ANAmcl*.



ANA 2.x circuit within the environment

Results task a):

(a1) Representation

Beside the graphical definition of the membership functions, the rules are written within a fuzzy rule editor, which is part of *ANAfuzzy 2.5*. Syntax checks and checks for valid identifiers are done automatically. The common rule base for FC1 and FC2 reads:

```
# rules 1-5
IF (ex2==p3) AND (x1==n1) THEN u=p8;
IF (ex2==p3) AND (x1==p1) THEN u=p7;
IF (ex2==p3) AND (x1==p2) THEN u=p5;
```

```

IF (ex2==p3) AND (x1==p3) THEN u=p3;
IF (ex2==p3) AND (x1==p4) THEN u=n1;
# rules 6-10
IF (ex2==p2) AND (x1==n1) THEN u=p7;
IF (ex2==p2) AND (x1==p1) THEN u=p6;
IF (ex2==p2) AND (x1==p2) THEN u=p4;
IF (ex2==p2) AND (x1==p3) THEN u=p3;
IF (ex2==p2) AND (x1==p4) THEN u=n1;
# rules 11-15
IF (ex2==p1) AND (x1==n1) THEN u=p7;
IF (ex2==p1) AND (x1==p1) THEN u=p5;
IF (ex2==p1) AND (x1==p2) THEN u=p3;
IF (ex2==p1) AND (x1==p3) THEN u=p2;
IF (ex2==p1) AND (x1==p4) THEN u=n1;
# rules 16-20
IF (ex2==n1) AND (x1==n1) THEN u=p4;
IF (ex2==n1) AND (x1==p1) THEN u=p3;
IF (ex2==n1) AND (x1==p2) THEN u=p2;
IF (ex2==n1) AND (x1==p3) THEN u=p1;
IF (ex2==n1) AND (x1==p4) THEN u=n1;
# rules 21-25
IF (ex2==n1) AND (x1==n1) THEN u=n1;
IF (ex2==n1) AND (x1==p1) THEN u=n1;
IF (ex2==n1) AND (x1==p2) THEN u=n1;
IF (ex2==n1) AND (x1==p3) THEN u=n1;
IF (ex2==n1) AND (x1==p4) THEN u=n1;

```

The setup of the fuzzy operators and methods has been set globally via a dialog window.

(a2 and a3) Visualization of FC1 and FC2

All computations were done using *ANA* 2.5 on a PC with 80486 100 MHz CPU. $ta_{fc1} = 39$ seconds, $ta_{fc2} = 4.5$ seconds, $ratio = 8.67$. There is direct support for singletons. Three dimensional surfaces can be generated from an ASCII file exported from *ANA* with either *GNUPLOT* or *MATLAB*. Three dimensional built in graphics for *ANA* will be available soon.

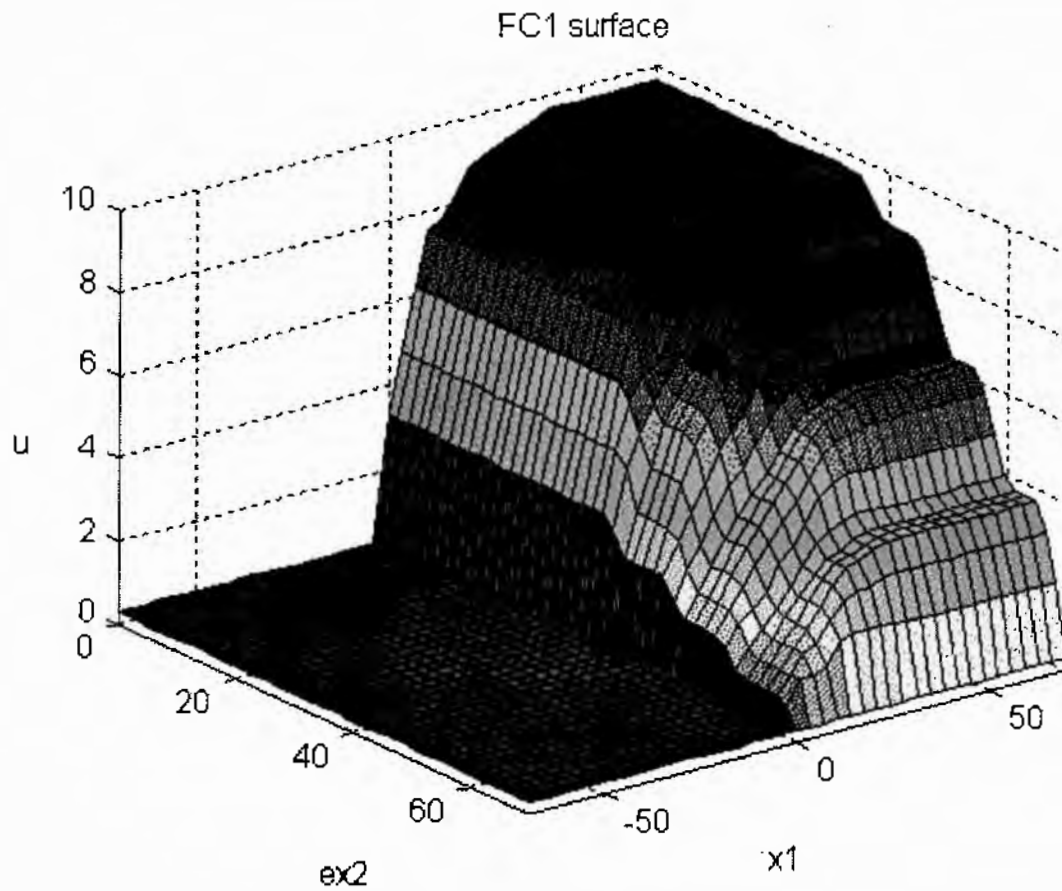


Figure 1: FC1 surface

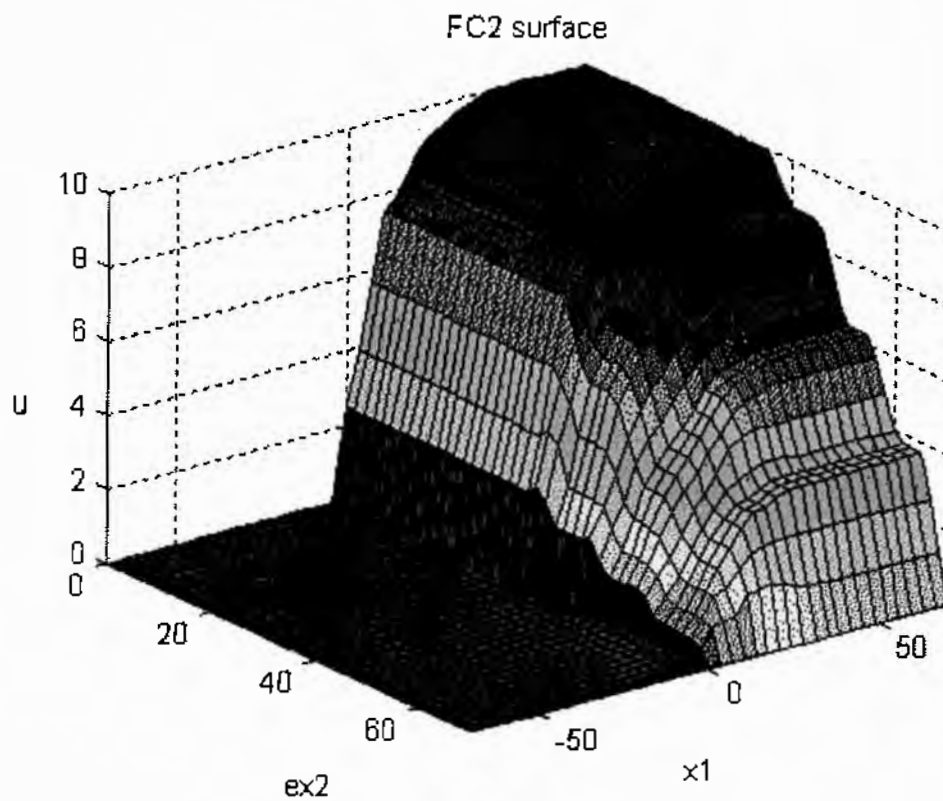


Figure 2: FC2 surface (singleton)

Results task b):

(b1) and (b2) results in $tb_{fc1} = 24$ seconds, $tb_{fc2} = 5$ seconds and $ratio = 4.8$.

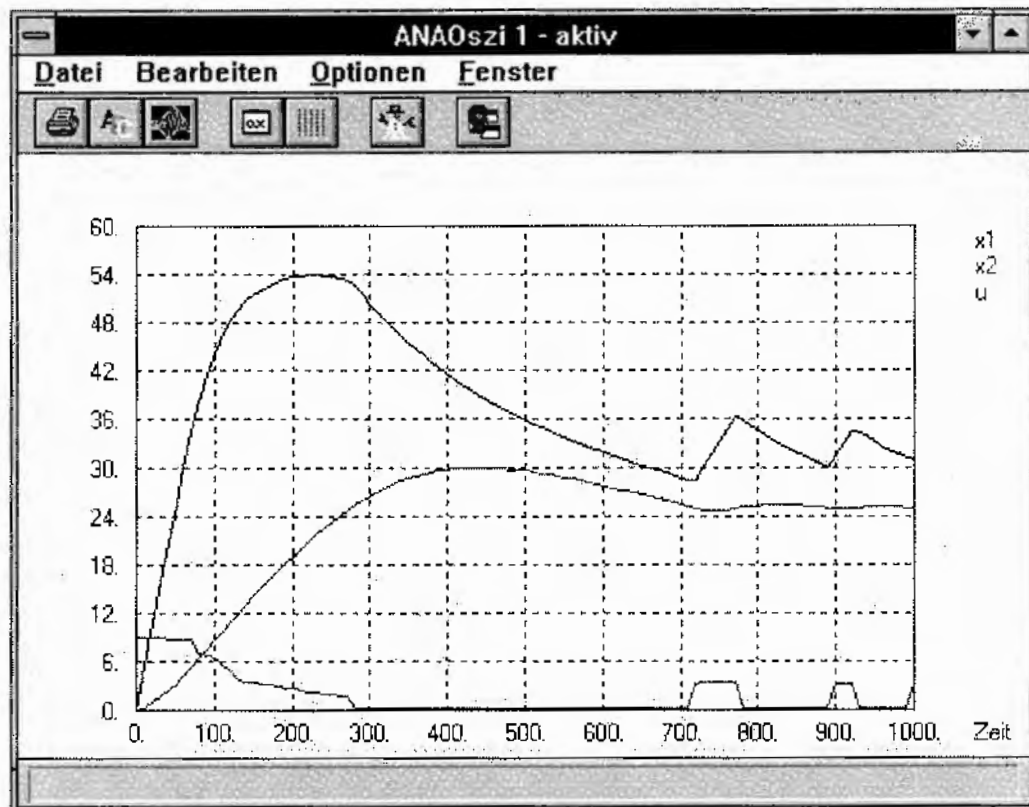


Figure 3: Simulation of whole system using FC1

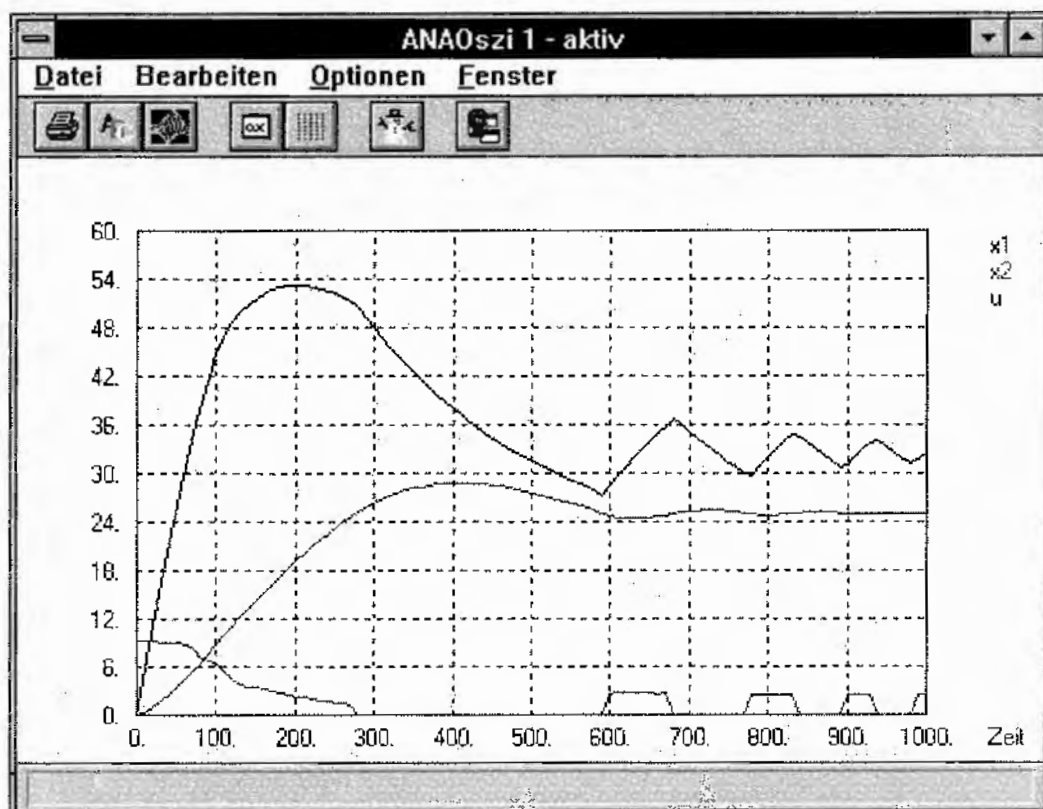


Figure 4: Simulation of whole system using FC2

Results task c):

(c1) FAM interference

Weighted rules can be defined using the WITH clause e.g. IF (ex2==p3) AND (x1==p2) THEN u=p5 WITH 0.1; The computation time $t_{c_{fc3}} = 4.5$ seconds.

(c2) Outstanding features

No more than shown above.

References

[Goldynia96] *Comparison 7 - ANA 2.x*, SNE 16, 1996, p. 35

The ANA 2.x system is free of charge. The base system can be obtained using anonymous ftp to [<ftp://ftp.iert.tuwien.ac.at/ana2>](ftp://ftp.iert.tuwien.ac.at/ana2). There is also information concerning ANA 2.x on our WWW server [<http://www.iert.tuwien.ac.at/ana2>](http://www.iert.tuwien.ac.at/ana2).

Johannes W. Goldynia

Institute for Electrical Control Engineering, Vienna University of Technology

Gusshausstr. 27-29/375, A-1040 Vienna

Tel. +43-1-58801 5220, Fax +43-1-5058907

email: goldynia@iert.tuwien.ac.at

[\[BACK to the ANA 2.x homepage\]](#)

Comparison 9 - SIMULINK

SIMULINK is the simulation *blockset* on top of MATLAB. It is a block-oriented simulation system with a graphical frontend. Models are composed using blocks with defined inputs and outputs, which are connected by lines. A rich set of standard blocks can be found in predefined libraries. Additionally *s-functions*, which are MATLAB interpreter scripts, can be used to implement blocks of higher complexity.

Description: The model of the Two Tank System was built up by combining the MATLAB *s-function* *s_tank* (figure 2), which describes the tank dynamics, and the function *sfis* inside the block *Fuzzy-Controller*. The last function was built with the aid of the MATLAB *Fuzzy Toolbox*.



Fig. 1: SIMULINK Model

```
function [sys, x0] = s_tank(t,x,u,flag)
% return xdot
if x(2) < 16
    r2 = 1.2;
else
    r2 = 1;
end
x1dot = 0.067*u - v1*0.06624*sqrt(abs(x(1) -
    -x(2))) * sign(x(1) - x(2));
x2dot = -r2*v2*0.0605*abs(x(2))^0.43 + v1*0.06624*
    *sqrt(abs(x(1) - x(2))) * sign(x(1) - x(2));
sys = [x1dot, x2dot];
...
```

Fig. 2: s-function *s_tank*

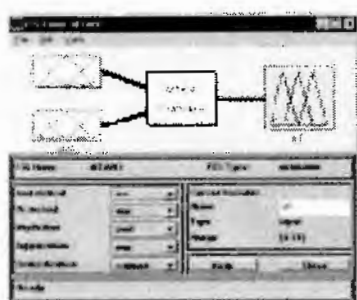


Fig. 3: FIS Editor

and Defuzzification can be defined by selecting predefined methods as well as by choosing customized ones. Rules are defined in a *Rule Editor* where one can choose between different formats of description called *verbose*, *symbolic* or *indexed*. The definition of the membership function can also be done in a special editor. The result is a matrix, describing the different parameters and

methods that has to be loaded into the model using the *readfis* command.

Task a2) The visualisation of the surface (figure 4) can be achieved directly within the *FIS Editor*. The corresponding computing time on a PC 80486 with 100 MHz was: $t_{sfic} = 21$ seconds.

Task a3) The *Fuzzy Toolbox* does not support singletons, so this task is omitted. Direct programming in MATLAB would be possible but inconvenient.

Task b1) The result of the simulation run is shown in figure 5 (computing time $t_{fic} = 22$ seconds).

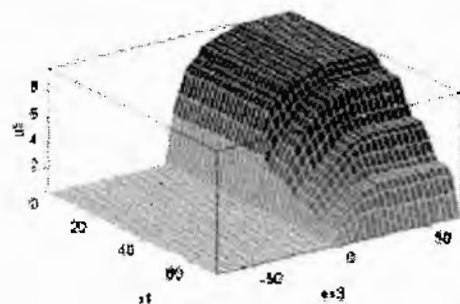


Fig. 4: Surface generated with the FIS Editor

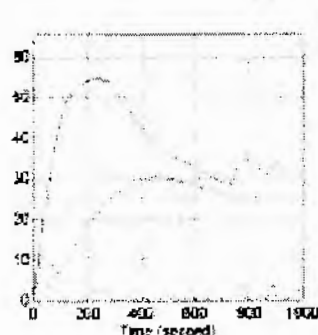


Fig. 5: Result of the run with FC1

Task b2) see Task a3)

Task c1) The *Fuzzy Toolbox* does not provide the necessary features for describing weighted rules.

Task c2) An outstanding feature of the *Fuzzy Toolbox* are the graphical capabilities of the *FIS Editor*. The *Rule Viewer* can display the whole fuzzy inference process based on the interference diagram. The *Surface Viewer* helps to visualise the output of the system versus any one or two inputs to the system. Furthermore, the system is capable of working either with *Mamdani* or *Sugeno*-style inferences. In addition to the polygonal shapes for defining membership functions used in this example a lot of continuous shapes for more specific applications are available. *ANFIS*, the *Adaptive Neuro Fuzzy Inference System*, enlarges the features of the toolbox to learning systems, tuned with a back-propagation algorithm.

Johannes W. Goldynia, Institute for Electrical Control Engineering, Vienna University of Technology, Guss-hausstr. 27-29/375, A-1040 Vienna, Tel. +43-1-58801 5220, Fax +43-1-5058907, Email: goldynia@iert.tuwien.ac.at

Comparison 9 - MATRIX_x / SystemBuild

SystemBuild is an interactive, graphical tool for building models of nonlinear systems. Continuous, discrete, multi-rate, enabled and triggered systems can be modeled in a hierarchical manner. The SystemBuild simulation environment generates information on the response and performance of the model using a wide range of integration algorithms.

SystemBuild belongs to the MATRIX_x Product Family which provides tools able to deal with all the steps of a project development: analysis, modeling and simulation, automatic code generation, hardware integration and test, automatic documentation generation.

Model description: The model is composed of two parts (figure 1): the plant modeled with elementary blocks of the predefined library and the Controller using the Fuzzy Module. This module is a specific block of SystemBuild in which the user can design its fuzzy controller via an editor.

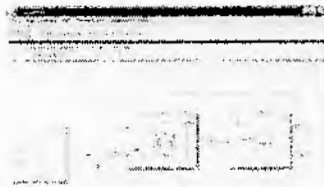


Figure 1

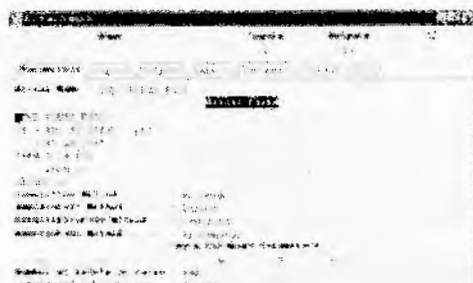


Figure 2

Results task a): (a1) The Fuzzy dialog box (figure 2) acts as a front end interface to the fuzzy logic system. The dialog's principal uses are selecting algorithms (connection, implication, defuzzification, aggregation), parameters, rules and data editing. Standard fuzzy algorithms are implemented in the Fuzzy block.

(a2) and (a3) The visualization of the FC1 surface (figure 3) is realized using an Xmath function. The computation time (PC Pentium with 120 MHz) is: $ta_{fc1} = 104$ seconds. The fuzzy block does not support singletons for the definition of membership functions so the emulation used for the singletons was a very narrow tri-

angular function. The FC2 surface is shown on figure 4. computation time $ta_{fc2} = 100$ seconds, ratio = 1.04.

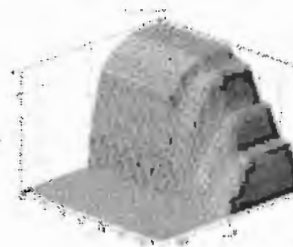


Figure 3

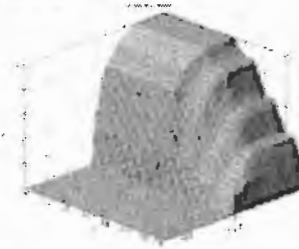
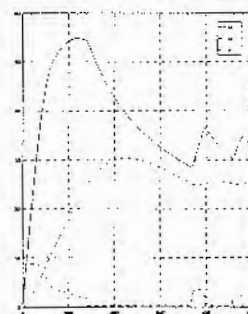
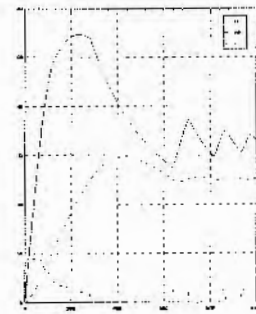


Figure 4

Task b : (b1) and (b2): Results of the simulation runs are shown figure 5. Computation time: $tb_{fc1} = 1.42$ second, $tb_{fc2} = 1.39$ second, ratio = 1.02.



Simulation run for FC1



Simulation run for FC2

Figure 5

Task c: (c1) FAM interference: the weight is specified using the keyword "weight" in the rule definition. The computation time is $tc_{fc3} = 1.26$ second.

(c2) The other features for this fuzzy block are:

- Specification of User Defined method for connection, implication, defuzzification and aggregation.
- Choice of the optimization preference: memory, speed or compromise. All the simulation runs above were done using speed optimisation. The computation time for **task (b1)** would be 26 seconds for memory optimization and 4 seconds for compromise, for **task (b2)** 25.4 seconds, 3.9 seconds.

Sébastien Cantini, ISI France, BP 274, F-78147 Vélizy Cedex, France, Tel: +33 1 34 58 26 26, email: scantini@isi.fr, WWW: <http://www.isi.com/>

Comparison 9 – ACSL Fuzzy Control Programmed at Model Level

ACSL is a classical continuous CSSL-type simulator. Models may be defined by means of equations in **ACSL Model** or graphically by means of blocks in the **Graphic Modeller**. ACSL models are compiled, using FORTRAN. Simple experiments may be done in ACSL's runtime interpreter, experiments of any kind may be done in **AMATH**, an environment of matrix language type. Although ACSL offers no direct support for fuzzy systems, any type of FORTRAN code may be embedded, thus providing the means for programming a fuzzy controller – mainly table lookups and matrix operations. For this solution ACSL 11.4.1 / AMATH 1.2 was used.

Model Description: The model equations (ODEs) in the **derivative section** in the following (abbreviated) ACSL model describe the continuous plant. The sampled data control is implemented in the **discrete section controller**, sampling data each t_s seconds. In this discrete section any kind of discrete control may be implemented – in this case a discrete fuzzy control. First fuzzification is done by evaluating the table functions $n1 \times 1, \dots$ describing the membership functions (defined in the **initial section**). Interference is simply programmed in nested loops, which determine min or max, resp. using the rules (defined in a matrix **rule**). Defuzzification is done by explicit programming of the centre of gravity in a loop.

```

initial
  constant x10=0, x20=0, tend=1000, p=1.5, .
  constant rule= 9, 8, 8, 5, 1, . . . . .
  table n1x1, 1, 4 / 0., 35., 40.0, 70.0, .
  table p1x1, 1, 6 / 0., 35., . . . . .
derivative
  if (x2 .lt. 16) then r = r1 else r = r2
  ax2 = exp(0.43*dlog(abs(x2)))
  f = 0.06624*v1*sqrt(abs(x1-x2))*sign(x1-x2)
  x1= integ ( 0.067 * u - f, x10)
  x2= integ ( f - 0.0605*r*v2*ax2, x20)
  ex2 = x2s - x2; end
discrete controller
  interval ts = 1.
  fx1(1)=n1x1(x1); fx1(2)=p1x1(x1); . . .
do label2 i=1,5
do label2 j=1,5
  fu(rule(i,j))=max(fu(rule(i,j)),
    min(fex2(6-i),fx1(j)))
label2..continue
do label3 i=1,8
sx=(fu(i)+0.25*fu(i+1))/(fu(i)+fu(i+1))
off=(i-1)*1.25
iy = iy + fu(i)*(sx-(sx*sx)/2.)+fu(i+1)
  *((1.25-sx)-(1.25-sx)*(1.25-sx)/2.)
ixy = ixy + fu(i)*(sx*off+sx*sx* . . . . .
label3..continue
u=ixy/iy; end;
  
```

Fuzzification
Interference
Defuzzification

This kind of implementation makes use of some ACSL operators, but it's mainly FORTRAN code, and in defuzzification it is depending on the type of membership functions (triangle membership function for FC1) because of

the analytical computation of the centre of gravity. In case of singletons (FC2) the code becomes much easier:

```

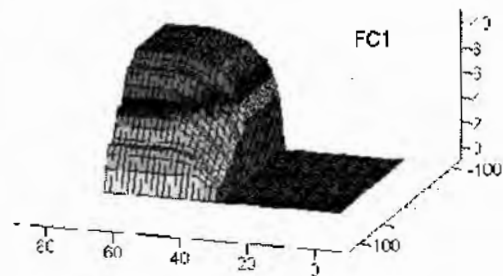
do label3 i=1,9
  u=u+(i-1)*1.25*fu(i); sfu=sfu+fu(i)
label3..continue
u=u/sfu
  
```

Weighting rules (FC3) are implemented easily by multiplying in the interference with a weighting matrix:

```
fu(rule(i,j))=max(fu(. . . . .)*weight(i,j))
```

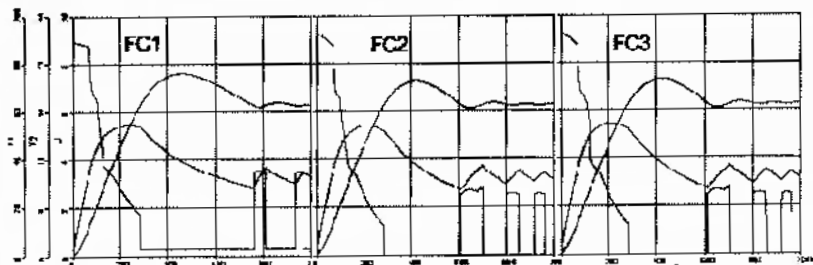
It is clear, that this kind of "manual" implementation is not very comfortable, but i) it is a very slim code without any overhead, ii) it is very fast, and iii) singletons and other discontinuous membership functions can be used without problems.

Results. *Task a)* tests the performance of the implementation by computing and visualising the 3-dimensional surface of the fuzzy controller FC1 and FC2, resp. on a 41×41 grid. The surface is plotted in AMATH (see fig.) by a standard command, the data are computed in a "dummy" ACSL model consisting only of initial and discrete section, evaluating in a loop the discrete section. Times for computation on a Pentium 166, 32 MB, are $t_{FC1}=0.98$ and $t_{FC2}=0.91$. As these times differ due to inaccurate measurement etc., the ratio 1.07 is not really significant.



Task b, c. Simulation of the whole system (task h) with FC1 and FC2 (see fig. at bottom) result in $t_{FC1}=2.68$ and $t_{FC2}=2.61$, resp. (ratio ~ 1.02). Weighting rules neither change the results nor the computation times essentially ($t_{FC3}=2.68$).

F. Breiteneker, ARGESIM/SIMTECH, Dept. Simulation Techniques, TU Vienna, email: Felix.Breiteneker@tuwien.ac.at



Comparison 9 – ACSL

Fuzzy Control Programmed at Environment Level

ACSL is a classical continuous CSSL-type simulator. Models may be defined by equations in ACSL Model (or graphically in the **Graphic Modeller**) and are compiled (on a FORTRAN base). ACSL's power was increased in 1996 by **AMATH**, an interpretative environment of matrix language-type with graphical features. With AMATH e.g. complex experiment control and hybrid modelling can be implemented comfortably. ACSL Model and AMATH offer no direct support for fuzzy control. But as AMATH is a matrix-type environment, fuzzy control (fuzzification, interference, and defuzzification) can be implemented simply (as well as a toolbox could be implemented easily).

Model Description: The model under consideration is a sampled data system. Each t_s seconds data are sampled, and the new control u is calculated by means of fuzzy control. In a classical implementation approach, the data sampling interrupts the "continuous" solution of the system governing ODEs at model level. The approach in this solution (with ACSL Model 11.4.1, AMATH 1.2) controls data sampling with calculation of the fuzzy control in the environment level (calling the model in a loop (start), as shown in the following "main" m-file and ACSL model file:

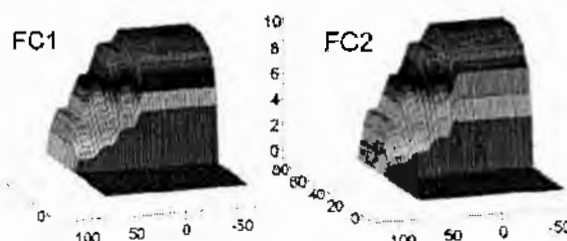
```
rule=[9 8 8 5 1; 8 7 6 4 1; ..... AMATH main
xl=zeros(1,1001);x2=zeros(1,1001);...
load(@file='c9amath.prx',@format='model')
xl(1)=0;x2(1)=0;ex2(1)=25; TEND=1;
for t=1:1000
    fx=fuzzify([xl(t) ex2(t)]);
    fu=interf(fx, rule, weight);
    u(t)=defuzzy(fu, 1);
    X10=xl(t); X20=x2(t); U=u(t);
    !start
    xl(t+1)=X1;x2(t+1)=X2;ex2(t+1)=25-X2;
end
model c9amath ACSL model file
if ( x2.lt.16.0 ) then r=1.2 else r=1
ax2 = exp(0.43*dlog(abs(x2)))
f = 0.06624*v1*sqrt(abs(x1-x2))*sign(x1-x2)
x1 = integ (0.067 * u - f, x10)
x2 = integ (f - 0.0605* r * v2 * ax2, x20)
end
```

The m-file `fuzzify` evaluates the input membership functions, `interf` performs the interference by matrix manipulations (in case of FC3 simply by multiplying with the matrix weight), and `defuzzy` calculates the control u by means of the center of gravity (in case of FC1 by calculating the centre of gravity by the analytical formula, and in case of FC2 by a simple weighted sum):

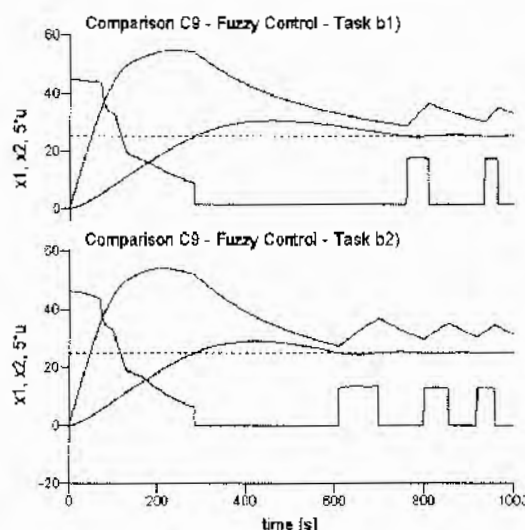
```
function fx=fuzzify(x) AMATH Fuzzification
nlx1=[0 35 40 70; 1 1 0 0];
plx1=[0 35 40 45 50 70; .....
fx(1,1)=interp1(nlx1(1,:),nlx1(2,:),x(1));
fx(1,2)=interp1(plx1(1,:),plx1(2,:),x(1));
return; end
```

```
function fu=interf(fx,rule,weight) AMATH Interference
fu=zeros(1,9);
for i=1:5; for j=1:5
    fu(rule(i,j))=max(fu(rule(i,j)),
        min(fx(1,j),fx(2,6-i))*weight(i,j));
end; end; return; end;
```

Results. Task a) A simple m-file calls in a nested loop 41 x 41 times the calculation of the fuzzy control, and a standard command plots the fuzzy control surface: Computation times on a Pentium 166, 32 MB, are $t_{FC1}=680$ and $t_{FC2}=677$ sec., resp. Due to the interpretative calculation these times are relatively high; they are mainly consumed in loops and function evaluations, so that difference between FC1 and FC2 does not really matter (giving ratio ~ 1).



Task b, c). Simulation of the whole system (using the main m-file) with FC1, FC2 and FC3 result in $t_{FC1}=420$ s, $t_{FC2}=410$, and $t_{FC3}=411$ sec., resp.; As results with FC2 and FC3 do not differ significantly, only figures for FC1 and FC2 are given below.



Remark: The sampled data nature and control of the overall system raises the question how important the continuous modelling of the plant really is. For investigation, the "continuous" plant model in ACSL (integrated with stepsize $h=0.001$, RK4) was replaced by a simple Euler step (stepsize $h=1$) in the AMATH m-file. As result, the relative difference in x_2 lies in a bandwidth of a relative error of 4×10^{-5} . Simulation times do not change essentially !!

F. Breiteneker, M. Lingl, ARGESIM/SIMTECH, Dept. Simulation Technique, TU Vienna, email: Felix.Breiteneker@tuwien.ac.at

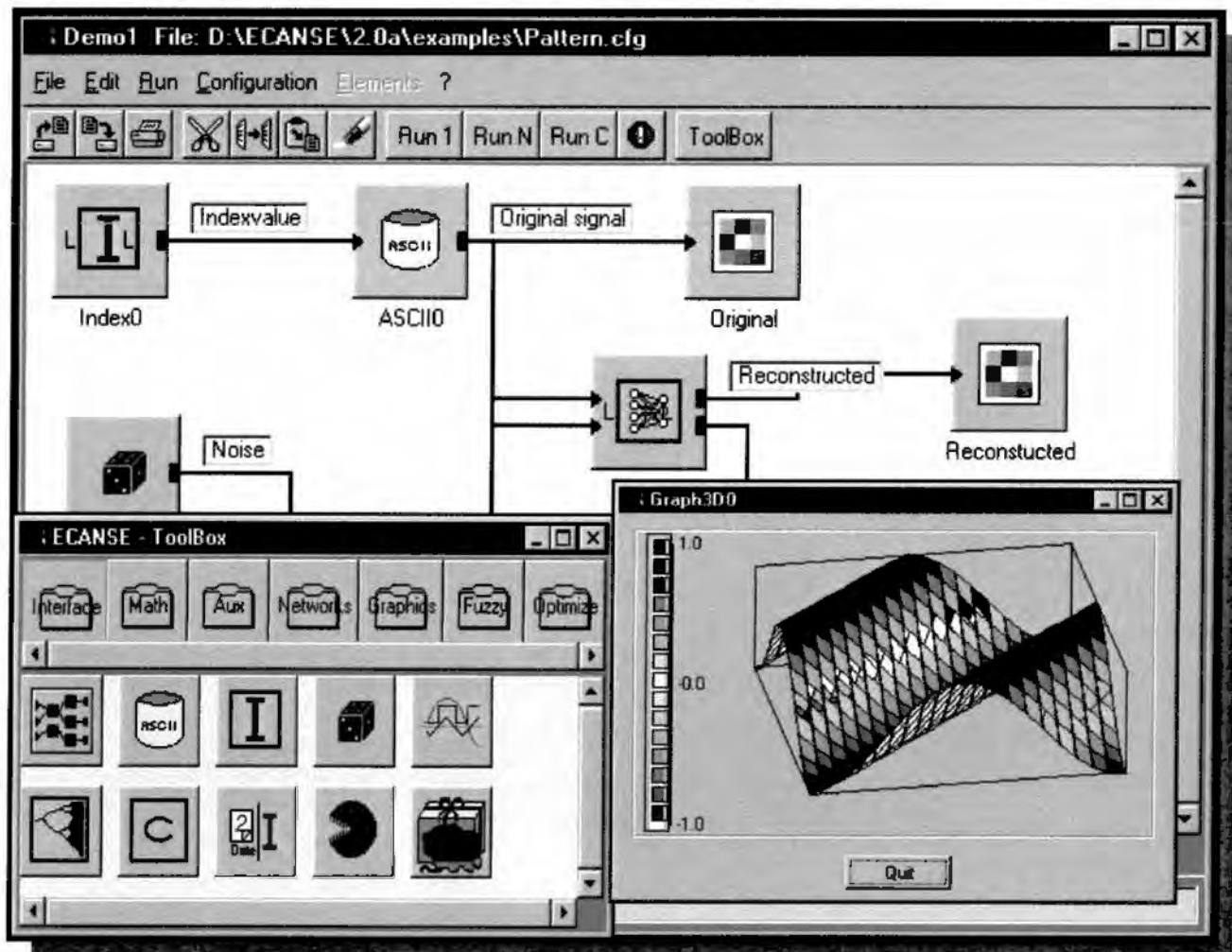
ANHANG

SIEMENS



ECANSE

Soft Computing Studio



Networked Thinking.
With Siemens
Program and System Engineering





ECANSE Where to apply

ECANSE® helps to solve your problems and to save your money in application domains like

- **Software Engineering**
e.g.: generic framework, component integration platform
- **Electrical Engineering:**
e.g.: power control systems, load forecast, hydro dam monitoring, filtering in AGC
- **Finance:**
e.g.: portfolio optimization, data segmentation, credit worthiness prediction
- **Space Engineering:**
e.g.: satellite image analysis, pollution monitoring system
- **Environmental Preservation:**
e.g.: ground level ozone prognosis
- **Chemical Engineering:**
e.g.: process control
- **Signal Processing:**
e.g.: neural filtering, mutual redundancy
- **Mechanical Engineering:**
e.g.: classification systems
- **Biology:**
e.g.: forest condition analysis
- **Marketing:**
e.g.: sales forecast, data segmentation
- and a big variety of **research & development** applications



ECANSE Easy Handling

Just one mouse click and you have realized a neural network with several thousands of internal connections. You may combine or compare in your application such challenging technologies as

- neural networks,
- fuzzy logic,
- genetic algorithms,
- expert systems,
- chaos theoretical methods,

and for some of our customers most important

- your existing programs

and of course

- your new ideas.

ECANSE is an open platform.

ECANSE is a testground for your ideas.

ECANSE is a graphical programming tool.

ECANSE supports you to realize your ideas faster and with less probability of errors due to the simplicity and clear structure of your programs which are composed by simple mouse clicks and drag-and-drop of icons.

ECANSE is a universal tool. Our costumers use ECANSE as a design tool, simulation tool, process control tool, data analysis tool, forecast tool, rapid prototyping tool, visualization tool, or as an platform to compare different approaches.



ECANSE Ready-to-Run Objects

ECANSE and all its objects are internally realized in the object-oriented C++ code. This guarantees efficient and fast execution of algorithms, fulfilling the ISO 9001-standards. Beside numerous basic objects ECANSE offers:

- MLP multilayer perceptrons and backpropagation learning
- Kohonen networks
- Hopfield networks
- fuzzy logic
- filter, statistics
- genetic algorithms
- matrices and optimization library SCRLapack, Blas
- interface to arbitrarily formatted ASCII-files
- ODBC/SQL interface to databases
- dynamic visualization and monitoring objects
- sensitivity analysis
- chaos theoretical methods
- script objects enable control or automation of ECANSE (e.g. for optimization)
- sub-configurations: arbitrarily many levels of abstraction

By double-clicking on the object icon, a dialogue box opens where you may change parameters for your purpose.

Every parameter or variable may be assigned to be input or output of the object.

Software Reuse
Generic Framework
Visual Programming
Component Integration Platform



System
Requirements

- IBM compatible PC platforms
Windows NT, Windows 95
- 32 MB main memory,
60 MB harddisk space
- MS Visual C++ Compiler:
optional for adding user
specific objects/code to
*ECANSE programmer's
version* (extendible)

ECANSE
Consulting Center

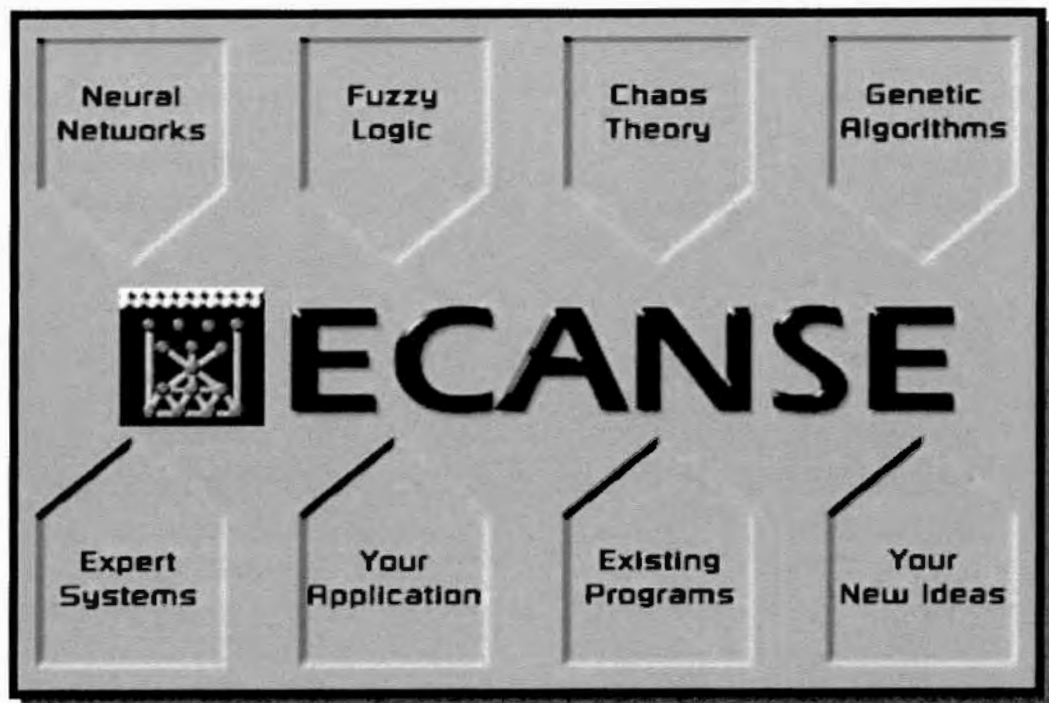
Either you may decide if you enjoy
to work with the comfortable graph-
ical user interface of ECANSE or
you want our ECANSE specialists
to work out your customized solu-
tion.

We offer professional services in

- consulting
- feasibility studies
- complete project development
- hotline support

Cooperations

- ÖFAI, Austrian Research Insti-
tute for Artificial Intelligence,
Vienna: Statistical and
Knowledge Based Methods in
ECANSE
- University of Technology,
Vienna, Austria: Dept. of
Communication and Radio
Frequency Engineering:
Chaos- and Information Theo-
retical Methods in ECANSE
- Siemens Corporate Research,
Princeton, USA: Matrices and
Optimization Library
(LAPACK, BLAS)



ECANSE® Soft Computing Studio is an open environment



Intro to Neural Networks

Artificial Neural Networks are designed similarly to the human brain. A neural network consists of a large number of primitive units, called neurons (e.g. each of these neurons is just capable to add all inputs and to nonlinearly propagate the result). The information is stored in form of the interconnection weights between these neurons.

A neural network can learn complex relationships from given examples (training phase) and can afterwards apply its "knowledge" to new input data (generalization).

Main application areas: pattern recognition, speech recognition, classification problems, forecast applications, nonlinear modelling ...



Intro to Fuzzy Logic

Fuzzy Logic is a method to deal with fuzzy terms like "warm" or "cold" instead of crisp values like "27.3°C" or "15.1°C". A person without computer experience is able to state rules by using these fuzzy terms. The set of rules is called fuzzy rule base. One of the main advantages of fuzzy logic is that other non-experts can read and understand the fuzzy rules and can easily change some fuzzy rules in order to improve the system behavior.

Main application areas: automatic control of complex systems (e.g.: anti-blocking system for cars, autofocus for cameras), parameter tuning of neural networks ("Neuro-Fuzzy"), fuzzy classification, fuzzy scheduling ...



Intro to Genetic Algorithms

Genetic Algorithms are an optimization method. The basics of Darwin's evolution theory ("survival of the fittest") are adapted for computer algorithms.

In a first step a so called "population" of random parameters is generated. For each so called "individuals" of this population the system performance is tested. The best of these individuals are selected for reproduction (mutation, recombination, copying) generating a second generation of individuals. After some generations an optimal solution is evolved.

Main application areas: optimization in general, parameter optimization, optimization of computer programs (genetic programming) ...

Contact : **Siemens AG Austria**
 PSE NLT, ECANSE® Sales
 Gudrunstrasse 11
 A-1100 Vienna

Phone: (+43 1) 1707 - 45985
FAX: (+43 1) 1707 - 56256
email: ecanse@siemens.at
<http://www.siemens.at/pse/ecanse>

Issued by:

Programm- und Systementwicklung
A-1100 Vienna/Austria
Gudrunstraße 11

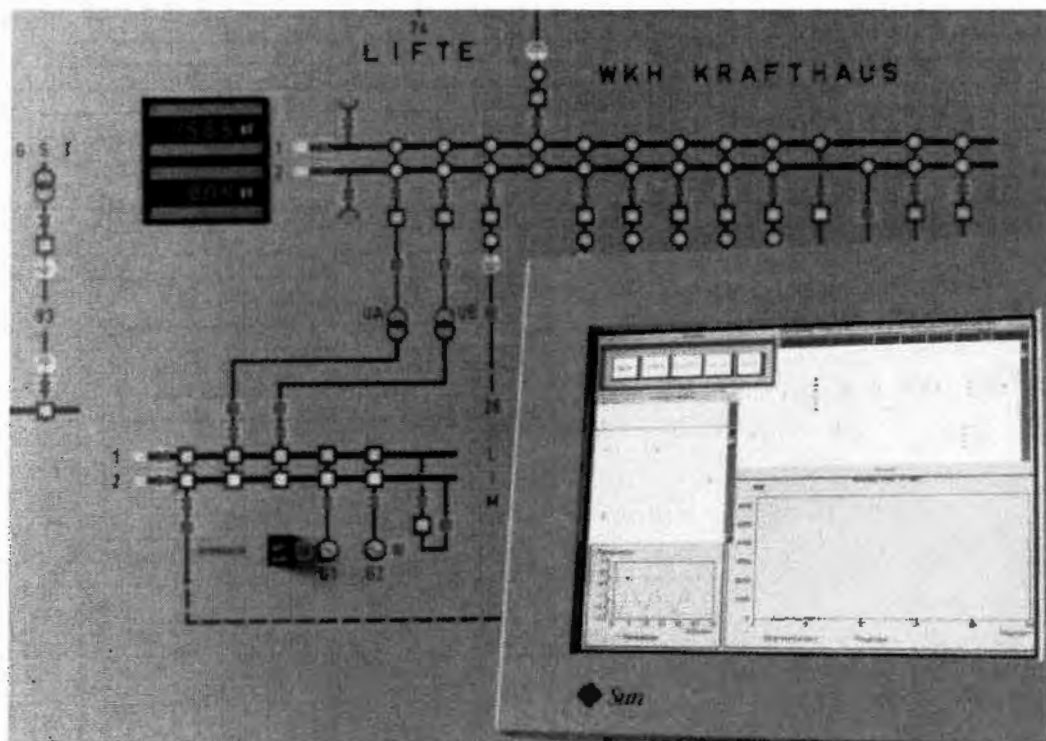
☎ +431 17 07/463 38

FAX +431 17 07/563 94

ECANSE

Lastprognose mit Neuronalen Netzen

Projektinformation



Die Betriebskosten von Energieerzeugungsanlagen sowie auch die Produktionskosten elektrischer Energie werden durch die Genauigkeit der Lastvorhersage signifikant beeinflusst.

Die Lastprognose mit neuronalen Netzen auf Basis ECANSE® zeichnet sich durch eine hohe Prognosegenauigkeit aus. Der durchschnittliche absolute Fehler liegt zwischen 1 und 2 %, d.h. durch den Einsatz dieser Methode kann die Prognosegenauigkeit im Vergleich zu klassischen Methoden (multiple Regression, Kalman Filter, ARIMA) verdoppelt werden. Die Prognose inkludiert spezielle Tage (z.B. Feiertage, Urlaubszeit) und ist im Stunden-, Halbstunden und Viertelstundenraster möglich.

Stichworte

Vorraussage des Energieverbrauches,
Optimierung des Kraftwerkeinsatzes,
Genauigkeitserhöhung der Prognose,
Kostensenkung.

Bitte wenden!

Vernetztes Denken.
Mit Siemens Programm- und
Systementwicklung



Lastprognose mit Neuronalen Netzen

Einsatzgebiet:

Damit Energieversorgungsunternehmen (EVU's) den Einsatz ihrer Kraftwerke optimieren können, sind sie an einer möglichst genauen Abschätzungen des in naher Zukunft zu erwartenden Energieverbrauchs einer Region interessiert.

Ökonomische Faktoren (Langzeittrends), zeitlich bedingte Einflüsse (Feiertage, Urlaube), Wetter und spezielle Ereignisse (Streik, TV-Programm) beeinflussen den regionalen Energieverbrauch. Lastprognosesysteme berücksichtigen diese Trends und ermöglichen dadurch den EVU's ein effektiveres Management der Kraftwerke. So ist es beispielsweise wichtig, ein Wärmekraftwerk rechtzeitig vor dem eigentlichen Strombedarf hochzufahren.

Zielsetzung:

Ziel ist es, durch die Verwendung neuer Technologien, eine Steigerung der Prognosegenauigkeit zu erreichen. Ein System von mehreren Neuronalen Netzen wird angewendet, um eine kurzfristige Lastprognose zu erstellen. Der Prognosezeitraum erstreckt sich bis zu 3 Wochen.

Vorteile durch ECANSE:

Bei den bisherigen praktischen Umsetzungen konnte mit ECANSE die Prognosegenauigkeit im Vergleich zu klassischen Methoden (multiple Regression, Kalman Filter, ARIMA) verdoppelt werden. Die Genauigkeit der Lastvorhersage hat einen signifikanten Einfluß auf die Betriebskosten von Energieerzeugungsanlagen und auch die Produktionskosten elektrischer Energie. Prognosefehler verursachen im Falle einer zu hohen Vorhersage des Energieverbrauchs Kosten beim ungerechtfertigten Hochfahren von Energieerzeugungssystemen, im Falle einer zu niedrigen Vorhersage wird die Reservehaltung erschwert.

- Hohe Prognosegenauigkeit (durchschnittlicher absoluter Fehler nur zwischen 1 und 2%),
- schnelle Anpassungsfähigkeit an veränderte Situationen (z.B. Sommerzeitwechsel),
- Lastprognose auch für spezielle Tage (z.B. Feiertage, Urlaube),
- Lastprognose z.B. im Stunden-, Halbstunden- und Viertelstundenraster möglich,
- Prognoselaufzeiten ≤ 1 sec.

Realisierung:

Die Anwendung erfolgt in 2 Phasen:

1. In der Lernphase wird das ECANSE-Prognosesystem auf in die Vergangenheit zurückreichende Eingangsdaten (Tag, Ferien, Ferientage, Temperatur, Lichteinfall, elektrischer Energieverbrauch) trainiert.
2. In der Prognosephase wertet ECANSE den korrespondierenden Tagestyp (Wochen-, Feiertag oder Wochenende) aus, nimmt eine Korrektur aufgrund Trends im Energieverbrauch und Wetter vor und erstellt eine Vorhersage.

Ansprechpartner: ECANSE®-Team
Siemens AG Österreich
PSE NLT2
Gudrunstraße 11
A-1100 Wien, Österreich

Tel.: (+43-1) 1707-45985
Fax: (+43-1) 1707-56256

email: ecanse@siemens.at
<http://www.siemens.at/pse/ecanse/ecanse.html>

SIEMENS

PowerProphet

Financial Forecasting Software



What is PowerProphet ?

PowerProphet® is a **financial forecasting software** that helps you determine in what direction the financial markets will move.

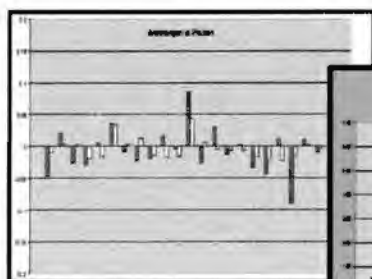
PowerProphet works as an **Excel-Add-In**, with an easy to handle user interface.

PowerProphet makes use of the newest cutting-edge technologies like

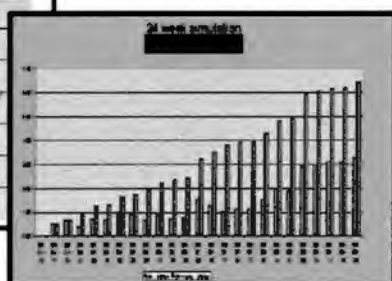
- **Neural Networks and**
- **Fuzzy Logic**

Neural networks are used to forecast the direction of share indices and exchange rates. Fuzzy rules are used to model expert knowledge in simple rules.

PowerProphet works with **Reuters-data** from Reuters 2000/3000 Terminals.



Percentage change



Comparison with benchmark

PowerProphet Easy Handling

PowerProphet offers an **easy to use interface**: standard Excel Sheets are used as user interface. By making one mouse click you get the forecast of the major indices one day, one week or one month ahead.

PowerProphet offers **two kind of forecasts**: standard and user-defined forecast.

For the standard version we do **data-mining**, which searches for hidden relationships in historical financial data (**genetic algorithm**).

No neural network, fuzzy logic and genetic algorithm expertise is needed. We take care of building the market models for you.

So all you have to do is letting PowerProphet forecast the direction of the index.

PowerProphet Application Fields

In the current release the following share indices can be forecasted:

- **Dow Jones Industrial**
- **German DAX**
- **FTSE 100**
- **Nikkei 225**

Besides the following exchange rates can be forecasted:

- **Dem/\$**
- **\$/GBP**
- **Yen/\$**

PowerProphet can also be used for generating and optimizing fuzzy rules.

You can generate a model based on macro- or micro-economical data easily, because the know-how can be directly represented in rules: e.g.:

IF (inflation increases) AND (utility index decreases) THEN Exchange Rate decreases
IF (yield increases) THEN FX increases

To obtain the best results, some parameters of the fuzzy-system (certainty, membership) can be optimized through a genetic algorithm.

Share Index Forecasts

Exchange Rate Forecasts

Rule Generation



Portfolio Optimization

PowerProphet offers in addition to the forecasts portfolio optimization. During portfolio optimization an optimal trade off between risk and return is searched.

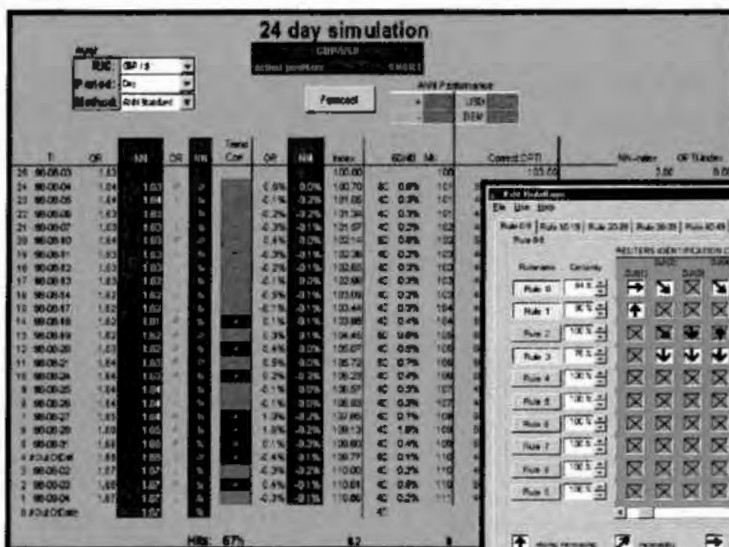
Traditional methods use quadratic optimization to find good risk-return tradeoffs. We offer better solutions by applying advanced techniques such as genetic algorithms.

System Requirements

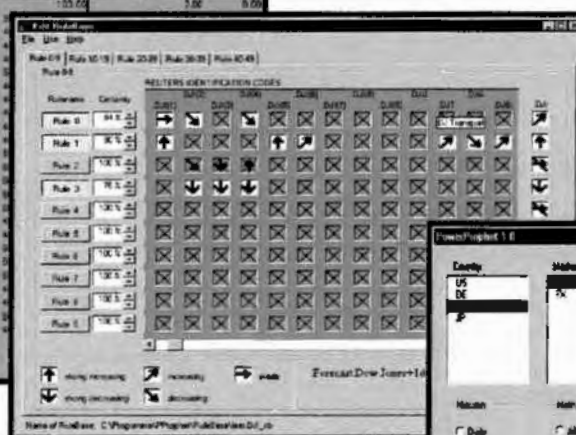
- Intel Pentium 133 or faster with 32 MB main memory and 60 MB harddisk space
- Windows NT 4.0 SP3 or higher
- Excel 97 SR1 or higher
- Reuters 2000/3000 Terminal

Cooperations with external partners

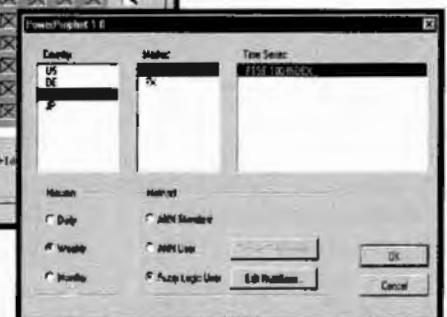
- ÖFAI, Austrian Research Institute for Artificial Intelligence, Vienna: Statistical and Knowledge Based Methods in ECANSE
- University of Technology, Vienna, Austria: Dept. of Communication and Radio Frequency Engineering: Chaos- and Information Theoretical Methods in ECANSE
- Siemens Corporate Research, Princeton, USA



24 day GBP/\$ simulation



Fuzzy rule editor



Function wizard



Intro to Neural Networks

Artificial Neural Networks are designed similarly to the human brain. A neural network consists of a large number of primitive units, called neurons (e.g. each of these neurons is just capable to add all inputs and to nonlinearly propagate the result). The information is stored in form of the interconnection weights between these neurons.

A neural network can learn complex relationships from given examples (training phase) and can afterwards apply its "knowledge" to new input data (generalization).

Main application areas: pattern recognition, speech recognition, classification problems, forecast applications, nonlinear modelling ...



Intro to Fuzzy Logic

Fuzzy Logic is a method to deal with fuzzy terms like "warm" or "cold" instead of crisp values like "27.3°C" or "15.1°C". A person without computer experience is able to state rules by using these fuzzy terms. The set of rules is called fuzzy rule base. One of the main advantages of fuzzy logic is that other non-experts can read and understand the fuzzy rules and can easily change some fuzzy rules in order to improve the system behavior.

Main application areas: automatic control of complex systems (e.g.: anti-blocking system for cars, autofocus for cameras), parameter tuning of neural networks ("Neuro-Fuzzy"), fuzzy classification, fuzzy scheduling ...



Intro to Genetic Algorithms

Genetic Algorithms are an optimization method. The basics of Darwin's evolution theory ("survival of the fittest") are adapted for computer algorithms.

In a first step a so called "population" of random parameters is generated. For each so called "individuals" of this population the system performance is tested. The best of these individuals are selected for reproduction (mutation, recombination, copying) generating a second generation of individuals. After some generations an optimal solution is evolved.

Main application areas: optimization in general, parameter optimization, optimization of computer programs (genetic programming) ...

Issued by:

Program and System Engineering

Power System Control

Gudrunstraße 11

A-1100 Vienna, Austria

Phone: 0043-1-1707-46360

FAX: 0043-1-1707-56256

email: power.prophet@siemens.at

<http://www.siemens.at/pse/PProphet>

©1998 by Siemens AG

All rights reserved

Soft Computing in Modellbildung und Simulation - ECANSE, TU-Wien, 03.12.1998, Teilnehmerliste

Dipl.-Ing.	Almeder	Christian	.		Vogelsanggasse 37/11	A-1050	Wien	calmeder@osiris.tuwien.ac.at
Dipl.-Ing. Dr.	Bayerl	Manfred	Büro für Technischen Umweltschutz		Morizgasse 4	A-1060	Wien	bftu+bo@magnet.at
Ao. Prof. Dr.	Bischof	Horst	Technische Universität Wien	Inst. f. Rechnergestützte Automation	Treittlstraße 1	A-1040	Wien	horst.bischof@tuwien.ac.at
Dipl.-Ing.	Bracio	Boris R.	Technische Universität Clausthal-Zellerfeld	Inst. f. Informatik	Erzstraße 1	D-38678	Clausthal-Zellerfeld	bbracio@osiris.tuwien.ac.at
Prof. Dipl.-Ing. Dr.	Breitenecker	Felix	.		Wiedner Hauptstraße 8-10	A-1040	Wien	Felix.Breitenecker@tuwien.ac.at
	Garn	Wolfgang	.		Blumenweg 30	A-2201	Gerasdorf	
Dipl.-Ing.	Görtler	Gregor	Technische Universität Wien	Inst. f. Maschinen- und Prozeßautomatisierung	Gußhausstraße 25-29	A-1040	Wien	gregor.goertler@tuwien.ac.at
Dipl.-Ing., MSc.	Hajrizi	Edmond	Technische Universität Wien	Inst. f. Fertigungstechnik	Karlsplatz 13	A-1040	Wien	hajrizi@mail.ift.tuwien.ac.at
	Hammermüller	Klaus	.					
Dr.	Hertl	Stefan	.					
	Holzer	Mario	.		Gymnasiumstraße 85/454	A-1190	Wien	
Dr	Katzenberger	Wolfgang E.	Wirtschaftsuniversität Wien		Weibelstrasse 4	A-1228	Wien	
Dipl.-Ing.	Kiss	Claus	Technische Universität Wien	ARGESIM / SIMTECH - Abt. Simulationstechnik	Wiedner Hauptstraße 8-10	A-1040	Wien	jeanluc@ws3.atv.tuwien.ac.at
Dr.	Kleemayer	Karl	Universität für Bodenkultur	Inst. f. Wildbach- und Lawinenschutz	Peter Jordanstraße 82	A-1190	Wien	klee@edv1.boku.ac.at
Dipl.-Ing.	Klug	Markus	.		Erlachgasse 82/5	A-1100	Wien	mklug@osiris.tuwien.ac.at
	Kozek	.	Technische Universität Wien	Inst. f. Maschinen- und Prozeßautomatisierung	Gußhausstraße 25-29	A-1040	Wien	kozek@impa.tuwien.ac.at
	Kubin	Gernot	Technische Universität Wien	Inst. f. Nachrichtentechnik	Gußhausstraße 25-29	A-1040	Wien	

Soft Computing in Modellbildung und Simulation - ECANSE, TU-Wien, 03.12.1998, Teilnehmerliste

Dr.	Kührer	Martin	Siemens AG Wien	Abt. PSE NLT SV	Gudrunstr. 11	A-1100	Wien Martin.Kuehrer@siemens.at
Dr.	Lieven	Karl	MIT GmbH	Management Intelligenter Technologien GmbH	Promenade 9	D-52076	Aachen kl@mitgmbh.de
Dipl.-Ing.	Lingl	Martin	.		Favoritenstraße 49-53/1/5	A-1100	Wien mlingl@osiris.tuwien.ac.at
	Luckinger	Gustav	Semperit		Wienersdorferstr. 20-24	A-2514	Traiskirchen
	Lutter	Clemens	.				
Dipl.-Ing.	Mladenka	Alice	Siemens AG Wien	Abt. PSE NLT SV	Gudrunstr. 11	A-1100	Wien Alice.Mladenka@siemens.at
D.I. Dr.	Müller	Herbert	Technische Universität Wien	Inst. f. Elektrische Anlagen	Gußhausstraße 25-29	A-1040	Wien
Dr.	Noisser	Robert	Technische Universität Wien	Inst. f. Elektrische Regelungstechnik	Gußhausstraße 25-29	A-1040	Wien robert@iert.tuwien.ac.at
	Rahmi	Shabnam	Technische Universität Wien	ARGESIM / SIMTECH - Abt. Simulationstechnik	Wiedner Hauptstraße 8-10	A-1040	Wien
	Redlein	Alexander	Technische Universität Wien	Inst. f. Automation	Treitlstraße 1	A-1040	Wien
OR Dr.	Reich	Gregor	Universität Wien	Inst. f. Analytische Chemie	Währinger Straße 38	A-1090	Wien
	Scheikl	Joachim	.		karolinengasse 13/10	A-1040	Wien joxg@osiris.tuwien.ac.at
	Seits	Klaus	Technische Universität Wien	ARGESIM / SIMTECH - Abt. Simulationstechnik	Bründlberggasse 23	A-2125	Neubau kseits@osiris.tuwien.ac.at
Dr.	Sturm	Monika	Siemens AG Wien	Abt. PSE NLT SV	Gudrunstr. 11	A-1100	Wien Monika.Sturm@siemens.at
Univ.- Doz. Dr.	Suda	Martin	Austrian Research Centers	Seibersdorf GmbH.	Forschungszentrum	A-2444	Seibersdorf martin.suda@arcs.ac.at
	Wasinger	Irmgard	Atominstitut	Arbeitsgruppe Photobiophysik	Turmburgg. 2/33	A-1060	Wien e8925005@stud1.tuwien.ac.at
	Wassermann	Johann	.		Adolf Loosgasse 12/17/1/4	A-1210	Wien jw@mdmtws3.tuwien.ac.at
	Wassertheurer	Sigi	.		Bräuhäusgasse 42/16	A-1050	Wien
	Wetzer	Michael	.				

Soft Computing in Modellbildung und Simulation - ECANSE, TU-Wien, 03.12.1998, Teilnehmerliste

Wibmer

Michael

.

Lange Gasse 44/12

A-1080

Wien

mwibmer@osiris.tuwien.ac.at

Seminare zu Modellbildung und Simulation

Über die Seminarreihe

Die Intention der schon 1991 begonnenen Reihe der Seminare zu Modellbildung und Simulation ist es, in Theorie und Praxis sowohl verschiedene Lösungsansätze in einzelnen Problemkreisen wie auch spezielle Softwaretools vorzustellen, über deren Einsatzmöglichkeiten zu informieren und Erfahrungen auszutauschen.

Zu den Vorträgen werden schriftliche Unterlagen in Form von **Seminarberichten** bereitgestellt. Pausen mit bereitgestelltem Buffet bzw. Kaffee fördern die Kommunikation und den Erfahrungsaustausch zwischen den Seminarteilnehmern.

Nähere Informationen und eine Bestellmöglichkeit für Berichte bisheriger Seminare finden sich im Web unter

<http://www.argesim.org/seminare>.

Folgende Seminare

S52

EUROPEAN ERUDIT MEETING: SOFT COMPUTING
Fr, 4. Dez. 1998

Das "ERUDIT Meeting" steht in engem Zusammenhang mit dem am Vortag, dem 3.12.1998, stattfindenden Seminar "Soft Computing in Modellbildung und Simulation DataEngine - ENCASE - MATLAB". ERUDIT steht für "European Network in Uncertainty Techniques Developments for Use in Information Technology", welches im Rahmen des EU Programmes "DG III Industry - Esprit Programme as a Network of Excellence" ins Leben gerufen wurde.

Dieses Meeting wird organisatorisch lokal von ARGESIM betreut, die inhaltliche und fachliche Betreuung (Vortrag, Teilnahme, etc.) wird von der Verwaltungsstelle des EU-Programmes in Aachen durchgeführt:

ERUDIT Service Center
c/o ELITE Foundation, Promenade 8, D-52076 Aachen
Tel: +49-241-2408-6969, Fax: +49-241-2408-94582
Email: erudit@mitgmbh.de
WWW: <http://ss-m3.mitgmbh.de/erudit/crudit.html>.

S53

MODELLBILDUNG UND SIMULATION IN DER ABFALLENTSORGUNG
Do, 28. JÄNNER 1999

Für das Sommersemester 1999 sind vier Seminare vorgesehen. Schwerpunkte werden dabei verstärkt für die anwendungsspezifische Präsentation verschiedener Methodologien in Ansätzen der Modellbildung und Simulation gesetzt als in der besprechung verschiedener Merkmale einzelner Softwareprodukte. Damit soll ein Trend vorgezeichnet werden, der sich in Zukunft verstärkt fortsetzen soll. So wurden bereits im Vorjahr gute Erfahrungen mit diesen anwendungsorientierten Seminaren gemacht: Grundlagen und Implementationen für Modellbildung und Simulation eines bestimmten Anwendungskreises werden über Simulatorgrenzen hinaus diskutiert, unter Berücksichtigung verschiedener methodologischer Ansätze.

Auch das gegenwärtige Seminar zielt verstärkt in diese Richtung: "Modellbildung und Simulation in der Abfallentsorgung" stellt Modelle und Implementierungen aus dem Bereich der lokalen und globalen Abwasserreinigung, der Entsorgungsplanung und ähnlichen einschlägigen Problem Bereichen vor. Dabei wird ein Bogen von Themen von Anwendungen für kontinuierlich zu behandelnde Probleme über diskrete Simulationen bis hin zur Planung von Anlagen gespannt. U.a. wird auch eine im Haus speziell zu diesem Zweck entwickelte SIMULINK Bibliothek vorgestellt.

S54

MATLAB SEMINAR
Do, 11. März 1999