

Forschungszentrum Karlsruhe
Technik und Umwelt

Wissenschaftliche Berichte
FZKA 5765

FZK-Seminarreihe
„Aktuelle Forschungs-
gebiete in der Mathematik“
Seminarbeiträge 1995

C. P. Hugelmann, R. Seifert, T. Westermann
Institut für Angewandte Informatik

Mai 1996

FORSCHUNGSZENTRUM KARLSRUHE
Technik und Umwelt

Wissenschaftliche Berichte

FZKA 5765

FZK-Seminarreihe
“Aktuelle Forschungsgebiete in der Mathematik“
Seminarbeiträge 1995

Herausgeber:
C.P. Hugelmann, R. Seifert, T. Westermann *
Institut für Angewandte Informatik

* Fachhochschule Karlsruhe

Forschungszentrum Karlsruhe GmbH, Karlsruhe

1996

**Als Manuskript gedruckt
Für diesen Bericht behalten wir uns alle Rechte vor**

**Forschungszentrum Karlsruhe GmbH
Postfach 3640, 76021 Karlsruhe**

ISSN 0947-8620

FZK-Seminarreihe: Aktuelle Forschungsgebiete in der Mathematik

Seminarbeiträge 1995

Zusammenfassung

1995 wurde im FZK die Seminarreihe über aktuelle Forschungsgebiete in der Mathematik fortgesetzt. Ziel dieser Seminarreihe ist, im FZK anwendungsorientierte, aktuelle Forschungsthemen aus der Mathematik zu präsentieren. Übersichtsvorträge sollen Einblicke in moderne Methoden und Verfahren der Mathematik ermöglichen. Die Vorträge standen daher wieder im engen Zusammenhang mit praxisbezogenen Anwendungen. Organisiert wurde die Seminarreihe von Claus-Peter Hugelmann (FZK, HIK), Rolf Seifert (FZK, IAI) und Thomas Westermann (FH Karlsruhe).

Im vorliegenden Bericht sind die Seminarbeiträge in schriftlicher Form zusammengefaßt.

FZK-Seminar Series on Selected Topics in Mathematics

Seminar reports 1995

Summary

In 1995 the FZK seminars on selected topics in applied mathematics were continued. The goal was to demonstrate the importance of applied mathematics and to present current research areas in mathematics. Survey lectures were to give an insight in modern methods and methodologies. The seminars were organized by Claus-Peter Hugelmann (FZK, HIK), Rolf Seifert (FZK, IAI), and Thomas Westermann (FH Karlsruhe).

This report contains the collection of the seminar papers.

Inhaltsverzeichnis

Vorwort	5
Vorstellung	7
N. Henze (Universität Karlsruhe)	9
Einige Schwierigkeiten im Umgang mit bedingten Wahrscheinlichkeiten	
Vorstellung	21
W. Eppler (Forschungszentrum Karlsruhe)	23
Neuro-Fuzzy-Systeme in Theorie und Anwendung	
Vorstellung	47
D.E. Ulmet (FHT Esslingen)	49
Design mit Bezier-Splinekurven in der CAD/CAM-Entwicklung	
Vorstellung	67
F. Grabmeier (IBM Heidelberg)	69
Computeralgebra-Algorithmen zur Symbolischen Integration	
Vorstellung	115
H. Gemmeke (Forschungszentrum Karlsruhe)	117
Fuzzy-Logik und ihre Anwendung	
Inhalt der früheren Seminarbände	147

Vorwort

Bereits zum drittenmal wurde die Seminarreihe "Aktuelle Forschungsgebiete in der Mathematik - Praxisbezogene Anwendungen" durchgeführt.

Ziel dieser Reihe ist, im Forschungszentrum Karlsruhe anwendungsorientierte, aktuelle Forschungsthemen aus der Mathematik zu präsentieren. Gedacht ist an Übersichtsvorträge, die Einblicke in moderne Methoden und Verfahren der Mathematik ermöglichen. Mit dieser Seminarreihe soll auch der Kontakt zu externen Forschungseinrichtungen erweitert und vertieft werden, um weitere kompetente Wissenschaftler als Gesprächspartner für das Forschungszentrum Karlsruhe zu gewinnen. Darüber hinaus soll aber auch ein breiter Gedankenaustausch innerhalb des Zentrums ermöglicht werden.

Angesprochen sind somit alle mathematisch-interessierte Mitarbeiter, die sich neuen mathematischen Verfahren und Methoden aufgeschlossen zeigen oder selbst an mathematischen Fragestellungen arbeiten. Praktisch kann sich aber jeder ohne tiefere Vorkenntnisse davon überzeugen, daß Mathematik keine trockene bezugslose Disziplin, sondern vielmehr eine lebendige Wissenschaft mit vielen überraschenden Querverbindungen, Witz und Eleganz ist.

Auch in der Seminarreihe 95 wurde ein breites Spektrum mathematischer Methoden vorgestellt. Neben einem Vortrag aus dem Gebiet der Statistik bildeten zwei Vorträge aus dem Themenfeld Neuronale Netze und Fuzzy-Logic einen gewissen Schwerpunkt. Referate zu den Bereichen Computeralgebra und Splinefunktionen rundeten die Vortragsreihe ab.

Ebenso vielfältig war auch der Personenkreis der Vortragenden. Neben zwei Referenten aus dem Forschungszentrum konnten drei Vortragende aus Industrie, Fachhochschule und Universität gewonnen werden, wodurch wieder wertvolle Kontakte zu externen Fachleuten und Einrichtungen geknüpft werden konnten. Allerdings referierten auch 1995 wieder nur männliche Wissenschaftler, obwohl sich die Organisatoren auch in diesem Bereich um Ausgewogenheit bemühten.

Im vorliegenden Bericht sind die Seminarbeiträge für 1995 in schriftlicher Form zusammengefaßt. Alle Beiträge werden von den Herausgebern durch eine kurze Vorstellung der Referenten eingeleitet.

Die Seminarreihe wird auch im Jahr 1996 fortgesetzt. Die aktuellen Themen können den Aushängen oder dem Terminkalender entnommen werden. Anregungen, Themen- und Vortragsvorschläge werden von den Organisatoren Claus-Peter Hugelmann (HIK(ehmals HDI), Tel. 07247/82-4798), Rolf Seifert (IAI, Tel. 07247/82-4411) und Thomas Westermann (FH Karlsruhe, Tel. 0721/925-1296) gerne entgegengenommen.



Prof. Dr. Norbert Henze

Prof. Dr. Norbert Henze wurde am 12.9.1951 geboren und studierte von 1969 bis 1975 Mathematik mit Nebenfach Informatik an den Universitäten Hannover und Barcelona. 1981 promovierte er, 1986 folgte die Habilitation in Hannover. Von 1987 bis 1990 vertrat er Professuren in Göttingen und Giessen, bis er schließlich 1991 den Lehrstuhl für Mathematische Stochastik an der Universität Karlsruhe übernahm. Seine Arbeitsgebiete sind Multivariate Statistik, Anpassungstest und Grenzwertsätze.

Einige Schwierigkeiten im Umgang mit bedingten Wahrscheinlichkeiten

von Norbert Henze

*Institut für Mathematische Stochastik, Universität Karlsruhe
Englerstraße 2, 76128 Karlsruhe*

1 Einleitung

Bedingte Wahrscheinlichkeiten modellieren die Aussicht auf das Eintreten eines Ereignisses unter einem Vorwissen über das Eintreten eines bestimmten anderen Ereignisses. Dabei entstehen häufig scheinbar widersprüchliche Aussagen, welche zu der oft geäußerten Meinung beitragen, mit Statistik ließe sich alles (oder auch nichts) beweisen. In dieser Arbeit werden exemplarisch einige klassische Beispiele zum Thema bedingte Wahrscheinlichkeiten zusammengestellt und die darin auftretenden scheinbaren Widersprüche geklärt.

2 Das Ziegenproblem und seine Verwandten

In der amerikanischen Spielshow „Let's make a deal“ ist als Hauptpreis ein Auto ausgesetzt. Hierzu sind auf der Bühne drei verschlossene Türen aufgebaut. Hinter einer rein zufällig ausgewählten Tür befindet sich der Hauptpreis, hinter den beiden anderen jeweils eine Ziege. Der Kandidat wählt eine der Türen, beispielsweise Tür 1, aus; diese bleibt aber vorerst verschlossen.

Der Spielleiter, welcher weiß, hinter welcher Tür das Auto steht, öffnet daraufhin mit den Worten: „Soll ich Ihnen mal etwas zeigen?“ eine der beiden anderen Türen, z.B. Tür 3, und eine Ziege schaut ins Publikum.

Der Kandidat hat nun die Möglichkeit, bei seiner ursprünglichen Wahl zu bleiben oder die andere verschlossene Tür (in unserem Beispiel Nr.2) zu wählen. Er erhält dann letztlich den Preis hinter dieser Tür.

In der Kolumne „Ask Marilyn“ des amerikanischen Wochenmagazins „Parade“ erklärte die Journalistin Marilyn vos Savant, daß ein Wechsel des Kandidaten zu Tür 2 dessen Chancen im Vergleich zum Festhalten an Tür 1 verdoppeln würde.

Dies war die Geburtsstunde des „Ziegenproblems“ im Jahr 1991, denn die Antwort von Frau Marilyn bescherte der Redaktion der „Parade“ eine Flut

von ungefähr 10000 Leserbriefen gegenteiliger Meinungen (interessanterweise tauchen jedoch die Spielshow „Let's make a deal“ und die Lösung des hier vorgestellten Problems schon 1975 auf (siehe [8],[9])). Das Problem wurde sowohl in der amerikanischen als auch in der deutschen Öffentlichkeit heiß diskutiert (siehe z.B. [3],[4],[6],[10],[13]). Es gab „Standhafte“, welche ihre ursprüngliche Wahl beibehalten wollten, „Randomisierer“, welche sich nur mittels eines Münzwurfs zwischen den verbleibenden Türen entscheiden mochten, und „Wechsler“, die ihre ursprüngliche Wahl verwerfen wollten.

Ca. 90 % der Zuschriften an Frau Marilyn spiegelten die Meinung wider, die Chancen auf den Hauptgewinn hinter Tür 1 hätten sich durch den Hinweis des Moderators von 1 zu 2 auf 1 zu 1 erhöht. Diese Meinung wurde mehrheitlich auch von akademisch vorgebildeten Lesern (u.a. auch Mitgliedern mathematischer Fakultäten) vertreten. Frau Marilyn blieb bei ihrer Aussage und mußte sich zum Teil deftige Beschimpfungen gefallen lassen. Es war die Rede von „typischen Laienfehlern“, „haarsträubendem Unsinn“, einer „peinlichen Angelegenheit“ und einem „Ärgernis“. Sie sei „eine dumme Törlin, welche die Krise der mathematischen Schulbildung noch vertiefe“.

Man empfahl ihr auch, unter dem Stichwort „bedingte Wahrscheinlichkeiten“ in einem beliebigen einführenden Lehrbuch zur Stochastik nachzulesen, um ihren Laienfehler einzusehen und zu erkennen, daß ihre Antwort „klar im Widerspruch zur Wahrheit stünde“.

Bevor wir diese Empfehlung aufgreifen, ist es angebracht, sich die „laienhafte“ Argumentation von Frau Marilyn in Erinnerung zu rufen. Diese Argumentation war wie folgt: Die Wahrscheinlichkeit, daß sich das Auto hinter Tür 1 befindet, ist $1/3$. Die Wahrscheinlichkeit, daß es sich hinter einer der beiden anderen Türen befindet, ist somit $2/3$. Öffnet der Moderator eine dieser beiden Türen, so steht die Tür fest, hinter welcher das Auto mit Wahrscheinlichkeit $2/3$ ist. Folglich verdoppelt „Wechseln“ die Chancen auf den Hauptgewinn.

Dieses Argument mag für viele einleuchtend sein, und es tritt noch klarer hervor, wenn man die Erfolgsaussichten der Strategien eines „Wechslers“ (dieser schwenkt immer von der zuerst genannten auf die nach dem Hinweis des Moderators verbleibende andere geschlossene Tür um) und eines „Standhaften“ (dieser bleibt unbeeinflusst vom Hinweis des Moderators immer bei seiner ursprünglichen Wahl) gegenüberstellt.

Der Standhafte gewinnt dann (und nur dann!) den Hauptgewinn, wenn sich dieser hinter der ursprünglich gewählten Tür befindet, und die Wahrscheinlichkeit hierfür ist $1/3$. Ein Wechsler hingegen gewinnt das Auto genau dann, wenn er zuerst auf eine der beiden „Ziegen-Türen“ zeigt (die Wahrscheinlichkeit hierfür ist $2/3$), denn nach dem Öffnen der anderen Ziegen-Tür durch den Moderator führt die Wechsel-Strategie ja automatisch zur „Auto-Tür“.

Überzeugend, nicht wahr? Für viele Kritiker von Frau Marilyn geht dieses Argument jedoch überhaupt nicht auf die konkrete Situation des Kandidaten ein. Dieser erhalte doch nach seiner Entscheidung für Tür 1 vom Modera-

tor freundlicherweise Tür 3 als Ziegen-Tür „ausgesondert“. Diese Information stelle für den Kandidaten eine Ausschluß-Bedingung für die Auto-Tür dar, und jeder normalbegabte Zwölfkläßler wisse doch, daß nach Ausschluß einer von drei gleichwahrscheinlichen Fällen die verbleibenden beiden Möglichkeiten die gleiche (bedingte) Wahrscheinlichkeit, nämlich jeweils $1/2$, besitzen.

Es sollte nicht schwer sein, viele Mitmenschen von der „Einsichtigkeit“ beider dem „gesunden Menschenverstand“ entsprungene Argumente zu „überzeugen“. Dies dürfte zu anregenden Diskussionen führen und Neugier auf die Auflösung des Paradoxons wecken.

Zur theoretischen Untermauerung ihres Einwandes gegen die Argumentation von Frau Marilyn führten viele mathematisch vorgebildete Kritiker folgendes Argument ins Feld:

Bekanntlich besteht ein endlicher Wahrscheinlichkeitsraum aus einer endlichen Menge Ω (diese beschreibt die möglichen Ergebnisse eines Zufallsexperiments) und einem Wahrscheinlichkeitsmaß P , welches jedem Ereignis A (Teilmenge A von Ω) eine Wahrscheinlichkeit $P(A)$ zuordnet. Die bedingte Wahrscheinlichkeit von A unter der Bedingung, daß ein Ereignis B eingetreten ist, wird durch den Quotienten

$$P_B(A) := P(A|B) := \frac{P(A \cap B)}{P(B)}$$

definiert. Bei festem Ereignis B ist $P_B(A)$ als Funktion von A wieder ein Wahrscheinlichkeitsmaß auf Ω (genauer: auf den Teilmengen von Ω). Anschaulich erfolgt durch die Mengenfunktion $P_B(\cdot)$ eine Neubewertung aller Ereignisse hinsichtlich der Aussicht ihres Eintretens, nachdem bekannt ist, daß das Ereignis B stattgefunden hat.

Im Spezialfall, daß das Wahrscheinlichkeitsmaß P die Gleichverteilung auf Ω ist, also $P(A) = |A|/|\Omega|$ ($A \subset \Omega$) gilt (wie üblich sei dabei $|M|$ die Anzahl der Elemente einer Menge M), folgt

$$P_B(A) = \frac{|A \cap B|/|\Omega|}{|B|/|\Omega|} = \frac{|A \cap B|}{|B|}$$

und somit

$$P_B(A) = \frac{|A|}{|B|} \tag{1}$$

für jedes Ereignis A mit $A \subset B$. Folglich ist das Wahrscheinlichkeitsmaß P_B (bei Einschränkung auf Teilmengen von B) die Gleichverteilung auf B .

Die Kritiker von Frau Marilyn kannten offenbar diese elementaren Betrachtungen über bedingte Wahrscheinlichkeiten und schlossen nun (etwas vorschnell) wie folgt weiter: Wir setzen $\Omega = \{1, 2, 3\}$, wobei die Zahl j für die Nummer der Tür steht, hinter der sich das Auto befindet. Weiterhin nehmen wir für P eine Gleichverteilung auf Ω an. Das Öffnen von Tür 3 durch den

Moderator ist gleichbedeutend damit, daß das Ereignis $B = \{1, 2\}$ eingetreten ist. Nach Formel (1) gilt $P_B(\{1\}) = P_B(\{2\}) = 1/2$, und somit hat sich die Wahrscheinlichkeit, daß Tür 1 die Auto-Tür ist, von $1/3$ auf $1/2$ erhöht.

Das Vorschnelle an dieser „Beweisführung“ ist, daß sie einen wichtigen Umstand außer acht läßt, nämlich die Tatsache, daß der Kandidat Tür 1 gewählt hat und **diese Tür vom Moderator nicht geöffnet werden darf**. Die Kritiker von Frau Marilyn hätten dann recht, wenn der Spielleiter **vor dieser Wahl des Kandidaten** Tür 3 als Ziegen-Tür zu erkennen gegeben hätte. Nach Wahl von Tür 1 durch den Kandidaten gibt es für den Spielleiter folgende Möglichkeiten:

- Ist das Auto hinter Tür 2, so **muß** er Tür 3 öffnen.
- Ist das Auto hinter Tür 3, so **muß** er Tür 2 als Ziegen-Tür enttarnen.
- Ist das Auto hinter Tür 1, so hat er die Wahl, Tür 2 oder Tür 3 zu öffnen.
Wir nehmen (vorläufig) an, daß er diese Wahl rein zufällig trifft.

Die Situation ist also diejenige eines zweistufigen Experiments, dessen erster Teil in der rein zufälligen Zuordnung des Hauptgewinns zu einer der drei Türen besteht. Die Wahl von Tür 1 durch den Kandidaten (welcher nicht zu außersinnlicher Wahrnehmung fähig sei) geschieht ohne Beschränkung der Allgemeinheit. Tür 1 besitzt nur die Rolle einer „Tabu-Tür“ für den Spielleiter. Nach Kenntnis von Tabu- und Auto-Tür öffnet der Spielleiter nach obiger Regel eine Ziegen-Tür (zweites Telexperiment). Dieser „Ausschluß-Mechanismus“ ist in Abbildung 1 durch ein Baumdiagramm veranschaulicht.

Bezeichnen A_j das Ereignis, daß sich das Auto hinter Tür j befindet ($j = 1, 2, 3$) und C das Ereignis, daß der Spielleiter Tür 3 öffnet, so ergibt sich aus $P(A_1) = P(A_2) = P(A_3) = 1/3$ und den bedingten Wahrscheinlichkeiten

$$P(C|A_1) = \frac{1}{2}, \quad P(C|A_2) = 1, \quad P(C|A_3) = 0 \quad (2)$$

die interessierende Wahrscheinlichkeit $P(A_1|C)$ mit Hilfe der Bayes-Formel zu

$$P(A_1|C) = \frac{P(A_1)P(C|A_1)}{\sum_{j=1}^3 P(A_j)P(C|A_j)} = \frac{\frac{1}{2}}{\frac{1}{2} + 1 + 0} = \frac{1}{3}.$$

Wegen $P(A_1) = \frac{1}{3}$ besagt dieses Ergebnis, daß die Ereignisse A_1 und C stochastisch unabhängig sind.

Das Öffnen von Tür 3 durch den Spielleiter vergrößert also keinesfalls die Chance, daß sich das Auto hinter Tür 1 befindet — ganz im Einklang mit den „laienhaften“ Überlegungen von Frau Marilyn. Da analog zu oben

$$P(A_2|C) = \frac{1}{\frac{1}{2} + 1 + 0} = \frac{2}{3}$$

gilt (die Ereignisse A_2 und C sind **nicht** stochastisch unabhängig), verdoppelt „Wechseln“ in der Tat die Aussicht auf den Hauptgewinn. ..

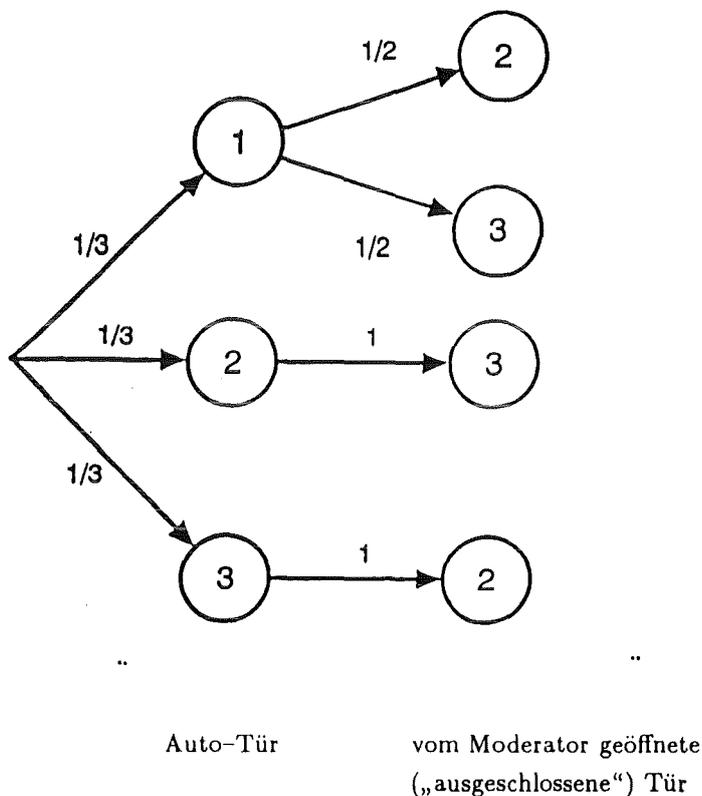


Abbildung 1: Baumdiagramm („Ausschluß-Mechanismus“) für das Ziegenproblem (Kandidat wählt Tür 1)

Eine Verallgemeinerung der Fragestellung ergibt sich, wenn der Moderator für den Fall, daß sich das Auto hinter Tür 1 befindet, Tür 3 mit der Wahrscheinlichkeit p (und Tür 2 mit der Wahrscheinlichkeit $1 - p$) öffnet. In Abänderung von (2) gilt dann $P(C|A_1) = p$, und die Bayes-Formel liefert

$$P(A_2|C) = \frac{P(A_2)P(C|A_2)}{\sum_{j=1}^3 P(A_j)P(C|A_j)} = \frac{1}{p+1}$$

Da dieser Ausdruck (unabhängig von p) mindestens $\frac{1}{2}$ ist, lohnt sich ein Wechsel also auch dann, wenn der Spielleiter mit einer gewissen Präferenz zufällig zwischen den Türen 2 und 3 wählt.

Eine „Gefangenen-Version“ des Ziegenproblems hat Martin Gardner 1959 vorgestellt (siehe [2]). In dieser makabren Variante gibt es drei zum Tode verurteilte Gefangene, von denen einer (durch das Los bestimmt) begnadigt wird. Der Wärter weiß, um welchen Gefangenen es sich dabei handelt, darf aber

nichts mitteilen. Auf mehrfaches Drängen des Gefangenen Nr. 1 gibt er diesem die Information: „Ich darf zwar über Dich keine Aussage machen, kann Dir aber mitteilen, daß Nr. 3 hingerichtet wird.“

Ersetzen wir das Auto durch die Begnadigung, die Ziege durch den Galgen, den Spielleiter durch den Wärter und Tür 1 durch den Gefangenen Nr. 1, so offenbart sich die begriffliche Äquivalenz von Gefangenenproblem und Ziegenproblem. Unter der Annahme, daß der Wärter für den Fall, daß Nr. 1 tatsächlich begnadigt wird, rein zufällig einen der beiden anderen Gefangenen „ausschließt“, folgt, daß sich die Aussicht von Nr. 1 auf die Begnadigung aufgrund der Information des Wärters nicht vergrößert hat.

Nahezu identisch mit dem Ziegenproblem ist auch das klassische „Schubladen-Paradoxon“ von J. Bertrand (siehe [1]), bei dem drei gleich aussehende Kästen mit je zwei Schubladen gegeben sind. In jeder Schublade liegt eine Münze; im ersten Kasten Gold-Gold, im zweiten Silber-Silber, im dritten Gold-Silber. Es wird rein zufällig eine Schublade ausgewählt. In dieser befindet sich eine Goldmünze. Mit welcher (bedingten) Wahrscheinlichkeit ist in der anderen Schublade dieses Kastens eine Silbermünze?

Auch hier sind gefährliche stochastische Fallstricke ausgelegt. Natürlich ist durch die gegebene Teilinformation einer von den 3 ursprünglich gleichwahrscheinlichen Kästen, nämlich der zwei Silbermünzen enthaltende Kasten, „aussortiert“ worden.

Wer jedoch vorschnell meint, die verbleibenden beiden Möglichkeiten seien gleichwahrscheinlich und somit die gesuchte bedingte Wahrscheinlichkeit gleich $1/2$, unterliegt dem gleichen Trugschluß wie die meisten Kritiker von Frau Marilyn. Entscheidend ist auch hier, nach welchem „stochastischen Mechanismus“ der Ausschluß eines Falles erfolgt, und dieser Mechanismus ist uns schon vom Ziegen- und Gefangenenproblem her bekannt. Hierzu interpretieren wir den Gold-Silber-Kasten als Hauptgewinn und die beiden übrigen Kästen als Nieten. Falls einer der beiden „Nieten-Kästen“ ausgewählt wird, so erfolgt durch Öffnen einer seiner beiden Schubladen auf jeden Fall der Ausschluß der anderen Niete. Bei Wahl des Gold-Silber-Kastens werden hingegen die beiden Nieten mit gleicher Wahrscheinlichkeit $1/2$ ausgeschlossen (je nachdem, welche Schublade gezogen wird).

Es liegt also das Baumdiagramm von Abbildung 1 vor, wobei im ersten Telexperiment die Wahl des Kastens (Tür 1 entspricht dem Gold-Silber-Kasten, welcher in keinem Fall ausgeschlossen wird) erfolgt und aufgrund des zweiten Telexperiments (rein zufällige Auswahl einer Schublade) der Ausschluß einer Niete vorgenommen wird. Nach unseren früheren Überlegungen bleibt also die Wahrscheinlichkeit dafür, daß der ausgewählte Kasten der Hauptgewinn ist, auch nach Öffnen einer seiner beiden Schubladen gleich $1/3$.

Die Analogie zum Ziegenproblem wäre vollkommen, wenn ein Moderator aus den beiden nicht gewählten Kästen tatsächlich die „aussortierte“ Niete

entfernen würde und danach die Möglichkeit des „Wechsels“ gegeben wäre.

Natürlich sind auch verschiedene Varianten des Schubladen-Paradoxons denkbar. Eine Möglichkeit besteht z.B. darin, drei Spielkarten zu benutzen, von denen eine beidseitig weiß, die zweite beidseitig rot und die dritte weiß-rot ist. Zur Osterzeit bieten sich drei äußerlich gleichartige Ostereier aus Pappe an, deren Innenseiten weiß-weiß, rot-rot bzw. weiß-rot sind. Dem Öffnen einer Schublade entspricht dann das Sichtbarmachen einer Kartenseite durch blindes Hervorziehen einer Karte unter einem schwarzen Tuch bzw. dem Öffnen eines rein zufällig gewählten Eies und Betrachten einer Innenseite.

3 Das Bridge-Paradoxon

Beim „Bridge“ erhalten 4 Spieler je 13 Karten. Eine Farbe wird vor Beginn des Spiels als „Trumpf“ bestimmt. Es sei bekannt, daß die Spieler I und II zusammen 6 Trümpfe besitzen (bedingendes Ereignis B), wobei die genaue „Trumpfverteilung“ zwischen I und II natürlich 6/0, 5/1, 4/2, 3/3, 2/4, 1/5 oder 0/6 sein kann. Unter der Annahme, daß alle Kartenverteilungen gleichwahrscheinlich sind (wobei wir ohne Beschränkung der Allgemeinheit die vereinfachende Annahme machen, daß die Menge der an Spieler I und II zusammen ausgeteilten 26 Karten, darunter 6 Trümpfe, **bekannt** und nur die Aufteilung dieser 26 Karten auf beide Spieler vorzunehmen ist), folgt dann durch Abzählen der günstigen und möglichen Fälle (hypergeometrische Verteilung!) z.B.

$$P_B(\text{„Trumpfverteilung } 3/3\text{“}) = \frac{\binom{6}{3} \binom{20}{10}}{\binom{26}{13}} = \frac{286}{805} \quad (3)$$

und

$$P_B(\text{„Trumpfverteilung } 4/2 \text{ oder } 2/4\text{“}) = 2 \cdot \frac{\binom{6}{4} \binom{20}{9}}{\binom{26}{13}} = \frac{390}{805}. \quad (4)$$

Im folgenden ist nur wichtig, daß die „3/3-Verteilung“ im Vergleich zur „4/2- oder 2/4-Verteilung“ die **kleinere** Wahrscheinlichkeit besitzt.

Das Spiel beginnt, und in den ersten beiden Stichen wird jeweils Trumpf ausgespielt. Es stellt sich heraus, daß die Spieler I und II in beiden Stichen „Trumpf bedient haben“, was natürlich die Trumpfverteilungen 6/0, 5/1, 1/5 und 0/6 im nachhinein ausschließt.

Nach diesen beiden Stichen besitzen die Spieler I und II je 11 Karten und zusammen 2 Trümpfe. Die Wahrscheinlichkeit dafür, daß sich bei rein zufälliger Aufteilung von 22 Karten, von denen 2 Trümpfe seien (bedingendes Ereignis C), auf 2 Spieler zu je 11 Karten die Trumpfverteilungen 1/1 bzw. 2/0 oder 0/2 ergeben, sind

$$P_C(\text{„Trumpfverteilung } 1/1\text{“}) = \frac{\binom{2}{1} \binom{20}{10}}{\binom{22}{11}} = \frac{11}{21} \quad (5)$$

und

$$P_C(\text{„Trumpfverteilung } 2/0 \text{ oder } 0/2\text{“}) = 2 \cdot \frac{\binom{2}{2} \binom{20}{9}}{\binom{22}{11}} = \frac{10}{21}. \quad (6)$$

Das anscheinend Widersprüchliche an diesem Resultat ist, daß die Trumpfverteilung 1/1 eine **größere Wahrscheinlichkeit** als die Verteilung 2/0 oder 0/2 besitzt, obwohl sie doch (durch „Abwerfen“ von 2 Trümpfen) aus der im Vergleich zur 4/2- oder 2/4-Kombination **unwahrscheinlicheren** Trumpfverteilung 3/3 stammt (siehe [11], S. 18–19).

Der Schlüssel zur Auflösung dieses Paradoxons liegt in den Gleichungen

$$\binom{6}{3} \cdot \binom{20}{10} \cdot 9 = \binom{6}{2} \cdot \binom{4}{2} \cdot \binom{2}{1} \cdot \binom{20}{10} \quad (7)$$

und

$$\binom{6}{4} \cdot \binom{20}{9} \cdot 6 = \binom{6}{2} \cdot \binom{4}{2} \cdot \binom{2}{2} \cdot \binom{20}{9}, \quad (8)$$

welche die Spielverläufe beim Abwerfen von Trümpfen berücksichtigen. In Gleichung (7), welche die Trumpfverteilungen 3/3 und 1/1 miteinander verknüpft, gibt der Binomialkoeffizient $\binom{20}{10}$ die Anzahl der Möglichkeiten an, aus den 20 „Nicht-Trümpfen“ 10 für Spieler I auszuwählen. Nachdem Spieler I und II jeweils 3 Trümpfe erhalten haben (hierfür gibt es $\binom{6}{3}$ Möglichkeiten), besitzt jeder Spieler 3 Möglichkeiten, nach Abwerfen von 2 Trümpfen einen Trumpf in der Hand zu behalten. Dies liefert den Faktor 9 auf der linken Seite von (7). Der erste (zweite) Faktor auf der rechten Seite von (7) ist die Anzahl von Möglichkeiten, aus den 6 Trümpfen (restlichen 4 Trümpfen nach Zuteilung von 2 Trümpfen an Spieler I) diejenigen beiden auszuwählen, welche Spieler I (Spieler II) abwirft. Von den verbleibenden beiden Trümpfen erhält jeder Spieler einen, was den dritten Faktor ergibt.

In gleicher Weise verbindet Gleichung (8) die Trumpfverteilungen 4/2 und 2/0. Hier werden für Spieler I aus 6 Trümpfen 4 ausgewählt ($\binom{6}{4}$ Möglichkeiten), wobei 6 Möglichkeiten vorhanden sind, nach Abwerfen von 2 Trümpfen 2 Trümpfe auf der Hand zu behalten. Dies erklärt die linke Seite von (8). Die Bedeutung der beiden ersten Faktoren auf der rechten Seite von (8) ist die gleiche wie in Gleichung (7). Da die übrigbleibenden beiden Trümpfe Spieler II gegeben werden müssen, gibt es hierfür nur eine ($= \binom{2}{2}$) Möglichkeit.

Dividiert man nun Gleichung (7) durch die mit dem Faktor 2 multiplizierte Gleichung (8), so folgt unter Beachtung von (3), (4), (5) und (6)

$$\begin{aligned} & \frac{9}{6} \cdot \frac{P_B(\text{„Trumpfverteilung } 3/3\text{“})}{P_B(\text{„Trumpfverteilung } 4/2 \text{ oder } 2/4\text{“})} \\ &= \frac{9}{6} \cdot \frac{286}{390} = \frac{9}{6} \cdot \frac{11}{15} = \frac{11}{10} \\ &= \frac{P_C(\text{„Trumpfverteilung } 1/1\text{“})}{P_C(\text{„Trumpfverteilung } 2/0 \text{ oder } 0/2\text{“})}, \end{aligned}$$

was den scheinbaren Widerspruch auflöst.

4 Das Simpson–Paradoxon

Ein Paradoxon, welches im Zusammenhang mit bedingten Wahrscheinlichkeiten wichtige Einsichten vermittelt, ist das klassische Simpson–Paradoxon (siehe z.B. [5],[7],[12]). Zu seiner Formulierung benötigen wir eine Zerlegung eines Grundraums Ω in paarweise disjunkte Ereignisse K_1, \dots, K_n sowie Ereignisse A und B . Dabei bezeichne im folgenden $\bar{B} = \Omega \setminus B$ das Gegenereignis zu B .

Das Simpson–Paradoxon liegt dann vor, wenn neben den Ungleichungen

$$P(A|B \cap K_j) > P(A|\bar{B} \cap K_j), \quad \text{für jedes } j = 1, \dots, n \quad (9)$$

„paradoxerweise“ die umgekehrte Ungleichung

$$P(A|B) < P(A|\bar{B}) \quad (10)$$

erfüllt ist. Wegen

$$P(A|B) = \sum_{j=1}^n P(K_j|B) \cdot P(A|B \cap K_j) \quad (11)$$

$$P(A|\bar{B}) = \sum_{j=1}^n P(K_j|\bar{B}) \cdot P(A|\bar{B} \cap K_j) \quad (12)$$

(Berechnung der bedingten Wahrscheinlichkeiten $P_B(A)$ bzw. $P_{\bar{B}}(A)$ mit Hilfe der Formel von der totalen Wahrscheinlichkeit) ist die Möglichkeit des Auftretens des Simpson–Paradoxons eine **mathematische** Banalität. Entscheidend für die Gültigkeit von (10) ist, daß die „Gewichte“ $P(K_j|B)$ in (11) gerade für diejenigen j „klein“ sein können, für die $P(A|B \cap K_j)$ „groß“ ist (und umgekehrt). Andererseits kann $P(K_j|\bar{B})$ in (12) gerade für diejenigen j „groß“ sein, für die $P(A|\bar{B} \cap K_j)$ „groß“ ist (ohne natürlich (9) zu verletzen) und umgekehrt.

Das Reizvolle am Simpson–Paradoxon ist sein Auftreten bei realen Daten, wobei die Interpretationsmöglichkeiten natürlich von den jeweiligen Rahmenbedingungen abhängen.

Als Beispiel betrachten wir die in Tabelle 1 aufgeführten Anzahlen im Zusammenhang mit auslaufenden Abonnements der Zeitschrift „American History Illustrated“ (siehe [5], [12]).

Monat	Kategorie des laufenden Abonnements					Σ
	K_1	K_2	K_3	K_4	K_5	
Januar						
ausl. Abo.	3594	18364	20862	2986	149	45955
Verläng.	2918	14488	4343	1783	13	23545
Anteil	0.812	0.789	0.208	0.597	0.087	0.512
Februar						
ausl. Abo.	884	5140	864	2224	45	9157
Verläng.	704	3907	122	1134	2	5869
Anteil	0.796	0.760	0.141	0.510	0.044	0.641

Tabelle 1: Verlängerungen auslaufender Abonnements der Zeitschrift „American History Illustrated“ im Jahr 1979

Von der Abonnement-Verwaltung ist jeder Abonnent einer von 5 Kategorien $1, \dots, 5$ je nach dem Grund für sein laufendes Abonnement zugeteilt (bzgl. der genauen Definition der Kategorien siehe [12]).

Im Januar 1979 wurden von den in der Kategorie 1 auslaufenden 3594 Abonnements 2918 verlängert, was einer Erneuerungsrate von 0.812 entspricht. Analog sind die Zahlen für die übrigen Kategorien zu verstehen. Über alle Kategorien summiert betrug der Anteil der Verlängerungen im Januar 0.512 (rechte Spalte). Obwohl im Februar die Erneuerungsrate in jeder Kategorie unter dem entsprechenden Januar-Wert liegt, ist die globale Erneuerungsrate im Februar mit 0.641 deutlich höher als im Januar.

Setzen wir Ω (bzw. B) für die Menge aller im Januar oder Februar (bzw. im Januar) auslaufenden Abonnements, K_j für die Menge aller Abonnements aus Ω in Kategorie j ($j = 1, \dots, 5$), A für die Menge aller verlängerten Abonnements aus Ω und P für die Gleichverteilung auf Ω , so liegt das Simpson-Paradoxon (9),(10) vor. Der Grund hierfür ist im wesentlichen in der geringeren Gewichtung der Kategorien 1 und 2 im Januar verglichen mit dem Monat Februar sowie in der hohen Gewichtung der Kategorie 3 im Januar zu suchen.

Als weiteres Beispiel zum Simpson-Paradoxon sind in Tabelle 2 das Gesamt-Bruttoeinkommen sowie die daraus gezahlte Einkommenssteuer der Jahre 1974 und 1978 in den U.S.A., aufgeschlüsselt nach Einkommensklassen, angegeben (siehe [5],[12]).

Jahreseinkommen (pro Person in \$)	Einkommen (in 1000 \$)	gezahlte Steuer (in 1000 \$)	durchschnittlicher Steueranteil
1974			
< 5000	41 651 643	2 244 467	0.054
5000 bis 9999	146 400 740	13 646 348	0.093
10000 bis 14999	192 688 922	21 449 597	0.111
15000 bis 99999	470 010 790	75 038 230	0.160
≥ 100000	29 427 152	11 311 672	0.384
Insgesamt	880 179 247	123 690 314	0.141
1978			
< 5000	19 879 622	689 318	0.035
5000 bis 9999	122 853 315	8 819 461	0.072
10000 bis 14999	171 858 024	17 155 758	0.100
15000 bis 99999	865 037 814	137 860 951	0.159
≥ 100000	62 806 159	24 051 698	0.383
Insgesamt	1 242 434 934	188 577 186	0.152

Tabelle 2: Einkommenssteuer in den U.S.A. 1974 und 1978

Obwohl der durchschnittliche Steueranteil in jeder Einkommenskategorie von 1974 auf 1978 gesunken ist, hat sich die durchschnittliche Steuerbelastung insgesamt von 14,1% auf 15,2% erhöht. Dies ist in diesem Fall natürlich auf den Effekt zurückzuführen, daß viele Dollar nach 4 Jahren in einer höheren Einkommenskategorie erwirtschaftet wurden und sich somit die Gewichte der Kategorien entscheidend verändert haben.

Als mathematisches Modell kann in diesem Beispiel Ω (bzw. B) für die Menge der 1974 oder 1978 (bzw. 1974) als Einkommen erzielten einzelnen Dollar und A für die Teilmenge der 1974 oder 1978 an den Fiskus gezahlten „Steuer-Dollar“ gewählt werden. Jeder eingenommene Dollar ist dabei einer der 5 Einkommens-Kategorien zuzurechnen. Wählen wir P als Gleichverteilung auf Ω , so gilt wieder (9) und (10), also das Simpson-Paradoxon.

An diesem Beispiel wird auch der durch (wissentliches oder unwissentliches) Verschweigen gewisser Aspekte erzielbare politische Effekt deutlich. Wäre zwischen den Jahren 1974 und 1978 eine Steuerreform durchgeführt worden, so könnte sich die Regierung die Abnahme der durchschnittlichen Steuerbelastung in jeder Einkommenskategorie als Erfolg an ihre Fahnen heften. Die Opposition hingegen könnte und würde nur mit der unleugbaren Tatsache Politik machen, daß die globale durchschnittliche Steuerbelastung zugenommen hat. So lügt man mit Statistik!

Literatur

- [1] *Bertrand, J.*: Calcul des Probabilités. 3.Auflage. Chelsea Publ. Comp., New York 1907.
- [2] *Gardner, M.*: Mathematical games. Scientific American 201, October (1959) 180–182, November (1959) 188.
- [3] *Gillmann, L.*: The car and the goats. Amer. Math. Monthly 99 (1992) 3–7.
- [4] *Lo Bello, A.*: Ask Marilyn: the mathematical controversy in Parade Magazine. Math. Gaz. 75 (1991) 275–277.
- [5] *Newson, G.*: Simpson's paradox revisited. Math. Gaz. 75 (1991) 290–293.
- [6] *v. Randow, G.*: Das Ziegenproblem. Denken in Wahrscheinlichkeiten. Rowohlt Taschenbuch Verlag, Reinbeck bei Hamburg 1992.
- [7] *Samuels, M.L.*: Simpson's Paradox and Related Phenomena. J. Amer. Statist. Assoc. 88 (1993) 81–88.
- [8] *Selvin, S.*: A problem in probability. Amer. Statist. 29 (1975) 67.
- [9] *Selvin, S.*: On the Monty Hall problem. Amer. Statist. 29 (1975) 134.
- [10] *Stewart, I.*: Das Versteck der Andromeda. Spektrum der Wissenschaft 11/91 (1991) 12–16.
- [11] *Szekely, G.*: Paradoxes in Probability Theory and Mathematical Statistics. R. Deidel Publ. Comp., Dordrecht 1986.
- [12] *Wagner, C.H.*: Simpson's Paradox in Real Life. Amer. Statist. 36 (1982) 46–48.
- [13] Schönheit des Denkens. Der Spiegel 34/91 (1991) 212–213.



Dr. Wolfgang Eppler

Dr. Wolfgang Eppler studierte von 1980 an Informatik an der Universität Stuttgart mit dem Schwerpunkt Hardware und im Nebenfach Elektrotechnik und schloß mit dem Diplom im September 1986 ab. Von 1987 bis 1993 war er wissenschaftlicher Mitarbeiter am Institut für Rechnerentwurf und Fehlertoleranz der Universität Karlsruhe und am Forschungszentrum Informatik (FZI) Karlsruhe. Dort arbeitete er im Bereich der Automatisierung, des Schaltkreisentwurfs und im Bereich der Neuronalen Netze, der Fuzzy Logik und der Regelungstechnik. Schließlich promovierte er mit dem Thema „Vorstrukturierung Neuronaler Netze mit Fuzzy Logik“. Seit Oktober 1993 ist er Nachwuchswissenschaftler im Forschungszentrum Karlsruhe in der Hauptabteilung Prozeßautomatisierung und Elektronik.

Neuro-Fuzzy-Systeme in Theorie und Anwendung

W. Eppler, H. Gemmeke

EINLEITUNG	24
PRINZIPIELLE KOPPELUNGSARTEN VON NEURO-FUZZY	24
Stärke der Verflechtung	25
Nicht gekoppelte Neuro-Fuzzy-Systeme	25
Schwach gekoppelte Neuro-Fuzzy-Systeme	26
Eng gekoppelte Neuro-Fuzzy-Systeme	26
Ziel der Koppelung bei Neuro-Fuzzy-Systemen	26
Adaptives propositionales System	26
Interpretierbares prozedurales System	27
NEURO-FUZZY-SYSTEME	28
NNDFR (Hayashi 1989)	28
FINE (Eppler 1990)	29
ANFIS (Jang 1991)	32
BIOFAM (Kosko 1992)	33
GARIC (Berenji und Khedar 1992)	34
NEFCON (Nauck 1993)	35
FuNe (Halgamuge 1993)	36
FAGNIS (Cechin, Eppler, Gemmeke 1994)	36
NEURO-FUZZY-ANWENDUNGEN	38
Diagnose von Leber-Funktionsstörungen	38
Regelung eines Sofortdampferzeugers	39
Regelung eines Schweißroboters	40
Automatik-Getriebe	40
Roboterarm	41
Regelung eines Interferometerspiegels	41
Roboter-Navigation	42
Regelung einer Klimaanlage	43
RESUMEE	44
LITERATUR	45

Einleitung

Neuronale Netze und Fuzzy Logik sind völlig verschiedene Ansätze, die zunächst wenig miteinander zu tun haben. Bei Neuronalen Netzen wird die Grobstruktur, d.h. der Typ und die Topologie des Netzes mit der Anzahl der Neuronen vom Entwickler vorgegeben. Die Feinstruktur wird durch einen Lernalgorithmus und große Datenmengen automatisch bestimmt. Die erreichte Genauigkeit ist umgekehrt proportional zur Datenmenge. In der Fuzzy Logik ist kein Lernen definiert. Die Verarbeitung von Fuzzy-Systemen basiert nicht auf Datenmengen, sondern auf Regeln, die von einem Entwickler oder Experten stammen. Bei der Entwicklung der beiden System-Arten spielen also komplementäre Eigenschaften eine Rolle: Neuronale Netze beziehen ihr Wissen in erster Linie aus Daten, Fuzzy-Systeme setzen einen Experten voraus. Höchstens in einer Hinsicht haben die beiden System-Arten eine vergleichbare Zielsetzung: in der Behandlung von Ähnlichkeiten. Neuronale Netze bilden durch Trainingsmuster eine ihnen angemessene innere Struktur heran, die es anschließend erlaubt, bis dahin unbekannte Testmuster zu verarbeiten. Die Verarbeitung der Testmuster, die in ihrer Struktur Ähnlichkeiten zu den Trainingsmustern aufweisen, ergibt ähnliche Resultate im Vergleich zu denen der Trainingsmuster. Mit der Fuzzy Logik werden nahe beieinander liegende Punkte des Eingaberaums ähnlich - d.h. in nahe beieinanderliegende Punkte des Ausgaberaums - abgebildet und verarbeitet. Dadurch vermeidet diese Logik Sprünge und scharfe Kanten, die nebeneinander liegende Punkte eindeutig voneinander trennen. Auch hier können wieder nahe beieinanderliegende Punkte als Daten mit ähnlicher Struktur gedeutet werden, die durch überlappende Zugehörigkeitsfunktionen und damit auch Regeln und die interpolierenden Eigenschaften des Defuzzifizierens ähnlich verarbeitet werden. Dies ist der qualitative Unterschied zur Verarbeitung mit konventionellen Systemen, die auf der klassischen zweiwertigen Logik basieren und mit harten Fallunterscheidungen arbeiten. Selbstverständlich können Ähnlichkeiten auch mit anderen konventionellen Verfahren behandelt werden, wie stochastischen Methoden und Algorithmen, welche harte Entscheidungen durch kontinuierliche Funktionen ersetzen. In vielen Fällen sind diese Methoden natürlich weiterhin sinnvoll und manchen Neuro-Fuzzy-Ansätzen überlegen. Das vorliegende Kapitel über Neuro-Fuzzy zeigt aber, daß Neuro-Fuzzy-Systeme eine einheitliche Methode liefern können, mit Problemen der beschriebenen Art umzugehen, ohne jedesmal die gesamte Theorie stochastischer Funktionen bemühen und an das jeweilige System anpassen zu müssen. Wir werden sehen, daß es dabei unterschiedliche Zielsetzungen gibt, je nachdem, ob die zeitliche Veränderung und Adaption des Systems eine Rolle spielt oder ob das aus großen Datensätzen entwickelte System für den Betrachter oder Anwender durchsichtiger und verstehbarer gestaltet werden soll.

Neuronale Netze und Fuzzy Systeme sind sich also, wenigstens was ihr Ziel bei der Behandlung von Ähnlichkeiten betrifft, ähnlich. In bezug auf andere Merkmale verhalten sie sich ziemlich entgegengesetzt. Zur Übersicht kann hier auf Tab.1 verwiesen werden, die verdeutlicht, wie sich beide Systeme zueinander oft komplementär verhalten (basiert auf Gupta und Rao (1994), korrigiert und erweitert nach Goos, Eppler und Gemmeke). Gerade in der Komplementarität liegt die Stärke der Neuro-Fuzzy-Systeme, weil beide Systeme auf gegensätzliche Weise bei gleichen Problemstellungen einsetzbar sind.

Prinzipielle Koppelungsarten von Neuro-Fuzzy

Seit 1989 zum erstenmal Neuronale Netze mit Fuzzy Logik in Verbindung gebracht wurden (Hayashi et al. 1989, Kuncicky und Kandel 1989, Yamakawa und Tomoda 1989), sind seither unter dem Stichwort „Neuro-Fuzzy“ recht unterschiedliche Arbeiten veröffentlicht worden. Seit der ersten Tagung zu Neuro-Fuzzy, die unseres Wissens der IEEE-Workshop „Neuro-Fuzzy Control in Muroran in Japan 1993 war, wird dieser Begriff gar als Oberbegriff für alle Systeme verwendet, die Neuronale Netze *oder* Fuzzy Systeme einsetzen. In diesem Kapitel werden Neuro-Fuzzy-Systeme nicht in dieser verallgemeinerten Weise verstanden, sondern ausschließlich als Koppelung von Neuronalen Netzen *und* Fuzzy Systemen. Selbst dann gibt es noch eine große Vielfalt von Systemen, bei denen es schwerfällt, sie als Neuro-Fuzzy-Systeme zu bezeichnen. Ein komplexes System, das an irgendeiner Stelle ein Neuronales Netz enthält und an irgendeiner anderen Stelle Fuzzy-Regeln verwendet, wird im folgenden als „nicht gekoppeltes“ Neuro-Fuzzy-System bezeichnet. Weitere Stufen, die sich nach der Stärke der Verflechtung der beiden Systemarten richten, sind dann „schwach gekoppelte“ oder „stark gekoppelte“ Neuro-Fuzzy-Systeme. Sie werden im folgenden etwas genauer erläutert. Aber nicht nur hinsichtlich ihrer Verflechtungsstärke, auch nach ihrer Zielsetzung unterscheiden sich die Systeme. Man kann hier zwei Hauptgattungen feststellen: Systeme, deren Beschreibung und Spezifikation genau festliegt (propositional), die sich aber flexibel (adaptiv) auf Umgebungsänderungen einstellen sollen, werden im folgenden als „adaptive propositionale Systeme“ behandelt. Systeme, die durch eine automatische Analyse von Datenmengen entstanden sind, und hierzu zählen neben Neuronalen Netzen auch stochastische Verfahren, Cluster-Erkennungsverfahren und Prognose-Werkzeuge, bzw. Systeme, die neue Daten einfach nach den Gesetzmäßigkeiten alter Daten *prozedural* verarbeiten, sind für den Anwender oft wenig vertrauenerweckend. Er sieht außer der Beobachtung der Eingangs- und Ausgangsdaten vom Verarbeitungsprozeß nichts und kann ihn daher auch nicht nachvollziehen, d.h. nicht *interpretieren*. Systeme, die sowohl prozedural als auch interpretierbar sind, sind also wünschenswert und werden im folgenden als „interpretierbare prozedurale Systeme“ bezeichnet.

Fähigkeiten	Merkmale	Fuzzy-System	Neuronales Netz
Wissensakquisition	<i>Fakten</i>	menschliches Expertenwissen	Trainingsdaten, Musterbeispiele
	<i>System</i>	kein Modell erforderlich	kein Modell erforderlich
	<i>Implementierung</i>	manuell durch Interaktion zwischen Mensch und Maschine	automatische Lernalgorithmen
Struktur	<i>System</i>	vom Entwickler festgelegt	Lernen der Feinstruktur aus Daten, Bilden der Grobstruktur durch andere Methoden (z.B. Evolution)
	<i>Problem</i>	durch Regeln vorgegeben	weitgehend unbekannt
	<i>Verständnis</i>	einfache Interpretation	Black Box
Unsicherheit	<i>Information</i>	quantitativ und qualitativ	quantitativ
	<i>Ziel</i>	Klassifikation, Approximation, Entscheidungsfindung	Klassifikation, Approximation, Entscheidungsfindung, Optimierung
Verarbeitung	<i>Mechanismus</i>	sequentielle heuristische Suche	parallele Berechnung
	<i>Geschwindigkeit</i>	niedrig	hoch
	<i>Berechnung</i>	global	lokal
Adaption	<i>Fehlertoleranz</i>	groß	sehr groß
	<i>Lernen</i>	extern durch Induktion, iterativ durch Experten	Anpassung von synaptischen Gewichten

Tab.1 Komplementarität von Neuronalen Netzen und Fuzzy Logik

Stärke der Verflechtung

Die Stärke der Verflechtung Neuronaler Fuzzy-Systeme sagt nichts über deren Leistungsfähigkeit aus. Der Grund, weshalb stark gekoppelte Systeme im allgemeinen dennoch einen stärkeren Reiz auf uns ausüben, liegt eher darin, daß in einem Neuro-Fuzzy-System die gesamten Vorteile von Neuronalen Netzen und Fuzzy Systemen auf ein Problem angewandt werden können. Im Fall eines nicht gekoppelten Systems wird auf ein Teilproblem ein Neuronales Netz angewandt und bei einem anderen Teilproblem ein Fuzzy-System eingesetzt. Die Vorteile beider Methoden können nicht die volle synergetische Wirkung entfalten. Trotzdem werden aus Gründen der Einfachheit mehr nicht gekoppelte Neuro-Fuzzy-Systeme eingesetzt als stärker miteinander verflochtene Systeme.

Nicht gekoppelte Neuro-Fuzzy-Systeme

Ein nicht gekoppeltes Neuro-Fuzzy-System besteht aus mindestens zwei völlig voneinander getrennten Teilen. Meist handelt es sich ohnehin um ein größeres, komplexes System, das aus vielen verschiedenen Komponenten zusammengesetzt ist. Ohne weitere Vorkehrungen ist hier bei der Verwendung eines Standard-Fuzzy-Systems dieses nicht adaptiv und das verwendete Neuronale Netz ohne zusätzliche Vorkehrungen nicht interpretierbar. Ein Beispiel für ein solches System ist die Regelung einer Getriebeübersetzung bei einem Auto mit Automatikgetriebe (Hayashi et al. 1993). Hier werden innerhalb des Gesamtsystems zwei Funktionen durch Fuzzy-Regeln approximiert. Eine Funktion schätzt die Last des Autos in Abhängigkeit von der Geschwindigkeit und der Beschleunigung der Antriebswelle. Eine andere Funktion soll die Absicht des Fahrers voraussagen, indem die Änderung des Gaspedals beobachtet wird. Schließlich wird im Gesamtsystem ein Neuronales Netz verwendet, das aus einer Reihe von Informationen (Last des Autos, Absicht des Fahrers, Veränderung des Gaspedals, Geschwindigkeit des Autos) den geeigneten Gang bei einem Fünfganggetriebe wählt. Zwei der Eingangsparameter des Neuronalen Netzes stammen von den Fuzzy-Systemen. Damit werden zwar die strukturellen Stärken der Fuzzy-Lösung und der adaptive Charakter der neuronalen Lösung an den Teilproblemen genutzt, wo strukturelles Wissen in Form von Regeln vorhanden ist bzw. adaptive Eigenschaften der Lösung erforderlich sind, aber auf eine volle Synergie beider Werkzeuge wird verzichtet.

Typisch für solche nicht gekoppelten Neuro-Fuzzy-Systeme ist der ad-hoc-Charakter des Zusammenspiels der Neuro- und Fuzzy-Teile. Die Beziehung zwischen ihnen kann nicht verallgemeinert werden und muß für jede Anwendung neu konzipiert werden.

Schwach gekoppelte Neuro-Fuzzy-Systeme

Bei schwach gekoppelten Neuro-Fuzzy-Systemen ist eine Beziehung zwischen den beiden Systemteilen vorhanden und kann auch in unterschiedlichen Anwendungen beibehalten werden. Ein solches System kann als Block angesehen werden, wobei jedem einzelnen Teil eine bestimmte Aufgabe zufällt. Häufig anzutreffende Konstellationen sind hier die Aufteilung in Vorverarbeitung, Hauptteil und eventuell Nachbearbeitung oder die Beeinflussung von Parametern des einen Systems durch das andere System. Im ersten Fall kann ein Fuzzy-System die Vorverarbeitung von Daten übernehmen, indem die rohen Eingangsdaten durch eine Fuzzifizierung in eine geeignetere und kompaktere Repräsentation übersetzt werden. Es folgt ein Neuronales Netz für die Hauptverarbeitung, dem schließlich ein Defuzzifizierungsalgorithmus nachgeschaltet sein kann. Beispiele dafür sind Bedzak und Daca (1995) oder teilweise Parten (1993), der für die Vor- bzw. Nachverarbeitung auch neuronale Realisierungen vorsieht. In diesem neuronalen Fall ist sein Ansatz dem eng gekoppelten Neuro-Fuzzy-System zuzuordnen. Den zweiten Fall mit der Parameteradaption eines Systems durch ein anderes haben wir dann, wenn beispielsweise die Parameter eines Fuzzy-Systems, wie z.B. die Breite oder Lage der involvierten Zugehörigkeitsfunktionen durch ein Neuronales Netz optimiert werden. Ein anderer Fall besteht dann, wenn ein System durch ein Neuronales Netz identifiziert wird, das wiederum ein Fuzzy-System beeinflusst. Zu dieser Kategorie zählt der Neuro-Fuzzy-Controller FAGNIS von Cechin und Eppler (1994), bei dem Fuzzy-Regeln aus einem Neuronalen Netz automatisch entwickelt werden. Solche Systeme sind schwach gekoppelte Neuro-Fuzzy-Systeme, weil sich die beiden Teile eng aufeinander beziehen und der eine ohne den anderen nicht verstanden werden kann. Sie sind nicht eng gekoppelt, weil das Neuronale Netz eindeutig vom Fuzzy-System getrennt ist und aus einer eigenständigen Implementierung besteht.

Eng gekoppelte Neuro-Fuzzy-Systeme

Bei eng gekoppelten Neuro-Fuzzy-Systemen kann nicht mehr zwischen einem neuronalen und einem Fuzzy-Teil unterschieden werden. Beide zusammen bilden einen Algorithmus, der sowohl Merkmale von Neuronalen Netzen als auch Fuzzy-Systemen aufweist. Eine Methode ist die direkte Implementierung von Fuzzy-Regeln in Neuronalen Netzen, wobei die Struktur und die Bestandteile (Neuronen, Lernalgorithmen) konventioneller Neuronaler Netze jeweils mehr oder weniger stark verändert werden. Beispiele hierfür sind die später behandelten Systeme von Eppler (1990), Jang (1991), Berenji (1992) oder Nauck (1993). Zu dieser Kategorie von Systemen gehören auch solche, die beispielsweise Neuronale Netze nur für die Entwicklung eines Systems verwenden, um hinterher das gelernte Wissen in einem Fuzzy-System zu implementieren (z.B. Kosko (1990)). Auch der umgekehrte Weg wurde beschritten. Ein Ansatz, der beide Richtungen zuläßt, ist in Eppler (1993) beschrieben. Andere Systeme ordnen Neuronale Netz-Teile so an, daß sie als Komponenten eines Fuzzy-Systems gedeutet werden können, d.h. sie bestehen aus einer Fuzzifizierung, Inferenz und Defuzzifizierung. Beispiele dafür sind von Hayashi (1989) oder Parten (1993) gegeben.

Ziel der Koppelung bei Neuro-Fuzzy-Systemen

Alle Neuro-Fuzzy-Systeme lassen sich mit mehr oder weniger Mühe in die oben vorgestellten drei Kategorien einordnen. Es können noch weitere Unterscheidungskriterien eingeführt werden, die beispielsweise die Systeme nach der Art der Fuzzy-Regeln zum Sugeno-Typ oder Mamdani-Typ oder nach der Art der Neuronalen Netze, d.h. nach den verwendeten speziellen Neuronen oder Lernalgorithmen einteilen. Diese Unterscheidungen sind aber weniger grundsätzlich. Wichtiger erscheint eine Einteilung in bezug auf die Anwendung des Neuro-Fuzzy-Systems zu sein, da meistens von dieser Richtung her die Entscheidung für das eine oder andere System getroffen wird. Dabei steht der Anwender selten vor der Situation, daß er bei der Implementierung ganz von vorne anfängt und in seinen Entscheidungen völlig frei ist. Oft existieren schon konventionelle Lösungen für das gestellte Problem, die aber in ihrer Qualität verbessert werden müssen, oder eine ähnliche Aufgabe wurde in einer verwandten Anwendung schon einmal gelöst und soll auf das bestehende Problem übertragen werden. Wenn Neuro-Fuzzy-Systeme eingesetzt werden sollen, sind die Hauptgründe dafür meist die Adaptionfähigkeit und die Verständlichkeit, also eine Interpretierbarkeit des Systems. Oft ist das eine gegeben, während das andere wünschenswert erscheint. So können Neuro-Fuzzy-Systeme in *adaptive propositionale* Systeme, d.h. das interpretierbare System soll adaptiv werden oder in *interpretierbare prozedurale* Systeme, d.h. das adaptive System soll interpretierbar sein, eingeteilt werden. Für das Verständnis des Systems ist es wichtig, welches dieser Ziele anvisiert wird. Manche Systeme streben beide Ziele gleichrangig an.

Adaptives propositionales System

Ein weit verbreiteter Glaube geht davon aus, daß Fuzzy-Systeme an sich schon adaptiv sind (z. B. CD-ROM-Lexikon von Bertelsmann unter dem Stichwort „Fuzzy Logik“). Wie wir bisher gesehen haben, ist dies nicht der Fall. Bei Fuzzy-Systemen wird immer wieder hervorgehoben, wie benutzerfreundlich das implementierte Wissen geändert werden kann. Bei einer Änderung der Umgebung des Systems können die Fuzzy-Regeln oft schnell

angepaßt werden. Dennoch gibt es viele Bereiche, in denen es wünschenswert ist, diese Anpassung automatisch, ohne Eingriff eines Experten, vorzunehmen. Hierfür muß eine automatische Methode gefunden werden, welche die einströmenden Daten soweit analysiert, daß entweder ganz neue Regeln entdeckt und alte gelöscht werden oder die bestehenden Zugehörigkeitsfunktionen abgeändert werden. Oft bestehen die Zugehörigkeitsfunktionen aus Basisfunktionen mit zwei oder mehr Parametern, aus denen alle zulässigen Funktionen bestimmt werden können (z.B. die Gaußfunktion als Basisfunktion mit den Parametern Mittelpunkt, Standardabweichung und Amplitude).

Ein Beispiel für ein propositionales System ist FAMOUS (Fuzzy Associative Memory Organizing Units System) von Ushida et al. (1994), das das Lernen auf einer konzeptuellen Ebene im Gegensatz zu der numerischen Ebene erreichen will. In einer beispielhaften Anwendung werden Gesichtsausdrücke erkannt und neu generiert. Die Verarbeitung geschieht mittels Fuzzy-Regeln, die Adaption der Regeln greift auf die neuronale und damit numerische Ebene zurück. Das Ziel des Systems ist es, die von menschlichen Experten entworfenen Regeln automatisch an die realen Systembedingungen anzugleichen. Andere Systeme, wie die später beschriebenen Ansätze ANFIS, FINE oder NNDFR fallen ebenfalls in diese Kategorie.

Ein großes Problem bei vielen Systemen dieser Kategorie ist die Erzeugung der Trainingsmuster. Werden Standard-Fuzzy-Regeln der Sugeno- bzw. Mamdani-Art verwendet, ist der Einsatzbereich des Neuro-Fuzzy-Systems meist in der Regelungstechnik. Die Eingänge sind dann Größen wie z.B. die Abweichung vom Sollzustand, die Änderung dieser Abweichung, usw. . Ausgänge sind Stellgrößen, die ein verändertes Verhalten der Regelstrecke hervorrufen. Da die Regelstrecke unbekannt ist, ist auch der Zusammenhang zwischen Stellgrößen und durch sie hervorgerufene Zustandsänderungen unbekannt. Soll der Regler, der aus dem Neuro-Fuzzy-System besteht, optimiert werden, müssen die Trainingsmuster sowohl den Systemzustand als auch die Stellgröße beinhalten. Die Stellgröße ist aber unbekannt. Es stellt sich also die Frage, wie sinnvolle Trainingsmuster generiert werden können. Eine Möglichkeit besteht darin, daß man sich die Muster von einem bereits gut arbeitenden Regler besorgt. So können z.B. Trainingsmuster erzeugt werden während ein Mensch eine Maschine regelt. Hinterher kann der mit diesen Mustern trainierte Neuro-Fuzzy-Regler eventuell diese Regelungsaufgabe übernehmen. Die derart konstruierte Regelung wird aber nie besser sein als die vorgegebenen Trainingsmuster! Ein bekanntes Beispiel ist das Rückwärts-Einparken eines Lastwagens mit Anhänger, das dem Fahrer viel Übung und Geschicklichkeit abverlangt. Zum erstenmal wurde dieses Beispiel von Nguyen und Widrow (1990) für einen rein neuronalen Regler angewandt, später löste Kosko (1992) mit einem Neuro-Fuzzy-Regler ein ähnliches Problem. Diese Art von Problemen ist allerdings bei Regelungsaufgaben, die es im allgemeinen mit sehr schnellen Vorgängen zu tun haben, selten. Die Automatisierung langsamer Prozesse, die von menschlichen Operateuren gesteuert werden, fällt meist nicht in das Gebiet der Regelungstechnik, sondern in die Automatisierungstechnik. Dort wird die Regelung im allgemeinen von Expertensystemen oder anderen wissensbasierten Verfahren vorgenommen. Für die übrigen, zu der Regelungstechnik zählenden Aufgaben, bleibt das Problem der Mustergenerierung.

Ein eleganter Ausweg zeigt sich, wenn ein Modell der Regelstrecke mit einbezogen wird. Dieses Modell ist zwar meistens unbekannt, kann aber durch adaptive Verfahren automatisch erzeugt oder verbessert werden. Hierzu bieten sich wieder insbesondere Neuronale Netze an. Für die Erzeugung der Trainingsmuster gibt es an dieser Stelle keine Probleme, da die Muster aus dem Zustand zur Zeit t und dem Folgezustand zur Zeit $t+1$ bestehen. Diese Größen sind leicht aus der realen Regelstrecke zu ermitteln. Liegt ein Modell der Regelstrecke vor, kann der Systemfehler am Ausgang berechnet und durch das Streckenmodell zurückpropagiert werden, so daß der Fehler der Stellgröße bei gegebenem Modell der Strecke ermittelt wird. Damit kann nun der eigentliche Neuro-Fuzzy-Regler optimiert werden. An der Stelle soll nicht weiter in diese Problematik eingedrungen werden; es finden sich auch hier noch zahlreiche Probleme ein, z.B. bei Systemen höherer Ordnung, die den Regelungsfehler nicht in einer Zeitstufe eliminieren können, oder der große Berechnungsaufwand bei hintereinandergeschaltetem Regler und Modell, usw. (siehe Cechin et al. 1994). Wichtig ist zu erkennen, daß Systeme ohne Einbeziehung eines Prozeßmodells nur über einen sehr eingeschränkten Anwendungsbereich verfügen. Das Prozeßmodell kann ohne tiefere Kenntnis der mathematischen Beschreibung des Modells aus den Meßdaten des vorliegenden Prozesses numerisch gelernt werden.

Interpretierbares prozedurales System

Die Interpretierbarkeit prozeduraler Systeme wird immer wieder dann gefordert, wenn an ein System sehr hohe Erwartungen an seine Zuverlässigkeit gestellt werden. Viele Entwickler gehen davon aus, daß Neuronale Netze in sicherheitsrelevanten Bereichen nicht eingesetzt werden können, weil niemand versteht, was in ihnen intern vorgeht. Einen Einblick in solche Systeme bekommt man noch am ehesten, wenn man verschiedene Datenmengen bearbeiten läßt und die Ergebnisse mit dem gewünschten vergleicht; ein Fehlermaß festlegt und den Abbildungsfehler danach bestimmt; oder mit Testmustergruppen die Generalisierungsfähigkeit des Systems feststellt. Die Struktur des Systems selbst wird dadurch aber nicht durchsichtiger, weshalb meist ein Rest von Mißtrauen in das System bleibt. Man muß sich allerdings klarmachen, daß dies sich bei anderen Software-Systemen nicht anders verhält. Die interne, meist nicht-hierarchische Struktur mit gegenseitigen Abhängigkeiten von Prozeduren und globalen Datenstrukturen überfordert den Entwickler und Anwender, so daß er sein System

nicht vollständig und im einzelnen versteht. Es ist einfach zu komplex. Er sieht sich aber grundsätzlich in der Lage, jede Stelle seines Systems mit entsprechendem Aufwand zu verstehen. Bei einem Neuronalen Netz sieht er dies nicht. Unsere These ist, daß dieser Unterschied auf die Sicherheit des Systems keinen Einfluß hat. Die mathematischen Gleichungen zu einem gegebenen Neuronalen Netz sind bekannt. Es macht keinen Unterschied, ob man ein Neuronales Netz mit n Verbindungsgewichten oder ein Optimierungsproblem n -ter Ordnung verstehen will. Sind aber einige Parameter überflüssig, redundant oder voneinander abhängig, sorgt eine Interpretation für ein erleichtertes Verständnis.

Es gibt aber noch einen weiteren Grund dafür, Neuronale Netze interpretierbar zu gestalten. Man möchte wissen, was das Neuronale Netz aus den Daten gelernt hat, um eventuell selber weitere Eingriffe in das System zu machen. Es ist utopisch anzunehmen, daß ein Neuronales Netz jede erdenkbare Struktur erfinden und den Daten anpassen kann. Meistens sind die Grenzen der Adaptionfähigkeit eines Neuronalen Netzes auf den ersten Blick sichtbar. Eine Interpretation des Netzes kann dann aber wertvolle Hinweise liefern, wie das Neuronale System geändert werden muß, um es besser an die Daten anpassen zu können. Das heißt, die Weiterentwicklung des Systems wird durch eine Interpretierbarkeit des Systems erleichtert. Bisher können uninterpretierte Neuronale Netze nur verbessert werden, wenn sie iterativ (durch „trial and error“) den verfügbaren Daten Schritt für Schritt angepaßt werden.

Ein System, bei dem die Interpretierbarkeit des Neuronalen Netzes im Vordergrund steht, ist FINE. Im Gegensatz zu den meisten anderen Systemen können beliebige Neuronale Netze interpretiert werden, die nicht nur einer bestimmten Form gehorchen wie z.B. ANFIS, das ebenfalls interpretierbar ist, aber kein allgemeines Neuronales Netz darstellt. Bei allgemeinen Neuronalen Netzen stellt sich allerdings das Problem, daß komplizierte Problemlösungen unübersichtlichen Beschreibungen entsprechen und im Extremfall die Interpretation des Netzes nicht mehr übersichtlicher ist als die Beschreibung des Netzes durch mathematische Formeln, mit denen das Netz implementiert ist. Dann ist es einfacher, Neuronale Netze zu verwenden, deren Struktur nicht so kompliziert werden kann, wie beispielsweise Radiale Basis-Funktionsnetze (RBF). Ein solcher Ansatz ist in Strefezza und Dote (1993) beschrieben. Ein Netz besitzt nur eine verdeckte Schicht von Neuronen, deren Aktivierungsfunktionen durch Gaußfunktionen beschrieben werden. Im einfachsten Fall der RBFs sind die Verbindungsgewichte alle auf 1 gesetzt und lediglich zwei Parameter der Gaußfunktion (Mittelwert und Standardabweichung) dem Lernalgorithmus unterworfen. Hier existieren also wesentlich weniger freie Parameter wie in einem sigmoiden Feedforward-Netz, was die Variationsvielfalt und damit auch die Interpretation des Netzes erleichtert.

Neuro-Fuzzy-Systeme

NNDFR (Hayashi 1989)

Das System NNDFR (Neural Network Driven Fuzzy Reasoning) von Hayashi besteht aus mehreren Neuronalen Netzen vom Typ des mehrschichtigen Perzeptrons. Jede Fuzzy-Regel besteht aus einem eigenen Neuronalen Netz, zusätzlich werden die Bedingungsteile aller Regeln in einem Netz ausgewertet. Die Fuzzy-Regeln sind von folgender Form:

$$\text{if } f_i(x_1, x_2, \dots, x_n) = A_i \text{ then } y_i = u_i(x_1, x_2, \dots, x_n)$$

Das Blockdiagramm von NNDFR ist in Abb. 1 gezeigt. Die Funktion f_i wird nicht, wie sonst üblich, durch eine T-Norm eingeschränkt, sondern ganz allgemein durch das Neuronale Netz NN_{cond} angegeben. NN_{cond} bekommt als Eingabe die n Eingangswerte und liefert als Ausgabe r Zugehörigkeitswerte zu r Fuzzy-Prädikaten A_r . Die Anzahl der Regeln r muß fest vorgegeben werden. Hayashi geht von einer vollständigen Trainingsmustermenge aus, die von einem anderen Regler stammen muß (siehe oben: Problem der Generierung der Trainingsmuster). Von diesen Daten verwendet er zunächst nur den Teil mit den Eingabewerten und klassifiziert sie durch einen konventionellen Cluster-Algorithmus. Jeder Cluster entspricht dann einer Fuzzy-Menge. So kann er jedes einzelne Muster einem von r Fuzzy-Prädikaten zuordnen. Mit dieser neu gebildeten Mustermenge trainiert er das Netz NN_{cond} . Anschließend werden die Konklusionen NN_i der Regeln trainiert. Dazu teilt Hayashi die vollständige Mustermenge in die einzelnen Cluster auf und trainiert für jeden Cluster ein eigenes Neuronales Netz. Am Schluß kann noch eine Optimierung durchgeführt werden, die auf ad-hoc-Methoden basiert: einzelne Eingangsparameter werden zufällig weggelassen und dann gestrichen, falls die Ausgabe nahezu unverändert bleibt. Die Ausgänge der Konklusionsnetze werden mit den Ausgängen des Bedingungsnetzes NN_{cond} multipliziert und die Resultate der r Regeln aufaddiert. Diese Operation entspricht der Defuzzifizierung bei Fuzzy-Reglern und liefert die auszugebende Stellgröße.

Der Einfluß der Fuzzy-Logik ist bei diesem Ansatz sehr gering. Sie gibt lediglich die äußere Struktur des Systems vor, wobei der Autor leider verschweigt, worin die Vorteile dieser Struktur liegen. Ohne die Verwendung der Fuzzy-Schreibweise berechnet sich die Ausgabe des Systems durch

$$y = \sum_{i=1}^r NN_{cond,i}(x_1, \dots, x_n) \cdot NN_i(x_1, \dots, x_n)$$

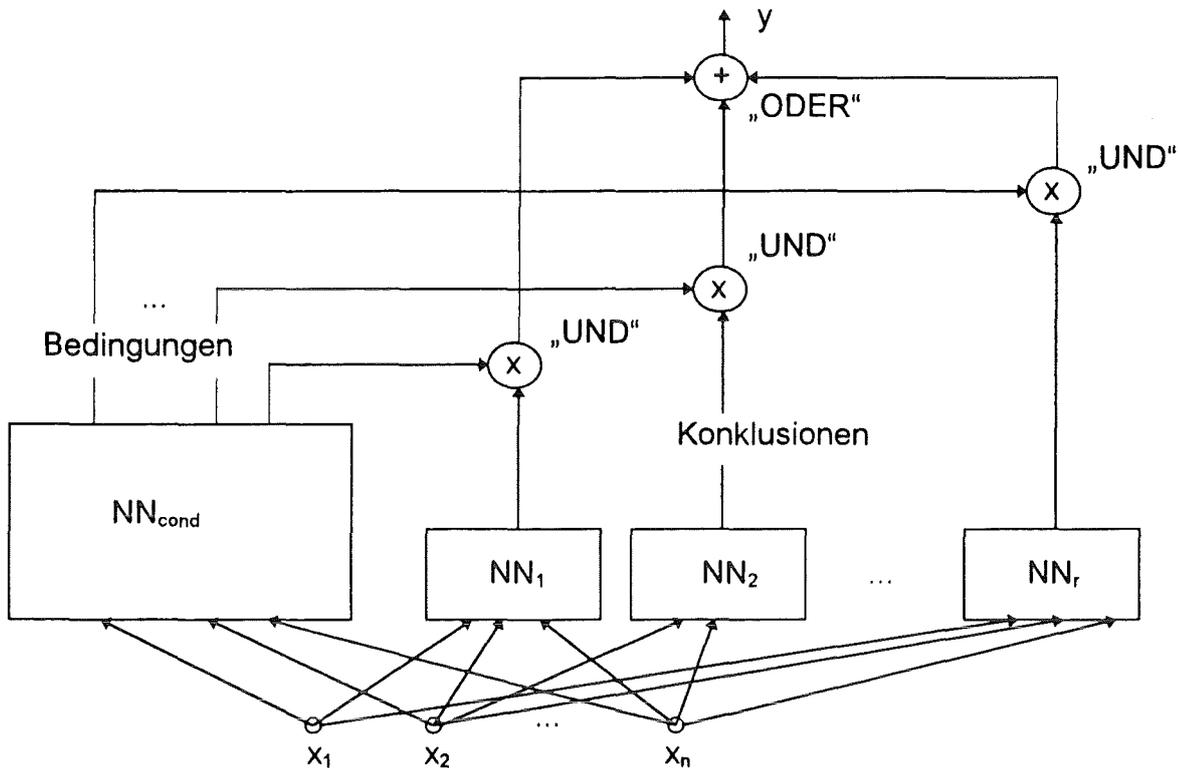


Abb.1 Blockdiagramm von NNDFR

Das System wurde bisher zur Regelung eines balancierten Stabs eingesetzt, aber auch in Klassifizierungsaufgaben verwendet (Takagi und Hayashi 1991). Es wurden dabei Sauerstoffkonzentrationen in der Bucht von Osaka vorhergesagt und die Rauheit von polierten keramischen Oberflächen abgeschätzt.

FINE (Eppler 1990)

Das Neuro-Fuzzy-System FINE (Fuzzy Initialized Neural Engine) besteht hier aus einem Feedforward-Netz mit unterschiedlich vielen Schichten (Abb.2). Alle Neuronen sind vom selben Typ. Als Aktivierungsfunktion wird die Sigmoid-Funktion verwendet. Die erste Schicht des Netzes repräsentiert die Eingangsvariablen des Systems. Es stehen zwei Möglichkeiten offen, diese Variablen im Neuronalen Netz zu implementieren.

(a) In der kontinuierlichen Kodierung wird jede Variable durch genau ein Neuron repräsentiert. Der Wertebereich der Variablen wird auf das Intervall $[0,1]$ normiert. Somit besitzen alle Eingangsvariablen denselben Wertebereich, was sich wegen der stärkeren Unabhängigkeit von den initialisierten Gewichtswerten auf die Lerngeschwindigkeit günstiger auswirkt. Die zweite und die dritte neuronale Schicht repräsentieren Fuzzy-Prädikate, die aus Sigmoidfunktionen zusammengesetzt werden. Sigmoid-förmige Fuzzy-Zugehörigkeitsfunktionen, wie sie etwa für die Prädikate „sehr klein“ bzw. „sehr groß“ gebraucht werden, können in einer neuronalen Schicht mit einem Neuron realisiert werden. Gaußförmige Funktionen, wie beispielsweise für das Prädikat „mittel groß“, benötigen zwei Schichten mit insgesamt drei Neuronen (siehe Abb. 2 rechts oben, wo Neuronen die Fuzzy-Prädikate „Links“, „Mitte“, „Rechts“ repräsentieren). Das dritte Neuron subtrahiert dabei eine Sigmoidfunktion mit größerem Schwellwert von einer mit kleinerem Schwellwert. Die Ausgänge der zweiten bzw. dritten Schicht liefern die Zugehörigkeitswerte, die sich ergeben, wenn das dem Neuron entsprechende Fuzzy-Prädikat auf eine Eingangsvariable angewandt wird. Diese Neuronen führen also eine Fuzzifizierung durch.

(b) Eine zweite Möglichkeit für die Kodierung der Eingabevariablen ist die sogenannte topologische Kodierung. Hier werden pro Variable mehrere Neuronen verwendet, so daß jedes Neuron für einen bestimmten Bereich des Wertebereichs der Variablen steht. Der Vorteil dieser Anordnung wirkt sich hauptsächlich beim Trainieren des Neuronalen Netzes aus. Diese Kodierung ist wesentlich flexibler als die kontinuierliche Kodierung und verfängt sich auch nicht so schnell in lokalen Minima. Die topologische Kodierung kann als Vorstufe zur Repräsentation der Eingabe mit Fuzzy-Prädikaten angesehen werden. Dazu ist es hilfreich, sich die Fuzzifizierung der zweiten neuronalen Schicht nicht als eine Operation in bezug auf ein Fuzzy-Prädikat vorzustellen, sondern die gesamte Schicht als transformierte Repräsentation der Eingabe anzusehen. In dieser Sichtweise kann dann eine topologische Kodierung als Repräsentation der Eingabe gesehen werden, deren einzelne Neuronen als Fuzzy-

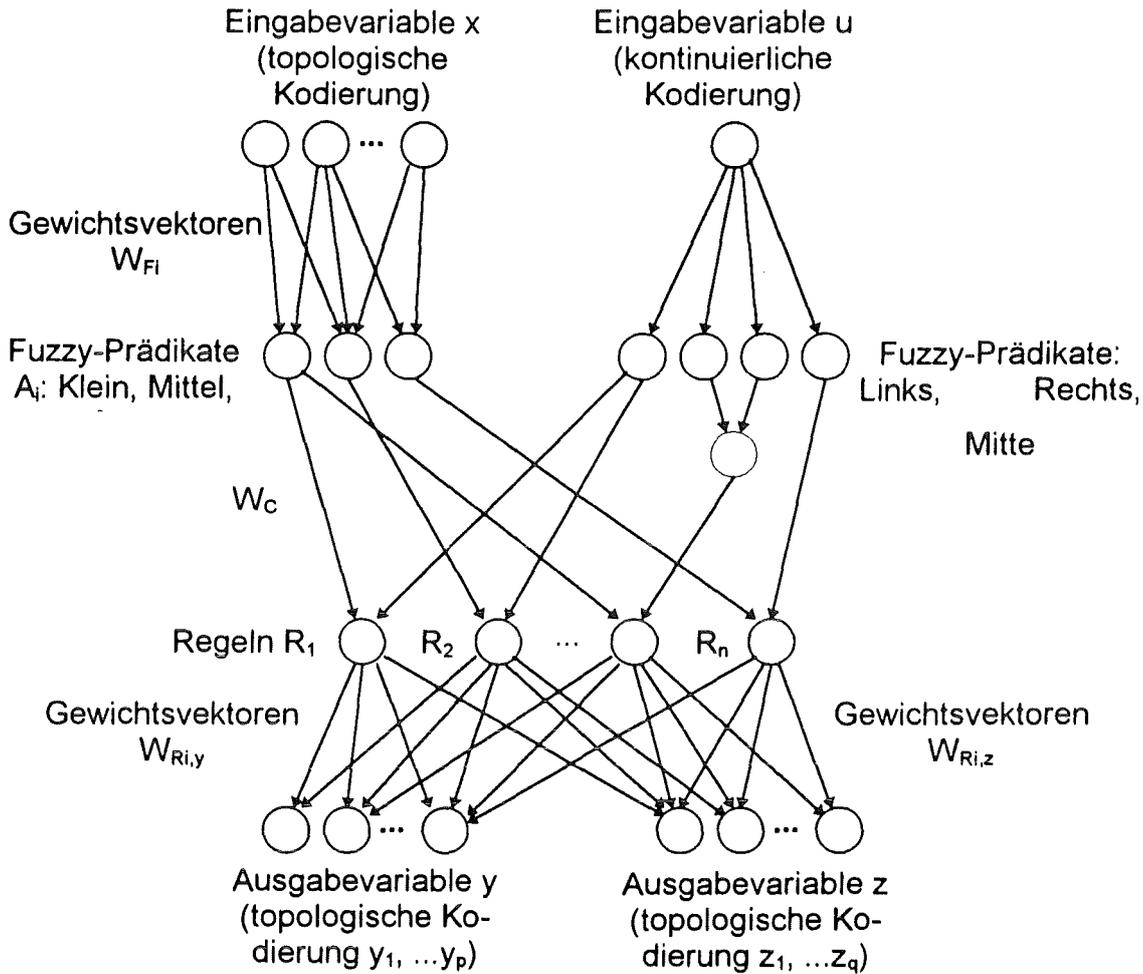


Abb.2 Struktur von FINE

Prädikate interpretiert werden können. Angenommen, wir haben eine Variable mit dem Wertebereich $[0,10]$ und repräsentieren diesen Bereich mit 10 Neuronen. Dann wird der Wert 2.77 durch folgenden Vektor repräsentiert: $(0 \ 0.23 \ 0.77 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0)$. Das Fuzzy-Prädikat „2“ besitzt hier eine Zugehörigkeitsfunktion, die im Intervall $[1,2]$ von 0 nach 1 linear ansteigt und im Intervall $[2,3]$ von 1 nach 0 linear abfällt, also eine Dreiecksfunktion darstellt. Die Weiterverarbeitung in der nächsten Schicht geschieht etwas anders als im Falle der kontinuierlichen Kodierung. In dieser Schicht sind die Fuzzy-Prädikate der Regelbedingungen repräsentiert. Sie benötigen nur eine Schicht, unabhängig davon, ob sie eine sigmoide oder Gauß-förmige Zugehörigkeitsfunktion verwenden. Entscheidend ist, daß sie folgender Gleichung genügen:

$$\mu_A(x) = \sigma(x \cdot w_F),$$

wobei $\mu_A(x)$ die Zugehörigkeitsfunktion des Prädikats A, angewandt auf die Muster x ist, $\sigma(\cdot)$ die Aktivierungsfunktion eines Neurons darstellt und w_F der Gewichtsvektor zwischen den Eingangsneuronen x_i und dem Neuron A ist. Wenn x in der topologischen Kodierung vorliegt, existiert genau ein i mit $x_i=1$. Für alle $j \neq i$ ist $x_j=0$. Damit ist die Aktivierungsfunktion $\sigma(\cdot)$ umkehrbar, denn es gilt:

$$x \cdot w_F = x_i \cdot w_{Fi} = w_{Fi}$$

und

$$\mu_A(x_i) = \sigma(w_{Fi}).$$

Die Umkehrung lautet dann:

$$w_{Fi} = \sigma^{-1}(\mu_A(x_i)).$$

In einer weiteren Schicht, welche die Regeln des Systems darstellt, werden die aus Schicht zwei bzw. drei resultierenden Fuzzy-Werte nach den Bedingungsanteilen der Regeln miteinander verknüpft. Die resultierende Aussteuerung des Bedingungsanteils einer Regel liefert den Grad $\mu_{Cond,i}$, mit dem eine Regel erfüllt ist:

$$\mu_{Cond,i} = \sigma \left(\sum w_{C,i} \mu_{A,i} - Bias \right).$$

(z.B. $w_{C,i}=4$, $Bias=6$ entspricht „UND“ bei zwei Eingängen). Oft sind die Verknüpfungen Fuzzy-Konjunktionen, es können jedoch beliebige andere logische Funktionen ebenfalls verwendet werden. Selbst XOR-Verknüpfungen sind hier möglich, wobei dann zwei neuronale Schichten verwendet werden müssen. Als Fuzzy-Konjunktionen können alle diejenigen Operatoren benutzt werden, welche die T-Norm erfüllen. Gebräuchliche T-Normen sind die Multiplikation und die Minimumbildung. Ein Gesetz der T-Norm ist die Linearität, d.h. $T(1,a)=a$. Wenn ein Eingang des Gatters, soll das Ergebnis dem zweiten Eingangswert entsprechen. Beim vorliegenden System wurde darauf geachtet, ein homogenes Neuronales Netz zu verwenden. Alle Neuronen sind deshalb vom selben Typ und verwenden dieselbe nichtlineare Aktivierungsfunktion. Aus diesem Grund kann das Gesetz der Linearität hier nicht gelten. Die Nichtlinearität kann allerdings in der Nähe des Schwellwerts der Aktivierungsfunktion in Grenzen gehalten werden. In Grenzen bedeutet, daß die Aktivierungsfunktion in einem Bereich um den Bias eine lineare Funktion sehr gut annähert. Darüberhinaus wirkt sich die Abweichung von der Linearität im Regelteil von Standard-Fuzzy-Controllern kaum aus, da meistens nur ein bis zwei logische Stufen hintereinander geschaltet werden.

In der nächsten neuronalen Schicht befinden sich die Fuzzy-Prädikate der Regel-Konklusionen. Diese Prädikate sind mit den Variablen der Ausgangsschicht verbunden. Die Kopplung zwischen diesen beiden Schichten hat Ähnlichkeit mit der Kopplung der ersten mit der zweiten Schicht, wenn die topologische Kodierung verwendet wird. Die Ausgangsvariablen werden in der weiter oben geschilderten topologischen Kodierung dargestellt. Die Beziehung zwischen einem Fuzzy-Prädikat B einer Regelkonklusion und der zugehörigen Variablen ist

$$\mu_B(y) = \sigma(\mu_{Cond}(x) \cdot w_R).$$

In dieser Beziehung taucht das neue Fuzzy-Prädikat B' auf, das nur in dem Fall mit B identisch ist, wenn $\mu_{Cond}(x) = 1$ ist. $\mu_{Cond}(x)$ gibt an, mit welchem Grad die Bedingung einer Regel erfüllt ist. Im Falle eines einzigen Fuzzy-Prädikats A im Bedingungsanteil einer Regel gilt $Cond = A$, im Falle der Konjunktion mehrerer Fuzzy-Prädikate A_i ergibt sich Cond aus der Schnittmenge dieser Prädikate. Weiterhin - aber dieser Fall wird erst später dargestellt - kann B' aus einer Überlagerung mehrerer Prädikate B_1, \dots, B_n bestehen, wenn nämlich mehrere Regeln (im allgemeinen zu unterschiedlichen Graden) gleichzeitig aktiv sind. Der Gewichtsvektor w wird am einfachsten bestimmt, wenn für den Augenblick $\mu_{Cond}(x) = 1$ angenommen wird. Obige Gleichung vereinfacht sich dann zu

Diese Beziehung beschreibt den Fall, daß nur eine Regel R_j die Ausgabevariable y aussteuert, d.h. für die volle Erfüllung der Regel R_j $\mu_{Cond}(x)=1$ gilt:

$$\mu_B(y_k) = \sigma(w_R),$$

so daß sich die Gewichte aus

$$w_R = \sigma^{-1}(\mu_B(y_k)).$$

ergeben. y_k sind die Stützpunkte der topologischen Kodierung von y. Durch diese Formel sind die Gewichte w_R über die Fuzzy-Zugehörigkeitsfunktion $\mu_B(y_k)$ der Ausgangsvariablen y gegeben. Damit wird die Kompilierung einer vorgegebenen Fuzzy-Ausgangsvariablen in die Parameter eines Neuronalen Netzes ermöglicht. Im Falle der Aktivierung mehrerer Regeln R_j , die darüber hinaus im allgemeinen zu unterschiedlichen Graden gleichzeitig aktiv sind, gilt:

$$B' = B_1 + \dots + B_n$$

und

$$\mu_{B'}(y) = \sigma(\mu_{Cond_1}(x) \cdot w_{R,1} + \dots + \mu_{Cond_n}(x) \cdot w_{R,n})$$

wobei die einzelnen Gewichtsvektoren unabhängig voneinander nach obigem Verfahren berechnet werden.

Die Umsetzung der Fuzzy-Regeln in die Gewichte eines Neuronalen Netzes bewirkt ein Neuronaler Fuzzy-Compiler. Als Eingabe erhält er eine Datei, die aus vier Blöcken besteht: Definition der linguistischen Variablen und der Fuzzy-Prädikate; Auflistung der Fuzzy-Regeln; Definition der Ein- und Ausgabevariablen; Repräsentation der Variablen im Neuronalen Netz. Ein Beispiel für die Definition der linguistischen Variablen „Temp“ mit ihrem Wertebereich [425, 625] und einem darauf sich beziehenden Fuzzy-Prädikat „Normal“ mit einer dreiecksförmigen Zugehörigkeitsfunktion (Werte im Bereich kleiner als 500 sind 0, im Bereich größer als 550 sind sie 1, im Bereich [500, 525] steigt die Funktion von 0 auf 1 linear an, im Bereich [525, 550] fällt sie linear) ist:

```

VAR Temp MIN 425 MAX 625 # Temperatur in Grad Celsius
PRED Normal POINTS 500, 525, 525, 550 # Prädikat Normal mit Maximum bei 525

```

Ein Beispiel für eine Regel aus der Regelbasis ist:

```

IF Temp IS Niedrig OR Druck IS Tief THEN U IS Negativ

```

Für die Implementierung im Neuronalen Netz muß für jede Variable definiert werden, mit welcher Anzahl an Neuronen sie im Neuronalen Netz repräsentiert sein soll (im folgenden Beispiel: NUM). Die Kodierung kann entweder linear oder logarithmisch erfolgen, wobei beim Logarithmus das Zentrum und die Stärke der Dehnung anzugeben ist. Um beim späteren Lernvorgang ein besseres Lernverhalten zu erzielen, können pro Prädikat oder Regel mehrere Neuronen vorgesehen werden (im Beispiel: RED = Zahl der Neuronen).

```

VAR Temp NUM 100 # 100 Neuronen für Variable Temp
LOGARITHM CENTER 525 VARIANCE 0.25 # logarithmische Kodierung
RED Temp: Hoch 5 # 5 Redundanzneuronen für Prädikat
# Hoch der Variablen Temp

```

Der Neuronale Fuzzy-Compiler erzeugt drei Dateien als Ausgabe, die vom Neuronalen Fuzzy-Controller unmittelbar benutzt werden. Eine Datei gibt die Topologie des Netzes an, eine weitere enthält die Verbindungsgewichte und die dritte Datei enthält Informationen für die Ein-/Ausgabe des Reglers. Die beispielsweise mit Backpropagation optimierten Gewichte lassen sich anhand der angegebenen Formeln wieder in ein Fuzzy-Entscheidungssystem zurücksetzen.

ANFIS (Jang 1991)

Das von Jang 1991 vorgestellte Modell ANFIS (Adaptive Network based Fuzzy Inference System) ist ein Beispiel für ein eng gekoppeltes Neuro-Fuzzy-System. Fuzzy-Regeln werden hier sehr direkt in einem Neuronalen Netz implementiert. Jang verwendet Fuzzy-Regeln der Sugeno-Form, d.h. die Bedingung einer Regel enthält ein oder mehrere Fuzzy-Prädikate, die Konklusion einer Regel besteht aus einer linearen Verknüpfung der Eingangsvariablen:

Typische Regel: $\text{if } x_1=A \text{ and } x_2=B \text{ then } y=a_1x_1+a_2x_2+a_3.$

Das Neuronale Netz besteht aus sechs Schichten, die allerdings nicht homogen aufgebaut sind (siehe Abb. 3). Die Neuronen verschiedener Schichten verwenden jeweils verschiedene Aktivierungsfunktionen. Bisher galt für die Aktivierung eines Neurons folgende Funktion:

$$f(x_1, x_2, \dots, x_n) = \frac{1}{1 + e^{-\sum_{j=1}^n w_j x_j + \theta_j}}$$

Jetzt sind es je nach Schicht verschiedene Aktivierungsfunktionen, wie

$$f(x) = \frac{1}{\sqrt{2\pi} \sigma} e^{-\frac{1}{2} \frac{(x-m)^2}{\sigma^2}}$$

oder

$$f(x_1, x_2) = x_1 \cdot x_2.$$

Die erste Schicht implementiert die Eingangsvariablen, deren kontinuierlicher Wertebereich von den Eingangsneuronen unmittelbar repräsentiert wird. Die zweite Schicht besteht aus Neuronen, welche die Fuzzy-Zugehörigkeitsfunktionen von Fuzzy-Prädikaten realisieren. Diese Zugehörigkeitsfunktionen entsprechen direkt den Aktivierungsfunktionen der Neuronen. Es sind Gauß-Funktionen, die mit zwei Parametern versehen sind: dem Mittelpunkt und der Breite der Funktion. Die Neuronen der zweiten Schicht sind mit denen der ersten nicht voll vernetzt, sondern sind immer nur mit einem Eingangsneuron verknüpft. Dies entspricht der Tatsache, daß der Wertebereich einer Eingangsvariablen auf die Fuzzy-Prädikate der zweiten Neuronen-Schicht abgebildet wird. Die Schwelle m in der Gaußfunktion bestimmt dabei die Zentrierung des Prädikats auf der Eingangsvariablen. Wird der Wertebereich einer Eingangsvariablen x in die Fuzzy-Prädikate „klein“ bzw. „groß“ zergliedert, so gibt es genau zwei neuronale Verbindungen von dem Eingabeneuron x zu zwei verdeckten Neuronen der zweiten Schicht. Die Ausgaben dieser Schicht entsprechen den Zugehörigkeitswerten der Eingabe in bezug auf ein bestimmtes Fuzzy-Prädikat. Zum Beispiel liefert das erste Neuron der zweiten Schicht, das für das Fuzzy-Prädikat „klein“ steht, als Ergebnis den Zugehörigkeitswert $\mu_{\text{klein}}(x)$. Die dritte Schicht besteht aus den Regelneuronen, die eine lineare Aktivierungsfunktion verwenden. Diese Schicht muß so groß sein, daß sämtliche Kombinationen der

Fuzzy-Prädikate der verschiedenen Eingangsvariablen eine eigene Regel und damit ein eigenes Neuron darstellen. Die aus der vorhergehenden Schicht resultierenden Zugehörigkeitswerte werden in der dritten Schicht lediglich multipliziert. Es folgt eine Schicht, die mit der vorhergehenden Schicht voll vernetzt ist. Die Verbindungen zwischen den Neuronen dieser Schichten geben das relative Gewicht der Regeln an, so daß Neuronen, die genauso wie ihre Vorgängerneuronen für einzelne Regeln stehen, teilweise unterdrückt werden. Jedes Neuron dieser Schicht ist mit genau einem Neuron der fünften Schicht verbunden. Dort wird der Konklusionsteil der Regel berechnet und mit dem Gewichtswert des vorausgehenden Neurons multipliziert. Das letzte Neuron (einziges Neuron der sechsten Schicht) führt lediglich noch eine Summation über die Neuronen der vorhergehenden Schicht aus. Eine Defuzzifizierung ist bei Regeln des Sugeno-Typs nicht nötig. Das Lernen des Reglers geschieht nur in einzelnen Schichten, die richtigen Zugehörigkeitsfunktionen müssen von Anfang an bekannt sein, während die Parameter a_1, a_2, a_3 der Konklusionen gelernt werden.

BIOFAM (Kosko 1992)

Das BIOFAM-System (Basic Input Output Fuzzy Associative Map) ist nur eines unter vielen, das Kosko in einem Buch (Kosko 1992) vorschlägt. Es ist allerdings das einzige System, das Kosko (und später auch andere) auf Anwendungen ansetzt und soll deshalb hier berücksichtigt werden.

BIOFAMs führen eine Abbildung von einem einzelnen Eingabewert auf eine Fuzzy-Verteilung durch (siehe Abb. 4). Für jeden Term der Regelprämissen wird ein eigenes FAM (Fuzzy Associative Map) gebildet, das eine Regel bis auf die Verknüpfung der Prämissenterme repräsentiert. Kosko verknüpft die Ergebnisse hinterher außerhalb der FAMs durch eine Minimumbildung und erreicht so denselben Effekt, wie wenn er die Prämissenterme von vornherein miteinander verknüpft hätte. Dies geschieht für jede der Regeln getrennt, so daß die Ergebnisse der Minimumbildung sämtlicher Regeln anschließend überlagert werden müssen. Dies erledigt BIOFAM durch eine Addition.

Da es zunächst nicht so nahe liegt, logische Verknüpfungen in der Prämisse der Fuzzy-Regeln durch Verknüpfung der Regelkonklusionen zu lösen, soll diese Vorgehensweise etwas genauer erläutert werden. Die Regel

$$\text{if } (x=A) \text{ and } (y=B) \text{ then } (z=C)$$

wird in BIOFAM durch

$$(\text{if } x=A \text{ then } z=C) \text{ and } (\text{if } y=B \text{ then } z=C)$$

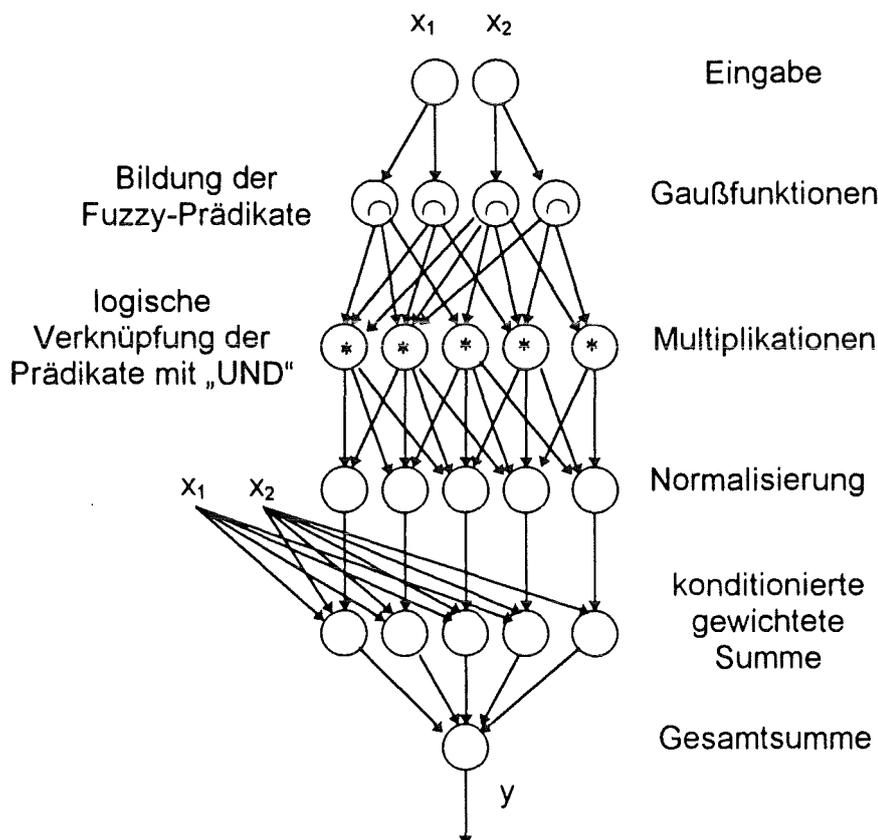


Abb. 3 Struktur von ANFIS

gelöst. An dieser Stelle zeigt sich deutlich, wie sich die Fuzzy-Inferenz „if A then C“ von der klassischen logischen Implikation „A->C“ unterscheidet. Mit der logischen Implikation führt die obige Umformung zu einem Widerspruch, wie man leicht an der Belegung (A wahr, B falsch, C falsch) sehen kann. Wird dagegen die Fuzzy-Inferenz durch eine T-Norm (entspricht „A and C“) gebildet, stimmt die Umformung.

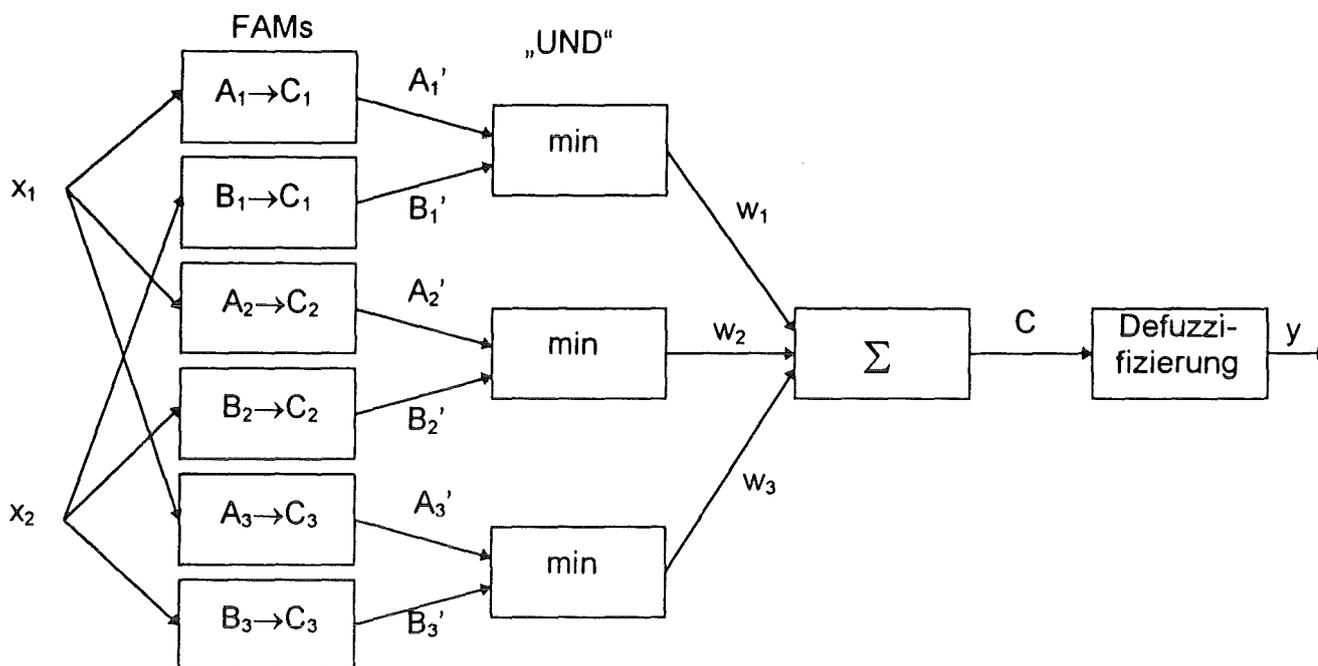


Abb. 4 Struktur von BIOFAM

Die FAMs können dabei durch zwei verschiedene Algorithmen gebildet werden. Einer davon verwendet ein Neuronales Netz. Kosko teilt zunächst für beide Algorithmen den Wertebereich des Eingabe- und des Ausgaberaums seines Fuzzy-Systems in n bzw. m verschiedene Intervalle ein. Diese Intervalle entsprechen den Fuzzy-Prädikaten. Die Ränder der Intervalle sind scharf, d.h. ihnen entsprechen rechteckige Zugehörigkeitsfunktionen. Für den ersten Algorithmus wird eine mehrdimensionale Matrix definiert, wobei jede Eingangsvariable und jede Ausgangsvariable einer Dimension entspricht. Die Dimensionen sind in die angegebenen Intervalle unterteilt. Die Daten der Mustermenge sind Punkte in diesem Raum und belegen jeweils ein Feld in der Matrix. Der Algorithmus zählt lediglich die Muster pro Matrixfeld, so daß die Matrix Häufigkeiten angibt, wie oft die Daten in bestimmten Intervallen (=Fuzzy-Prädikat) liegen. Die k häufigsten Felder legen k Fuzzy-Regeln fest. Der zweite Algorithmus ist ein neuronaler Lernalgorithmus (DCL - Differential Competitive Learning), der dem Competitive Learning von Grossberg (1976) stark ähnelt. Eingänge des Neuronalen Netzes sind eine Eingangs- plus die Ausgangsvariable des Systems (bei mehreren Ausgangsvariablen müssen diese unabhängig voneinander in mehreren Systemen implementiert werden), Ausgänge des zweischichtigen Netzes sind die einzelnen Felder der Matrix und damit die Abbildungsregeln des Systems. Wird die Zahl der Ausgangsneuronen von vornherein auf k begrenzt, bilden sich nur die k „stärksten“ Regeln heraus. Es ist zu beachten, daß diese „starken“ Regeln durch größere Datenmengen in einem Bereich zustande kommen. Besonders in der Regelungstechnik ist dies aber nicht gleichbedeutend mit „wichtigen“ Regeln, die auch in Bereichen mit wenigen Datenpunkten liegen können. Kosko muß deshalb seine automatisch gefundenen Regeln teilweise mit Regeln aus wenig repräsentierten Datenbereichen ergänzen. Durch den Lernalgorithmus nehmen die Gewichtsvektoren zwischen diesen beiden Schichten die Mittelpunkte von Clustern häufig vorkommender Muster an. In etwa entsprechen sie den k häufigsten Feldern der Matrix des zuerst angegebenen Algorithmus.

GARIC (Berenji und Khedar 1992)

Das GARIC-System (Generalized Approximate Reasoning based Intelligent Control) stammt von Berenji und Khedar (1992). Es gehört zu den eng gekoppelten Neuro-Fuzzy-Systemen und verwendet für die Implementierung eines Fuzzy-Systems mehrere Neuronale Netze (siehe Abb. 5). Logisch ist das System in zwei Teile geteilt: das Aktionsnetz (ASN - Action Selection Network) und das Bewertungsnetz (AEN - Action-State Evaluation Network). Das ASN ist ein Fuzzy-Regler aufgebaut aus speziellen Neuronen in fünf Schichten, deren Verbindungen ohne Gewichte fest verdrahtet sind. Die Eingangsschicht besteht aus Neuronen, welche die Eingangsvariablen repräsentieren. Die zweite Schicht besteht aus Fuzzy-Prädikaten, wobei der Ausgang eines Neurons der Zugehörigkeit eines Eingabewertes zu dem betreffenden Fuzzy-Prädikat angibt. Die Aktivierungsfunktion entspricht also der Zugehörigkeitsfunktion des Fuzzy-Prädikats. Wie im ANFIS-System

bestimmen drei Parameter die Form der Zugehörigkeitsfunktion. Allerdings werden hier keine Gauß-Funktionen verwendet, sondern Dreiecksfunktionen. Die Parameter geben den Punkt des Maximums, die Breite links vom Maximum und die Breite rechts vom Maximum an. Die dritte Schicht enthält die Regeln, welche die verschiedenen Bedingungssteile miteinander kombiniert. Logisch gesehen handelt es sich hier um ein logisches UND, das in der Fuzzy-Theorie mit der T-Norm umrissen wird. Kandidaten dafür sind üblicherweise die Minimum-Funktion oder die Multiplikation. Berenji definiert hier eine neue Funktion SOFTMIN, die er wegen ihrer Differenzierbarkeit (diese erfordert der Lernalgorithmus) der reinen Minimum-Funktion vorzieht:

$$SOFTMIN(x_1, \dots, x_n) = \frac{\sum_{i=1}^n x_i e^{\alpha x_i}}{\sum_{i=1}^n e^{\alpha x_i}}$$

SOFTMIN enthält einen freien Parameter α , der im einen Extrem ($\alpha=0$) dem arithmetischen Mittel entspricht und am anderen Ende ($\alpha=\infty$) die gewöhnliche Minimum-Funktion approximiert. Diese Regelschicht ist sowohl mit der nächsten als auch übernächsten Schicht verbunden. Die nächste Schicht enthält die Fuzzy-Prädikate des Konklusionsteils der Regel. Im GARIC-System wird auf die aufwendige Defuzzifizierung verzichtet und die Fuzzy-Zugehörigkeitsfunktionen der einzelnen Regeln werden nicht miteinander überlagert, um anschließend einen einzelnen Wert zu ermitteln. Sondern für jede Regel wird gesondert ein scharfer Wert aus der Fuzzy-Zugehörigkeitsfunktion der Konklusion bestimmt (in Schicht 4). Diese Prozedur wird von Berenji LMOM (Local Mean Of Maximum) bezeichnet. Das Neuron der letzten Schicht berechnet eine gewichtete Summe der Aktivitäten aus den beiden vorangegangenen Schichten und gibt das Aktionssignal auf das System. Hier ist die Version von Segers et al. (1994) wiedergegeben. Berenji und Khedar (1992) fügen vor der Ausgabe der Aktion noch eine Zufallsvariable ein, die hier im weiteren vernachlässigt wird.

Das Bewertungsnetz besteht aus einem Neuronales Netz mit einer verdeckten Schicht, wobei die Eingangsschicht zusätzlich direkt mit der Ausgangsschicht verbunden ist (sog. shortcuts). Dieses Netz liefert ein Verstärkungssignal, das sowohl für die Ausgabe des Systems verwendet wird als auch zum Lernen des Systems beiträgt. Das Verstärkungssignal, das noch durch eine externe Verstärkung modifiziert wird, übernimmt während des Lernverfahrens die Rolle des (rückpropagierten) Fehlers, womit die Parameteränderungen der Neuronen in AEN und ASN bestimmt werden. Damit hat das Bewertungsnetz AEN die Rolle eines Modells für die Zustandsfunktion bzw. den Fehler des Systems.

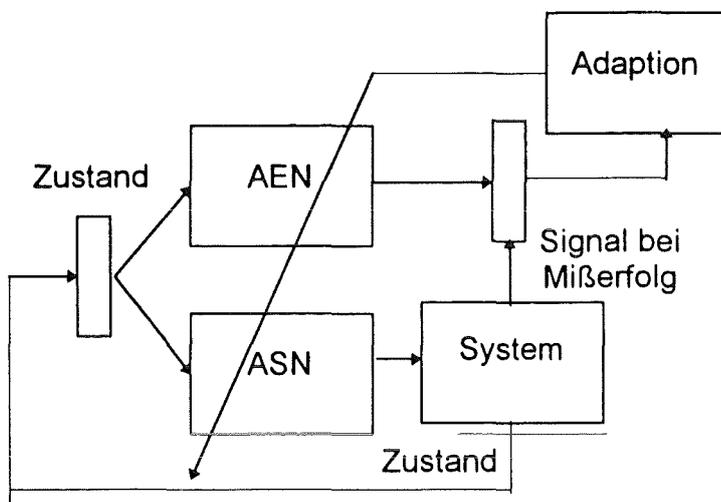


Abb.5 Vereinfachte Struktur von GARIC, mit AEN (Action Evaluation Network) und ASN (Action Selection Network), nach Segers et al. (1994)

NEFCON (Nauck 1993)

NEFCON (NEural Fuzzy CONTroller) ist ein dreischichtiges Neuronales Netz mit Spezialneuronen. Die Besonderheit stellen die Gewichte dar, die keine Zahlenwerte, sondern Fuzzy-Mengen sind. Sie sind teilweise gekoppelt, d.h., wenn durch den Lernalgorithmus Gewichtsänderungen durchgeführt werden, verändern sich solche Gewichte um den gleichen Betrag (siehe Abb. 6). Die Fuzzy-Prädikate der Ausgangsschicht des Neuronales Netzes (d.h. der Konklusionen der Regeln) müssen monoton sein, d.h. sie dürfen keine Dreiecksform oder Gaußglockenform haben. Dies ist wichtig, weil die Autoren für die Gewichts Anpassung nicht Backpropagation oder ein anderes gebräuchliches Gradientenverfahren benutzen, sondern einen speziellen Lernalgorithmus vorschlagen. Wenn für eine Regel (also ein verdecktes Neuron) festgestellt wird, daß sie die Aktion des Systems

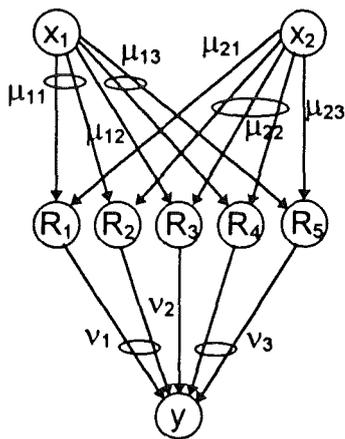


Abb. 6 Ein Beispiel für NEFCON mit 2 Eingangsgrößen und 5 Regeln

verbesserte, wird die Zugehörigkeitsfunktion ihrer Prämissen verbreitert. Sie bekommen dadurch stärkeres Gewicht. Im gegenteiligen Fall wird die Breite der glockenförmigen Zugehörigkeitsfunktion verringert. Gleichzeitig wird die monotone Ausgangsfunktion, die eine Gerade darstellt, im Erfolgsfall steiler, bei einem Mißerfolg flacher gemacht. Eine Verschiebung der Zugehörigkeitsfunktionen wird nicht in Betracht gezogen. Ein Problem dieses Ansatzes ist die Beurteilung der Aktion. Dafür muß eine Gütefunktion definiert werden und ihre Beurteilung auf die einzelnen Regeln zurückgeführt werden. Für jeden einzelnen Lernschritt muß angegeben werden, ob das Vorzeichen der resultierenden Aktion mit dem Vorzeichen der optimalen Aktion übereinstimmt. Je nachdem bekommt der zurückzupropagierte Fehler einen Vorzeichenwechsel oder nicht. Diese Vorschrift ist nicht so schwierig zu erfüllen wie eine genaue Angabe der optimalen Aktion. Trotzdem dürfte in vielen nichtlinearen Systemen höherer Ordnung das Vorzeichen der optimalen Aktion nicht so leicht zu ermitteln sein. Alternativ zu dem beschriebenen Lernen geben die Autoren einen Algorithmus an, um eine neue Regelbasis zu bilden. Aber auch in diesem Falle muß das Vorzeichen der optimalen Aktion bekannt ist.

FuNe (Halgamuge 1993)

Der Fuzzy-Neural Controller FuNe von Halgamuge (1993) besteht aus vier Blöcken: Eingabe-, Fuzzifizierung-, Regel- und Ausgabe-Block. Jeder Block kann wieder mehrere neuronale Schichten beinhalten (Abb.7). Die Neuronen der Eingabeschicht entsprechen den Eingangsvariablen des Systems. Der Fuzzifizierungsblock besteht aus zwei Schichten. In der ersten Schicht befinden sich Neuronen mit Sigmoidfunktionen als Ausgabefunktionen, mit denen sich monotone Zugehörigkeitsfunktionen für Prädikate wie etwa „weit rechts“ oder „sehr klein“ darstellen lassen. Prädikate wie „um Null herum“, „mittel“ oder „geradeaus“ erfordern eine zusätzliche neuronale Schicht mit linearen Ausgangsfunktionen, um glockenförmige Zugehörigkeitsfunktionen als Kombination zweier Sigmoidfunktionen herzustellen. Der Regel-Block besteht aus drei Schichten: die erste berechnet das Supremum seiner Eingaben, den aktuellen Zugehörigkeitswerten der Fuzzy-Prädikate. Das heißt, daß ein Neuron dieser Schicht den am stärksten aktivierten Fuzzy-Wert des Fuzzy-Prädikats übernimmt. Für jede Eingabe-Variable existiert ein solches Neuron für die Supremum-Berechnung. In der nächsten Schicht werden Maxima und Minima über die Supremum-Neuronen berechnet. In einer weiteren Schicht werden diese Ergebnisse mit den Zugehörigkeitswerten der Fuzzy-Prädikate der Regelbedingungen verknüpft. Hierfür wird als T-Norm wieder das Minimum verwendet.

Das System kann aus Daten mit a priori unbekannter Struktur eine Beschreibung in Form von Fuzzy-Regeln liefern. Dafür ist eine zweite Phase nötig. Die erste Phase verwendet ein Netzwerk wie oben beschrieben. Am Ende des Trainings mit den Lerndaten werden die Gewichte zwischen Regelblock und Ausgangsschicht in einer Tabelle gespeichert. Daraufhin wird der Regelblock gänzlich neu strukturiert, so daß er nur noch eine Schicht mit Regeln enthält. Die Bedingungen der Regeln setzen sich entweder aus konjunktiv oder aus disjunktiv verknüpften Termen zusammen. Die Tabelle wird verwendet, um die Gewichte der neuen Konfiguration des Regelblocks zu bestimmen.

FAGNIS (Cechin, Eppler, Gemmeke 1994)

Dieses System ist speziell auf Anwendungen in der Regelung von nichtlinearen Prozessen ausgerichtet. Die bisher vorgestellten Systeme werden in der überwiegenden Zahl auch in diesem Bereich eingesetzt. Diese Systeme übernehmen im allgemeinen die Funktion des Reglers. Bei FAGNIS (Fuzzy Automatically Generated Neural Inferred System) übernehmen konventionelle Zustandsregler (Riccati-Regler) den Regelungsteil und einem Neuronalen Netz fällt die Aufgabe zu, den teilweise unbekanntem Prozeß zu identifizieren (siehe Abb. 8).

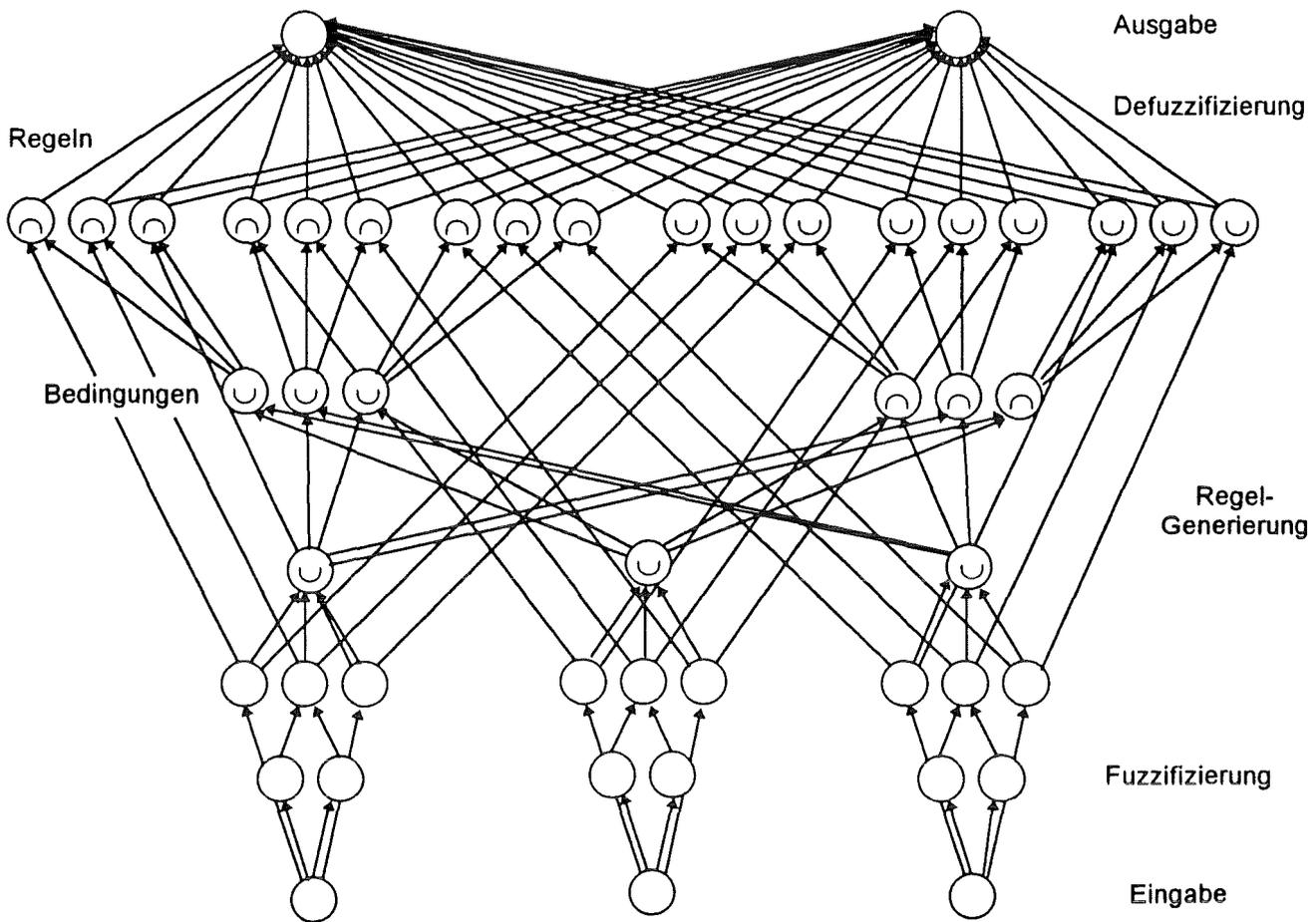


Abb. 7 Ein Beispiel für FuNe, das Trainingsnetz

Dieses System geht also nach Prinzipien der klassischen Regelungstechnik vor: zuerst wird vom realen Prozeß ein Modell entwickelt und dann aus dieser Modellbeschreibung ein dazugehöriger Regler entworfen. Der Unterschied in FAGNIS ist, daß die Modellbeschreibung des Systems automatisch bzw. numerisch geschieht und kein menschlicher Experte nötig ist. Dadurch kann sich das System veränderlichen Prozessen und Umgebungen leichter anpassen. Das notorische Problem der Generierung von Trainingsmustern vereinfacht sich entscheidend, da bei der Systemidentifikation die Trainingsmuster aus den Ein- und Ausgängen des Prozesses bestehen, die lediglich abgelesen werden müssen. Die Eingabe des Netzes besteht aus dem aktuellen Prozeßzustand und den Stellgrößen, die den Prozeß beeinflussen. Die Ausgabe des Netzes besteht aus dem Folgezustand des Prozesses bei den angelegten Stellgrößen. Das Neuronale Netz ist im allgemeinen sehr klein, mit einer verdeckten Schicht und wenigen sigmoiden Neuronen. Eine geringe Zahl von freien Parametern garantiert einen schnellen Lernvorgang und hält die Zahl der erforderlichen Trainingsmuster niedrig. Die Ausgangsneuronen sind linear.

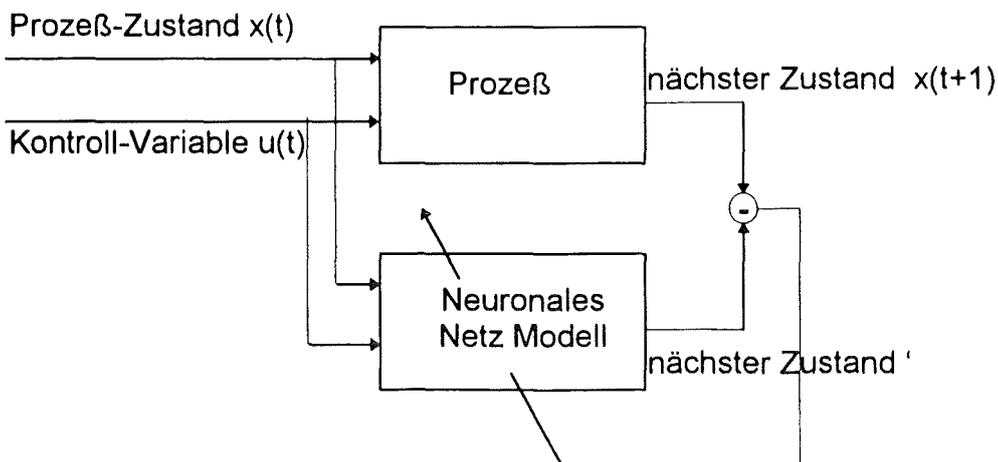


Abb. 8 Prozeßidentifikation durch FAGNIS

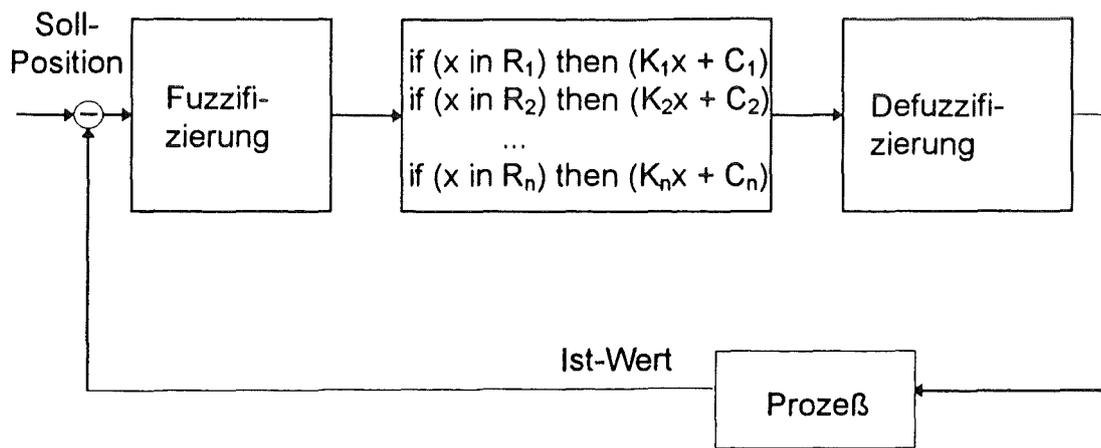


Abb. 9 Struktur von FAGNIS

Nach der Systemidentifikation wird das nichtlineare Neuronale Netz linearisiert. Dies geschieht, indem die nichtlineare Ausgangsfunktion eines Neurons in lineare Stücke zerlegt wird. Der Arbeitspunkt des Systems befindet sich dann immer innerhalb eines linearen Gebiets des Systems. Insgesamt setzt sich das System aus mehreren linearen Gebieten zusammen. Für jedes der Gebiete läßt sich die Systemgleichung aus dem Zustand der Neuronenaktivitäten berechnen. Die Systemgleichung

$$X(t+1) = A_i X(t) + B_i U(t) + C_i$$

(mit i dem Index des linearen Gebiets, X dem Prozeßzustand, U der Stellgröße und A , B , C den Systemmatrizen) ist wiederum Grundlage für die Berechnung eines linearen Zustandsreglers

$$U(t) = K_i X(t) + C_i$$

mit K und C den Regler-Matrizen eines Riccati-Reglers. Ein Problem eines solchen stückweise linearen Reglers ist die mangelnde Stabilität bei den Steigungsänderungen der Zustandsfunktion an den Bereichsgrenzen zwischen einem linearen Gebiet und dem nächsten. Dieses Manko kann zwar nicht vollständig behoben, aber entscheidend abgeschwächt werden, wenn die Übergänge fließend gestaltet werden. Die einfachste Möglichkeit, dies zu tun, ist die Verwendung der Fuzzy-Logik. Die verschiedenen Gebiete werden im Bedingungsteil der Regeln abgeprüft, die zugehörigen Riccati-Regler stellen die Konklusionen der Regeln dar (siehe Abb. 9). Ohne Verwendung der Fuzzy-Logik, aber unter Verwendung ihrer Syntax, können die Gebiete als Prädikate mit einer rechteckigen Zugehörigkeitsfunktion angesehen werden. Ist der Wert der Zustandsvariablen des Systems in einem bestimmten linearen Gebiet, ist die Zugehörigkeitsfunktion 1, sonst 0. Wird die Fuzzy-Logik angewandt, ist die Zugehörigkeitsfunktion glockenförmig und überlagert sich mit den Funktionen der angrenzenden Gebiete. Dadurch können im Randbereich mehrere Fuzzy-Regeln aktiv werden, die unterschiedliche Riccati-Regler kreieren. Die Defuzzifizierung geschieht einfach durch Überlagerung dieser linearen Regler. Instabilitäten werden durch diese Methode nicht unmöglich gemacht, aber weitgehend ausgeschlossen.

Neuro-Fuzzy-Anwendungen

Diagnose von Leber-Funktionsstörungen

Hayashi (1991). Das Fuzzy Neuronale Expertensystem FNES realisiert zwei Funktionen: die Generalisierung der von Trainingsdaten stammenden Information und die Extraktion von Fuzzy-Regeln aus dem verwendeten Neuronalen Netz (siehe Abb.10). Dieses System wurde in der medizinischen Diagnose eingesetzt. Vier verschiedene Lebererkrankungen sollten aufgrund von neun verschiedenen biochemischen Tests und dem Geschlecht der Untersuchten festgestellt werden. Physiologen untersuchten 373 Patienten und diagnostizierten deren Lebererkrankung. Diese Daten dienen dem Training eines Neuronalen Netzes (FNES - „Fuzzy Neural Expert System“). Nach dem Training konnte das Netz 100% der Daten richtig zuordnen. Es konnten 48 Fuzzy-Regeln extrahiert werden, von denen 12 eine der Erkrankungen bestätigten und 36 die betreffenden Erkrankungen ausschlossen. Anschließend wurde eine Testmenge von Daten erstellt, bei der wiederum Physiologen 163 Patienten behandelten. Von dieser Testmenge konnten vom Neuronalen Netz 75.5% korrekt klassifiziert werden. Eine parallel implementierte Diskriminanten-Analyse schnitt weit schlechter ab.

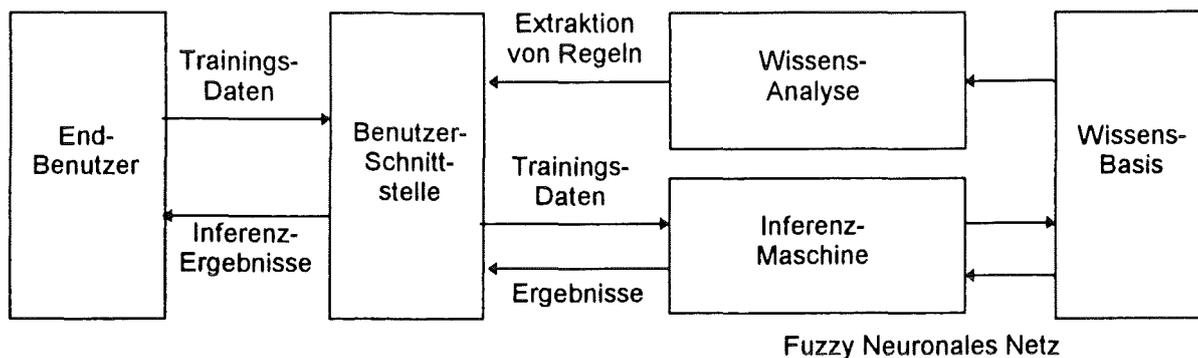


Abb.10 Blockdiagramm des Fuzzy Neuronales Systems FNES mit automatischer Regelextraktion

Das verwendete Neuronales Netz verwendet keine kontinuierlichen Neuronen, sondern ternäre, d.h. die Aktivierungsfunktion liefert drei Werte: -1, 0, +1. Deshalb ist auch nicht das Gradientenverfahren Backpropagation anwendbar, sondern es wird ein Lernalgorithmus von Gallant (1990) verwendet. Die Eingabe des Netzes sieht zwei unterschiedliche Arten von Eingängen vor: scharfe und unscharfe Zellgruppen. Die Neuronen einer scharfen Zellgruppe sind immer gleich aktiviert, und zwar mit -1 bzw. mit +1. Die Neuronen von unscharfen Zellgruppen geben eine Thermometerkodierung an, wobei von links beginnend positive Einsen stehen, die nach rechts fortschreitend irgendwann von negativen Einsen abgelöst werden. Der Übergang vom positiven zum negativen Bereich gibt quasi den Füllstand auf einem Zahlenstrahl an, weshalb Hayashi diese Darstellung auch „Level Set Representation“ nennt. Das Netz enthält eine Zwischenschicht und direkte Verbindungen zwischen Eingangs- und Ausgangsschicht. Die Extraktion der Fuzzy-Regeln geschieht dadurch, daß die Eingänge nach einem bestimmten Algorithmus solange mit einem bestimmten Muster belegt werden, bis der betrachtete Ausgang des Netzes als „wahr“ oder „falsch“ klassifiziert werden kann.

Regelung eines Sofortdampferzeugers

(Eppler 1993). Energieunternehmen haben das Problem, sich bei der Stromerzeugung ständig nach dem Stromverbrauch richten zu müssen, da nicht genügend Speicherkapazität zur Verfügung steht. Zu Spitzenlastzeiten müssen deshalb innerhalb kurzer Zeit geeignete Stromerzeuger zugeschaltet werden können. Dafür bieten sich H_2/O_2 -Dampferzeuger an, die ein schnelles Anfahrverhalten bei hohem Wirkungsgrad besitzen und deren Leistung sich in weiten Grenzen variieren läßt (siehe Abb.11). Bei einem solchen System sind drei Größen zu regeln: der Dampfmassenstrom von Wasserstoff, Sauerstoff und Wasserdampf, der Druck in der Brennkammer und die Temperatur. Ein Modell des Prozesses wird durch ein Differentialgleichungssystem 13. Ordnung beschrieben. Um dieses System zu regeln wurde anfangs ein nichtlinearer Zustandsregler entworfen. Dafür waren Vereinfachungen des Systemmodells nötig, wie die Reduktion in ein Differentialgleichungssystem dritter Ordnung und die Vernachlässigung des Totzeitvolumens zwischen Gasventil und Einblaskopf. Das aus dem Differentialgleichungssystem hergeleitete Regelgesetz benötigt Parameter, deren Herleitung einen enormen Zeitaufwand verursacht. Bedingt durch die hohe Komplexität des Systems können durch die Rechenungenauigkeit verursachte Rechenfehler, insbesondere bei der Invertierung der Systemmatrix, nicht mehr einfach ignoriert werden. Die vielen Nichtlinearitäten der Kennlinien der Ventile bzw. der Dampftemperatur sowie die starke Vermaschung des Mehrgrößensystems verursachen einen enormen Aufwand bei der herkömmlichen Reglersynthese. Die Lösung des Problems mit dem Neuro-Fuzzy-Regler FINE zeigt, daß diese Methode mit weit weniger Aufwand und einigen zusätzlichen Vorteilen angewandt werden kann.

Zunächst wurden einige Fuzzy-Regeln vom Mamdani-Typ definiert, die anschließend in der Simulation von Hand noch verbessert wurden. Diese Regeln wurden dann durch den FINE-Compiler in die Gewichte eines Neuronales Netzes transferiert. Zu diesem Zeitpunkt funktioniert das Neuronale Netz genauso wie der Fuzzy-Regler, aber es besteht zusätzlich die Möglichkeit der Adaption. FINE setzt grundsätzlich voraus, daß die Trainingsmuster von einem Menschen oder einem anderen guten Regler stammen. Beim vorliegenden Prozeß konnte, da beide Möglichkeiten nicht existierten, auf ein halbautomatisches Verfahren zurückgegriffen werden. Ein Experte bewertete den Regelungsverlauf und gab Hinweise, in welche Richtung die Zugehörigkeitsfunktionen verschoben werden mußten. Mit dieser Information konnten neue Trainingsmuster hergestellt und das Netz damit trainiert werden. Damit war es immerhin möglich, den Regelungsverlauf zu beschleunigen und einen nicht-akzeptablen Überschwinger zu beseitigen. Der Neuro-Fuzzy-Regler wurde in dieser Anwendung also zur Feinabstimmung eines Fuzzy-Reglers verwendet. Starke Systemveränderungen können mit diesem System allerdings nicht adaptiert werden.

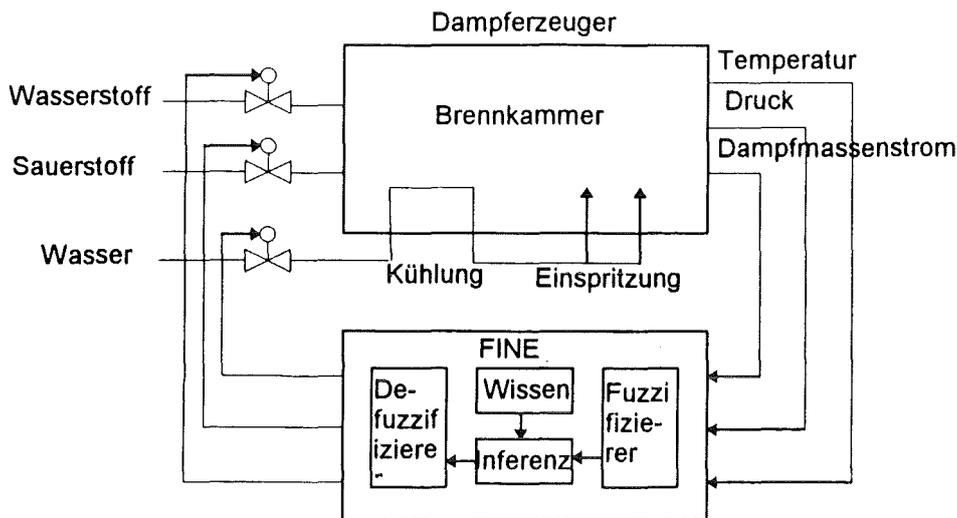


Abb.11 Regelung eines Dampferzeugers durch FINE

Regelung eines Schweißroboters

Ma (1993). Ein Schweißroboter mit fünf Achsen soll möglichst genau entlang der Schweißnaht fahren. Der Abstand von der gefahrenen Linie zu der gewünschten Linie wird mit Hilfe einer CCD-Kamera festgestellt. Dieser Abstand und die Abstandsänderung dienen als Eingabe für einen Fuzzy-Regler, der daraus eine Richtungsänderung berechnet. Die Parameter des Fuzzy-Controllers werden durch ein Neuronales Netz eingestellt. Die Zugehörigkeitsfunktionen der Fuzzy-Prädikate werden jeweils als stückweise linear angenommen, d.h. entweder dreiecksförmig („geradeaus“) oder in Form einer Rampe („links“, „rechts“) angeordnet. Freie Parameter dieser Zugehörigkeitsfunktionen sind die Anfangs- und Endpunkte der Dreiecks- und Rampenfunktionen. Diese Funktionen werden durch unterschiedliche Neuronale Netz-Ausgaben gedehnt oder gestaucht. Dem Neuronalen Netz fällt allerdings nur die einfache Aufgabe zu, aus drei Abstandswerten eine Richtungsänderung abzuschätzen. Das Netz generiert die Parameter für den Fuzzy-Regler nicht direkt. Die Richtungsänderung muß über eine Tabelle in die freien Parameter des Fuzzy-Reglers übersetzt werden. Leider müssen durch diesen Umweg auf irgendeine Weise die Werte der Tabelle bestimmt werden. Das Verfahren dazu wird nicht angegeben. Der Grund für dieses etwas umständliche Verfahren liegt im Problem der Mustergenerierung für das Neuronale Netz, das klassifizierte und bewertete Beispiele verlangt. Diese Beispiele gibt es hier nicht, da das Neuronale Netz offline trainiert wird und die Trainingsdaten nicht während des Regelungsvorganges erzeugt und bewertet werden.

Die Anforderungen des Systems werden durch den Neuro-Fuzzy-Regler erfüllt. Der Schweißroboter kann mit einer Geschwindigkeit von 3 cm/sec auch Kurven und rechtwinklige Knicke mit einer sehr geringen Abweichung nachfahren.

Automatik-Getriebe

Hayashi (1993). Bei einem Auto mit Automatikgetriebe (Hayashi et al. 1993) soll die Getriebeübersetzung geregelt werden. In erster Linie wird die Belastung des Antriebs aus einer Menge von Status-Sensoren abgeschätzt. Ebenso wird das Verhalten des Fahrers vorhergesagt. Beide Aufgaben werden durch Fuzzy-Regeln beschrieben. Im ersten Fall ist die Antriebsbelastung abhängig von der Geschwindigkeit und der Beschleunigung der Antriebswelle. Für die Vorhersage des Fahrerverhaltens wird die Änderung des Gaspedals beobachtet. Die Fuzzy-Regeln besitzen die Standard-Form von Mamdani. Das Ergebnis der beiden Fuzzy-Blöcke wird zusammen mit Parametern wie der Veränderung des Gaspedals und der Geschwindigkeit des Autos in ein Neuronales Netz eingespeist. Das Netz besitzt eine verdeckte Schicht mit neun Neuronen. Die Ausgabe des Netzes bestimmt direkt den geeigneten Gang des Fünfganggetriebes, mit dem das Auto bei der aktuellen Last geschaltet werden sollte. Ein erfahrener Fahrer fährt mit dem Auto solange, bis genügend Trainingsdaten für das Neuronale Netz entstanden sind.

Um Daten zu bekommen wurden Testfahrten an einem Berg mit veränderlicher Last durchgeführt. Im Vergleich zu einer konventionellen Methode ergaben sich Verbesserungen im Benzinverbrauch, den Abgaswerten und im sanfteren Fahrverhalten.

Roboterarm

Segers (1994). Der Greifer auf einem rotierenden Roboterarm ist entlang des Arms radial verschiebbar und soll bei verschiedenen Rotationsgeschwindigkeiten des Roboterarms an einer bestimmten Radius-Position von einer Kraft F gehalten werden. Als Regler wird das Neuro-Fuzzy-System GARIC verwendet. Regelgrößen sind die Abweichung der Greiferposition von der Sollposition und deren Änderung in der Zeit, die Stellgröße des Systems ist eine Kraft, die aufgebracht werden muß, um der Zentrifugalkraft des Systems entgegenzuwirken. Das Aktionsnetz (ASN) fuzzifiziert die beiden Eingabegrößen jeweils durch fünf Fuzzy-Prädikate, welche die entsprechenden Zugehörigkeitswerte an die Regelschicht weitergeben. Diese besteht aus 25 Neuronen, d.h. jede mögliche Kombination der Prädikate wird durch eine Regel abgedeckt. Die Stellgröße wird ebenfalls durch fünf Fuzzy-Prädikate fuzzifiziert, so daß die vierte neuronale Schicht aus fünf Neuronen besteht. Die Ausgangsschicht besteht aus genau einem Neuron, das die Stellgröße darstellt. Sie wird direkt auf den Prozeß geschaltet. Das Bewertungsnetz (AEN) erhält als Eingabe zu den zwei Regelgrößen noch zusätzlich einen Bias. Die verdeckte Schicht besteht aus einer unbestimmten Anzahl von Neuronen (während des Experimentes von 5 bis 20 variiert), die Ausgangsschicht enthält als einziges Neuron das Bewertungssignal. Die Zugehörigkeitsfunktionen der Prämissen der Regeln werden vorinitialisiert und später nicht mehr verändert. Die Adaption des Systems betrifft nur die Fuzzy-Prädikate der Regel-Konklusionen, die anfangs alle zu Null initialisiert werden. Im Laufe der Adaption der Regelung bilden sich drei Fuzzy-Prädikate heran, die ihren Mittelpunkt ungleich Null haben. Nach 21

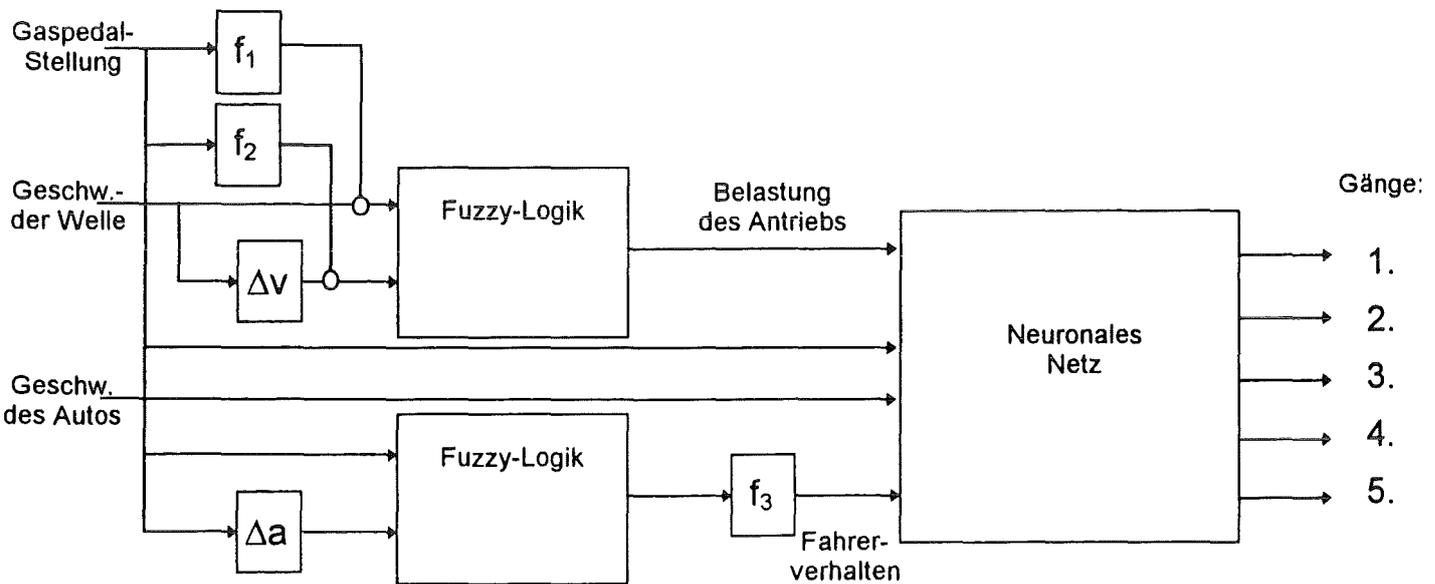


Abb.12 Getriebeübersetzung durch Fuzzy-Logik und Neuronales Netz

Versuchen hat das System die Aufgabe gelernt. Allerdings läßt sich nach dieser Methode der Fehler nicht sehr stark verbessern und die Genauigkeit läßt zu wünschen übrig. Sobald ein stabiler Zustand des Systems erreicht ist, hört es auf zu lernen. Die anfängliche Regelbasis muß sinnvoll initialisiert sein. GARIC lernt keine neuen Regeln, sondern verändert lediglich die bestehenden. Ein weiteres Problem stellt die starke Abhängigkeit von den frei einstellbaren Parametern (Lernrate, Anzahl der verdeckten Neuronen, ...) dar. Auf kleine Schwankungen reagiert das System sehr sensibel.

Regelung eines Interferometerspiegels

(Cechin, Eppler, Gemmeke 1995). In einem Ballon-Experiment, ausgeführt am Forschungszentrum Karlsruhe, werden Spurengase in der Stratosphäre untersucht. Die Meßgeräte, in der Hauptsache ein Interferometer zur Aufnahme von Emissionsspektren der zu untersuchenden Gase, befinden sich in der Gondel des Ballons. Das Interferometer untersucht Schicht für Schicht der Stratosphäre, wozu es zeitweise konstant auf einen Horizontpunkt gerichtet sein muß. Durch starke Winde (bis zu 200 km/h) kommt der Ballon in Schwingung und rotiert. Die Rotation wird durch ein Pivot-Element ausgeglichen, die Schwingungsstabilisierung besorgt die Regelung eines durch einen Motor beweglichen Spiegels. Durch die tiefen Temperaturen in der Stratosphäre verändern sich eventuell einige Parameter des Spiegelsystems (wie Reibung, Unwucht, Trägheit). Da die Bedingungen für eine optimale Reglereinstellung in der Stratosphäre also anders sind als auf der Erde, ist eine adaptive Regelung unumgänglich. Während einer Messung sollte nicht adaptiert werden, um die Stabilität des Systems nicht zu gefährden. Die Abweichung der gewünschten Spiegelstellung von der tatsächlichen wird - zusammen mit anderen Daten - permanent zur Erde gefunkt. Wird die Abweichung zu groß, wird am Ende eines

Meßzyklus von der Erde aus eine Adaptionphase gestartet. Der Ballonflug dauert durchschnittlich 10 Stunden und enthält viele Meßzyklen. Für eine einzelne Adaptionphase stehen maximal zwei Minuten zur Verfügung.

Für die Regelung dieses Systems wurde FAGNIS eingesetzt, das durch seine Systemidentifikation und das Offline-Training die gewünschten Anforderungen erfüllt. Der Zustand des Systems, eines Gleichstrommotors, wird durch vier Zustandsvariablen beschrieben. Dieses System vierter Ordnung verhält sich unter Normalbedingungen zwar einigermaßen linear, doch werden die Nichtlinearitäten unter den extremen Bedingungen der Stratosphäre größer. Sie sind wegen der hohen notwendigen Genauigkeit ($\Delta\theta \approx 140 \mu\text{rad}$) und dem großen dynamischen Bereich ($\approx 10^4 \cdot \Delta\theta$) nicht zu vernachlässigen. Die Zustandsgrößen sind Position, Winkelgeschwindigkeit, Winkelbeschleunigung und Hysterese. Das Neuronale Netz zur Systemidentifikation wird vorstrukturiert, da über das System schon einiges bekannt ist. Es wird eine neuronale Zwischenschicht mit zwei Neuronen verwendet. Lineare Verknüpfungen zwischen Ein- und Ausgängen des Neuronalen Netzes kommen dadurch zustande, daß die Eingangsschicht direkt mit der Ausgangsschicht verbunden ist. Für die ersten drei Zustandsgrößen ist die Differentialgleichung in erster Näherung bekannt:

$$J\ddot{\theta} = M - f\dot{\theta}$$

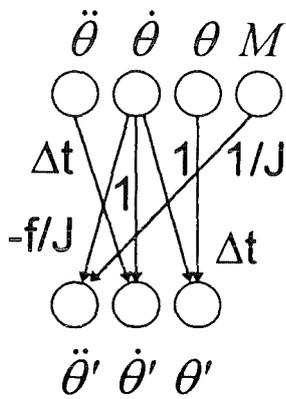


Abb.13 (a) ideale Vorstrukturierung des Neuronalen Netzes nach den näherungsweise bekannten Differentialgleichungen

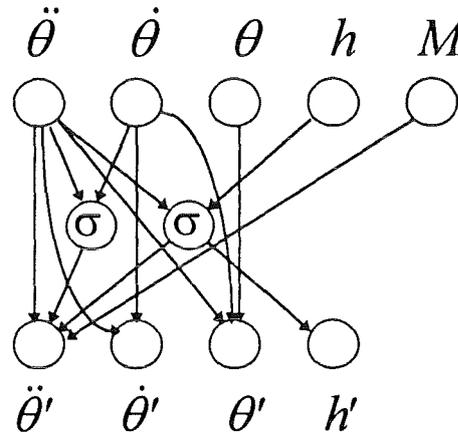


Abb.13 (b) tatsächliche Vorstrukturierung des Neuronalen Netzes

mit J dem Trägheitsmoment, M dem Drehmoment des Motors und f der Reibungskonstante. Daraus lassen sich die Gleichungen

$$\begin{aligned} \theta &= \theta + \dot{\theta} \cdot \Delta t \\ \dot{\theta} &= \dot{\theta} + \ddot{\theta} \cdot \Delta t \\ \ddot{\theta} &= M/J - f/J \cdot \dot{\theta} \end{aligned}$$

formulieren und das Neuronale Netz vorstrukturieren (siehe Abb.13a). Es zeigte sich, daß ein vollständig vernetztes Netz sehr viel schwerer lernt und sich oft in lokalen Minima verfängt. Nach einer sorgfältigen Analyse der Modellabweichungen und der vollvernetzten Lösung wurden eine zusätzliche Zustandsvariable h (Hysterese des Motors) sowie zwei sigmoide Neuronen in der Zwischenlage eingeführt (siehe Abb.13b).

Die Erzeugung der Trainingsmuster geschieht durch eine Folge von unterschiedlichen Sprüngen, die auf den Spiegelmotor aufgeschaltet werden. Insgesamt werden in einer Minute 200 neue Muster erzeugt. Das vorstrukturierte Netz lernt diese Muster innerhalb von 100 bis 300 Zyklen und benötigt dazu eine weitere Minute, so daß innerhalb von zwei Minuten die gesamte Lernphase abgeschlossen ist. Auf diese Weise schafft es der Regler, die Spiegelposition immer innerhalb der vorgeschriebenen Toleranzgrenze von $140 \mu\text{rad}$ zu halten.

Roboter-Navigation

Pan et al. (1995). Ein Roboter soll sich autonom in einer Fabrikhalle bewegen und verschiedene Dinge erledigen. Kernstück seiner Regelung ist ein Fuzzy-Controller, der seine Eingaben über verschiedene Module bezieht. Die Module bestehen teilweise aus Neuronalen Netzen. Ein Modul ist dafür verantwortlich, in einem Flur von den beiden Wänden möglichst den gleich großen Abstand zu haben. Dafür werden zwei Neuronale Netze verwendet,

die als Eingabe verschiedene Gebiete der Hough-Transformierten des Kamerabildes bekommen und in der Ausgabe eine Richtungsanweisung (40° links, geradeaus,...) produzieren. Ein anderes Modul ist für die Erkennung einer Verzweigung eines Flures zuständig. Ein Neuronales Netz gibt hierbei die Entfernung zur Verzweigung an, die ebenfalls aus der Hough-Transformierten des Kamerabildes abgeschätzt wird. Weitere Module sind für die Erkennung von Hindernissen und ganz speziellen Gegenständen, wie Türrahmen von geschlossenen Türen, sowie für die Pfadplanung zuständig. Für jede der Erkennungsaufgaben wird ein Neuronales Netz eingesetzt.

Der Fuzzy-Controller ist ein Echtzeit-Expertensystem, das Aktionen an die Steuerung des Roboters weitergibt. Die Regeln sind vom Mamdani-Typ, die Defuzzifizierung erfolgt nach dem Flächenschwerpunkt (Center of Area). Eine typische Regel ist wie folgt:

(rule-name rule-20

<i>IF</i>	(turn-angle	right-20)
	(distance-to-junction	far)
	(sonar-reading	no-obstacle)
<i>THEN</i>	(turn-angle	right-20)
	(distance-to-travel	long))

Das komplette System wurde auf einer VME-Bus-Karte mit einem MC68040 Mikroprozessor implementiert.

Regelung einer Klimaanlage

Pfeiffer (1995). Der Gebläsemotor einer Klimaanlage soll so geregelt werden, daß ein konstantes Klima herrscht. Der Zeitbereich des Systems ist so, daß es noch von Hand geregelt werden kann. Es sollte diese Regelung auch automatisch erreicht werden. Dafür wurde eine Neuro-Fuzzy-Regelung auf einem PC implementiert. Mit einer Echtzeit-Erweiterung von Matlab/Simulink und einem Analog-Digital-Konverter-Board werden Abtastzeiten von bis herunter zu 20 ms erreicht.

Als Neuro-Fuzzy-System wird ein Netzwerk aus Radialen Basisfunktionen verwendet. Die Eingangsschicht besteht wie immer aus den Eingabevariablen, es folgt eine Schicht von Neuronen mit Gaußfunktionen als Aktivierungsfunktionen. Über variable Gewichte ist diese Schicht mit einem linearen Ausgangsneuron verbunden. Dieses Netz entspricht einem Fuzzy-System mit Regeln der folgenden Form: Im Bedingungsteil werden die Eingangsvariablen fuzzifiziert und durch eine Konjunktion mit einander verknüpft. In der Konklusion der Regeln sind nur Fuzzy-Singletons erlaubt, d.h. keine vollständigen Fuzzy-Prädikate. Die Defuzzifizierung ist dementsprechend einfach. Der Konklusionswert wird entsprechend der T-Norm für die Inferenz mit dem Fuzzy-Wert der Bedingung multipliziert. Dann werden sämtliche Regeln aufsummiert und normiert.

Der Regler muß direkt durch gleichzeitige Angabe der Regel- und Stellgrößen trainiert werden. Dies leistet ein Mensch, der die Klimaanlage gerade noch von Hand regeln kann. Es wird eine Reihe von Sollwert-Sprüngen vorgegeben, die der menschliche Regler nachfahren muß. Auf diese Weise wird eine Musterdatei erstellt, mit der wiederum das Neuronale Netz trainiert werden kann. Durch die einfache Struktur des Netzes kann das Ergebnis direkt in eine Fuzzy-Schreibweise (mit Singletons als Konklusionen) übertragen und damit interpretiert werden. Die Qualität des Neuro-Fuzzy-Reglers entspricht direkt der des regelnden Menschen. Sie kann niemals besser werden.

Resumee

Es ist deutlich geworden, daß sich Neuronale Netze und Fuzzy Logik gegenseitig ergänzen können und durch ihre Kombination sich eine Vielfalt neuer Möglichkeiten eröffnet. Es ist daher berechtigt, Neuro-Fuzzy als eigenständige Methode zur Informationsverarbeitung in komplexen Systemen anzusehen. Es gibt derzeit eine Fülle unterschiedlicher Systeme, die Lösungsansätze in diesem Bereich darstellen. Die Tendenz ist da, möglichst einfach zu bedienende Werkzeuge anzubieten, mit denen ein unerfahrener und in Neuronalen Netzen und Fuzzy Logik unbelasteter Anwender schnell sein spezifisches Problem lösen kann. Es ist selbstverständlich, daß ein wenig spezifisches, allgemein anwendbares System nicht immer optimale Lösungen produziert. Andererseits kann kaum erwartet werden, daß ein für eine spezielle Anwendung zugeschnittenes System auch für andere Anwendungen benutzerfreundlich eingesetzt werden kann. Zwischen diesen beiden Polen bewegen sich auch die Neuro-Fuzzy-Systeme. In Zukunft muß bestimmt noch einiges geleistet werden, um für Neulinge in Neuronalen Netzen die Hemmschwellen zu senken, solche Werkzeuge einzusetzen. Es ist unzumutbar, daß sich Anwender vor dem Einsatz eines Neuro-Fuzzy-Tools zuerst die komplette Theorie der Neuronalen Netze und Fuzzy Logik aneignen müssen.

Mit dem Einsatz von Neuro-Fuzzy-Werkzeugen hat sich die Entwicklung adaptiver Systeme sehr stark vereinfacht. Adaptive Systeme gab es auch schon vorher, aber sie stellten immer Speziallösungen dar, die nicht von der einen Anwendung zur nächsten übertragen werden konnten. Diese Situation hat sich nun verändert: für ein adaptives System müssen lediglich die Voraussetzungen für die Anwendung Neuronaler Netze gegeben sein. Die Software kann als anwendungsunabhängiger Teil von einschlägigen Firmen oder über Internet als „Public Domain“ bezogen werden. Für jede neue Anwendung verbleibt der Aufwand für die Erstellung der Trainingsdaten, die in genügend großer Zahl vorhanden sein müssen. Diese Aufgabe ist keineswegs trivial, und die Erfahrung aus vielen Neuro-Fuzzy-Projekten zeigt, daß im Gegensatz zur sonst üblichen Programmierung hier der Aufwand in die Erzeugung geeigneter Datensätze zu investieren ist. Darüber hinaus ist die Festlegung der Netz-Topologie oder des Netztyps eine Aufgabe, die in vielen Fällen nicht ohne Expertenwissen auskommt. Ein noch nicht befriedigend gelöstes Problem ist sicherlich das in eine Anwendung integrierte neuronale Lernen, wenn das Neuro-Fuzzy-System nur einen Teil des Gesamtsystems ausmacht. Unabhängig von einzelnen Schwierigkeiten sind in den letzten Jahren enorme Fortschritte auf diesem Gebiet erzielt worden, wie nicht zuletzt die weiter vorne beschriebenen Anwendungen zeigen.

Literatur

- Bedzak, M., Daca, W., „Intelligent Control by Fuzzy Neural Network Method“, in Proceedings of EUFIT 1995, Aachen, Germany, pp. 1025 - 1029
- Parten, C., „A Self-Organizing Fuzzy Neurocontroller“, in Proceedings of IEEE Workshop on Neuro Fuzzy Control, Muroran, Japan, 1993, pp. 189 - 193
- Ushida, H., Imura, A., Yamaguchi, T., Takagi, T., „Conceptual-level-learning-based Refinement of a Human-motion Model“, in Proceedings of IEEE Workshop on Fuzzy Logic and Neural Networks/Genetic Algorithms, Nagoya, Japan, 1994
- Strefezza, M., Dote, Y., „Neuro-Fuzzy Motor Controller“, in Proceedings of IEEE Workshop on Neuro Fuzzy Control, Muroran, Japan, 1993, pp. 72 - 79
- Takagi, H., Hayashi, I., „NN-Driven Fuzzy Reasoning, Intern. Journal on Approximate Reasoning, 1991, Nr. 5, pp. 191 - 212
- Pan, J., Pack, D. J., Kosaka, A., Kak, A. C., „FUZZY-NAV: A Vision Based Robot Navigation Architecture“, in Proceedings of World Congress On Neural Networks (WCNN), Washington, 1995, pp. 602 - 607
- Pfeiffer, B.-M., „Imitation of Human Operators by Neuro-Fuzzy-Structures“, in Proceedings of EUFIT, Aachen, 1995, pp. 804 - 809
- Higgins, C. M., Goodman, R. M., „Learning Fuzzy Rule-Based Neural Networks for Control“, in Neural Information Processing Systems 5 (ed. Hanson et al.), Morgan Kaufmann Publishers, 1993, pp. 350 - 357
- Hayashi, Y., „A Neural Expert System ...“, in Neural Information Processing Systems 3 (ed. Hanson et al.), Morgan Kaufmann Publishers, 1991, pp. 578 - 584
- Teshnehlab, M., Watanabe, K., „A Fuzzy Gaussian Potential Neural Network“, in Proceedings of IEEE Workshop on Fuzzy Logic and Neural Networks/Genetic Algorithms, Nagoya, Japan, 1994, pp. 14 - 22
- Hayashi, K., „Neuro Fuzzy Optimal Transmission Control“, in Proceedings of IEEE Workshop on Neuro Fuzzy Control, Muroran, Japan, 1993, pp. 97 - 101
- Ma, P., Ohshima, K., „Neuro-Fuzzy Control in Tracking of 2 Dimensional Orbit“, in Proceedings of IEEE Workshop on Neuro Fuzzy Control, Muroran, Japan, 1993, pp. 80 - 83
- Gallant, S.I., „Perceptron Based Learning Algorithms“, IEEE Transactions on Neural Networks, 1(2), pp. 179 - 191
- Segers, F.P.G., Smits, J.G.M.M., „Tuning of a Fuzzy Logic Controller with GARIC“, in Proceedings of EUFIT 1994, Aachen, Germany, pp. 189 - 194
- Gupta, M.M., Rao, D.H., „On the principles of fuzzy neural networks“, in Fuzzy Sets and Systems 61 (1994), pp. 1 - 18, North-Holland
- Hayashi, I., Nomura, H., Wakami, N., „Artificial neural network driven fuzzy control and its application ...“, in Proc. of the Third IFSA Congress (ed. Bezdek), Seattle, 1989, pp. 610-613
- Kuncicky, D.C., Kandel, A., „A fuzzy interpretation of neural networks“, in Proc. of the Third IFSA Congress (ed. Bezdek), Seattle, 1989, pp. 113-116
- Yamakawa, T., Tomoda, S., „A fuzzy neuron and its application to pattern recognition“, in Proc. of the Third IFSA Congress (ed. Bezdek), Seattle, 1989, pp. 30-38
- Eppler, W., „Adaptive Control with a Prestructured Neural Fuzzy Controller“, Proceedings of IEEE Int. Workshop on Neuro-Fuzzy Control, 1993, Muroran (ed. Dote et al.), 222-228
- Cechin, A.L., Eppler, W., „Automatic design of a fuzzy controller from a neural process model“, Proceedings of 2nd European Congress on Intelligent Techniques and Soft Computing, Aachen, Germany, September 20-23, 1994, 752-756
- Eppler, W., „Prestructuring of neural networks with fuzzy rules“, in Proc. of the Neuro-Nimes (ed. J. Hérault), EC2, 1990, pp. 227 - 241

Eppler, W., „Neuronaler Fuzzy-Controller zur Regelung eines H₂O₂-Dampferzeugers“, in Technische Anwendungen von Fuzzy-Systemen (ed. K. Goser), VDE-Verlag 1992

Eppler, W., „Vorstrukturierung Neuronaler Netze mit Fuzzy-Logik“, VDI-Verlag, Fortschrittsberichte, Reihe 10, Nr. 266, 1993, 174 pp.

Jang, J.R., "ANFIS: Adaptive-Neural-Based Fuzzy Inference Systems", in IEEE Trans. Systems, Man and Cybernetics, 1991

Kosko, B., "Neural Networks and Fuzzy Systems", 1992, Prentice-Hall International, Inc., Englewood Cliffs, N.J.

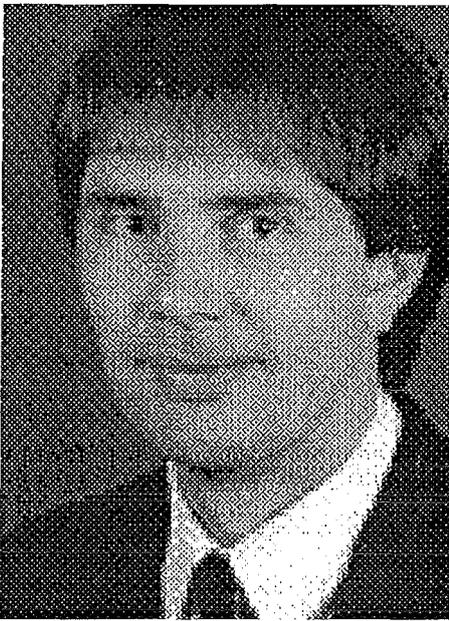
Berenji, H.R., Khedkar, P., „Learning and Tuning Fuzzy Logic Controllers Through Reinforcements“, in IEEE Trans. Neural Networks, 3:, pp. 724-740

Nauck, D., Klawonn, F., Kruse, R., „Neuronale Netze und Fuzzy-Systeme“, Vieweg, 1994

Halgamuge, S.H., Glesner, M., „Neural Networks in Designing Fuzzy Systems for Real World Applications“, in International Journal for Fuzzy Sets and Systems, North Holland, 1994

Nguyen, D., Widrow, B., „The Truck Backer-Upper: An Example of Self-Learning in Neural Networks“, in R. Eckmiller (ed.): Advanced Neural Computers, North Holland, 1990

Grossberg, S., „Adaptive pattern classification and Universal recoding: I. Parallel development and coding of neural feature detectors“, in Anderson et al. (ed.): Neurocomputing: Foundations of Research, pp. 245-258, MIT Press, 1976,1988



Prof. Dr. Dan Ulmet

Prof. Dr. Dan Ulmet studierte von 1976 bis 1981 Mathematik mit Nebenfach Informatik an der Universität Temeswar (Rumänien). Nach Abschluß des Studiums arbeitete er bis 1988 in einem Maschinenbau-Unternehmen und wurde 1989 wissenschaftlicher Angestellter am Mathematischen Institut der Universität Stuttgart, wo er 1990 mit der Arbeit "Nichtlineare elektrische Netzwerke" promovierte. Im Anschluß war er in der CAD/CAM-Entwicklung bei der Mercedes-Benz AG tätig. Seit dem Wintersemester 1993/94 ist er Professor für Mathematik an der Fachhochschule für Technik in Esslingen.

CAD Systeme stellen bei der industriellen Entwicklung und dem Design von Bauteilen ein unverzichtbares Hilfsmittel dar. Wurden die Untersuchungsgegenstände zunächst nur zweidimensional erfaßt, setzt man heute dreidimensionale Modelle routinemäßig ein. Möglich wurde diese Entwicklung durch immer leistungsfähigere Rechner, aber auch durch spezielle Algorithmen zur Erfassung der enormen Datensätze. Diese Algorithmen beruhen zumeist auf Spline-kurven und -flächen. Einen Sonderfall nehmen hierbei die Bezier-Spline-Kurven ein; sie sind geometrisch einfach zu konstruieren und besitzen algebraisch eine einfache Struktur.

GRUNDLAGEN DER CAD/CAM ENTWICKLUNG MIT SPLINEKURVEN - EINE EINFÜHRUNG -

Dan - Eugen Ulmet

Fachhochschule Esslingen - Hochschule für Technik, Kanalstr. 33, 73728 Esslingen

ZUSAMMENFASSUNG

Splinekurven und Splineflächen haben sich in der CAD/CAM Entwicklung bei der Modellierung von Freiformgeometrie weltweit etabliert. Dabei spielen 'splines' in Bézier - Form, die unabhängig von den Ingenieuren P. Bézier (Firma Renault) und P. de Casteljau (Firma Citroën) vor etwa 35 Jahren entdeckt wurden, immer noch eine wichtige Rolle.

Es werden zunächst einige der wichtigsten geometrischen, analytischen und programmiertechnischen Eigenschaften von Bézier Splinekurven eingeführt. Abschliessend werden einige der aktuellen Splinemethoden in der akademischen und industriellen CAD/CAM Entwicklung vorgestellt.

1. EINLEITUNG

Vor etwa 35 Jahren hatten zwei junge Ingenieure, P. Bézier von der Firma Renault und P. de Casteljau von Citroën unabhängig voneinander eine einfache aber revolutionäre Idee, die es ermöglicht hat CAD/CAM Systeme für die Automobilkonstruktion zu entwickeln. Sie haben eine spezielle Kategorie von Kurven und Flächen entdeckt, mit der komplizierte Formen modelliert werden können und die sich insbesondere für die Darstellung und Manipulation innerhalb eines Computersystems hervorragend eignet - es handelt sich um die sogenannten Bézier Splinekurven und -flächen. Damit ist das Spezialgebiet CAGD (Computer Aided Geometric Design) entstanden.

Es ging darum mit Hilfe einer möglichst geringen Datenmenge entweder neue Formen zu modellieren (**Modelldesign**), oder für existierende Modelle (z.B. Tonmodelle) die wesentliche geometrische Information herauszufiltern,

um diese Formen später (eventuell in einem vergrößerten Maßstab) rekonstruieren zu können (**Modellrückführung**). Das Problem der **Datenreduktion** und der damit verbundenen **Rechenzeitoptimierung** ist nach wie vor eine der zentralen Anforderungen in der CAGD Entwicklung.

Was für geometrische Formen lassen sich mit wenigen Daten modellieren? Zunächst einfache Profile die aus **Geraden** und **Kreisbögen** (sogenannten 'primitives') bestehen und deshalb den Löwenanteil von technischen Zeichnungen bilden. Die benötigte Datenmenge (jeweils 2 Punkte bzw. 1 Punkt und eine Zahl) ist minimal, dafür lassen sich aber mit diesen geometrischen Elementen keine komplizierteren Formen (**Freiformgeometrie**) modellieren. Bézier und de Casteljau haben ein Geometrielement entdeckt - das sogenannte **Bézier Segment** - mit dessen Hilfe praktisch jede geometrische Form modelliert werden kann. Die Bézier Segmente können sozusagen als 'Atome' eines komplizierten Freiformmodells aufgefaßt werden. Ein Bézier Segment wurde von Bézier und de Casteljau mit Hilfe von 4 Punkten aufgebaut - eine Datenmenge die den Anforderungen der Datenreduktion im hohen Maße genügt. Die Konstruktion des Segmentes als eine im Intervall $[0, 1]$ parametrisierte Kurve ist als **Algorithmus von de Casteljau** bekannt.

Bezeichnen wir die 4 Punkte mit B_0, B_1, B_2, B_3 und $t \in [0, 1]$ sei ein Parameterwert. Im 1. Schritt des Algorithmus werden die 3 Zwischenpunkte B_0^1, B_1^1 , bzw. B_2^1 konstruiert, die die Strecken B_0B_1, B_1B_2 , bzw. B_2B_3 jeweils im Verhältnis t unterteilen. Im 2. Schritt wird dieses Unterteilungsverfahren mit den 3 Zwischenpunkten des 1. de Casteljau Schrittes wiederholt - die 2 neuen Zwischenpunkte sind B_0^2, B_1^2 . Die 3. de Casteljau Iteration liefert schließlich den Punkt B_0^3 , der das Segment $B_0^2B_1^2$ im Verhältnis t unterteilt (siehe Bild 1).

Wird nun der Algorithmus von de Casteljau für alle Parameterwerte $t \in [0, 1]$ durchgeführt, so entsteht eine parametrisierte Kurve, ein Bézier Segment. Die Kurve startet im Punkt B_0 tangential zur Strecke B_0B_1 und läuft in B_3 tangential zur Strecke B_2B_3 ein. Dabei verläuft das Segment im inneren der konvexen Hülle der 4 **Bezierpunkte** B_0, B_1, B_2, B_3 (zur Begründung siehe Abschnitt 2.2).

Programmiertechnisch lassen sich diese neuen Geometrielemente sehr einfach in der **CAD/CAM Entwicklung** in ein CAD/CAM Softwaresystem **implementieren**. Allein die 4 Bézier Punkte werden als **Geometriedaten** in eine DATEI gespeichert und der **Bauplan** des Bézier Segmentes - der Algorithmus von de Casteljau - wird als PROGRAMM codiert. In der **CAD/CAM Anwendung** wird dann beim Zugriff auf das Bézier Segment

das PROGRAMM zur Manipulation der DATEI aktiviert und das Segment berechnet. Anschließend kann das Bézier Segment am Bildschirm dargestellt und weiterbearbeitet werden. Dabei wird in der Regel jeweils auf die DATEI mit unterschiedlichen Programmen zugegriffen (siehe auch Abschnitt 2.4).

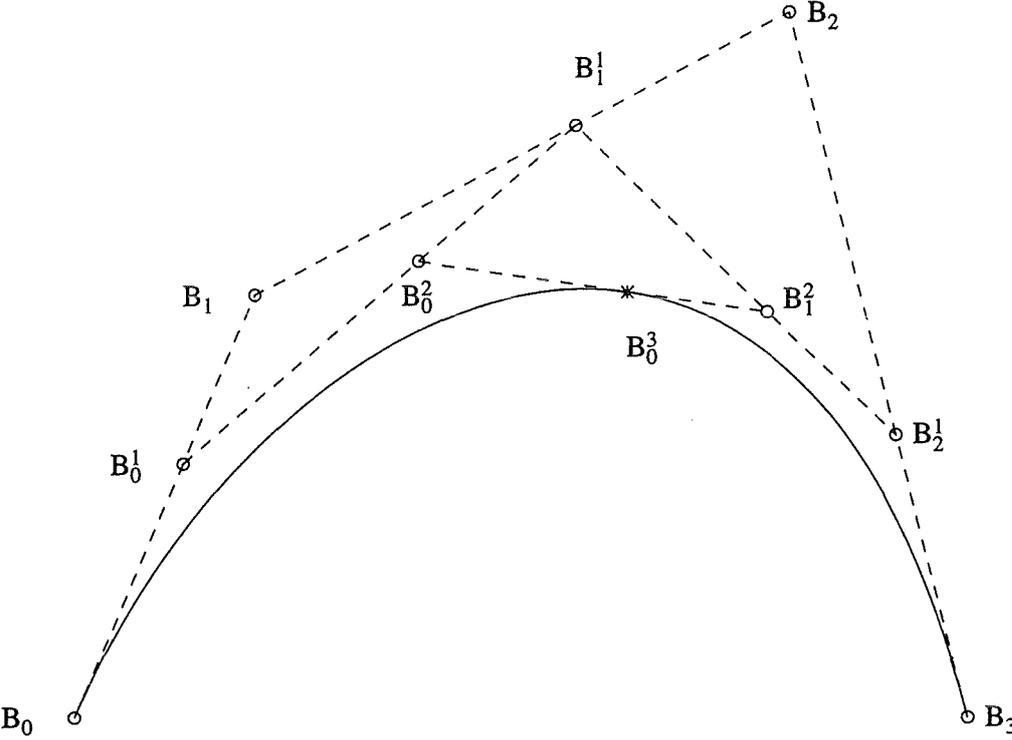


Bild1. Der Algorithmus von de Casteljau.

2. BÉZIER SEGMENTE

Bézier Segmente wurden ursprünglich durch einen **geometrischen** Algorithmus, dem Algorithmus von de Casteljau **rekursiv** definiert. Erst etwa 10 Jahre später wurde eine explizite **parametrische** Darstellung und der Zusammenhang zwischen Bézier Segmenten und Bernstein Polynomen von A. R. Forrest entdeckt (siehe [3]). Mit Hilfe dieser Parametrisierung können Bézier Segmente auch **analytisch** untersucht werden.

2.1. Die Parametrisierung

Ein Bézier Segment besitzt die folgende Parametrisierung:

$$\vec{x}(t) = \sum_{i=0}^n \vec{b}_i \cdot B_i^n(t), \quad t \in [0, 1], \quad (1)$$

wobei der Grad n des Bézier Segmentes gleich der Anzahl der de Casteljau Iterationen ist. (Für die im 1. Abschnitt beschriebenen Konstruktion ist $n = 3$). Die Bézier Koeffizienten \vec{b}_i sind die Ortsvektoren der Bézier Punkte B_i und

$$B_i^n(t) = \binom{n}{i} (1-t)^{n-i} t^i, \quad i = 0, 1, 2, \dots, n \quad (2)$$

sind die Bernstein Polynome vom Grad n . Wird der Algorithmus von de Casteljau anstatt mit 4 Punkten und 3 Iterationen wie im 1. Abschnitt analog mit $n+1$ Punkten und n Iterationen durchgeführt, so entsteht ein Bézier Segment vom Grad n mit der Parametrisierung (1). Für die weiteren Ausführungen werden wir den klassischen Fall der kubischen Bézier Segmente betrachten. Die meisten Ideen lassen sich problemlos für allgemeine Bézier Segmente vom Grad n übertragen ([3], [4]).

Die Aussagen des 1. Abschnittes über den Verlauf eines Bézier Segmentes können nun mit Hilfe der Parametrisierung (1) leicht nachgewiesen werden.

2.2. Der geometrische Verlauf

Für $n = 3$ ergibt sich zunächst nach (1) und (2) die Parametrisierung eines kubischen Bézier Segmentes:

$$\vec{x}(t) = \sum_{i=0}^3 \vec{b}_i B_i^3(t) = \vec{b}_0 \cdot (1-t)^3 + \vec{b}_1 \cdot 3(1-t)^2 t + \vec{b}_2 \cdot 3(1-t)t^2 + \vec{b}_3 \cdot t^3. \quad (3)$$

Damit kann nachgerechnet werden, daß $\vec{x}(0) = \vec{b}_0$ und $\frac{d\vec{x}}{dt}(0) = 3B_0\vec{B}_1$.

Das bedeutet, daß das Segment den Anfangspunkt B_0 und die Anfangstangente B_0B_1 besitzt. Analog läßt sich nachweisen, daß das Segment in B_3 tangential zu B_2B_3 einläuft.

Andererseits ist die Darstellung (3) eine **konvexe** Kombination der Bézier Koeffizienten weil die Bernsteinpolynome $B_i^3(t) \geq 0$ die folgende Eigenschaft haben:

$$\sum_{i=0}^3 B_i^3(t) = 1. \quad (4)$$

Deshalb verläuft das Bézier Segment in Inneren der konvexen Hülle der Bézier Punkte.

2.3. Designeigenschaften

Der Verlauf des Bézier Segmentes kann durch Modifikation der Lage der Bézier Punkte sehr flexibel gestaltet werden. Dadurch eignen sich Bézier Segmente hervorragend für **Designzwecke**. Wird ein Bézier Punkt B_i in eine andere Lage B_i^* verschoben, so bewirkt diese Modifikation eine Verformung des Bézier Segmentes, die am stärksten in einer Umgebung des Parameters $t = i/3$ ausgeprägt ist. Dieses kann mit Hilfe der Parametrisierung (3) analytisch nachgerechnet werden.

Zur Veranschaulichung, werden wir den Bézier Punkt B_1 entlang der Richtung B_0B_1 in die Lage B_1^* verschieben und feststellen, daß die ganze Kurve in Richtung von $B_1B_1^*$ (ähnlich wie eine Marionette) 'nachgezogen'

wird. Die Verformung ist im Bereich des Parameterwertes $t = 1/3$ am stärksten ausgeprägt.

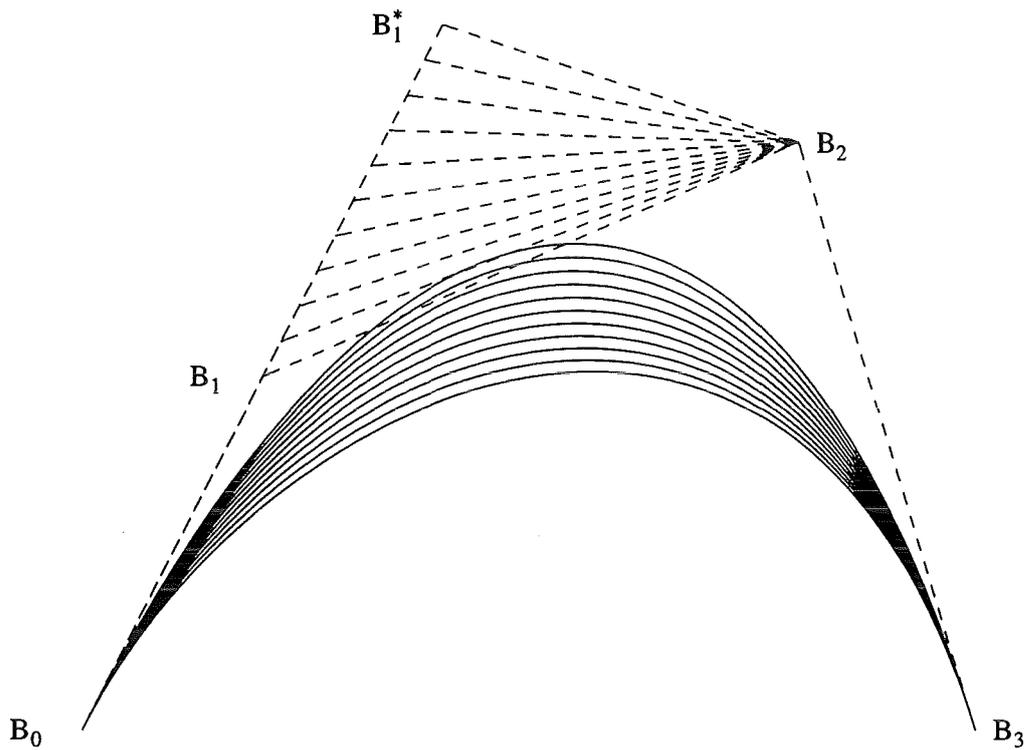


Bild 2. Designeigenschaften.

Im Bild 2 ist die Originalkurve mit einigen Zwischenstufen der Verformung zur Zielkurve abgebildet. Allerdings kann man diese Eigenschaft (die auch gelegentlich '**Gummihauteigenschaft**' genannt wird) am überzeugendsten interaktiv auf einem Grafikbildschirm simulieren.

2.4. Die affine Invarianz

Die schon im Abschnitt 2.2 erwähnte Eigenschaft (4) der Bernsteinpolynome sowie die Darstellung (3) zeigen auch, daß die Punkte des Bézier Segmentes **affine** Kombinationen der Bézier Punkte sind. Affine Kombinationen sind invariant gegenüber affinen Abbildungen. Das bedeutet, daß z. B. die Rotation der 4 Bézier Punkte eines Segmentes zur Bestimmung der Rotation des gesamten Bézier Segmentes ausreicht. Das rotierte Segment wird dann mit dem Algorithmus von de Casteljau aus den 4 Bildpunkten gerechnet (siehe auch die Bemerkungen zu den Implementierungsverfahren im Abschnitt 1). Das ist deshalb sinnvoll weil die Auswertung mit dem Algorithmus von de Casteljau (oder mit schnelleren Auswertungsverfahren) weniger Rechenzeit als die Multiplikation mit der Rotationsmatrix benötigt. Wenn man 'alle' Punkte (eine Punktauflösung mit gewünschter Dichte) des Bézier Segmentes rotieren müßte, würde das insgesamt viel mehr Zeit kosten.

So ermöglicht die 'Affinität' der Bézier Segmente eine beträchtliche **Rechnerzeitoptimierung** unter anderem auch bei den in der CAD/CAM Anwendung oft benutzten Manipulationen der Modelle wie Rotationen, Translationen und Zooms.

3. BÉZIER SPLINEKURVEN

Um kompliziertere Kurvenverläufe zu modellieren reicht in der Regel ein Bézier Segment (selbst höheren Grades als $n = 3$) nicht aus. Deshalb ist es sinnvoll diese als zusammengesetzte Kurven durch Aneinanderhängen von mehreren Bézier Segmenten mit niedrigem Grad, z.B. $n = 3$, zu modellieren. Dabei spielt die Qualität des Überganges an den Nahtstellen, wie z.B. Tangenten- oder Krümmungstetigkeit, eine entscheidende Rolle. Das führt zum Begriff einer Splinekurve.

Definition

Eine Bézier Splinekurve vom Grad n ist eine stückweise polynomial definierte Kurve

$$\vec{x}(t) = \sum_{i=0}^n \vec{b}_i^k B_i^n \left(\frac{t - t_{k-1}}{t_k - t_{k-1}} \right), \quad t_{k-1} \leq t < t_k, \quad k = 1, \dots, l, \quad (5)$$

die aus l Bézier Segmenten besteht und an den 'Nahtparametern' $t = t_k$, ($k = 1, 2, \dots, l-1$) $n-1$ mal stetig differenzierbar ist (C^{n-1} -stetig). Die Bézier Koeffizienten \vec{b}_i^k des k -ten Segmentes sind die Ortsvektoren der entsprechenden Bézier Punkte B_i^k , $i = 0, 1, \dots, n$; $k = 1, 2, \dots, l$.

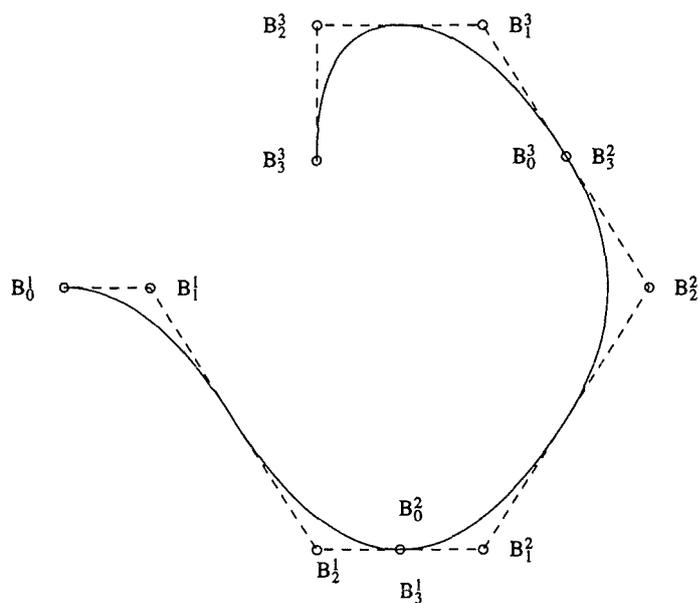


Bild 3. Eine kubische Bézier Splinekurve mit 3 Segmenten.

Bemerkungen

1) Die Bézier Punkte B_i^k , $k = 1, 2, \dots, l$, eines Splines mit l Segmenten sind nicht mit den Hilfspunkten B_i^j , $j = 1, 2, \dots, n$, des Algorithmus von de Casteljau für ein Bézier Segment zu verwechseln.

2) Für $n = 3$ erhält man die Definition kubischer Bézier Splinekurven, mit denen wir uns im folgenden befassen werden. Die meisten Ideen können allerdings auch für Bézier Splinekurven höheren Grades übertragen werden.

3) Die einzelnen Bézier Segmente s_k der Bézier Splinekurve sind im allgemeinen jeweils in Intervallen $I_k = [t_{k-1}, t_k]$ unterschiedlicher Länge $l_k = t_k - t_{k-1}$, $k = 1, 2, \dots, l$ definiert ('nichtuniforme' Splines). Die Parameter t_k , $k = 0, 1, \dots, l$ werden Parameterknoten genannt. Dabei sind t_0 und t_l die Randparameter der (nicht geschlossenen) Kurve und t_k die Parameter der 'Nahtpunkte' B_3^k für $k = 1, 2, \dots, l-1$. Für C^0 -stetige Kurven gilt offensichtlich für die Nahtstellen die Bedingung $B_3^k = B_0^{k+1}$, $k = 1, 2, \dots, l-1$.

4) Zur Berechnung der einzelnen Bézier Segmente mit Hilfe der Bernstein Polynome wird der **globale** Parameter $t \in I_k$ zuerst durch den normierten **lokalen** Parameter $u = \frac{t - t_{k-1}}{t_k - t_{k-1}} \in [0, 1]$ ersetzt.

5) Die **analytische** Bedingung der C^2 Stetigkeit für kubische Bézier Splinekurven hängt auch von der Parametrisierung ab und fordert mehr als die **geometrische** Bedingung der Tangenten- und Krümmungstetigkeit (GC^1 und GC^2 Stetigkeit).

Die C^1 (und C^2) Stetigkeitsbedingung an einem 'Nahtparameter' $t = t_k$ kann mit Hilfe der relativen Lage der entsprechenden Nahtstelle B_3^k (zwischen den Bézier Segmenten s_k und s_{k+1}) und den benachbarten Bézier Punkten B_2^k (und B_1^k) sowie B_1^{k+1} (und B_2^{k+1}) charakterisiert werden:

Satz

- 1) Eine Bézier Splinekurve ist in $t = t_k$, C^1 -stetig, wenn die Nahtstelle B_3^k und die benachbarten Bézier Punkte B_2^k und B_1^{k+1} kollinear sind, und B_3^k die Strecke $B_2^k B_1^{k+1}$ im selben Verhältnis $v_k = l_k : l_{k+1}$ unterteilt wie der Nahtparameter t_k die benachbarten Intervalle I_k und I_{k+1} .

II) Eine Bézier Splinekurve ist in $t = t_k$, C^2 -stetig, wenn die C^1 Stetigkeitsbedingung I erfüllt ist und es einen Punkt C_k gibt, der die folgenden Eigenschaften besitzt:

- a) C_k ist kollinear mit B_2^k und B_1^k , und B_2^k unterteilt die Strecke $B_1^k C_k$ im Verhältnis v_k .
- b) C_k ist kollinear mit B_1^{k+1} und B_2^{k+1} , und B_1^{k+1} unterteilt die Strecke $C_k B_2^{k+1}$ im Verhältnis v_k .

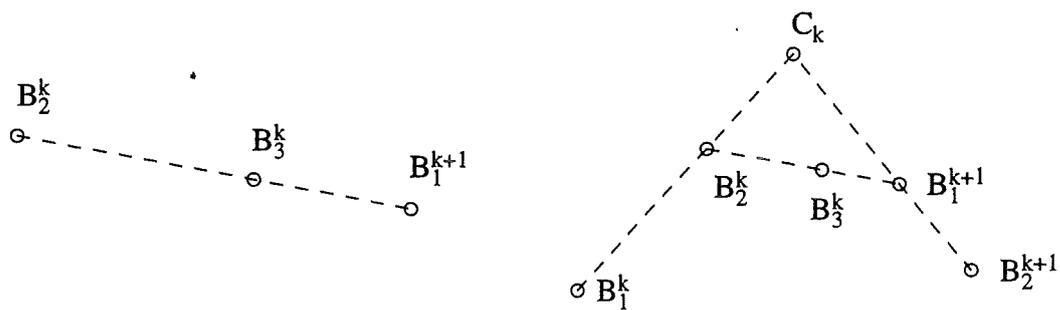


Bild 4. C^1 und C^2 Stetigkeitsbedingungen.

Bemerkungen

1) In **Kurzform** bedeutet C^1 Stetigkeit, daß eine Nahtstelle B_3^k die zwei benachbarten Bézier Punkte im Verhältnis v_k der entsprechenden Parameterintervalle 'interpoliert' und C^2 Stetigkeit bedeutet, daß es darüberhinaus einen Punkt C_k gibt, der die zwei benachbarten Bézier Punktepaare im selben Verhältnis v_k 'extrapoliert'.

2) Die Punkte'konstellationen' im Bild 4 können auch im Bild 1 erkannt werden. Hier entsteht durch den Algorithmus von de Casteljau eine C^3 - stetige Bézier Splinekurve mit 2 Segmenten, die in den Intervallen $[0, t]$ und $[t, 1]$ parametrisiert sind.

3) Der Beweis des Satzes basiert auf der Darstellung der 1. und 2. Ableitung einer Bézier Splinekurve und auf etwas Indexrechnung ([4]).

4) Die geometrischen Bedingungen I und II des Satzes ermöglichen eine bemerkenswerte **Datenreduktion**. Wenn man für jedes Segment die 4 Bézier Punkte speichern würde, so wäre eine Datenmenge von $N = 4 * l$ Punkten nötig.

Berücksichtigt man nun zunächst die C^0 Stetigkeit der Bézier Splinekurve, so reicht es die Nahtstellen nur einmal zu speichern. Die Datenmenge wäre dann $N_0 = 3 * l + 1$.

Die C^1 Stetigkeitsbedingung I ermöglicht es weiter die $l - 1$ Nahtpunkte auszusparen. Diese können durch lineare Interpolation ihrer Nachbarn berechnet werden. Dadurch wurde die Datenmenge auf $N_1 = 2 * l + 2$ Punkte reduziert.

Wenn man schließlich noch die C^2 Bedingung II nutzt, so kann man nachweisen, daß es reicht die ersten 2 Bézier Punkte des 1. Segmentes, die $l - 1$ Hilfspunkte C_k sowie die letzten 2 Bézier Punkte des letzten Segmentes zu speichern. Damit wird die minimale Punkteanzahl $N_2 = l + 3$ 'herausgefiltert' (im allgemeinen $l + n$ Punkte). Diese Punkte werden mit $D_i, i = 0, 1, \dots, l + 2$ bezeichnet und **B-Spline Punkte** genannt. Alle Bézier Punkte können aus den B-Spline Punkten mit Hilfe linearer Interpolation rekonstruiert werden (siehe [4]). Im Bild 5 sind die B-Spline Punkte $B_0^1, B_1^1, C_1, B_2^2, B_3^2$ mit x markiert; die Bézier Punkte sind mit o dargestellt.

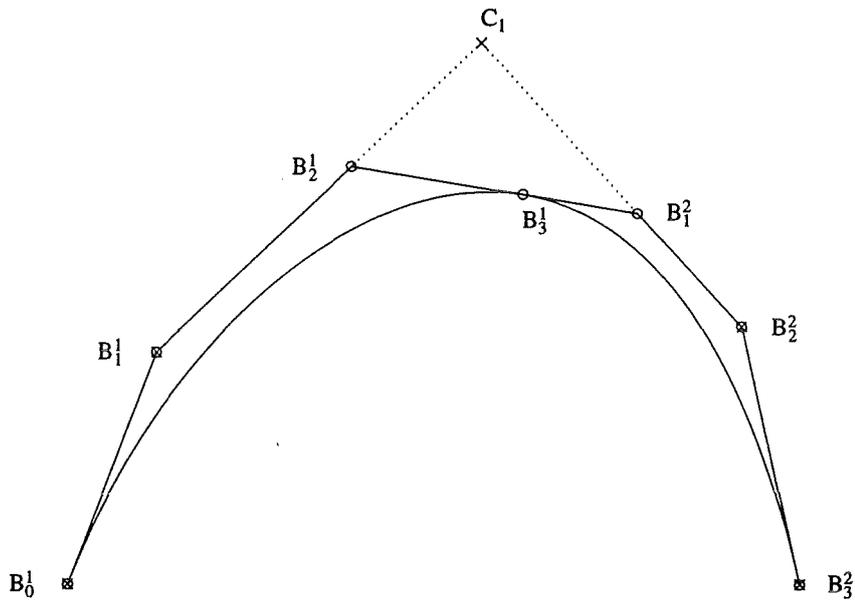


Bild 5. Bézier Punkte (o) und B- Spline Punkte (x).

4. B- SPLINE KURVEN

Die oben erwähnten geometrischen Überlegungen zur Datenreduktion führen zu dem analytischen Begriff einer B- Spline Kurve. Dabei handelt es sich um eine andere analytische **Darstellung** des selben geometrischen **Objektes** der von der ursprünglich eingeführten Bézier Splinekurve defi-

niert wurde. Die Darstellung erfolgt nun, anstatt in der Basis der Bernstein Polynome B_i^n , in der Basis der sogenannten B- Spline Funktionen N_i^n , wobei n der Grad der betrachteten Kurve ist.

Es werden zunächst die Parameterknoten s_j , für $j = 0, 1, \dots, l + 2 \cdot n - 2$ wie folgt eingeführt:

$$s_0 = s_1 = \dots = s_{n-1} = t_0 \quad (t_0 \text{ ist ein } n\text{-facher Knoten});$$

$$s_{n+i-1} = t_i, \quad i = 1, 2, \dots, l - 1 \quad (\text{einfache Knoten});$$

$$s_{l+n-1} = s_{l+n} = \dots = s_{l+2 \cdot n - 2} = t_l \quad (t_l \text{ ist ein } n\text{-facher Knoten}).$$

Die Randparameter t_0 und t_l werden damit n -fach in den Knotenvektor \vec{s} eingefügt. Die anderen Knoten sind einfach.

Eine B- Spline Kurve wird mit Hilfe der Knoten s_j im selben Intervall $[s_{n-1}, s_{l+n-1}] = [t_0, t_l]$ wie die Bézier Splinekurve parametrisiert.

Definition

Eine B- Spline Kurve mit l Segmenten vom Grad n wird definiert durch:

$$\vec{x}(s) = \sum_{i=0}^{l+n-1} \vec{d}_i N_i^n(s), \quad s \in [s_{n-1}, s_{l+n-1}]. \quad (6)$$

Die B- Spline Funktionen N_i^n sind rekursiv definiert als:

$$N_i^0(s) = \begin{cases} 1 & : \quad s_i \leq s < s_{i+1} \\ 0 & : \quad \text{sonst,} \end{cases}$$

$$N_i^n(s) = \frac{s - s_i}{s_{i+n} - s_i} N_i^{n-1}(s) + \frac{s_{i+n+1} - s}{s_{i+n+1} - s_{i+1}} N_{i+1}^{n-1}(s), \quad i = 0, 1, \dots, l + n - 1.$$

Bemerkungen

1) Die analytische Definition einer B- Spline Kurve ist wesentlich 'technischer' als die Definition einer Bézier Splinekurve. Der damit verbundene zusätzliche Programmieraufwand bei der Implementierung von B- Spline Kurven in ein CAD/CAM System wird aber nicht nur durch die schon erwähnte beträchtliche **Datenreduktion** rechtfertigt. Bei den in der CAGD Anwendung oft praktizierten **Nachbearbeitungen** der Kurven, die meist durch Modifikation der B- Spline Punkte implementiert sind, werden auch geometrische Eigenschaften wie Tangenten- und Krümmungsstetigkeit beibehalten. Das ist bei der Modifikation der Bézier Punkte nicht gesichert, weil dabei die Bedingungen des C^1/C^2 Stetigkeitssatzes (insbesondere die Kollinearität) im allgemeinen verletzt werden.

2) Die C^1 und C^2 Stetigkeitsbedingungen an den Nahtstellen sind implizit in der Definition einer B- Spline Kurve durch die Wahl des Knotenvektors \vec{s} (mit den entsprechenden Vielfachkeiten) mit Hilfe der B- Spline Funktionen N_i^n verschlüsselt.

Mit Hilfe der B- Spline Funktionen werden auch allgemeinere Kurven definiert, bei denen unterschiedliche Stetigkeitseigenschaften an den Nahtstellen vorliegen. Das wird im wesentlichen durch die Erhöhung der Vielfachheit der Nahtparameter erreicht. Bei der Vielfachheit r eines Parameterknotens liegt dann C^{n-r} Stetigkeit an der entsprechenden Nahtstelle vor. Solche Kurven sind in der CAD/CAM Anwendung stärker verbreitet als die maximal glatten B- Spline Kurven. Sie sind in der Praxis im Fachjargon unter dem Namen SPLINES bekannt. Die Differenzierbarkeitsordnung IDIF eines SPLINES ist die Ordnung der höchsten stetigen Ableitung an den Nahtstellen. Wenn z.B. ein SPLINE einen Knick besitzt, so ist IDIF=0; falls IDIF=1, so ist der SPLINE an allen Nahtstellen mindestens C^1 stetig.

3) In der CAD/CAM Konstruktion wird darauf geachtet, daß die erzeugten SPLINES so 'glatt' wie möglich sind, insbesondere daß Tangenten- und Krümmungsstetigkeit vorliegt. Darüberhinaus ist es wünschenswert, daß im Krümmungsdiagramm keine unnötigen Vorzeichenwechsel vorkommen. Damit lassen sich unerwünschte Krümmungsänderungen (sogenannte **Schwingungen**) der Kurve vermeiden. In der Praxis spielen vor allem kubische SPLINES eine besondere Rolle. SPLINES höheren Grades benötigen mehr Speicherplatz und besitzen eventuell zu viele unerwünschte Schwingungen.

5. AUSBLICK

I. Die Ideen die wir für Kurven in den bisherigen Abschnitten im Überblick vorgestellt haben, lassen sich relativ direkt auch für Flächen übertragen. Das wird durch die sogenannte **Tensorprodukt** Methode ermöglicht.

Definition

Ein 'Tensorprodukt Bézier Flächenpatch' vom Grad n in u und Grad m in v wird definiert als:

$$\vec{x}(u, v) = \sum_{i=0}^n \sum_{j=0}^m b_{ij} \vec{B}_i^n(u) B_j^m(v), \quad (u, v) \in [0, 1] \times [0, 1].$$

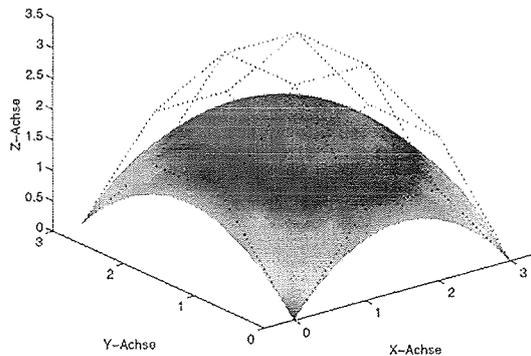


Bild6. Ein bikubisches Flächenpatch mit Bézier Netz.

Eine Bézier Spline Fläche wird aus mehreren Patches aufgebaut und ist $C^{n-1} \times C^{m-1}$ - stetig bei den Übergängen zwischen den einzelnen Patches. In der Praxis werden die Stetigkeitsbedingungen bei Bedarf etwas abgeschwächt und allgemeinere Flächen, sogenannte SURFS, erzeugt.

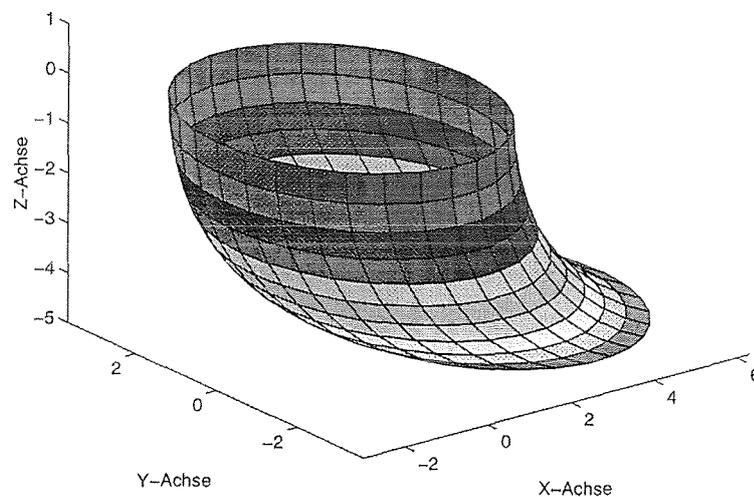


Bild7. Eine SURF.

II. Die Verfeinerung der CAD/CAM Grundlagen, Entwicklung und Anwendung ist ein kontinuierlicher Prozess, der in der letzten Zeit besonders rasant fortschreitet. Dem Sprung von den hier beschriebenen Bézier Splines zu B- Splines folgte relativ kurz danach die Einführung der NURBS (nicht uniforme rationale B- Splines). NURBS sind zwar zur Zeit 'the state of the art' im CAD/CAM Bereich, doch neue Ideen wie 'parametric design', 'variational design' usw. machen die Entwicklung auf diesem Gebiet weiterhin

interessant und spannend.

III. Literatur

1. C. de Boor, *Spline Toolbox for use with MATLAB*, The Math Works, 1992.
2. J. Hoschek (Ed.), *Freeform Tools in CAD Systems*, Teubner Stuttgart, 1991
3. J. Hoschek, D. Lasser, *Grundlagen der geometrischen Datenverarbeitung*, Teubner Stuttgart, 1992.
4. G. Farin, *Curves and Surfaces for CAGD*, Academic Press, 1993.
5. L. Piegl, W. Tiller, *The NURBS Book*, Springer, 1995.
6. D. E. Ulmet, *Geodesic Offsets of Spline Curves on Spline Surfaces - An Industrial Perspective*, erscheint 1996.



Dr. Johannes Grabmeier

Dr. Johannes Grabmeier diplomierte im Fach Mathematik an der Technischen Universität München. 1981 wurde er Assistent an der Universität Bayreuth, wo er 1986 im Bereich der Darstellungsgeometrie promovierte. Im Anschluß wechselte er zu IBM Deutschland und arbeitet an dem Computeralgebra System AXIOM. Er zeichnet sich verantwortlich für viele AXIOM-packages, wie z.B. über Darstellungstheorie, Permutationsgruppen, endliche Felder. Dr. Grabmeier ist Mitautor des Buches "AXIOM: The Scientific Computation System", Springer Verlag 1992 (Hrg. R.D.Jenks, R.S.Sutor).

Computeralgebra Systeme sind attraktive Werkzeuge in der Mathematik, um Rechnungen symbolisch auf einem Rechner durchzuführen. Eines der Herzstücke aller Computeralgebra Systeme ist die symbolische Integration. Hierbei gilt es zu entscheiden, ob es zu einer vorgegebenen elementaren Funktion eine elementare Stammfunktion existiert oder zu zeigen, daß es keine solche geben kann. Zu dieser sehr komplizierten Aufgabenstellung gibt es einen Entscheidungsalgorithmus, der im Rahmen der Theorie der Differentialalgebra diskutiert wird. Der Vortrag "Computeralgebra-Algorithmen zur symbolischen Integration" soll einen Einstieg in die Probleme der symbolischen Integration liefern und anhand von Beispielen die Lösungsalgorithmen beschreiben.

Eine Einführung in die Computeralgebra-Algorithmen zur symbolischen Integration

Johannes Grabmeier

Zusammenfassung

Eines der zentralen Gebiete der Computeralgebra ist die symbolische Integration. Während die in den klassischen Analysisvorlesungen diskutierten Methoden wie Substitution und partielle Integration von heuristischer Natur sind, – der dort oft vorgestellte *Algorithmus* zur Integration rationaler Funktionen mittels Partialbruchzerlegung hingegen unterstellt die Kenntnis der Faktorisierung des Nenners über den reellen Zahlen, – werden in der Computeralgebra Entscheidungsverfahren untersucht und implementiert:

Man finde algorithmisch zu jeder vorgegebenen elementaren Funktionen eine elementare Stammfunktion oder beweise (algorithmisch), daß es keine solche geben kann.

Zur Lösung so gestellter Fragen wird insbesondere an die algorithmischen Methoden der Mathematik des letzten Jahrhunderts (z.B. E. HERMITE) angeknüpft.

Während G.H. HARDY am Anfang dieses Jahrhunderts noch vermutete, daß es für dieses Problem keine Lösung geben kann, wurde 1969 von R.H. RISCH ein Entscheidungsverfahren angegeben. Viele weitere Arbeiten im Anschluß daran aber haben erst die Implementierung für gewisse Teilklassen von Funktionen z.B. in den Computeralgebra-Systemen MAPLE oder AXIOM ermöglicht. Insbesondere sind hier die Arbeiten von B.M. TRAGER, M. ROTHSTEIN und M. BRONSTEIN zu nennen.

Die Intention dieser Arbeit ist es durch eine elementare Einführung in dieses faszinierende Gebiet zur Rezeption dieser Entwicklungen beizutragen. Dies ist eine sowohl herausfordernde wie m.E. notwendige Aufgabe. Es ist nämlich das Spektrum der für die Integrationsalgorithmen notwendigen mathematischen Methoden und Gebiete sehr breit. Angesichts der erfreulicherweise ständig wachsenden Verbreitung und Nutzung von Computeralgebra-Systemen besteht die Gefahr, daß die Kluft zwischen Anwendung und Kenntnis, was algorithmisch im Computer passiert, größer ist als bei anderen Teilgebieten der Computeralgebra. Es ist daher die Absicht mit dieser Übersichtsarbeit besonders im pädagogischen Bereich die Behandlung zumindest ausgewählter Teile dieser Algorithmen zu erleichtern und eine Hinführung zur demnächst erscheinenden ersten Monographie [10] zu diesem Thema von M. BRONSTEIN zu geben.

Inhaltsverzeichnis

1	Einleitung	72
2	Probleme mit den herkömmlichen Methoden	73
2.1	Regelbasierte Heuristiken	73
2.2	Geschlossene Darstellung eines Integrals	73
2.3	Ist die Integration wirklich linear?	74
2.4	Entscheidbarkeit des Integrationsproblems	74
2.5	Ein „Algorithmus“ zur Integration rationaler Funktionen	75
2.6	Integraltafeln	75
3	Integration rationaler Funktionen	76
3.1	Reduktion durch den euklidischen Algorithmus	76
3.2	Partialbruchzerlegung und strukturelle Einsichten	76
3.3	Die Forderung nach rationalen Algorithmen	76
3.4	Quadratfreie Faktorisierung des Nennerpolynoms	77
3.5	Hermiteische Reduktion	78
3.6	Varianten ohne Partialbruchzerlegung von Mack und von Ostrogradsky-Horowitz	79
3.7	Der Rothstein-Trager-Algorithmus für den logarithmischen Anteil	81
3.8	Erinnerung an die Resultante	82
3.9	Ein rationaler Algorithmus zur Bestimmung des Integrals einer rationalen Funktion	83
3.10	Reelle Darstellungen	84
4	Differentialalgebra und elementare Erweiterungen	86
4.1	Kann der Algorithmus für die Integration rationaler Funktionen auf allgemeinere Situationen übertragen werden?	86
4.2	Differentialalgebra	86
4.3	Erweiterung durch Logarithmen, Exponentialfunktionen und algebraischen Funktionen	87
4.4	Warnung bei der Interpretation für reelle Funktionen	88
4.5	Monome – in diesem Kontext transzendente Elemente, deren Ableitung ein Polynom ist und die die Konstanten nicht verändern	88
4.6	Wann sind Exponentialfunktionen und Logarithmen transzendent?	89
4.7	Elementare Funktionen	89
4.8	Verlust der Teilerfremdheit eines irreduziblen Nenners mit seiner Ableitung	89
4.9	Das Monoid der speziellen Polynome	90
4.10	Faktorisierung nach speziellen und normalen Anteilen	91
4.11	Normale Nenner, Reduktion nach Hermite und einfache Funktionen	91
4.12	Reduzierte Funktionen: der Nenner ist speziell	92
4.13	Faktorisierung und Bewertungen	92
4.14	Der Satz von Liouville	93
5	Integration transzendenter Funktionen	97
5.1	Allgemeine Struktur des Algorithmus	97
5.2	Diagramm zur Integration transzendenter Funktionen	98
5.3	Residuenkriterium von Rothstein und Trager für einfache Funktionen	99
5.4	Polynomreduktion für nicht-lineare Monome auf Polynome höchstens vom Grad des Ableitungspolynoms minus zwei	99

5.5	Der Logarithmus im speziellen Fall – Integration von Polynomen primitiver Monome	100
5.6	Die Exponentialfunktion im speziellen Fall – Integration von Laurentpolynomen und die Differentialgleichung von RISCHE	101
5.7	Die Tangensfunktion im speziellen Fall – gekoppelte lineare Differentialgleichungen	102
5.8	Die Risch-Norman-Methode	102
6	Die Integration algebraischer Funktionen	103
6.1	Normierte Darstellung einer algebraischen Funktion	103
6.2	Ganzheitsbasen	104
6.3	Konstruktion von Ganzheitsbasen	104
6.4	Ganzheitsbasen für einfache Radikalerweiterungen	105
6.5	Große Grade der polynomialen Zählerkoeffizienten – Polstelle im Unendlichen	106
6.6	Simulation von Hermite	106
6.7	Der logarithmische Anteil	107
7	Ausblick	108
7.1	Übersichtsarbeiten	108
7.2	Spezielle Funktionen	108
7.3	Definite Integration	108
7.4	Symbolisches Lösen von Differentialgleichungen	109
8	Dank	109

1 Einleitung

Die wichtigsten allgemein bekannten und benutzten Methoden zum Auffinden von Stammfunktion von Funktionen sind *partielle Integration* *Substitution von Variablen* und im Falle der rationalen Funktionen die *Partialbruchzerlegung* nach einer Faktorisierung des Nenners. Zusätzlich kann oft die Linearität des Integrals ausgenutzt und der Hauptsatz der Differential- und Integralrechnung angewandt werden. Schließlich greift man auch auf Integrationstabellen zurück.

Mit der Unterrichtung und Einübung dieser aufgezählten Techniken erschöpft sich in der Regel nicht nur der Mathematikunterricht in den Schulen zu diesem Thema, sondern zumeist auch eine Analysisvorlesung für Ingenieur- oder Mathematikstudenten.

In diesem Aufsatz wird der Einsatz des Computers für diese Problemstellung diskutiert und eine elementare Einführung in die algorithmische Integrationstheorie gegeben, wie sie heute in vielen Computeralgebra-Systemen verwendet wird. Erstaunlich dabei ist, daß sie – obgleich diese gänzliche verschiedenen und andersartigen Methoden bereits ihre Wurzeln im letzten Jahrhundert ([31], [49], [44]) haben und auch für den Fall rationaler Funktionen durchaus elementaren Charakter haben – bislang kaum Einzug in aktuelle Vorlesungs- und Unterrichtsinhalte gefunden haben und nur in einigen wenigen Spezialvorlesungen zur Computeralgebra gelehrt werden.

Die algebraischen(!) Methoden, die sich hier erfolgreich bewährt haben, werden heute dem Gebiet der Computeralgebra – siehe [13] oder den Übersichtsartikel [28] – mit seinen exakten und symbolischen Verfahren zugeordnet. In der Computeralgebra, die nach R. LOOS ([11])

der Teil der Informatik *und der Mathematik*¹ ist, der *algebraische Algorithmen* entwickelt, analysiert, implementiert und anwendet,

sieht man es als Ziel an, für Probleme wie der Bestimmung von Stammfunktionen ein Entscheidungsverfahren anzugeben und nicht einfach nur die eingangs aufgelisteten Methoden von heuristischer Natur zu implementieren. Eine adäquate und präzise Formulierung dieser Problemstellung findet sich im Buch [23] von J.H. DAVENPORT et al.:

Zu je zwei Funktionenklassen \mathcal{F} und \mathcal{G} , deren Ableitungen \mathcal{G}' alle in \mathcal{F} liegen, gebe man einen Algorithmus an, der zu jeder vorgelegten Funktion f aus \mathcal{F} eine Funktion g aus \mathcal{G} mit Ableitungen $g' = f$ berechnet, falls ein solches g existiert, oder aber automatisch beweist, daß ein solches g nicht in \mathcal{G} existiert.

Im Abschnitt 3 werden wir für die Situation, daß \mathcal{F} die Klasse der rationalen Funktionen in x ist und \mathcal{G} diejenige Klasse von Funktionen, die sich durch rationale Ausdrücke in x und \log ausdrücken lassen einen solchen Algorithmus diskutieren. Im Rest der Arbeit geht es im wesentlichen um die Situation, daß sowohl \mathcal{F} als auch \mathcal{G} die Klasse der elementaren Funktionen ist.

¹Diese Ergänzung stammt vom Autor.

2 Probleme mit den herkömmlichen Methoden

2.1 Regelbasierte Heuristiken

Bereits in den sechziger Jahren – siehe etwa [65] und [47] – wurden Computerimplementierungen durchgeführt, die einerseits die klassischen Methoden am Computer als heuristische, regelbasierte Implementierungen realisierten, andererseits aber bereits auch Methoden der algorithmischen Integrationstheorie benutzten.

Wie oft in solchen Situationen ist man bei einer Implementierung der *Heuristiken* partielle Integration und Substitution von Variablen bei vielen vorgelegten Funktionen sehr schnell in der Lage, eine Stammfunktion zu finden, andererseits hat man aber nach n erfolglosen Versuchen – auch eine Computerrealisierung muß hier eine Schranke einbauen, wengleich diese höher sein kann als bei einem Schüler, der im Abitur die *richtige* Substitution sucht – nie Einsicht bekommen, ob es vielleicht keine *geschlossene* Darstellung des Integrals gibt.

2.2 Geschlossene Darstellung eines Integrals

Hier entsteht sofort die nächste Problemstellung. Was soll man unter *geschlossener Darstellung* eines Integrals verstehen? Es ist allgemein bekannt, – und wir werden im Abschnitt 5.6 auf Seite 101 zeigen, warum das so ist – daß die Gaußsche Fehlerfunktion $\int e^{-x^2} dx$ keine *elementare* Darstellung besitzt. Häufig wird die Formulierung *geschlossene Darstellung* synonym zur Aussage benutzt, daß das Integral eine *elementare* Funktion ist. *Elementare Funktionen* sind solche, die sich rekursiv durch rationale Ausdrücke in x, i, \exp und \log über den rationalen Zahlen \mathbb{Q} sowie durch Bildung algebraischer Funktionen y konstruieren lassen. Man beachte, daß mit \exp und i wegen $\cos(x) = \frac{1}{2}(e^{ix} + e^{-ix})$ und $\sin(x) = \frac{1}{2i}(e^{ix} - e^{-ix})$ auch die trigonometrischen Funktionen enthalten. Man sehe aber dazu auch den Abschnitt 3.10 auf Seite 84, in dem wir reelle Darstellungen eines Integrals diskutieren.

Ein wichtiges Beispiel für algebraische Funktionen, das heißt Funktionen y für die es ein bivariates Polynom F gibt, dessen Koeffizienten elementare Funktionen sein dürfen, so daß $F(x, y) = 0$ gilt, sind die Wurzelfunktionen, die einer Gleichung wie etwa $y^n - \sin\left(\frac{\log(1+x)}{\tan x}\right) = 0$ genügen. Ein nicht-triviales Beispiel für eine algebraische Funktion ist $y := \tan(\operatorname{atan}(x)/3)$ mit $0 = F(x, y) = y^3 - 3xy^2 - 3y + x$ und Integral²

$$\frac{8 \log \left(3 \tan \left(\frac{\arctan(x)}{3} \right)^2 - 1 \right) - 3 \tan \left(\frac{\arctan(x)}{3} \right)^2 + 18 x \tan \left(\frac{\arctan(x)}{3} \right)}{18}$$

Für eine mathematisch ausführlichere Diskussion des Begriffes *elementare Funktion* im Kontext von Differentialalgebra sehe man weiter unten im Abschnitt 4.2 auf Seite 86 und z.B. in [10].

Unsere Aufgabenstellung läßt sich also dahingehend präzisieren, daß wir sowohl für \mathcal{F} als auch für Klasse der möglichen Stammfunktionen \mathcal{G} die Klasse der elementaren Funktionen wählen. Da die Funktion $\exp(x^2) = e^{x^2}$ nicht elementar integrierbar ist, müßte also ein Entscheidungsverfahren der Computeralgebra gemäß dem Anspruch der oben gegebenen Definition einen Beweis erbringen, daß es keine elementare Funktion g geben kann, die $g'(x) = e^{x^2}$ erfüllt.

Bemerkenswert ist auch die Tatsache, daß es nur Nuancen im Integranden sein können, die eine elementar integrierbare Funktion von einer nicht elementar integrier-

²Mit AXIOM berechnet, weder aktuelle Versionen von MAPLE noch MATHEMATICA können diese Funktionen symbolisch integrieren.

baren Funktionen unterscheiden. Beispielsweise unterscheiden sich die beiden folgenden Funktionen nur durch das Vorzeichen einer Konstanten:

$$f_+ := \frac{x+1}{(x-2)(\sqrt{x^3+1})}, \quad f_- := \frac{x-1}{(x-2)(\sqrt{x^3+1})}, \quad (1)$$

es gilt jedoch $\int f_+ dx = -\frac{1}{3} \log \left(\frac{(6x+6)\sqrt{x^3+1} + x^3 + 12x^2 - 6x + 10}{x^3 - 6x^2 + 12x - 8} \right)$, während $\int f_- dx$ nicht elementar darstellbar ist.

2.3 Ist die Integration wirklich linear?

So natürlich uns der Begriff der elementaren Funktionen erscheint, mit der Linearität des Integrals verträgt er sich im allgemeinen nicht! Die Funktion $x^x + \log(x)x^x$ besitzt die elementare Stammfunktion $x^x (= e^{x \log x} = \exp(x \log x))$, während die Summanden x^x und $\log(x)x^x$ nicht elementar integrierbar sind. Wir können also die Linearität des Integrals nur eingeschränkt für unsere Problemstellung benutzen!

2.4 Entscheidbarkeit des Integrationsproblems

Auch die Frage nach der Entscheidbarkeit des gestellten Problems stellt sich: Kann es überhaupt einen solchen Algorithmus geben, der den gestellten Anforderungen genügt? In der Tat hatten Mathematiker wie G.H. HARDY, siehe [29] auf Seite 7, noch am Anfang dieses Jahrhunderts vermutet, daß das Problem für die elementaren Funktionen nie gelöst werden würde, wobei es offen bleiben muß, ob er das Problem für so schwierig erachtete oder aber bereits eine Vorstellung über die Unentscheidbarkeit von Problemen vorhanden war:

Complete answers to these questions have not and probably never will be given.

Interessanterweise hat der Autor diese Passage in der 2. Auflage [30] auf Seite 8 durch folgenden Satz ersetzt:

It would be unreasonable to expect complete answers to these questions.

Bei der Frage nach der Unentscheidbarkeit eines solchen Problems kommt es nun sehr auf die präzise Problemstellung an. D. RICHARDSON hat 1968 in [52] gezeigt, daß es für das Problem der Integration von Funktionen aus $\mathbb{R}(x, i, \pi, |\cdot|, \exp, \log)$ kein Entscheidungsverfahren geben kann. Man kann nämlich in diesem Funktionenraum eine Konstanten c konstruieren, für die der Test auf 0 unentscheidbar ist. Damit wird auch sofort die Frage nach der Integration der Funktion $ce^{(x^2)}$ unentscheidbar. Dieses Beispiel gibt zusätzlich einen Hinweis darauf, daß es nicht gelingen kann, solche Verfahren rein analytisch für den Raum der reellen Funktionen zu suchen. Weiter unten werden wir auf die notwendigen Algebraisierungen und die dann notwendigen richtigen Interpretationen für reelle oder komplexe Funktionen noch eingehen, siehe die Abschnitte 4.2 auf Seite 86 und 4.4 auf Seite 88.

Dennoch braucht man an dieser Stelle nicht den Mut verlieren. Um diesem Problem zu begegnen, werden in der Computeralgebra in der Regel nur *konstruktive Körper* betrachtet. Ein Körper K ist *konstruktiv*, wenn die Menge seiner Elemente *entscheidbar* ist. Das soll heißen, daß es einen Algorithmus gibt, der zu je zwei Elementen $a, b \in K$ entscheidet ob $a = b$ gilt oder nicht und wenn es Algorithmen für die Strukturabbildungen gibt, d.h. ein Algorithmen, die zu gegebenem $a, b \in K$ $a + b$, ab , $-a$ und zu $a \neq 0$ auch a^{-1} berechnen. Zu dieser Definition sehe man auch den Abschnitt 4.5 in [3].

In der Tat hat nun R.H. RISCH 1969 – siehe [53, 54] ein Entscheidungsverfahren angegeben, das zumindest theoretisch das Integrationsproblem für elementare Funktionen gelöst hat. Dieses Verfahren mit seinen seither erzielten wesentlichen Verbesserungen, die es erst für Computeralgebra-Systeme praktikabel gemacht haben, ist das Thema dieser Übersichtsarbeit.

2.5 Ein „Algorithmus“ zur Integration rationaler Funktionen

Bei den rationalen Funktionen wird oft ein „Algorithmus“ angegeben, der ohne weiteren Kommentar eine Faktorisierung des Nenners in irreduzible Polynome über den reellen Zahlen vom Grad 1 und 2 annimmt und dann mittels Partialbruchzerlegung und Auflisten der verschiedenen Fälle vorgibt, das Problem gelöst zu haben. Unser Haupteinwand hier ist, daß man über den reellen Zahlen in der Regel eben nicht exakt faktorisieren kann und damit bereits die Voraussetzung dieses „Algorithmus“ nicht erfüllbar ist. Die Übungsbeispiele sind typischerweise dann auch von der Gestalt, daß sich die Nenner bereits über den rationalen Zahlen durch Hinschauen in lineare und quadratische Faktoren zerlegen lassen. Wie wir in Abschnitt 3.5 auf Seite 78 sehen werden, ist auch eine (aufwendige) Faktorisierung des Nennerpolynoms in irreduzible Faktoren gar nicht nötig, um das Integral einer rationalen Funktion über den rationalen Zahlen zu bestimmen.

2.6 Integraltafeln

Schließlich ist auch die Lösung mit Integraltafeln nur mit großer Vorsicht zu behandeln. So haben etwa M. KLERER und F. GROSSMAN in [36, 37] Fehlerraten in gängigen Nachschlagewerken mit bis zu zweistelligen Prozentanteilen aufgedeckt. Für eine zweifach geschachtelte Wurzel wie

$$\frac{\sqrt{\sqrt{x^2 + a^2} + x}}{x}$$

hat J.H. DAVENPORT ([22]) nur eine einzige Tafel gefunden, die dieses Integral angibt – leider aber falsch. Das Computeralgebra-System AXIOM benötigt weniger als 2 Sekunden Rechenzeit³ um für die Fälle $a > 0$ als auch $a < 0$ reelle Stammfunktionen zu bestimmen, im ersten Fall ist das

$$\sqrt{a} \log \left(\frac{-2 \sqrt{a} \sqrt{\sqrt{x^2 + a^2} + x + \sqrt{x^2 + a^2} + x + a}}{\sqrt{x^2 + a^2} + x - a} \right) - 2 \sqrt{a} \arctan \left(\frac{\sqrt{\sqrt{x^2 + a^2} + x}}{\sqrt{a}} \right) + 2 \sqrt{\sqrt{x^2 + a^2} + x},$$

im zweiten Fall

$$\sqrt{-a} \log \left(\frac{-2 \sqrt{-a} \sqrt{\sqrt{x^2 + a^2} + x + \sqrt{x^2 + a^2} + x - a}}{\sqrt{x^2 + a^2} + x + a} \right) - 2 \sqrt{-a} \arctan \left(\frac{\sqrt{\sqrt{x^2 + a^2} + x}}{\sqrt{-a}} \right) + 2 \sqrt{\sqrt{x^2 + a^2} + x}.$$

Der Fall $a = 0$ ist übrigens hier nicht direkt ablesbar und muß extra bestimmt werden: $\int \frac{\sqrt{\sqrt{x^2 + x}}}{x} dx = 2 \sqrt{\sqrt{x^2 + x}}$. Das parameterabhängige Rechnen in der Computeralgebra ist Gegenstand von aktueller Forschung und noch weit von einer befriedigenden Lösung entfernt, für das Lösen von parameterabhängigen Gleichungen sehe man den Übersichtsartikel [71].

³auf einer IBM RS/6000 550

3 Integration rationaler Funktionen

3.1 Reduktion durch den euklidischen Algorithmus

Bei der Integration einer rationalen Funktion $f = \frac{z}{d} \in \mathbb{Q}(x)$ kann man auf Grund der Division mit Rest und des euklidischen Algorithmus annehmen, daß das Zählerpolynom $z \in \mathbb{Q}[x]$ und das Nennerpolynom $d \in \mathbb{Q}[x]$ teilerfremd sind und daß es Polynome p und a gibt mit

$$f = p + \frac{a}{d}, \quad \deg a < \deg d.$$

Weiterhin nehmen wir nun für unsere theoretische Vorüberlegungen an, daß eine Faktorisierung des Nennerpolynoms

$$d = \prod_{i=1}^s (x - \alpha_i)^{e_i} \quad (2)$$

über einem *Zerfällungskörper* vorliegt; das bedeutet, es sind genügend viele algebraische Zahlen – z.B. die α_i – dem Körper der rationalen Zahlen adjungiert, so daß eine Faktorisierung in lineare Polynome möglich ist. Beispielsweise ist $\mathbb{Q}(\sqrt{2})$ Zerfällungskörper für $x^2 - 2$ und $\mathbb{Q}(i)$ Zerfällungskörper für $x^2 + 1$.

3.2 Partialbruchzerlegung und strukturelle Einsichten

Mittels Partialbruchzerlegung und Elimination aller nicht-konstanten Zähler unter Hinzunahme von kleineren Potenzen im Nenner kann man Elemente $\alpha_{i,j}$ im Zerfällungskörper finden mit

$$f = p + \sum_{i=1}^s \sum_{j=1}^{e_i} \frac{\alpha_{i,j}}{(x - \alpha_i)^j},$$

und das Integral von f kann über einem Zerfällungskörper des Nenners leicht bestimmt werden:

$$\int f \, dx = \underbrace{\int p \, dx + \sum_{i=1}^s \sum_{j=2}^{e_i} \left(-\frac{1}{j-1}\right) \frac{\alpha_{i,j}}{(x - \alpha_i)^{j-1}}}_{\text{rationaler Anteil}} + \underbrace{\sum_{i=1}^s \alpha_{i,1} \log(x - \alpha_i)}_{\text{transzendenter Anteil}}.$$

Durch diese Überlegungen ist es klar, daß jede Stammfunktion einer rationalen Funktion in eine Summe eines *rationalen Anteils* und eines *transzendenten* oder *logarithmischen Anteils* darstellbar ist. Diese ist eine wichtige *strukturelle Erkenntnis* über die Gestalt des Integrals, die später noch eine zentrale Rolle spielen wird. Weiter ist festzustellen, daß die Polynome, die als Argumente der Logarithmen auftreten – auch nach Zusammenfassung mittels $\log(f) + \log(g) = \log(fg)$ – die Eigenschaft besitzen, quadratfrei zu sein. Dabei nennen wir ein Polynom *quadratfrei*, wenn es keine mehrfachen Nullstellen in einem Erweiterungskörper besitzt. Umgekehrt tragen nur die mehrfachen Nullstellen zum rationalen Anteil des Integrals bei!

3.3 Die Forderung nach rationalen Algorithmen

Ein einfaches Beispiel ist $\frac{x^5 - 2x^3 + x^2 - 1}{x^2 - 2} = \frac{1}{x^2 - 2} + x^3 + 1$ mit Zerfällungskörper $\mathbb{Q}(\sqrt{2})$ und Integral $-\frac{1}{2\sqrt{2}} \log(x + \sqrt{2}) + \frac{1}{2\sqrt{2}} \log(x - \sqrt{2}) + \left(\frac{1}{4}x^4 + x\right) = \frac{1}{2\sqrt{2}} \log\left(\frac{(x^2 - 2\sqrt{2} + 2)}{x^2 - 2}\right) + \left(\frac{1}{4}x^4 + x\right)$ gemäß der Partialbruchzerlegung $x^3 + 1 + \frac{\frac{1}{2\sqrt{2}}}{x - \sqrt{2}} - \frac{\frac{1}{2\sqrt{2}}}{x + \sqrt{2}}$.

Will man dagegen das nächste Beispiel

$$f := \frac{7x^{13} + 10x^8 + 14x^7 - 7x^6 - 4x^3 - 4x^2 + 3x + 3}{x^{14} - 2x^8 - 2x^7 - 2x^4 - 4x^3 - x^2 + 2x + 1}$$

von B.F. CAVINESS aus [12] mit dieser Methode anpacken, wird man sehr schnell feststellen, daß das Nennerpolynom irreduzibel über \mathbb{Q} ist. Wenn wir nun die Nullstellen des Nenners formal adjungieren, d.h. im Zerfällungskörper rechnen, ist zunächst das Nennerpolynom über dieser Erweiterung des Körpers der rationalen Zahlen zu faktorisieren. Das kann sehr aufwendig sein, da der Zerfällungskörper eines Polynoms vom Grad n i.a. den Körpererweiterungsgrad $n!$ besitzt. Es stellt sich hier folglich die Frage, ob man diese Faktorisierung nicht vermeiden kann.

Wird über $\mathbb{Q}(\sqrt{2})$ gerechnet, so erkennt man daß es zur Darstellung der Stammfunktion ausreicht, eine *algebraische Erweiterung*⁴ von \mathbb{Q} vom Grad 2 statt vom Grad 14! durchzuführen:

$$\int f dx = \frac{\sqrt{2} + 1}{2} \log(x^7 - \sqrt{2}x^2 - (\sqrt{2} + 1)x - 1) - \frac{\sqrt{2} + 1}{2} \log(x^7 + \sqrt{2}x^2 + (\sqrt{2} - 1)x - 1).$$

Das wirft zwei weitere Forderungen auf. Der Algorithmus zur Integration rationaler Funktionen sollte zum einen eine *minimale* Körpererweiterung bestimmen können, über der das Integral darstellbar ist. Zum anderen soll möglichst vermieden werden, in diesen Körpererweiterungen überhaupt zu rechnen. Ist dies gegeben, spricht man davon, daß ein *rationaler* Algorithmus vorliegt.

3.4 Quadratfreie Faktorisierung des Nennerpolynoms

Im Hinblick auf die Gestalt des logarithmischen Anteils ist es eigentlich nicht notwendig das Nennerpolynom über einem Zerfällungskörper in *lineare* Faktoren zu zerlegen. Vielmehr genügt es, das Teilerpolynom zu bestimmen, das alle einfachen Nullstellen von d enthält. Allgemeiner spricht man von einer *quadratfreien* Faktorisierung eines Polynoms d , wenn für eine natürliche Zahl m

$$d = \prod_{i=1}^m d_i^i$$

gilt und wenn im Polynom d_i alle i -fachen Nullstellen von d – aber natürlich nur einfach – aufgesammelt sind; d_i kommt dann genau i -fach in d vor. Weiterhin ist nach Konstruktion klar, daß diese quadratfreien Polynomteiler d_i paarweise teilerfremd sind, d.h. es gilt $\text{ggT}(d_i, d_j) = 1$ für $i \neq j$.

Wie können nun diese Teiler d_i sinnvoll konstruiert werden? Es würde ja hier sicher keinen Sinn machen, über dem Zerfällungskörper die vollständige Faktorisierung durchzuführen und dann die jeweils zusammengehörenden linearen Faktoren auszumultiplizieren. Auch haben wir die Frage nach den Koeffizienten der d_i noch zu klären. Welcher Erweiterungskörper der rationalen Zahlen ist für diese Koeffizienten notwendig? Zur Beantwortung dieser Fragen kann uns die generelle Forderung nach rationalen Algorithmen weiterhelfen! Der Schlüssel zum Erfolg liegt hier in der Bildung von Ableitungen, da bei jeder Ableitung ein Exponent um 1 verkleinert wird! Die Ableitung d' , berechnet mit der Produktregel nach der faktoriellen Form von d gemäß Gleichung (2), ist

$$d' = \sum_{i=1}^s e_i (x - \alpha_i)^{e_i - 1} \prod_{j \neq i} (x - \alpha_j)^{e_j}. \quad (3)$$

⁴D.h. ein Erweiterungskörper von \mathbb{Q} deren Elemente alle einer polynomialen Gleichung mit Koeffizienten aus \mathbb{Q} genügen. Ein Zerfällungskörper ist ein Spezialfall einer algebraischen Erweiterung.

Damit folgt

$$\text{ggT}(d, d') = \prod_{i=1}^s (x - \alpha_i)^{e_i - 1} = \prod_{i=2}^m d_i^{i-1}$$

und insbesondere sieht man mit derselben Überlegung, daß $\text{ggT}(d_i, d'_i) = 1$ gilt. Das Produkt aller quadratfreien Teiler d_i kann somit nur durch die *rationalen* Operationen des Bildens der Ableitung und des größten gemeinsamen Teilers sowie Polynomdivision mittels

$$\frac{d}{\text{ggT}(d, d')} = d_1 d_2 \dots d_n \quad (4)$$

bestimmt werden. Mit $\text{ggT}(d, d')$ anstelle von d kann das Verfahren iteriert werden. Im nächsten Schritt erhalten wir $d_2 d_3 \dots d_n$ und damit d_1 , dann d_2 u.s.w. Zusätzlich hat sich durch diese Konstruktion ergeben, daß alle Polynome d_i in $\mathbb{Q}[x]$ liegen und keine algebraische Erweiterung für die Koeffizienten notwendig ist! Insgesamt können wir also den folgenden Satz festhalten.

Satz 1 *Es gibt einen rationalen Algorithmus, der zu jedem Polynom die quadratfreie Faktorisierung liefert.*

Als Beispiel betrachten wir $d := x^8 + 6x^6 + 12x^4 + 8x^2$. Die quadratfreie Zerlegung ist $x^2(x^2 + 2)^3$, die man Schritt für Schritt so erhält: Im ersten Schritt ist der größte gemeinsam Teiler von d mit seiner Ableitung zu bilden: $x^5 + 4x^3 + 4x$. Damit ist das Produkt aller quadratfreien Faktoren gleich $\frac{x^8 + 6x^6 + 12x^4 + 8x^2}{x^5 + 4x^3 + 4x} = x^3 + 2x$. Mit dem größten gemeinsamen Teiler wird nun der nächste Schritt durchgeführt und der der größte gemeinsam Teiler mit seiner Ableitung $5x^4 + 12x^2 + 4$ bestimmt, hier $x^3 + x$. Die Division von $x^5 + 4x^3 + 4x$ durch dieses Polynom ergibt $d_2 d_3 = x^3 + 2x$ und damit die Erkenntnis, daß es hier keine einfachen Nullstellen gibt. Im letzten Schritt ist $\text{ggT}(x^3 + x), 3x^2 = x$ und damit $d_3 = \frac{x^3 + x}{x} = x^2 + 1$. Genau genommen, wissen wir an dieser Stelle nur, daß $d_3 d_4 \dots$ gleich $x^2 + 1$ ist. Aus Gradgründen oder aber durch einen weiteren Schritt des Algorithmus wird aber sofort klar, daß $d_3 = x^2 + 1$ ist. Insgesamt haben wir nun die quadratfreie Zerlegung vollständig bestimmt.

3.5 Hermite'sche Reduktion

Nachdem wir den Nenner quadratfrei faktorisiert haben, sind wir nun in der Lage die Partialbruchzerlegung auf diese Faktorisierung anzuwenden und erhalten Polynome a_i mit $\deg a_i < \deg d_i$ und

$$\int \frac{z}{d} dx = \int \frac{z}{\prod_{i=1}^m d_i^i} dx = \sum_{i=1}^m \int \frac{a_i}{d_i^i} dx. \quad (5)$$

Die Idee von E. HERMITE – siehe [31] oder [10] – ist es nun, Summanden, die direkt integriert werden können, abzuspalten und dadurch die Exponenten i sukzessive zu verkleinern, bis schließlich nur die Exponenten 1, d.h. die Beiträge für den logarithmischen Anteil, übrigbleiben. Auch hier wird wesentlich wieder der euklidische Algorithmus benutzt, so daß auch dieser Teilalgorithmus wieder rational ist. Mit der erweiterten Variante davon kann man für $i > 1$ Polynome b und c bestimmen mit

$$-\frac{a_i}{i-1} = b d'_i + c d_i, \quad (6)$$

da ja wegen der Quadratfreiheit von d_i die Beziehung $\text{ggT}(g_i, g'_i) = 1$ gilt. Das ist nun in der Tat sehr trickreich, da jetzt die folgenden Transformationen für die Polynome unter dem Integral möglich sind. Man beachte, daß bei der zweiten Gleichung der Term

$\frac{b'}{d_i^{i-1}}$ addiert und subtrahiert wird, damit bei der dritten Gleichung die Ableitungsregel $\left(\frac{b}{d_i^{i-1}}\right)' = \left(\frac{b'}{d_i^{i-1}} - \frac{(i-1)bd_i'}{d_i^i}\right)$ angewandt werden kann:

$$\begin{aligned} \frac{a_i}{d_i^i} & \stackrel{(8)}{=} - (i-1) \frac{bd_i' + cd_i}{d_i^i} \\ & = \left(\frac{b'}{d_i^{i-1}} - \frac{(i-1)bd_i'}{d_i^i}\right) + \frac{(1-i)c-b'}{d_i^{i-1}} \\ & = \left(\frac{b}{d_i^{i-1}}\right)' + \frac{(1-i)c-b'}{d_i^{i-1}} \end{aligned}$$

Der erste Summand der rechten Seite der letzten Gleichung hat einen Beitrag zum rationalen Anteil der Integrals, während das Integrationsproblem für den zweiten Summanden durch die Reduktion des Exponentes nach endlich vielen Iterationen dieser Idee beim Exponenten 1 ankommt und damit das Integral von der Gestalt $\int \frac{q}{d_i} dx$ nur noch einen Beitrag zum logarithmischen Anteil liefern kann. Man beachte dabei, daß der Grad des Zählerpolynoms stets kleiner dem Grad des Nennerpolynoms bleibt.

Wir betrachten als Beispiel die Funktion $\frac{a}{d}$ mit $a := x^7 - 24x^4 - 4x^2 + 8x - 8$ und dem Nenner $d := x^8 + 6x^6 + 12x^4 + 8x^2$, den wir schon im letzten Abschnitt in seine quadratfreien Faktoren x und $x^2 + 2$ zerlegt hatten. Die Partialbruchzerlegung liefert dann das Ergebnis $\frac{x-1}{x^2} + \frac{x^4-6x^3-18x^2-12x+8}{(x^2+2)^3}$. Im ersten Schritt der Reduktion gibt das $\frac{x-1}{x^2} = \left(\frac{1}{x}\right)' + \frac{1}{x}$ für den ersten Summanden. Für den quadratfreien Faktor $v := x^2 + 2$ mit Exponenten $i = 3$ und seine Ableitung $v' = 2x$ liefert der erweiterte euklidische Algorithmus, daß $-\frac{1}{2}(x^4 - 6x^3 - 18x^2 - 12x + 8) = (6x)Dv - (\frac{1}{2}x^2 + 3x - 2)v$ gilt. Damit ergibt sich mit dem bereits bestimmten rationalen Anteil $\frac{1}{x}$ nun durch Addition von $\frac{6x}{(x^2+2)^2}$ der neue rationale Anteil $\frac{x^4+10x^2+4}{x^5+4x^3+4x}$. Als neuen Zähler für den noch zu behandelnden, reduzierten Nenner v^2 erhalten wir $(1-3)(-\frac{1}{2}x^2 + 3x - 2) - 6 = x^2 - 6x - 2$. EUKLID liefert jetzt $-x^2 - 6x - 2 = (-x+3)Dv + v$. Damit wächst der rationale Anteil um $\frac{-x+3}{x^2+2}$ auf $\frac{3x^3+8x^2+6x+4}{x^5+4x^3+4x}$. Da der Zähler des jetzt noch übrigen Terms mit quadratfreiem Nenner v verschwindet, trägt dieser quadratfreie Faktor nichts zum logarithmischen Anteil bei. Also ist $\frac{a}{d}$ als Summe eine Ableitung einer rationalen Funktion $\frac{3x^3+8x^2+6x+4}{x^5+4x^3+4x}$ und einer rationalen Funktion $\frac{1}{x}$ mit quadratfreiem Nenner dargestellt. Wir haben also das Integral vollständig bestimmt:

$$\int \frac{x^7 - 24x^4 - 4x^2 + 8x - 8}{x^8 + 6x^6 + 12x^4 + 8x^2} dx = \frac{3x^3 + 8x^2 + 6x + 4}{x^5 + 4x^3 + 4x} + \log(x).$$

3.6 Varianten ohne Partialbruchzerlegung von Mack und von Ostrogradsky-Horowitz

Im letzten Abschnitt hatten wir ausgenutzt, daß nach Partialbruchzerlegung im Nenner nur noch d_i^i stand und daß $\text{ggT}(d_i, d_i') = 1$ war. Wir können die Partialbruchzerlegung vorab gänzlich vermeiden, wenn wir ausnutzen, daß es gar nicht wesentlich ist, daß im Nenner nur eine Potenz eines quadratfreien Polynoms stehen darf. Es genügt eine Situation zu betrachten bei der d_i und ein weiteres Polynom teilerfremd sind. Wenn wir im Falle eines festen $i \geq 2$ und $d_i \neq 1$ den Nenner

$$d = \prod_{j=1}^m d_j^j = uv^i \tag{7}$$

mit $v := d_i$ und dann natürlich $u := \frac{d}{v^i} = \prod_{j \neq i} d_j^j$ schreiben, so gilt ebenfalls $1 = \text{ggT}(uv', v)$. Mit EUKLID kann man wieder Polynome b und c mit $\text{deg}(b) < \text{deg}(v)$ konstruieren, so daß

$$-\frac{a}{i-1} = buv' + cv \tag{8}$$

gilt und wir mit derselben Rechnung nun

$$\frac{a}{uv^i} = \left(\frac{b}{v^{i-1}} \right)' + \frac{(1-i)c - ub'}{uv^{i-1}} \quad (9)$$

erhalten und dann das Spiel wieder neu mit kleineren Exponenten fortsetzen können.

In beiden Varianten dieses Algorithmus ist die Zahl der Operationen im schlimmsten Fall quadratisch in m . D. MACK in [46] geht noch einen Schritt weiter und erledigt in einer Iteration gleich die Reduktion der Exponenten größer 1 der quadratfreien Faktoren um 1! Das klappt, da im Falle von $m \geq 2$ die Polynome (!) $\frac{d_{(1)}d_{(2)}'}{d^{(2)}}$ und $d_{(2)}$ teilerfremd sind, wobei wir

$$d_{(j)} := d_j d_{j+1} \dots d_m \text{ und } d^{(j)} := \prod_{i=j}^m d_i^{i-j+1}$$

gesetzt haben, also insbesondere ist natürlich $d = d^{(1)}$, $d_{(j)}d^{(j+1)} = d^{(j)}$, $\text{ggT}(d, d') = d^{(2)}$ und $d^{(2)'} = (\sum_{i=2}^m (i-1) \prod_{j \neq i} d_j) d^{(3)}$. Wieder können wir mit dem euklidischen Algorithmus Polynome b und c konstruieren, die

$$a = b \left(-\frac{d_{(1)}d^{(2)'}}{d^{(2)}} \right) + cd_{(2)} \quad (10)$$

erfüllen. Wir erhalten

$$\frac{a}{d} = \left(\frac{b}{d^{(2)}} \right)' + \frac{c - d_1 b'}{d_1 d^{(2)}}$$

mit der HERMITESchen Umformung und wegen $d = d^{(1)} = d_{(1)}d^{(2)} = d_1 d_{(2)} d^{(2)}$. Man beachte, daß nun der Nenner $(d_1 d_2) d_3^2 \dots d_m^{m-1}$ des zweiten Summanden höchstens $m-1$ Potenzen von quadratfreien Faktoren enthält und dieses Verfahren ebenso iteriert werden kann.

Zu guter Letzt sei auch das Verfahren von M.W. OSTROGRADSKY [49] – von E. HOROWITZ [32] unabhängig davon 125 Jahre später wiederentdeckt – kurz erwähnt. Es ersetzt die Lösung der diophantischen⁵ Gleichung ähnlich wie (10) in b und c durch das Lösen eines linearen Gleichungssystems mit $\text{deg } d^{(2)} + \text{deg } d_{(1)} = \text{deg } d$ Gleichungen und ebensovielen Unbestimmten, den Koeffizienten von b und c . In der Tat sieht man aus den Überlegungen bei der Version von MACK, daß der Nenner des rationalen Anteils immer das Polynom $d^{(2)}$ teilt, der Nenner des restlichen Anteils mit quadratfreiem Nenner teilt natürlich das Polynom $d_{(1)}$. Folglich gibt es Polynome b und c gibt, deren Grade wegen $\text{deg } a < \text{deg } d$ o.E. den Bedingungen $\text{deg } b < \text{deg } d^{(2)}$ und $\text{deg } c < \text{deg } d_{(1)}$ unterliegen und für die

$$\frac{a}{d} = \left(\frac{b}{d^{(2)}} \right)' + \frac{c}{d_{(1)}} = \frac{b'}{d^{(2)}} - \frac{b(d^{(2)})'}{(d^{(2)})^2} + \frac{c}{d_{(1)}}$$

gilt, woraus

$$0 = a - bd_{(1)} + b \frac{d^{(2)'} d_{(1)}}{d^{(2)}} - cd^{(2)}.$$

folgt. Das Polynom auf der rechten Seite hat maximal den Grad $\text{deg}(d) - 1$. Die $\text{deg}(d)$ Koeffizienten der rechten Seite müssen also alle gleich 0 sein und das ergibt das Gleichungssystem. Das Verfahren ist sehr effizient für rationale Funktionen, läßt sich aber nicht so gut wie die HERMITESchen Verfahren verallgemeinern.

⁵D.h. es werden Lösungen in $\mathbb{Q}[x]$, nicht in $\mathbb{Q}(x)$ gesucht.

3.7 Der Rothstein–Trager–Algorithmus für den logarithmischen Anteil

Mit Division mit Rest und der Methode von E. HERMITE oder den Varianten davon ist es möglich, vollständig den rationalen Anteil des Integrals zu bestimmen. Damit bleibt nur noch das Problem der Integration des Spezialfalls

$$\int \frac{a}{d} dx = \int \sum_{i=1}^n \frac{c_i}{x - \alpha_i} dx = \sum_{i=1}^n c_i \log(x - \alpha_i), \quad (11)$$

mit quadratfreiem Nenner d und Konstanten c_i und α_i in einer algebraischen Erweiterung von \mathbb{Q} zu bewerkstelligen. Für die sogenannten *Residuen* c_i ⁶ haben B.M. TRAGER [66] und M. ROTHSTEIN [59] unabhängig die Formel

$$\text{res}_x(d, a - zd') = \prod_{i=1}^n (z - c_i)^{\epsilon_i} \in \mathbb{Q}[z] \quad (12)$$

für ein n und mit einer neuen Unbestimmten z gefunden. Der fundamentalen Bedeutung der Resultante res_x angemessen, haben wir ihr den nächsten Abschnitt 3.8 zur Erinnerung gewidmet. Was nun auf den ersten Blick wie ein Druckfehler aussieht – man erwartet ja zunächst $\mathbb{Q}(c_1, \dots, c_n)[z]$ als Polynomring für das Produkt –, ist in Wirklichkeit die schöne Erkenntnis, daß mit Hilfe der *rationalen* Konstruktion der Resultante (siehe Abschnitt 3.8) der transzendente Anteil des Integrals ebenso mit einem rein rationalen Algorithmus bestimmt werden kann! Kombiniert mit einer quadratfreien Zerlegung erhalten wir

$$\prod_{i=1}^n (z - c_i) \in \mathbb{Q}[z].$$

Mehr noch: die algebraische Erweiterung $\mathbb{Q}(c_1, \dots, c_n)$ von \mathbb{Q} ist die zur Darstellung des Integrals minimal notwendige Erweiterung!

Wir gehen zum Beweis dieses fundamentalen Resultats von einer beliebigen, also bzgl. der Körpererweiterung nicht notwendig minimalen Darstellung $\int \frac{a}{d} dx = \sum_{i=1}^n c_i \log v_i$ bzw. dazu äquivalent $a \prod_{i=1}^n v_i = (\sum_{i=1}^n c_i v_i' \prod_{j \neq i} v_j) d$ aus. Dabei sind die c_i s paarweise verschiedene algebraische Zahlen über \mathbb{Q} , die nicht notwendig linearen, aber nicht konstanten, quadratfreien Polynome v_i besitzen als Koeffizienten ebenso algebraische Zahlen über \mathbb{Q} , sie sind paarweise teilerfremd und ihre Leitkoeffizienten sind 1. Wir setzen $u_i := \prod_{j \neq i} v_j$ und erhalten mit Teilbarkeitsargumenten und der Teilerfremdheit von $\text{ggT}(a, d) = 1$ und $\text{ggT}(v_j, v_j' u_j) = 1$, daß sich der Nenner d und der Zähler a

$$d = \prod_{i=1}^n v_i \text{ und } a = \sum_{i=1}^n c_i v_i' u_i$$

durch v_i und u_i ausdrücken lassen.

Als nächstes wird gezeigt, daß

$$v_j = \text{ggT}(a - c_j d', d)$$

für alle j gilt. Auch das kann wieder mit Teilbarkeitsargumenten erreicht werden. Man zeigt $v_j | (a - c_j d')$, damit $v_j | \text{ggT}(a - c_j d', d)$ und schließlich $\text{ggT}(a - c_j d', v_h) = 1$ für alle $h \neq j$. Damit haben die Polynome $a - c_j d'$ und d gemeinsame Nullstellen, das

⁶Vgl. dazu Funktionentheorie und Laurentreihen: Koeffizient von $(x - \alpha_i)^{-1}$ der Entwicklung von $\frac{a}{d}$ um den Punkt α_i .

heißt es gilt $\text{res}(a - c_j d', d) = 0$ und insbesondere ist c_j eine Nullstelle des Polynoms $\text{res}_x(a - td', d)$.

Nun ist noch zu zeigen, daß jede Nullstelle c dieses Polynoms im Integral vorkommen muß. Sei also $\text{res}_x(a - cd', d) = 0$, dann haben d und $a - cd'$ eine gemeinsame Nullstelle, insbesondere ist $g := \text{ggT}(a - cd', d)$ nicht konstant. Jeder irreduzible Faktor p von g teilt genau ein v_j und damit auch $v'_i u_i$ für alle $i \neq j$, nicht aber $v'_j u_j$. Aus der Tatsache, daß $p \mid a - cd' = \sum_{i=1}^n (c_i - c) v'_i u_i$ teilt, folgt damit $p \mid (c_j - c) v'_j u_j$ und damit bleibt der Konstanten $c_j - c$ nichts anderes übrig als 0 zu sein.

Nun ist noch die Minimalität zu zeigen: Ist L eine minimale algebraische Erweiterung der rationalen Zahlen, die zur Darstellung des Integrals ausreicht, dann ist L Zerfällungskörper des Polynoms $\text{res}_x(a - zd', d) \in \mathbb{Q}[z]$. Man geht dabei von einer Darstellung des Integrals über L aus. Falls ein Polynom v als Argument des Logarithmus nicht quadratfrei ist, wird es faktorisiert und in eine Summe von Logarithmen verwandelt. Analog erreicht man die Teilerfremdheit. Konstante werden eliminiert, damit können die Polynome normiert werden. Dann werden alle Logarithmen mit gleichem Argument zusammengefaßt. Gilt schließlich noch, daß zwei Koeffizienten von Logarithmen gleich sind, so wird zusammengefaßt: $c \log v + c \log v' = c \log(vv')$, was an der Teilerfremdheit und der Quadratfreiheit nichts ändert. Damit sind die Voraussetzungen von oben hergestellt und die Behauptung folgt.

3.8 Erinnerung an die Resultante

Zur Illustration des oben Dargelegten betrachten wir das Beispiel

$$f := \frac{a}{d} := \frac{x^4 - 3x^2 + 6}{x^6 - 5x^4 + 5x^2 + 4}$$

aus [10]. Der Nenner ist quadratfrei und irreduzibel. Wir haben also die Resultante von $x^6 - 5x^4 + 5x^2 + 4$ und $a - zd' = (-6z)x^5 + x^4 + (20z)x^3 + (-3)x^2 + (-10z)x + 6$ bzgl. x zu bestimmen. Gemäß Definition der sylvesterschen Resultante zweier Polynome vom Grad m und n ist dies die Determinante der folgenden quadratischen $(m+n) \times (m+n)$ -Matrix, die n -mal die Koeffizienten des zweiten Polynoms und m -mal diejenigen des ersten Polynoms enthält, und zwar jeweils um ein Spalte nach rechts verschoben.

$$\det \begin{pmatrix} 1 & 0 & -5 & 0 & 5 & 0 & 4 & & & & \\ & 1 & 0 & -5 & 0 & 5 & 0 & 4 & & & \\ & & \ddots & & \\ -6z & 1 & 20z & -3 & -10z & 6 & & & & & \\ & -6z & 1 & 20z & -3 & -10z & 6 & & & & \\ & & \ddots & & \\ & & & & -6z & 1 & 20z & -3 & -10z & 6 & \end{pmatrix} = 45796(4z^2 + 1)^3. \quad (13)$$

Wir erhalten also das Ergebnis, daß die Residuen genau die Nullstellen von $4z^2 + 1$ sind, hier also $\frac{i}{2}$ und $-\frac{i}{2}$.

War die Resultante von SYLVESTER im 19. Jahrhundert eingeführt worden, um nicht-lineare Gleichungssysteme zu lösen – zwei Polynome besitzen genau dann gemeinsame Nullstellen in einem Erweiterungskörper, wenn Ihre Resultante 0 ist, – so spielt sie nun auch eine zentrale Rolle für die Integrationstheorie. Der bekannteste Spezialfall beim Lösen von Gleichungen ist die Diskriminante $\text{disk}(f) = \text{res}(f, f')$ beispielsweise bei quadratischen Gleichungen

$$\text{res}(ax^2 + bx - c, 2ax + b) = b^2 - 4ac.$$

Bei algebraischen Funktionen y mit $F(x, y) = 0$ bestimmen die Nullstellen der Diskriminante $\text{disk}_y(F)$ singuläre Punkte. Bei der Bestimmung einer Ganzheitsbasis im

Zusammenhang mit der Integration algebraischer Funktionen im Abschnitt 6.2 auf Seite 104 wird die Diskriminante eine herausragende Rolle spielen.

Die Lösungen zweier bivariater Gleichungen in x und y gelingt oft am schnellsten durch Berechnung ihrer Resultanten bzgl. x und y . Es sei an dieser Stelle auch erwähnt, daß die Berechnung der Resultante durch das *Subresultanten-Verfahren* von G.E. COLLINS und W.S. BROWN (siehe z.B. [24], Abschnitt 7.3.) wesentlich effizienter gestaltet werden kann, da dabei nur eine Folge von Polynomdivisionen nötig ist.

Nach diesem Ausflug aber zurück zur Integration von $\frac{x^4-3x^2+6}{x^6-5x^4+5x^2+4}$. Wir haben noch die Polynome unter den Logarithmen zu bestimmen:

$$\begin{aligned} \text{ggT}(d, a - cd') &= \text{ggT}(x^6 - 5x^4 + 5x^2 + 4, (-6c)x^5 + x^4 + (-20c)x^3 + \\ &\quad (-3)x^2 + (-10c)x + 6) \\ &= x^3 + 2cx^2 - 3x - 4c \end{aligned}$$

für beide Nullstellen c . Damit erhalten wir also das Ergebnis

$$\begin{aligned} \int \frac{x^4 - 3x^2 + 6}{x^6 - 5x^4 + 5x^2 + 4} dx &= \sum_{c|4c^2+1} c \log(x^3 + 2cx^2 - 3x - 4c) \\ &= \frac{i}{2} \log \left(\frac{x^3 + ix^2 - 3x - 2i}{x^3 - ix^2 - 3x + 2i} \right). \end{aligned} \quad (14)$$

3.9 Ein rationaler Algorithmus zur Bestimmung des Integrals einer rationalen Funktion

Der einzige Schönheitsfehler in Bezug auf Rationalität der Methoden ist noch die Bestimmung der Polynome unter den Logarithmen! Hier haben wir bislang der größten gemeinsamen Teiler im Polynomring $\mathbb{Q}(c)[x]$, im Beispiel mit $4c^2+1=0$, zu bestimmen gehabt. Auch hier helfen die Subresultanten von G.E. COLLINS und W.S. BROWN nach D. LAZARD, R. RIOBOO ([43]) und B.M. TRAGER weiter. Man kann zeigen, – siehe etwa [10] – daß das Polynom der Subresultanten-Folge vom Grad i gerade zum Residuum c_i gehört, wobei sich der Index i hier – anders als zuvor – auf das Polynom r_i bezieht, d.h. auf das Polynom, das alle i -fachen Nullstellen der Resultante aufammelt. Im Beispiel vorhin erhalten wir als Subresultantenfolge

$$\begin{aligned} &[2930944z^6 + 2198208z^4 + 549552z^2 + 45796, \\ &(-119840z^5 - 59920z^3 - 7490z)x - 23968z^4 - 11984z^2 - 1498, \\ &(-11200z^4 - 2604z^2 + 49)x^2 + 25600z^4 + 5952z^2 - 112, \\ &(800z^3 - 14z)x^3 + (-400z^2 + 7)x^2 + (-2440z^3 + 32z)x + 792z^2 - 16, \\ &(-60z^2 + 1)x^4 + 2zx^3 + (120z^2 - 3)x^2 + 26zx + 144z^2 + 6, \\ &-6zx^5 + x^4 + 20zx^3 - 3x^2 - 10zx + 6, \\ &x^6 - 5x^4 + 5x^2 + 4]. \end{aligned}$$

Die quadratfreie Zerlegung des ersten Eintrags der Liste – das ist die Resultante! – ist $45796(4z^2 + 1)^3$, das heißt nur $r_3 = 4z^2 + 1$ ist von 1 verschieden. Wir haben also den Teiler vom Grad 3 aus der Subresultantenfolge auszuwählen und die Nullstellen von $4z^2 + 1$ einzusetzen, bzw. damit zu reduzieren. Das Ergebnis ist $-214zx^3 + 107x^2 + 642zx - 214 = -214z(x^3 - \frac{1}{2}x^2 - 3x - \frac{1}{2}) = -214z(x^3 + 2zx^2 - 3x - 4z)$ und damit sind auch die Polynome unter den Logarithmen in Gleichung (14) mit rein rationalen Algorithmen bestimmt.

Insgesamt können wir die dargestellten Überlegungen in folgendem Satz zusammenfassen.

Satz 2 Der folgende Algorithmus zur Bestimmung des Integrals einer rationalen Funktion $f = \frac{z}{d} \in \mathbb{Q}[x]$ mit $\text{ggT}(z, d) = 1$ ist rational.

1. Mit Division von z durch d mit Quotient p und Rest a wird das Problem zunächst auf die Integration des Polynoms p sowie der Integration von $\frac{a}{d}$ mit $\deg a < \deg d$ reduziert.
2. Mit quadratfreier Faktorisierung des Nennerpolynoms d und Partialbruchzerlegung nach dieser Faktorisierung wird das Problem auf die Integration von rationalen Funktionen der Form $\frac{a}{d}$ mit quadratfreiem d zurückgeführt.
3. Mit der Methode von Hermite wird die Integration von $\frac{a}{d}$ mit quadratfreiem d auf das Problem der Integration von $\frac{a}{d}$ mit quadratfreiem d reduziert.
4. Die Integration von $\frac{a}{d}$ mit quadratfreiem d geschieht durch Berechnung der Resultante $\text{res}(d, a - zd')$ nach ROTHSTEIN und TRAGER und anschließender quadratfreier Faktorisierung. Die Nullstellen des Ergebnispolynomes sind die Koeffizienten der Logarithmen. Durch Benutzung eines Subresultanten-Algorithmus erhält man die Polynome unter der Logarithmen.

3.10 Reelle Darstellungen

Wenden wir diese Methoden auf die rationale Funktion $\frac{1}{x^2+1}$ an: Der Nenner ist quadratfrei und irreduzibel, die Resultante ist $\text{res}_x(x^2+1, 1-2zx) = 4z^2+1$, der größte gemeinsame Teiler in $\mathbb{Q}(c)[x]$ von x^2+1 und $-2cx+1$ ist $x+2c$, man beachte daß $0 = 4c^2+1$ gilt. Man erhält also $\int \frac{1}{x^2+1} dx = \frac{i}{2} \log(x-i) + \frac{-i}{2} \log(x+i)$ – und damit gerade nicht, was man erwartet, nämlich $\arctan(x^2+1)$. Es entsteht hier also das Problem, falls möglich, reelle Darstellungen eines Integrals zu finden.

Noch extremer wirkt sich das aus, wenn man an definiten Integralen interessiert ist. Hier wäre es fatal, wenn man einfach fehlerhaft, *obere Grenze eingesetzt minus untere Grenze eingesetzt*, rechnen würde, obgleich manch ein Computeralgebra-System das tut. M. BRONSTEIN macht das am Beispiel (14) aus Abschnitt 3.8 deutlich. Diese Funktion ist im Intervall $[1, 2]$ stetig und positiv, während $\frac{i}{2} \log(8+4i-6-2i) - \frac{i}{2} \log(8-4i-6+2i) - \frac{i}{2} \log(1+i-3-2i) + \frac{i}{2} \log(1-i-3+2i) = -\frac{5\pi}{4} + \arctan(\frac{1}{2}) \sim -3,46$. Komplexe Nullstellen, Verzweigungen des komplexen Logarithmus und nicht stetige Stammfunktionen sind hier die Probleme!

R. RIOBOO hat in [55] das folgende Methode angegeben um die auftretenden Logarithmen in Arkustangens, also in reelle Integrale umzuwandeln. Für $0 \neq u, v \in k[x]$ mit $u^2 + v^2 \neq -1$, seien ferner s und t Polynome mit $-tu + sv = \text{ggT}(u, v)$ gemäß Euklid. Dann gilt $i \frac{d}{dx} \log \left(\frac{u+vi}{u-vi} \right) = 2 \frac{d}{dx} \arctan \left(\frac{su+tv}{\text{ggT}(u,v)} \right) + i \frac{d}{dx} \log \left(\frac{s+it}{s-it} \right)$, was man durch Nachrechnen einsehen kann. Da es auf die Konstanten nicht ankommt, kann man aus dieser Gleichung ein iteratives Verfahren ableiten. In reellen Körpern gelingt es ferner die Summe der Logarithmen nach ROTHSTEIN-TRAGER immer durch Zusammenfassen der Real- bzw. Imaginärteile der komplex-konjugierten Nullstellen des quadratfreien Anteils der Resultante in Logarithmen der hier benötigten Form umzuwandeln.

In unserem Beispiel von oben ist $\int \frac{x^4-3x^2+6}{x^5-5x^4+5x^2+4} dx = \frac{i}{2} \log \left(\frac{(x^3-3x)+(x^2-2)i}{(x^3-3x)-(x^2-2)i} \right)$. Wir erhalten im ersten Schritt für $u := x^3-3x$ und $v := x^2-2$ die Koeffizienten $s = \frac{1}{2}x^2 - \frac{1}{2}$ und $t = \frac{1}{2}x$ und damit $\arctan \left(\frac{x^5-3x^3+x}{2} \right) + \frac{i}{2} \log \left(\frac{(x^2-x)+xi}{(x^2-x)-xi} \right)$. Im zweiten Schritt ergibt sich für $u := x^2-x$ und $v := x \arctan(x^3) + \frac{i}{2} \log \left(\frac{x+i}{x-i} \right)$, $s = x$ und $t = 1$ ist. Schließlich ist $\frac{i}{2} \frac{dx}{x} \log \left(\frac{x+i}{x-i} \right) = \frac{dx}{x} \arctan x$, so daß insgesamt die reelle und stetige Darstellung $\arctan \left(\frac{x^5-3x^3+x}{2} \right) + \arctan(x^3) + \arctan x$ des Integrals bestimmt wurde.

Natürliche ist nun auch gewährleistet, daß nun durch jetzt *korrekte* Anwendung des Hauptsatzes der Differential- und Integralrechnung das richtige Ergebnis der definiten Integration erzielt wird.

4 Differentialalgebra und elementare Erweiterungen

4.1 Kann der Algorithmus für die Integration rationaler Funktionen auf allgemeinere Situationen übertragen werden?

Wie wir gesehen haben ist das Problem der Integration rationaler Funktionen im Hinblick auf die Forderungen der Computeralgebra nach rationalen Entscheidungsverfahren vollständig gelöst. In diesem Abschnitt werden wir einige Ideen und Voraussetzungen zur algorithmischen Integration elementarer Funktionen, die nun auch exakt definiert werden, entwickeln. Wir werden die Schwierigkeiten und die dadurch notwendigen Modifikationen diskutieren, die entstehen, wenn man versucht, diese Algorithmen etwa auf die Situation der rationalen (!) Funktion

$$f := \frac{x - \tan x}{\tan^2 x} = \frac{x - t}{t^2} \in \mathbb{Q}(x)(t),$$

wobei $t := \tan x$ gesetzt wird, anzuwenden. Es liegt hier also eine rationale Funktion in t über $\mathbb{Q}(x)$ vor! Im Unterschied zur Situation einer rationalen Funktion in x mit Integration nach x haben wir hier keine Ableitung nach der Unbestimmten t der rationalen Funktion in t zu betrachten, sondern nach wie vor die Ableitung nach x , was sich für t wie folgt auswirkt:

$$t' = \frac{d}{dx}t = 1 + t^2,$$

während in der klassischen Situation

$$x' = \frac{d}{dx}x = 1$$

vorliegt. Diese Operation $'$ verhält sich nun in Bezug auf Funktionen in t wie eine Ableitung.

4.2 Differentialalgebra

J.F. RITT und E.R. KOLCHIN haben in den dreißiger und vierziger Jahren die Theorie der Differentialalgebren – und entsprechend Differentialringe und Differentialkörper – entwickelt und den Begriff der Ableitung zu dem einer formalen Derivation verallgemeinert, siehe [56] und [40]. Neben der Addition und Multiplikation von Elementen der Algebra R (z.B. Funktionsalgebren) gibt es eine Operation D mit den Eigenschaften

$$\begin{aligned} D(\alpha f + \beta g) &= \alpha Df + \beta Dg \\ D(f \cdot g) &= Df \cdot g + f \cdot Dg \end{aligned}$$

für alle Elemente f und g der Algebra und Koeffizienten α und β aus einem Koeffizientenkörper k über dem die Algebra definiert ist. Wir schreiben $\int f = g$, falls $Dg = f$ ist. Im Beispiel von oben haben wir die Differentialalgebra $\mathbb{Q}(x)(t)$ aller rationalen Funktionen in t mit Koeffizienten im Körper $\mathbb{Q}(x)$ und Derivation $D(f) = f' := \frac{d}{dx}f$. Die Elemente, deren Ableitung 0 ergibt, werden *Konstanten* der Differentialalgebra genannt und mit $C_D(R)$ bezeichnet. Im Falle von $\mathbb{Q}(x)$ und $D = \frac{d}{dx}$ sind das natürlich gerade die Elemente aus \mathbb{Q} . Eine besondere Rolle spielen die Konstanten bei Differentialkörpererweiterungen, siehe dazu Abschnitt 4.5.

4.3 Erweiterung durch Logarithmen, Exponentialfunktionen und algebraischen Funktionen

Die algebraischen und transzendenten Erweiterungen eines Differentialkörpers k zur Konstruktion von elementaren Funktionenklassen sind auf natürliche Weise Differentialalgebren. Exponentialfunktionen und Logarithmusfunktionen können jetzt abstrakt durch die gültigen Ableitungsregeln – jetzt zum Axiom erhoben – durch Körpererweiterungen modelliert werden. Ein Element t heißt *Exponentialfunktion*, wenn ein $b \in k$ existiert, so daß die Gleichung

$$Dt = Db \cdot t$$

gilt. In diesem Fall schreibt man auch $t = e^b = \exp b$. Ein Element t heißt *Logarithmusfunktion*, wenn ein $b \in k$ existiert, so daß die Gleichung

$$Dt = \frac{Db}{b}$$

gilt. In diesem Fall schreibt man auch $t = \log b$. Analog kann man etwa wie oben den Tangens $\tan(b)$ als Lösung einer Gleichung $Dt = (1 + t^2)Db$ definieren, der Arcustangens $t = \arctan(b)$ genügt der Gleichung $Dt' = \frac{Db}{1+b^2}$, jeweils für ein $b \in k$.

Man wird sich zunächst nun auch klarmachen müssen, wie sich die Derivation in Bezug auf die uns interessierenden Erweiterungen verhält. Mit der Ableitungsregel

$$D\left(\frac{a}{b}\right) = \frac{Da b - a Db}{b^2}$$

läßt sich die Derivation D einer nullteilerfreien Differentialalgebra R eindeutig auf den Quotientenkörper⁷ fortsetzen.⁸ Der Polynomring $R[x]$ wird beispielsweise durch die Ableitung der Koeffizienten eines Polynomes

$$\kappa_D\left(\sum_{i=0}^n a_i x^i\right) := \sum_{i=0}^n Da_i x^i$$

zur Differentialalgebra. Das ist aber nicht die einzige Möglichkeit, im Gegenteil, es ist fast alles möglich: Zu einer transzendenten Erweiterung $R(x)$ und einem beliebig vorgegebenem Element $w \in R(x)$ ist $\kappa_D + w \frac{d}{dx}$ eine Fortsetzung der Derivation D mit $Dx = w$. Im Falle von algebraischen Erweiterungen muß man noch zusätzlich voraussetzen, daß ein Differentialkörper (k, D) und eine separable Erweiterung⁹ $k \leq K$ vorliegt, dann gibt es genau eine Fortsetzung von D auf K und für $a \in K$ mit Minimalpolynom p gilt die Formel

$$Da = -\frac{\kappa_D(p)(a)}{p'(a)}.$$

Ein einfaches aber instruktives Beispiel ist die Wurzelfunktion $a(x) = \sqrt{x}$ d.h. a erfüllt die Gleichung $y^2 - x = 0$, also ist $k := \mathbb{Q}(x)$, $D := \frac{d}{dx}$, $K := k(a)$, das Minimalpolynom

⁷Das ist die formale Konstruktion von Brüchen wie bei den rationalen Zahlen aus den ganzen. Ein wichtiges Beispiel sind die rationalen Funktionen als Quotientenkörper der Polynome. Noch allgemeiner genügt es im Nenner nur Elemente aus einer multiplikativ abgeschlossenen Teilmenge zuzulassen eine Situation, die uns im Abschnitt 4.9 noch begegnen wird.

⁸Wir sprechen von einer Fortsetzung der Derivation D von R auf S , wenn S eine Ringerweiterung (Algebren-, Körper-) ist mit einer Derivation, die auf R mit D übereinstimmt und dann sinnvollerweise auch gleich mit D bezeichnet wird.

⁹Das heißt, daß die Minimalpolynome aller Elemente nur einfache Nullstellen haben, was nur im Falle von Charakteristik ungleich 0 schiefe gehen kann.

von a ist $p := y^2 - x$ und damit gilt $\frac{d}{dx} a = -\frac{\kappa \frac{d}{dx} (y^2 - x)(a)}{p'(a)} = -\frac{-1}{2a} = \frac{1}{2a}$, da sich hier natürlich der Ableitungsstrich p' auf die Unbestimmte y bezieht, die Koeffizienten 1 und x von y^2 und 1 aber nach x abgeleitet werden müssen. Wir sehen also, daß diese Konstruktion mit der bekannten Regel $\sqrt{x}' = \frac{1}{2\sqrt{x}}$ übereinstimmt.

4.4 Warnung bei der Interpretation für reelle Funktionen

Da wir nun die notwendige algebraische Modellierung unserer Problemstellung vorgenommen haben, ist es Zeit nochmals eindringlich auf die Problematik der richtigen Interpretation hinzuweisen, wenn man etwa die dadurch gewonnenen Ergebnisse auf reelle Funktionen mit Definitions- und Wertebereichen anwenden will. Ein drastisches Beispiel hierfür ist die Funktion $f := \arctan x + \arctan \frac{1}{x}$. Es gilt nämlich

$$(\arctan x + \arctan \frac{1}{x})' = \frac{1}{1+x^2} + \frac{1}{1+(\frac{1}{x})^2}(-\frac{1}{x^2}) = 0.$$

Das heißt insbesondere, daß $\int 0 dx = \arctan x + \arctan \frac{1}{x}$ vollkommen im Sinne unserer Definition des Integrationsproblems in Ordnung ist! Das aus der Integrationstheorie der Analysis bekannte Gesetz, daß sich zwei Stammfunktionen nur durch eine Konstante unterscheiden, kann hier natürlich nicht mehr gelten.

4.5 Monome – in diesem Kontext transzendente Elemente, deren Ableitung ein Polynom ist und die die Konstanten nicht verändern

Transzendente Elemente t wie der oben betrachtete Arkustangens, dessen Ableitung Dt ein Polynom in $k[t]$ ist, werden in diesem Kontext *Monome* oder genauer *D-Monome* genannt. Zusätzlich wird gefordert, daß sich der Teilkörper der Konstanten nicht durch Hinzufügen eines Monoms t ändert, d.h. $C_D(k) = C_D(k(t))$. Sie besitzen dann natürlich im Grad der Ableitung einen D -Grad \deg_D und ähnliches. Jedes Integral $t := \int a \notin k$ eines Elements $a \in k$ ist ein Monom, da $Dt = a$ sogar ein konstantes Polynom ist. Dasselbe gilt für die Spezialfälle $t := \log b = \int \frac{Db}{b}$ und $t := \arctan b = \int \frac{Db}{b^2+1}$ für $b \in k$. Diese Monome, deren Ableitungen konstante Polynome, d.h. aus k sind heißen *primitive Monome*. Exponentialfunktionen von Integralen von Elementen a aus k , auch *Hyperexponentialfunktionen* genannt, sind Beispiele für Monome, deren Ableitungspolynome vom Grad 1 sind: Für $a \in k$ sei $t := e^{\int a}$ ist $Dt = at$ und für den Spezialfall $t := e^b = e^{\int Db}$ ist $Dt = Dbt$ für $b \in k$. Für den Tangens eines Integrals $t := \tan \int a$ eines Elementes $a \in K$, auch *Hypertangens* genannt, gilt $Dt = Da + Da^2$. Ein weiteres Beispiel mit D -Grad 2 ist $t := \frac{DJ_\nu}{J_\nu}$, da die Besselfunktion J_ν die Differentialgleichung $x^2 D^2 y + x Dy + (x^2 - \nu^2)y = 0$ erfüllt. Daraus folgt $Dt = \frac{y D^2 y - (Dy)^2}{y^2} = -t^2 - \frac{1}{x}t - \frac{x^2 - \nu^2}{x^2}$, d.h. t erfüllt eine Riccatigleichung.

Ein Beispiel für eine nicht-triviale Konstante ist $\log 2$, die bei Adjunktion von $t := \log(2x)$ an $\mathbb{Q}(x, \log(x))$ auftritt: $D(\log(2x) - \log(x)) = \frac{1}{x} - \frac{1}{x} = 0$. Das bedeutet, die Funktion $\log(2x)$ ist zwar ein Logarithmus über k , jedoch kein Monom, da sich die Konstanten ändern: $\mathbb{Q} = C_D(k) < \mathbb{Q}(\log 2) \leq C_D(k(t))$. Eine hinreichende Bedingung für eine primitive Erweiterung wie ein Logarithmus ein Monom zu sein, ist die Bedingung, daß $a := Dt \in k$ keine Ableitung eines Elementes in k ist. Eine hinreichende Bedingung für eine Erweiterung mit einer Exponentialfunktion t monomial zu sein, ist, daß sich die logarithmische Ableitung $a := \frac{Dt}{t} \in k$ nicht wie eine n -te-Wurzel eines Elementes $0 \neq v \in k$ verhält, d.h. für keine natürliche Zahl n und $0 \neq v \in k$ gilt $na := \frac{Dv}{v}$. Man beachte, daß $v = u^{\frac{1}{n}}$ so eine Gleichung erfüllt: $nu := \frac{Dv}{v}$.

4.6 Wann sind Exponentialfunktionen und Logarithmen transzendent?

Um nun verschiedene Exponentialfunktionen und Logarithmen in einem möglicherweise sehr komplexen Ausdruck in Griff zu bekommen, ist es notwendig Strukturaussagen treffen zu können, die es ermöglichen, Monome von Konstante zu unterscheiden und den Ausdruck in gewisser Weise in eine Normalform zu transformieren.

Die Ideen dazu gehen zurück bis auf J. LIOUVILLE ins letzte Jahrhundert, wurden aber erst von M. ROSENBLIETH und RISC in [53] im Sinne der Computeralgebra präzisiert:

Satz 3 Über einem Körper k von Konstanten gilt folgendes:

Eine Exponentialfunktion e^a ist genau dann transzendent über $k(t_1, \dots, t_{n-1})$, wobei die t_i algebraischen Funktionen, Logarithmen $\log(b_i)$ oder Exponentialfunktionen e^{b_i} sind, wenn ihr Argument a nicht in der Form $c + \sum_i r_i b_i$ mit $c \in k$ und rationalen Zahlen r_i geschrieben werden kann.

Eine Logarithmus $\log a$ ist genau dann transzendent über $k(t_1, \dots, t_{n-1})$, wobei die t_i algebraischen Funktionen, Logarithmen $\log(b_i)$ oder Exponentialfunktionen e^{b_i} sind, wenn keine Potenz a^m ihres Arguments a in der Form $c \prod_i b_i^{n_i}$ mit $c \in k$ und ganzen Zahlen $m \neq 0$ und n_i geschrieben werden kann.

Ein einfaches Beispiel ist $k(x, \log(x+1))$. Zunächst ist $\log(x-1)$ transzendent über diesem Körper, da es keine Konstante $c \in k$ und Exponenten $0 \neq m, n_2 \in \mathbb{Z}$ geben kann mit $(x-1)^m = (x+1)^{n_2}$. Andererseits ist aber z.B. $\log(\sqrt{x+1})$ nicht transzendent. Wegen $x^2 - 1 = (x+1)(x-1)$ ist $\log(x^2 - 1)$ nicht transzendent über $k(x, \log(x+1), \log(x-1))$, ebensowenig wie $\log(x-1)$ nicht transzendent über $k(x, \log(x+1), \log(x^2 - 1))$ ist, da $(x+1) = (x-1)^{-1}(x^2 - 1)$ ist.

4.7 Elementare Funktionen

Nun kann auch der Begriff *elementare Funktion* präzise definiert werden. Sei $k \leq K$ eine Differentialkörpererweiterung. Ein Element $t \in K$ heißt *elementar* über k , wenn t entweder algebraisch über k ist, oder aber ein Logarithmus oder eine Exponentialfunktion über k ist. Ist so ein elementares Element auch ein Monom, dann heißt es natürlich *elementares Monom*. Eine Differentialkörpererweiterung $k \leq E$ heißt *elementare Erweiterung*, wenn es Elemente $t_1, \dots, t_n \in E$ gibt mit $E = k(t_1, \dots, t_n)$ und für alle i ist t_i elementar über $k(t_1, \dots, t_i)$.

4.8 Verlust der Teilerfremdheit eines irreduziblen Nenners mit seiner Ableitung

Wir haben uns in den letzten Abschnitten klar gemacht, daß es nun gilt, die Methoden der Integration von rationalen Funktionen über \mathbb{Q} in x mit der Ableitung $'$ nach der Variablen x und der Regel $x' = 1$ beispielsweise auf die Situation rationaler Funktionen über $\mathbb{Q}(x)$ in t mit der Ableitung $D = \frac{d}{dx}$ und der Regel $Dt = t^2 + 1$ zu übertragen. Dazu betrachten wir zwei Beispiele, einmal $f := \frac{x-t}{t^2}$, zum anderen $g := \frac{1}{(t^2+1)^2}$ und analysieren die Schritte des Algorithmus 2. Division mit Rest wäre sicher kein Problem in $\mathbb{Q}(x)[t]$ und ist auch hier gar nicht nötig. Die Nenner sind in beiden Fällen bereits quadratfrei faktorisiert und wir versuchen nun mit der Methode von Hermite den Exponenten zu reduzieren. Hier hatten wir benutzt, daß d und Dd teilerfremd sind. Im Falle von f haben wir hier $\text{ggT}(t, Dt) = \text{ggT}(t, t^2 + 1) = 1$ und erhalten

$$-\frac{x-t}{2-1} = -(x-t) = \underbrace{-(x-t)(t^2+1)}_b + \underbrace{(xt-t^2)t}_c \quad (15)$$

und damit wegen $Dt = D(-(x-t)) = -1 + t^2 + 1 = t^2$

$$\begin{aligned} \frac{x-t}{t^2} &= D\left(\frac{-(x-t)}{t}\right) + \frac{-xt + t^2 - t^2}{t} \\ &= D\left(\frac{-(x-t)}{t}\right) - x = -D\left(\frac{x}{t}\right) - x, \end{aligned} \quad (16)$$

also

$$\int \frac{x - \tan(x)}{\tan^2(x)} dx = -\frac{x}{\tan(x)} - \frac{x^2}{2}. \quad (17)$$

Das ging also gut. Ein Glück war hier insbesondere auch, daß wir kein Integral eines Polynoms in t bestimmen mußten. Im rationalen Fall hatten wir damit keinerlei Problem. Was aber passiert bei $\int t^4 - 2t^2 + xt + 1 dx$?

Widmen wir uns nun dem zweiten Beispiel und bestimmen den größten gemeinsamen Teiler von $1+t^2$ und $D(1+t^2) = 2tDt = 2t(1+t^2)$, also $\text{ggT}(d, Dd) = 1+t^2 \neq 1$. Der Trick von Hermite nun b und c zu bestimmen um mit $-\frac{1}{2t-1} = bDd + cd$ den Exponenten 2 im Nenner zu reduzieren, scheitert hier schon! Es liegt hier mit dem Nennerpolynom $t^2 + 1$ eine *spezielle* Situation vor, während im Falle von t die *normale* Methode anwendbar ist. Da Mathematiker sehr kreativ im Finden von passenden Namen sind, nennt man Polynome p in t aus $\mathbb{Q}(x)[t]$ mit $\text{ggT}(p, Dp) = 1$ *normale* Polynome, während Polynome mit $\text{ggT}(p, Dp) = p$ – oder äquivalent dazu $p|Dp$ – *spezielle* Polynome heißen. Daß der Nenner von g mit $Dt = t^2 + 1$ etwas zu tun hat, ist übrigens kein Zufall.

Eine Antwort wie in der Situation des Integrals eines Polynoms in $t = \tan(x)$ oder gar einer rationalen Funktion, deren Nenner eine Potenz des speziellen Polynoms $(t^2 + 1)$ ist, zu verfahren ist, werden wir in den Abschnitten 5.4 auf Seite 99 und 5.7 auf Seite 102 geben.

4.9 Das Monoid der speziellen Polynome

Für ein Monom t über k bzgl. der Derivation D sei mit \mathcal{S} die Menge aller Polynome bezeichnet, die ihre eigene Ableitung teilen:

$$\mathcal{S} := \{p \in k[t] : p|Dp\}.$$

Wegen $pq|(Dpq + pDq) = D(pq)$ ist diese Menge multiplikativ abgeschlossen, also eine Halbgruppe, wegen $1 \in k \subseteq \mathcal{S}$ sogar ein Monoid. Sie wird von k und den irreduziblen Elementen $\mathcal{S}^{\text{irr}} := \{p \in \mathcal{S} : p \text{ irreduzibel und Leitkoeffizient } 1\}$ erzeugt. \mathcal{S} ist natürlich auch bzgl. Teilerbildung abgeschlossen. Bei primitiven Monomen wie bei $t := \int a$ gilt natürlich $\mathcal{S}^{\text{irr}} = \emptyset$, für $t := e^{\int a}$ ist $\mathcal{S}^{\text{irr}} = \{t\}$, während im schon betrachteten Beispiel $t := \tan \int a$ die Menge \mathcal{S}^{irr} je nachdem ob $i \in k$ ist ein oder zwei Elemente besitzt: $\mathcal{S}^{\text{irr}} = \{p : p|(t^2 + 1)\}$.

Ein irreduzibles Polynom ist nun entweder speziell oder normal, da natürlich für $\text{ggT}(p, Dp)$ nur 1 oder p möglich ist. Vorausschauend berechnen wir hier auch gleich noch diesen größten Teiler, wenn eine Faktorisierung $p = \prod_{i=1}^m p_i^{e_i}$ in paarweise teilerfremde p_i vorliegt. Das Ergebnis ist

$$\text{ggT}(p, Dp) = \left(\prod_{i=1}^m p_i^{e_i-1} \right) \left(\prod_{i=1}^m \text{ggT}(p_i, Dp_i) \right), \quad (18)$$

was man sehr schnell durch Nachrechnen einsehen kann: Seien a und b teilerfremd, dann gilt $\text{ggT}(ab, D(ab)) = \text{ggT}(a, D(ab))\text{ggT}(b, D(ab)) = \text{ggT}(a, Da b + aDb)\text{ggT}(b, Da b + aDb) = \text{ggT}(a, Da)\text{ggT}(b, Db)$.

Das Produkt von paarweise teilerfremden normalen Polynomen ist wieder normal. Für ein elementares Monom t mit Ableitung $h(t) := Dt \in k[t]$ kann man die speziellen und normalen Polynome gut charakterisieren. Es ist in dieser Situation ein quadratfreies Polynom p genau dann normal, wenn die Ableitungen $D\alpha$ ungleich $h(\alpha)$ sind für alle Wurzeln α von p in einem Zerfällungskörper. Ein von 0 verschiedenes Polynom p ist dagegen speziell, wenn die Ableitung $D\alpha$ gleich der Einsetzung $h(\alpha)$ ins Ableitungspolynom h ist. Sind die Grundkörperelemente alle konstant, d.h. $Da = 0$ für alle $a \in k$, dann ergibt sich als unmittelbare Folgerung, daß ein von 0 verschiedenes p genau dann speziell ist, wenn es h teilt, während ein quadratfreies p genau dann normal ist, wenn $\text{ggT}(p, h) = 1$ ist.

Die Eigenschaft eines Polynoms, speziell bzw. normal zu sein, bleibt erhalten, wenn man den Koeffizientenring algebraisch erweitert. Die neuen Konstanten hängen aber bei einer Differentialkörpererweiterung stark von den speziellen Polynomen ab, da nach Kürzen sowohl der Zähler als auch der Nenner eines konstanten Elementes $c \in C_D(k(t))$ jeweils ein spezielles Polynom sein muß.

4.10 Faktorisierung nach speziellen und normalen Anteilen

Wie wir im Beispiel in Abschnitt 4.1 sahen, läßt sich die Reduktion der Exponenten der quadratfreien Faktoren eines normalen Nennerpolynomen ganz genau so wie im Fall einer rationalen Funktion durchführen. Nun gilt es also noch einen beliebigen Nenner eines Integranden in seine speziellen und normalen Bestandteile zu zerlegen. Zunächst ist festzuhalten, daß für $p \in k[t]$ der größte gemeinsame Teiler von p und seiner Ableitung $\frac{d}{dt}p$ nach t den größten gemeinsamen Teiler von p und der Derivation Dp für jede beliebige Derivation D teilt. Betrachtet man nun die irreduziblen Faktoren von p getrennt nach normalen n_i und die speziellen s_j Polynomen und wendet man die Formel aus Gleichung (18) sowohl für D als auch für $\frac{d}{dt}$ an, so kürzt sich der erste Faktor sowieso weg, während wegen der Normalität $\prod_i \text{ggT}(n_i, Dn_i) = 1$ gilt, wegen der Irreduzibilität aber auch $\prod_i \text{ggT}(n_i, \frac{d}{dt}n_i) = 1$ und $\prod_j \text{ggT}(s_j, \frac{d}{dt}s_j) = 1$. Damit bleibt gerade $\prod_j \text{ggT}(s_j, Ds_j) = \prod_j s_j$, das Produkt aller quadratfreien irreduziblen Teilerpolynome, übrig! Für quadratfreie Nenner liefert das sofort die gewünschte Faktorisierung in den speziellen Anteil und in den normalen Anteil. Durch Iteration liefert das nun wieder einen rationalen Algorithmus, der jede von 0 verschiedene rationale Funktion f aus $k(t)$ auf eindeutige Weise in

$$f = f_p + f_s + f_n$$

zerlegt, wobei $f_p \in k[t]$ der *polynomiale* Anteil ist – man erhält ihn durch Division mit Rest wie immer –, $f_s = \frac{b}{d_s}$ mit Polynomen $b, d_s \in k[t]$, $\deg b < \deg d_s$ der *spezielle Anteil*, da d_s speziell mit Leitkoeffizient 1 ist, und $f_n = \frac{c}{d_n}$ mit Polynomen $c, d_n \in k[t]$, $\deg c < \deg d_n$ der *normale Anteil*, da d_n normal mit Leitkoeffizient 1 ist. Dies wird durch Partialbruchzerlegung nach dem speziellen und normalen Anteil erreicht.

4.11 Normale Nenner, Reduktion nach Hermite und einfache Funktionen

Auf f_n kann nun die Hermitsche Reduktion angewandt werden und man bekommt die Darstellung $f_n = Dq + g$ mit einem Polynom $q \in k[t]$ und einer rationalen Funktion $g \in k(t)$ mit quadratfreiem, normalen Nenner. So eine Funktion heißt *einfach*, diese Bezeichnung ist in Analogie zu rationalen Funktionen mit einfachen endlichen¹⁰ Polstellen, d.h. mit quadratfreiem Nenner, gewählt. Sie $f := \frac{x-t}{t^2}$ mit $k := \mathbb{Q}(x)$, $t := \tan(x)$, $Dx = 1$, $Dt = t^2 + 1$. Um die Potenz 2 des Nenners zu reduzieren, setzen

¹⁰D.h. die Polstellen sind aus k .

wir $v := t$ und $u := 1$. Der erweiterte Euklidische Algorithmus löst die Gleichung $bDv + ct = b(t^2 + 1) + ct = \frac{x-t}{1-2} = t - x$ durch $b := -x$ und $c := 1 + xt$. Also folgt $f = D \frac{b}{v^2-1} + \frac{(1-2)c-uDb}{uv^2-1} = D \frac{-x}{t} + \frac{-1-xt+1}{t} = -\frac{x}{t} - x$. Damit ist $\int \frac{x-t}{t^2} dx = -\frac{x}{\tan x} - \frac{x^2}{2}$.

4.12 Reduzierte Funktionen: der Nenner ist speziell

Eine rationale Funktion mit $f = f_s$, nennen wir f *reduziert bzgl. D* . Die Menge der reduzierten Elemente bezeichnen wir mit $k[t]_S$. Sie ist gleich der Lokalisierung¹¹ von $k[t]$ bezüglich des Monoids der speziellen Polynome. Sie ist sogar eine Differentialringweiterung von $k[t]$. Hier ist die Analogie zu sehen zu den Polynomen, also rationale Funktionen, die keine endlichen Pole besitzen, sondern höchstens einen Pol im Unendlichen oder eben in unserer allgemeineren Situation Pole, die durch speziellen Nenner bestimmt sind. Im Falle von $t := \tan(x)$ ist die rationale Funktion $\frac{t}{t^4+2t^2+1}$ reduziert, also ein Element von $k[t]_S$.

4.13 Faktorisierung und Bewertungen

Anders als Polynomringe kann man algebraischen Funktionen – diesen werden wir uns im Abschnitt 6 auf Seite 103 widmen – im allgemeinen nicht eindeutig in irreduzible Objekte faktorisieren, der Begriff der Teilbarkeit und dann natürlich größte gemeinsame Teiler machen hier keinen Sinn. Dennoch spielen Polstellen und Nullstellen – die man im klassischen Fall der rationalen Funktionen durch diese Operationen unter Kontrolle bekommt – und das Verhalten im Unendlichen eine große Rolle bei der Integration.

Als Ersatz dient nun die Bewertungstheorie. Man abstrahiert die Eigenschaften der Exponenten eines gegebenen Primelementes p in einem Ringelement, also eine Primzahl bei den ganzen Zahlen, oder ein irreduzibles Polynom. Wir bewerten sozusagen ein Element x mit dem Exponenten $\nu_p(x)$ von p . Beispielsweise ist $\nu_5(1000) = 3$, da $1000 = 2^3 5^3$, oder $\nu_{x^2+1} \left(\frac{x^{10}+4x^8+2x^7+6x^6+6x^5+5x^4+6x^3+3x^2+2x+1}{x^{10}+3x^8+3x^6+x^4} \right) = \nu_{x^2+1} \left(\frac{(x^2+1)^2(x^3+x+1)^2}{(x^2+1)^3 x^4} \right) = -1$. Sei also $\nu : R \rightarrow \mathbb{Z} \cup \infty$ mit $\nu(x) = \infty \Leftrightarrow x = 0$, $\nu(xy) = \nu(x) + \nu(y)$, $\nu(x+y) \geq \min(\nu(x), \nu(y))$ jeweils für alle x, y . Bei rationalen Funktionen gibt es außer den schon angesprochenen Exponenten irreduzibler Polynome nur noch eine weitere Bewertung, die das Verhalten der Funktion im Unendlichen beschreibt:

$$\nu_\infty \left(\frac{a}{d} \right) = \deg(a) - \deg(d)$$

für $a \neq 0$.

Bei algebraischen Funktionen betrachtet man die Bewertungen mit Hilfe des Konzepts der Bewertungsringe $k(x) < \mathcal{O} < k(x, y)$, die durch die Eigenschaft, daß für jedes $z \in k(x, y)$ entweder $z \in \mathcal{O}$ oder $z^{-1} \in \mathcal{O}$ gilt, definiert sind. Diese Ringe sind lokale Ringe, d.h. sie besitzen ein eindeutig bestimmtes maximales Ideal P , das ein Hauptideal ist, d.h. von einem Element p erzeugt wird. Das maximale Ideal P wird auch *Stelle* genannt. Ein Erzeuger p spielt die Rolle eines Primelementes, jedes Element $0 \neq z \in k(x, y)$ läßt sich nun eindeutig als $p^n u$ mit $n \in \mathbb{Z}$ und u invertierbar in \mathcal{O} darstellen. Da außerdem n unabhängig vom gewählten Primelement $p \in P$ ist, kann durch $\nu_P(z) := n$ für $z \neq 0$ und $\nu_P(0) := \infty$ eine zugehörige Bewertung definiert werden. Umgekehrt kann man jeder Bewertung ν eine Stelle $P := \{z : \nu(z) > 0\}$ und einen Bewertungsring $\mathcal{O}_\nu := \mathcal{O}_P := \{z : \nu(z) \geq 0\}$ zuordnen. Nun mag noch die Bezeichnung *Stelle* verwundern. In der Tat kann man jetzt bei gegebenem $z \in k(x, y)$

¹¹Das ist das allgemeine Konzept des Bruchrechnens, im Zähler sind Elemente von $k[t]$, im Nenner nur solche aus S zugelassen; siehe auch die Fußnote auf Seite 87.

ganz formal jeder Stelle P einen Wert $z(P)$ zuordnen. Das geschieht durch die Erkenntnis, daß $K := \mathcal{O}_P/P$ ein Restklassenkörper ist, da P ein maximales Ideal ist, und damit $z(P) := z + P \in K$ für $z \in \mathcal{O}_P$ und $z(P) := \infty$ für $z \notin \mathcal{O}_P$ definiert werden kann. Wenn wir nun noch als *Nullstellen der Ordnung n* solche Stellen P mit $z(P) = 0$ und $\nu_P(z) = n > 0$ bezeichnen und analog dazu *Polstellen der Ordnung $n > 0$* mit $z(P) = \infty$ und $\nu_P(z) = -n < 0$, sollte klar sein, daß damit eine Verallgemeinerung der Situation von rationalen Funktionen gelungen ist, die all die notwendigen Begriffsbildungen erlaubt. Man vergleiche dazu auch die in der Funktionentheorie konstruierten RIEMANNSchen Flächen auf denen die algebraischen Funktionen darauf keine mehrfachen Werte mehr besitzen – $\pm\sqrt{x}$! So gibt es etwa für ein z nur endlich viele Nullstellen und Polstellen. Für jedes $z \neq 0$ kann man sich nun alle Null- und Polstellen mit ihren Ordnungen als Koeffizienten als formale ganzzahlige Linearkombination $\sum_{\nu_P(z) \neq 0} \nu_P(z)P$ notieren. Die umgekehrte Frage, wann es zu einem solchen *Divisor*, einem Element dieser sogenannten *Divisorgruppe*, eine Funktion z gibt, die Nullstellen und Polstellen mit jeweils genau vorgeschriebener Ordnung hat, ist bei rationalen Funktionen sehr leicht, bei algebraischen Funktionen aber das Hauptproblem bei der Bestimmung des logarithmischen Anteils des Integrals, man sehe dazu weiter unten den Abschnitt 6.7.

4.14 Der Satz von Liouville

Wir haben im Abschnitt 3.2 gesehen, daß das Integral einer rationalen Funktion auch eine Summe von Logarithmen enthalten kann. Das ist auch deshalb plausibel weil die Ableitung einen Logarithmus vernichtet. Bei Quadraten oder höhere Potenzen von von Logarithmen $(\log^2 x)' = 2 \frac{\log x}{x}$ ist das schon anders. Die Exponentialfunktion gar ist berühmt dafür, daß ihr die Ableitung nichts anhaben kann.¹² Von den Bausteinen der von uns betrachteten elementaren Funktionenwelt fehlen in dieser Plausibilitätsbetrachtung noch die algebraischen Funktionen. Die Formel $y' = -\frac{\partial F(x,y)}{\partial x} / \frac{\partial F(x,y)}{\partial y}$ für die Ableitung einer implizit gegebenen Funktion y mit $F(x,y) = 0$ gibt uns aber keinen Anlaß anzunehmen, daß dabei y eliminiert wird, z.B. ist $(\sqrt{x})' = \frac{1}{2\sqrt{x}}$. Denkt man noch länger über dieses Phänomen nach, dann kann man schnell zu der Vermutung kommen, daß, wenn ein elementares Integral einer elementaren Funktion existiert, dieses eigentlich nur in der Form *rationale Funktion plus Summe von Logarithmen* sein kann.

Solche Überlegungen hat erstmals P.S. LAPLACE 1812 angestellt. N.H. ABEL (1802-1829) und A.M. LEGENDRE (1825-1832) haben für algebraische Funktionen genauere Untersuchungen dieser Art durchgeführt. Daß es hier nicht-elementare Stammfunktionen geben kann, war schon mit den *elliptischen Integralen* $\int \frac{1}{\sqrt{(1-x^2)(1-k^2x^2)}} dx$, $\int \frac{(1-k^2x^2)}{\sqrt{(1-x^2)(1-k^2x^2)}} dx$ und $\int \frac{1}{(1+hx^2)\sqrt{(1-x^2)(1-k^2x^2)}} dx$ für $0 < k < 1$ seit L.EULER im 18. Jahrhundert bekannt. J. LIOUVILLE (siehe Kapitel IX *Integration in Finite Terms* in der Biographie von J. LÜTZEN [44]) hat sich in den Jahren 1833 bis 1841 in elf Arbeiten intensiv mit dieser vermuteten strukturellen Einsicht in die Gestalt eines elementaren Integrals beschäftigt und benutzt. Er hat dieses nach ihm benannte *Prinzip von Liouville* für gewisse Klassen von Funktionen bewiesen. Sein Biograph J. LÜTZEN ([44], p. 420) faßt diese Entwicklung wie folgt zusammen.

Of course, it still requires some function theory to show that ordinary integration theory can be incorporated in the theory of differential fields,

¹²Das geht sogar in Mathematikerwitze ein: Zwei Mathematiker im Irrenhaus. Der eine geht mit einer Drohgebärde auf den anderen zu: „Geh mir aus dem Weg, sonst differenzier' ich Dich“, worauf der andere sich aber mit keinem Schritt bewegt und mit strahlendem Gesichtsausdruck erwidert: „Macht nichts, ich bin die e -Funktion!“.

but the theory has become entirely algebraic. In a way, one considers Rosenlicht's approach as being Liouville's algebraic approach pushed to its extremes, and in fact, several arguments are still close to Liouville's. The recent work of R.H. Risch, on the other hand, approaches the problem of integration in finite terms with algebraic geometrical methods and seems to be far away from Liouville's way of thinking.

Einen ersten modernen Zugang zu diesem Satz hat S. OSTROWSKI 1946 in [50] gegeben. In einer erweiterten algebraisierten Form – siehe M. Rosenlicht [58] –, kann man den Satz wie folgt aussprechen.

Satz 4 *Das Prinzip von Liouville.*

Sei K ein Differentialkörper (der Charakteristik 0)¹³ mit einem algebraisch abgeschlossenen Konstantenkörper C .¹⁴ Sei weiter $f \in K$ gegeben. Wenn es eine elementare Erweiterung E von K gibt, in der es ein g mit $Dg = f$ gibt, dann existiert eine natürliche Zahl n und Elemente v, u_1, \dots, u_n in K sowie Konstante c_1, \dots, c_n in C , so daß

$$f = Dv + \sum_{i=1}^n c_i \frac{Du_i}{u_i}$$

gilt.

Umformuliert mit Integralen heißt das natürlich gerade, daß in diesem Fall

$$\int f \, dx = v + \sum_{i=1}^n c_i \log u_i$$

gilt und daß u_i und v außer den c_i keine anderen Größen enthält, die nicht schon in f auftreten würden.

B.F. CAVINESS, B. SAUNDERS und M.F. SINGER haben den Satz noch weiter verallgemeinert, siehe [64]. Er gilt dann auch noch für Erweiterung mit speziellen Funktionen, siehe etwa [38, 39].

Die Formulierung dieses Satzes ist sicher nicht im Stil der Computeralgebra: Wir sprechen hier von der Existenz von Objekten, sagen aber nicht, wie man diese konstruieren kann. Das ist aber genau das Ziel unserer Aufgabe, Algorithmen zur Integration zu konstruieren. Der Satz wird nun benutzt, um in konkreten Situationen diesem Ziel näher zu kommen. Die *Meta-Erkenntnis*, die wir hier ziehen können, ist also,

daß gute konstruktive und angewandte Mathematik und speziell Computeralgebra sehr wohl theoretische und unkonstruktive Resultate benutzt und braucht, bei diesen aber nicht stehen bleibt, sondern diese einsetzt um konstruktive und algorithmische Resultate zu erzielen.

Der formale Beweis des Satzes von LIOUVILLE kann mit Induktion nach dem Grad m der Erweiterung $E = K(t_1, \dots, t_m)$ und Fallunterscheidung danach, ob $t = t_1$ algebraisch oder ein transzendentes elementares Monom über K ist, also ein Logarithmus oder eine Exponentialfunktion. Wesentlich geht dabei ein, daß in allen 3 Fällen, die Konstanten von $F := K(t_1)$ sich nicht verändern. Nach Induktionsvoraussetzung existieren v, u_1, \dots, u_n in F sowie Konstante c_1, \dots, c_n in C mit

$$f = Dv + \sum_{i=1}^n c_i \frac{Du_i}{u_i}. \quad (19)$$

¹³ Anders als bei dem bei uns immer zu Grunde gelegten Körper der rationalen Zahlen gibt es Körper, zu denen es eine Primzahl p gibt mit $p \cdot 1_K = 0_K$. Diese müssen hier ausgeschlossen bleiben.

¹⁴ D.h. alle polynomialen Gleichungen mit Koeffizienten in C können bereits über C gelöst werden. \mathbb{Q} ist wegen $x^2 - 2$ nicht algebraisch abgeschlossen, aber \mathbb{C} ist das natürlich.

Im transzendenten Fall zeigt man zunächst, daß v und damit Dv in $K[t]_S$ liegen müssen indem man nachweist, daß $\nu_p(v) \geq 0$ für alle normalen, irreduziblen Polynome $p \in K[t]$ gilt. Als nächstes zerlegt man die rationalen Funktionen $u_i \in F = K(t)$ in Potenzprodukte von normierten irreduziblen Polynomen aus $K[t]$ mit positiven oder negativen Exponenten und Koeffizienten w_i aus K , verwendet die logarithmische Ableitung $\frac{D(p_1^{e_1} \dots p_n^{e_n})}{p_1^{e_1} \dots p_n^{e_n}} = \sum_{i=1}^n e_i \frac{Dp_i}{p_i}$ und faßt zusammen. Die auftretenden normierten irreduziblen Polynome müssen allesamt speziell sein, ihre Summe bezeichnen wir mit h , die Summe $\sum_{i=1}^n c_i \frac{Dw_i}{w_i} \in K$ der logarithmischen Ableitungen der Koeffizienten mit g und die Gleichung (19) schreibt sich nun

$$f = Dv + g + h. \quad (20)$$

Ist nun t ein Logarithmus, d.h. es gibt ein $a \in K$ mit $Dt = \frac{Da}{a}$, dann ist jedes irreduzible Polynom normal und es kann keine solchen speziellen Terme geben. Insbesondere kann dann Dv nur ein Polynom sein. Ist $Dv = 0$ folgt die Behauptung sofort, andernfalls kann v nur von der Form $ct + b$, $Dc = 0$ und damit $Dv = Db + c \frac{Da}{a}$.

Im Falle, daß es ein $a \in K$ gibt mit $\frac{Dt}{t} = a$, also t die Exponentialfunktion $\exp a$, gibt es nur ein spezielles irreduzibles und normiertes Polynom, nämlich t . damit kann die Summe h nur einen Term besitzen, nämlich $h = d_1 \frac{Dt}{t} = d_1 a$, wobei $d_1 \in K$ ist.

Abschließend haben wir noch den Fall eines algebraischen Elements t zu betrachten. Da die Spurabbildung sp mit der Derivation D vertauscht und die Spur eine logarithmische Ableitung der logarithmische Ableitung der Norm N ist, können wir sp auf die Gleichung (19) anwenden und erhalten

$$[K(t) : K]f = \text{sp}(f) = D(v) + \sum_{i=1}^n c_i \frac{Nu_i}{u_i}$$

Wir setzen also $w := \sqrt[n]{\frac{v}{[K(t):K]}}$ und $w_i := N(u_i)$ und damit folgt die Behauptung.

Die Aussage des Satzes von Liouville kann noch etwas praktikabler gemacht werden. Üblicherweise liegt nämlich kein algebraisch abgeschlossener Konstantenkörper C vor. In diesem Fall kann mit technischen Argumenten, wie Körpereinbettungen und Spurbetrachtungen die Aussage so präzisiert werden:

Satz 5 *Das Prinzip von Liouville – kein algebraisch abgeschlossener Konstantenkörper. Sei K ein Differentialkörper (der Charakteristik 0) mit Konstantenkörper C . Sei weiter $f \in K$ gegeben. Wenn es eine elementare Erweiterung E von K gibt, in der es ein g mit $Dg = f$ gibt, dann existiert eine natürliche Zahl n , Elemente c_1, \dots, c_n aus einer algebraischen Erweiterung von C , $v \in K$ und Elemente u_1, \dots, u_n in $K(c_1, \dots, c_n)(t)$, so daß*

$$f = Dv + \sum_{i=1}^n c_i \frac{Du_i}{u_i}$$

gilt.

Eine wichtige Anwendung ist die Ableitung struktureller Aussagen bei der Frage der Existenz von elementaren Integralen von reduzierten Elementen $f \in k[t]_S$, d.h. von rationalen Ausdrücken im Monom t , deren Nenner aus S ist. In diesem Fall kann der Satz von Liouville wie folgt spezialisiert werden. Wesentlich ist, daß das Element v nicht nur in $K := k(t)$ liegt, sondern seinerseits bereits in $k[t]_S$ liegt und auch die Argumente u_i der Logarithmen sind speziell.

Satz 6 *Das Prinzip von Liouville – spezieller Fall.*

Sei k ein Differentialkörper (der Charakteristik 0), t ein Monom über k , $C = C_D(k(t))$ der Konstantenteilkörper und $f \in k[t]_S$ ein reduziertes Element. Wenn es eine elementare Erweiterung E von $K = k(t)$ gibt mit Konstantenteilkörper $C_D(E)$, der algebraisch über C ist, in der es ein g mit $Dg = f$ gibt, dann existiert eine natürliche

Zahl n , c_1, \dots, c_n aus einer algebraischen Erweiterung von C , $v \in k[t]_S$ und Elemente u_1, \dots, u_n in $k(c_1, \dots, c_n)(t)$, so daß

$$f = Dv + \sum_{i=1}^n c_i \frac{Du_i}{u_i}$$

gilt.

5 Integration transzendenter Funktionen

5.1 Allgemeine Struktur des Algorithmus

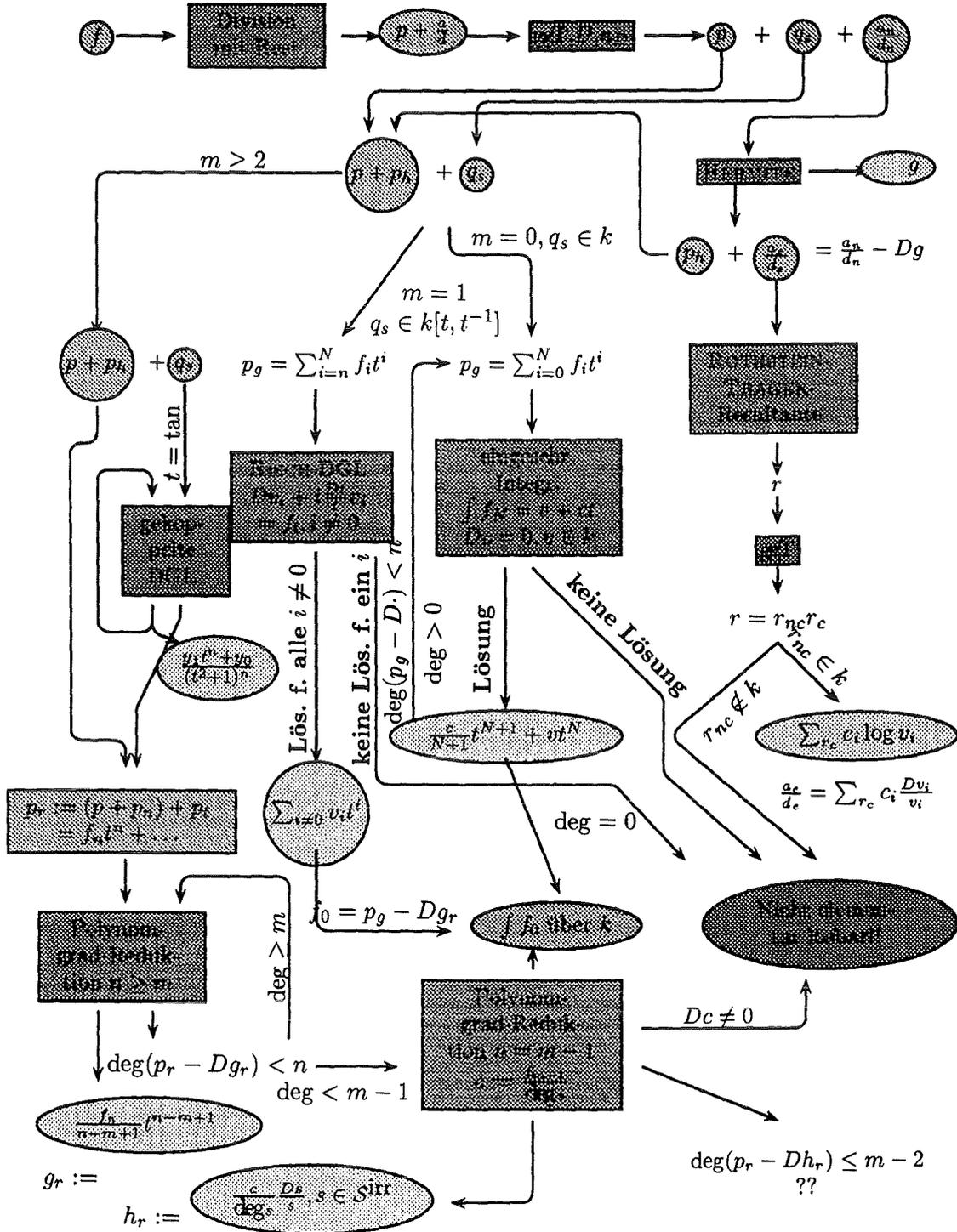
Eine vorgelegte elementare transzendente Funktion f wird man zunächst mit Exponentialfunktionen und Logarithmen t_i so darstellen, daß $f \in C(x, t_1, \dots, t_n)$ für einen Konstantenkörper C liegt. Rekursiv wird nun ein Monom t nach dem anderen eliminiert. Das geschieht so, daß man zunächst f als rationale Funktion in $t = t_n$ über dem Körper $k := C(x, t_1, \dots, t_{n-1})$ darstellt. Wie in Abschnitt 4.10 dargestellt, wird $f = f_p + f_s + f_n$ in seinen polynomialen Anteil f_p , seinen speziellen Anteil f_s und seinen normalen Anteil f_n zerlegt. Letzterer wird dann mit Hermite als Summe einer Ableitung Dq und einer einfachen rationalen Funktion f_e geschrieben. Liegt ein nicht-lineares Monom vor, dann kann der polynomial Anteil f_p unter Generierung einer weiteren Ableitung eines Polynoms so reduziert werden, daß der Grad kleiner als der Grad des Ableitungspolynoms Dt ist, genaueres dazu im Abschnitt 5.4.

So weit kann der Integrationsalgorithmus immer durchgeführt werden und damit eine Funktion zumindest teilweise elementar integriert werden. Was ist nun noch übrig? Zum einen ein einfacher Summand, zum anderen eine reduzierte rationale Funktion. Der einfache Summand ist für den logarithmischen Anteil zuständig. Mit Hilfe der ROTHSTEIN-TRAGER-Resultante, der Verbesserung von LAZARD-RIOBOO-TRAGER und dem Residuenkriterium – siehe Abschnitt 5.3 – findet man immer den logarithmischen Anteil. Zusätzlich behält man ein weiteres Polynom zum Integrieren übrig oder aber man erkennt an dieser Stelle, daß dieses Polynom, gleichgültig welches speziellen Polynom auch immer dazu addiert wird – wir haben uns ja immer noch nicht um f_s gekümmert! – schon nicht mehr elementar integrierbar ist. Schließlich haben wir uns noch um die polynomialen und reduzierten Anteile zu kümmern. Die Reduktion der Grade bzw. der Exponenten im Nenner ist abhängig von den Eigenschaften des Monoms t . In allen Fällen aber wird das Problem auf ein einfacheres Problem in einem kleineren Körper zurückgeführt. Im Abschnitt 5.5 werden primitive Monome wie Logarithmen behandelt. Die Reduktion geschieht durch Lösung eines eingeschränkten Integrationsproblems in k . Der Fall exponentieller Monome führt auf die Lösung der zugehörigen Differentialgleichungen von Risch über k . Das wird im Abschnitt 5.6 dargestellt. Schließlich wird im Abschnitt 5.7 der Fall von Tangensmonomen behandelt. Dies ist ein Alternative zur Darstellung des Tangens durch Exponentialfunktionen und der komplexen Einheit, komplexe Integration und anschließende Bestimmung eines reellen Integrals. Das zu lösende Basisproblem über k ist die Lösung eines gekoppelten Systems zweier linearer Differentialgleichung der Ordnung 1.

Wir haben die wichtigsten Teile dieses Algorithmus zur Veranschaulichung in einem Ablaufdiagramm zusammengestellt.

5.2 Diagramm zur Integration transzendenter Funktionen

Sei $f \in k(t)$, $Dt \in k[t]$, $m := \deg Dt$ und $C_D(k) = C_D(k(t))$.



5.3 Residuenkriterium von Rothstein und Trager für einfache Funktionen

Die Resultante von ROTHSTEIN-TRAGER kann auf die allgemeine Situation transzender Funktionen übertragen werden. An Stelle der quadratfreien Nenner einer allgemeinen rationalen Funktion tritt nun ein normales Polynom mit $\nu(q) \geq -1$ für einige normale irreduzible und von Null verschiedene Polynome q . Dabei wird mit Hilfe der Bewertungstheorie die Residuumsfunktion $\text{residue}_q(f)$ – siehe Abschnitt 4.13 – für ein solches irreduzibles, normales $0 \neq q \in k[t]$ durch $\pi_q(f \frac{q}{Dq}) \in k[t]/qk[t]$ für alle Funktionen aus $\frac{1}{q}\mathcal{O}_q$, also solche mit $\nu_q(f) \geq -1$, definiert, da natürlich $\mathcal{O}/\frac{1}{q}\mathcal{O}_q \cong k[t]/qk[t]$ gilt. Man beachte, daß wegen der Normalität von q die Ungleichung $\nu_q(Dq) \geq 0$ gilt, und damit $\nu_q(f \frac{q}{Dq}) \geq 0$. Zur Verallgemeinerung des Resultantensatzes von ROTHSTEIN-TRAGER sei nun $f \in k(t)$ ein einfaches Polynom, das sich als $p + \frac{q}{d}$ nach Division mit Rest darstellen läßt: $p, a, d \in k[t]$, $d \neq 0$, $\deg a < \deg d$ und $\text{ggT}(a, d) = 1$. Sei

$$r := \text{res}_z(a - zDd, d) \in k[z]$$

mit einer neuen Unbekannten z . Ein Element α ist eine Nullstelle der Resultante r genau dann, wenn α das Residuum $\text{residue}_q(f)$ von f bzgl. eines normalen irreduziblen Polynoms p ist. Als Konsequenz des Satzes von Liouville erhalten wir im Falle eines algebraisch abgeschlossenen Konstantenkörpers $C_D(k)$ die zusätzliche Bedingung, daß ein $h + f$ für einfache rationale Funktion f und jede reduzierte rationale Funktion aus $k[t]_S$ genau dann elementar integrierbar ist, wenn $\text{residue}_p(f)$ – betrachtet als Element in $k[t]!$ – eine Konstante für jedes normale, irreduzible Polynom p ist. Das kann nun so in ein Kriterium verarbeitet werden: Zur Integration einer einfachen Funktion $f = p + \frac{q}{d}$ wird die Resultante $r := \text{res}_z(a - zDd, d)$ in ihren speziellen und ihren normalen Anteil

$$r = r_s r_n$$

mit Hilfe der Derivation κ_D zerlegt. Man beachte, daß die Nullstellen von r_s genau die Nullstellen von r sind, die Konstante sind. Dann kann man den logarithmischen Anteil

$$g := \sum_{\{\alpha: r_s(\alpha)=0\}} \alpha \frac{Dg_\alpha}{g_\alpha}$$

mit

$$g_\alpha := \text{ggT}(a - \alpha Dd, d)$$

bestimmen. Es gilt $g \in k(t)$, der Nenner von g teilt d und $f - g$ ist einfach. Hat nun $f + h$ ein elementares Integral über $k(t)$ für irgendein $h \in k[t]_S$, dann muß $r_n \in k$ sein und $f - g$ ist sogar ein Polynom. Auch dazu gibt es wieder eine Variante mit Subresultanten.

Als Beispiel kann man etwa sehr leicht einsehen, daß der Integrallogarithmus $\text{lix} = \int \frac{1}{\log x} dx$ nicht elementar ist. Mit $t := \log(x)$ folgt $\text{res}_t(1 - \frac{x}{t}, t) = 1 - \frac{x}{t}$, die Nullstelle x ist keine Konstante!

5.4 Polynomreduktion für nicht-lineare Monome auf Polynome höchstens vom Grad des Ableitungspolynoms minus zwei

Unter nicht-linearen Monomen t verstehen wir natürlich solche, deren Ableitung ein Polynom vom Grad m größer 1 ist. Wir können dieses Polynom benutzen um jedes Polynom $f \in k[t]$ vom Grad größer oder gleich m als Summe einer Ableitung einer Polynoms $q \in k[t]$ und eines Polynoms $r \in k[t]$ vom Grad kleiner oder gleich $m - 2$ darzustellen, d.h.

$$\int f = q + \int r.$$

Sei $n := \deg f > 0$ und zunächst $n \geq m$. Weiter sei f_n der Leitkoeffizient von f sowie λ derjenige von Dt . Wir nutzen aus, daß $D(t^{n-m+1}) = t^{n-m}Dt = t^{n-m}(\lambda t^m + \dots)$ den Grad n hat. Mit einem geeigneten Koeffizienten $c \in k$ versehen, – der andere Summand $Dc \cdot t^{n-m+1}$ stört nicht, da ja nach Voraussetzung $m \geq 2$ gilt! – kann man das verwenden um den größten Term von f zu eliminieren. In der Tat, setzt man $c := \frac{f_n}{(n-m+1)\lambda} \in k$, dann hat $f - D(ct^{n-m+1})$ einen Grad kleiner n und das Verfahren kann für den nun höchsten Grad iteriert werden bis man einen Grad kleiner als m erreicht hat.

Gilt nun $n = m-1$ für das aktuelle noch zu integrierende Polynom f , dann ist noch ein weiterer Reduktionsschritt möglich. Wir haben $f_{m-1} \neq 0$ und setzen $c := \frac{f_{m-1}}{\lambda} \in k$. Der Satz von Liouville impliziert als notwendige Bedingung, daß $Dc = 0$ ist. In jedem Fall aber kann man t^{m-1} in f eliminieren. Es gilt nämlich $\deg(f - \frac{c}{\deg p} \frac{Dp}{p}) < m-1$, wobei p irgendein spezielles Polynom vom Grad größer 0 ist.

5.5 Der Logarithmus im speziellen Fall – Integration von Polynomen primitiver Monome

Der Prototyp eines primitiven Monoms ist der Logarithmus. Ist $t := \log a$ für $a \in k$, so gilt $Dt = \frac{Da}{a} \in k$. Da $S = k$ und deshalb $k[t]_S = k[t]$ gilt, spezialisiert sich der Satz 6 so: Wenn es zu $f \in k[t]$ ein elementares Integral gibt, dann existieren eine natürliche Zahl n , Konstanten c_1, \dots, c_n aus einer algebraischen Erweiterung des Konstantenkörpers $C_D(k)$, ein Polynom $v \in k[t]$ und Elemente u_1, \dots, u_n bereits alle in $k(c_1, \dots, c_n)$ mit $f = Dv + \sum_{i=1}^n c_i \frac{Du_i}{u_i}$. Wegen $Dt \in k$ kann damit v höchstens vom Grad $m+1$ sein, wobei $m := \deg f$ ist. Sei also $f := f_m t^m + f'$, $\deg f' < m$ und $v := v_{m+1} t^{m+1} + v_m t^m + v'$, $\deg v' < m$. Aus der Betrachtung der Exponenten $m+1$ und m folgt

$$Dv_{m+1} = 0 \text{ und } f_m = Dv_m + (m+1)Dt v_{m+1}.$$

Als notwendige Bedingung zur Entscheidung der Frage, ob f elementar integrierbar ist, haben wir also diese sogenannte *eingeschränkte Integrationsproblem* über k zu lösen. Hier sucht man zu gegebenem $f \in k$ und einem $w \in k$ ein $g \in k$ (!) und eine Konstante $c \in C_D(k)$ (!), sodaß $f = Dg - cw$ gilt (hier $f := f_m, w := -(m+1)Dt, c := v_{m+1}, g := v_m$). Kann dieses Problem in k algorithmisch gelöst werden, so gibt es zwei Möglichkeiten. Entweder wir finden Lösungen für $v_m \in k$ und $v_{m+1} \in C_D(k)$ und haben damit das Problem der Integration von f auf die Integration von $f - D(v_{m+1} t^{m+1} + v_m t^m)$, ein Polynom vom Grad kleiner als m , zurückgeführt. Das Verfahren kann iteriert werden bis schließlich t vollständig eliminiert ist. Im zweiten Fall gibt es keine Lösung des eingeschränkten Integrationsproblems, dann wissen wir aber, daß f nicht elementar integrierbar ist.

Bei der Integration von Potenzen t^m des Logarithmus $t := \log(x)$ über $k := \mathbb{Q}(x)$ kann immer $v_{m+1} := 0$ und $v_m := x$ gewählt werden. Man erhält die Rekursion $\int t^m = t^m - m \int t^{m-1}$ mit Anfangslösung $\int t^0 = x$, die die geschlossene Lösung $\int t^m = x \left(t^m + \sum_{i=1}^m (-1)^i \frac{m!}{(m-i)!} t^{m-i} \right)$ hat.

Das nächste Beispiel ist aus dem Buch [10] von M. BRONSTEIN. Sei $f := (\log x + \frac{1}{\log x})\text{li}x - \frac{x}{\log x}$ mit dem Integrallogarithmus $\text{li}x = \int \frac{1}{\log x} dx$. Sei $k := \mathbb{Q}(x, t_0)$ mit $t_0 := \log x$, $Dt_0 = \frac{1}{x}$ und $t := \text{li}x$ mit $Dt = \frac{1}{t_0} \in k$, d.h. der Integrallogarithmus ist ein primitives Monom über $\mathbb{Q}(x, \log x)$. Das Polynom f in t ist vom Grad 1, der Leitkoeffizient ist $t_0 + \frac{1}{t_0}$, wir haben also die Gleichung $t_0 + \frac{1}{t_0} = Dv_1 + 2\frac{1}{t_0}v_2$ in v_1 und v_2 mit der zusätzlichen Bedingung $Dv_2 = 0$ zu lösen. Setzt man $v_2 := \frac{1}{2}$, so ist das Problem eingeschränkt auf die Bestimmung von $\int \log x dx$, dessen Lösung mit derselben Methode $\int \log x dx = x \log x - x$ ist. Damit ergibt sich $f - \frac{1}{2}\text{li}(x)^2 - (x \log x - x)\text{li}x = -x \in k$, also $\int f dx = \frac{\text{li}(x)^2}{2} - (x \log x - x)\text{li}x - \frac{x^2}{2}$.

5.6 Die Exponentialfunktion im speziellen Fall – Integration von Laurentpolynomen und die Differentialgleichung von RISCH

Hier haben wir die Situation von Integranden aus $k[t]_S$ in der Situation $S^{\text{irr}} = \{t\}$, also $k[t]_S = k[t, t^{-1}]$, dessen Elemente *Laurentpolynome* genannt werden, zu betrachten. Für das Monom t gilt $Dt = at$ für ein $a \in k$. Sei also $f = f_M t^M + \dots + f_m t^m \in k[t, t^{-1}]$ mit ganzen Zahlen $m \leq M$ und $f_i \in k$. Unter der Annahme, daß eine Funktion mit einem elementaren Integral vorliegt, kann man nun den Satz 6 wie folgt spezialisieren. Die speziellen u_i können nur von der Gestalt $u_i = b_i t^{m_i}$ mit ganzen Zahlen m_i und $b_i \in k$ sein, analog gilt $v = \sum_{i=r}^R v_i t^i$ mit ganzen Zahlen $r \leq R$ und Koeffizienten $v_i \in k$. Damit folgt

$$\sum_{i=1}^n c_i \frac{Du_i}{u_i} = \sum_{i=1}^n c_i \frac{D(b_i t^{m_i})}{b_i t^{m_i}} = \frac{Dt}{t} \sum_{i=1}^n m_i c_i + \sum_{i=1}^n c_i \frac{Db_i}{b_i} \in k(c_1, \dots, c_n),$$

was nur sein kann, wenn alle Koeffizienten der t^i gleich 0 sind. Folglich ist $f - Dv \in k(c_1, \dots, c_n)$, also

$$\sum_{i=\min(r,m)}^{\max(M,R)} (f_i - Dv_i - iv_i \frac{Dt}{t}) t^i \in k(c_1, \dots, c_n).$$

O.E. können wir $v_i = 0$ für $i > M$ und $i < m$ annehmen. Eine notwendige Bedingung für die Existenz eines elementaren Integrals ist folglich, daß es für alle Gleichungen

$$Dv_i + (i \frac{Dt}{t}) v_i = f_i, \quad m \leq i < 0, \quad 0 < i \leq M$$

Lösungen in k gibt. Diese lineare Differentialgleichung erster Ordnung

$$Dy + \alpha y = \beta$$

über k mit $\alpha, \beta \in k$ heißt *Differentialgleichung von RISCH*. Um das Integrationsproblem von Funktionen aus $k(t)$ mit einem exponentiellen Monom t lösen zu können, ist es also notwendig, daß man die Differentialgleichung von RISCH über dem Grundkörper k lösen kann. M. BRONSTEIN hat in [8] einen rationalen Algorithmus dafür angegeben. Ältere Arbeiten zu diesem Thema sind [53], [33] und [20].

Nun das Beispiel, auf das wir alle schon lange gewartet haben. Sei $k := \mathbb{Q}(x)$, $t := e^{-x^2}$, $Dt = -2xt$. Die Frage nach $\int e^{-x^2}$ führt auf die Lösung der Differentialgleichung $Dy - 2xy = 1$ über $\mathbb{Q}(x)$. Sei $\frac{p}{q}$ eine Lösung mit $\deg q > 0$ und $\text{ggT}(p, q) \neq 1$. Dann hat q einen nicht-trivialen irreduziblen Faktor mit Exponenten n . Folglich enthält der Nenner von Dy diesen Faktor mit Exponenten $n + 1$, während der Nenner von $2xy$ diesen Faktor höchstens mit Exponenten n enthalten kann. Also kann y nur ein Polynom p sein. Wegen $\deg Dp \neq \deg(2xp)$ kann es aber auch dann keine Lösung geben.

Im nächsten Beispiel sei $k := \mathbb{Q}(x, \sin x, \cos x)$, $t := e^{\sin x}$, also $\frac{Dt}{t} = \cos x$. Zur Integration von

$$f := (2 \cos^2 x - \sin x) e^{2 \sin x} + (3x^2 + x^3 \cos x) e^{\sin x} + \frac{\cos x - x \sin x}{e^{\sin x}}$$

sind die drei Differentialgleichungen $Dv_2 + 2 \cos x v_2 = 2 \cos^2 x - \sin x$, $Dv_1 + \cos x v_1 = 3x^2 + x^3 \cos x$ und $Dv_{-1} - \cos x v_{-1} = \cos x - x \sin x$ zu lösen. Lösungen sind $v_2 = \cos x$, $v_1 = x^3$ und $v_{-1} = x \cos x$, folglich gilt

$$\int f dx = \cos x e^{2 \sin x} + x^3 e^{\sin x} + \frac{c \cos x}{e^{\sin x}}.$$

5.7 Die Tangensfunktion im speziellen Fall – gekoppelte lineare Differentialgleichungen

Hier betrachten wir den Fall eines Monoms mit $\deg Dt = 2$, genauer $Dt = a(t^2 + 1)$, also $t = \tan(f a)$ mit $t^2 + 1 \neq 0$ und $C_D(k) = C_D(k(t))$. Bedingungen dafür, wann so eine Tangensfunktion ein Monom ist, sind in [10] gegeben. Das Problem der Integration eines Elements aus $k(t)$ kann natürlich mittels der komplexen Einheit i auf den Fall von Exponentialfunktionen zurückgeführt werden. Ist man aber an reellen Darstellungen interessiert, so verzichtet man auf i und man kann deshalb annehmen, daß $t^2 + 1$ irreduzibel über k ist. Es gilt $S^{\text{irr}} = \{t^2 + 1\}$.

Ein allgemeines reduziertes Element ist aus $k[t, (t^2 + 1)^{-1}]$. Mit zum exponentiellen Fall analogen Überlegungen wird die Reduktion der Exponenten von $(t^2 + 1)^{-1}$ zurückgeführt auf die Lösung eines einfacheren Problems über dem Grundkörper k . Das Analogon zur Differentialgleichung von RISCH ist hier die Lösung zweier gekoppelter linearer Differentialgleichungen der Ordnung 1 über k . Sei dazu m der größte Exponent von $(t^2 + 1)^{-1}$ in f , d.h. $(t^2 + 1)^m f \in k[t]$ und sei weiter $(t^2 + 1)^m f = q(t^2 + 1) + a_1 t + a_0$ mit $a_1, a_0 \in k$ das Ergebnis der Division mit Rest, dann betrachtet man das System

$$\begin{pmatrix} Dy_1 \\ Dy_0 \end{pmatrix} + \begin{pmatrix} 0 & -2m \frac{Dt}{t^2+1} \\ 2m \frac{Dt}{t^2+1} & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_0 \end{pmatrix} = \begin{pmatrix} a_1 \\ a_0 \end{pmatrix}.$$

Gibt es keine Lösung, so hat f kein elementares Integral, andernfalls kann mittels $f - D\left(\frac{y_1 t + y_0}{(t^2 + 1)^m}\right)$ der größte Exponent m eliminiert werden. Das Verfahren wird iteriert bis entweder einmal die gekoppelte Differentialgleichung keine Lösung besitzt oder aber nur noch die Integration eines Polynoms zu bewerkstelligen ist.

Die Integration eines Polynoms hingegeben kann wegen $\deg Dt = 2$ vollständig mit den Methoden aus Abschnitt 5.4 auf die Integration eines Elementes aus k reduziert werden, bzw. man erkennt, daß es keine elementare Lösung geben kann.

Beispiel: $f := \frac{2 + \tan(x)^2}{1 + (\tan(x) + x)^2}$. Nenner q ist normal. Bilde dazu $\text{ggT}(q, Dq)$ und $\text{ggT}(q, \frac{\partial}{\partial t} Dq)$ und irreduzibel. Reduktion mir Hermite.

5.8 Die Risch-Norman-Methode

Kurz erwähnt sei noch die Methode von RISCH-NORMAN, die nicht in dieses Schema der rekursiven Elimination der Monome paßt. Hier werden alle Logarithmen und Exponentialfunktionen simultan behandelt und eine multivariate Faktorisierung des Nenners über einem algebraischen Abschluß des Konstantenkörpers durchgeführt. Die Idee ist, den Zähler des rationalen Anteil und die Koeffizienten der Logarithmen, von denen angenommen wird, daß als Argumente nur die irreduziblen Faktoren des Nenners auftauchen, a là HOROWITZ durch lineare Gleichungssysteme zu bestimmen. Damit dieses Vorgehen zulässig ist, sind die Körper, die die genannten und noch weitere technische Bedingungen wie Gradbeschränkung des Zählers, erfüllen, zu identifizieren, eine nicht gerade einfache Aufgabe. Literatur zu dieser Methode ist [48], [18] und [19]

6 Die Integration algebraischer Funktionen

Dies ist eines der schwierigsten Kapitel in der ganzen Theorie. Im Rahmen dieser Übersichtsarbeit können die Probleme und Schwierigkeiten, die hier auftreten nur ganz kurz gestreift werden. Die Arbeiten [54] von R.H. RISCH und [17] von J.H. DAVENPORT beschäftigen sich mit diesem Problem. Besonders empfehlenswert dazu ist die Doktorarbeit von B.M. TRAGER [67], die das Problem für den Fall des Grundkörpers $\mathbb{Q}(x)$ löst, und der wir in diesem Abschnitt folgen. Eine kurze Übersicht ist auch im Abschnitt 12.8 in [24] zu finden. M. BRONSTEIN verallgemeinert in [9] B.M. TRAGERS Algorithmus für die Situation des Grundkörpers $k(t)$ mit t ein Monom, das entweder ein Logarithmus oder eine Exponentialfunktion ist. Auf diesen allgemeinen Fall von gemischt algebraisch-transzendenten Funktionen können wir in dieser Arbeit nicht weiter eingehen.

6.1 Normierte Darstellung einer algebraischen Funktion

Die Probleme beginnen schon damit, daß es nicht ohne weiteres eine kanonische Darstellung einer rationalen Funktion $f(x, y) \in k(x, y)$ in der Unbestimmten x und in einer algebraischen Funktion y mit irreduziblem Polynom F mit $F(x, y) = 0$ gibt. Es reicht nämlich nicht aus zu fordern, daß der Zähler und der Nenner teilerfremde Polynom in $k(x)[y]$ sind. Beim einfachsten Beispiel $F(x, y) = y^2 - x$, also $y(x) = \pm\sqrt{x}$, kann man das wegen

$$f(x, y) = \frac{1}{y} = \frac{y}{x} \quad (21)$$

schon sehen.

Zunächst wird man aber noch erreichen wollen, daß F normiert ist, d.h. Leitkoeffizienten 1 hat. Falls das nicht schon der Fall ist, betrachtet man die algebraische Funktion ay , falls $a \in k(x)$ der Koeffizient des höchsten Potenz von y ist. Diese Funktion erzeugt natürlich denselben Funktionenkörper. Für die algebraische Funktion $y := f_+$ in Gleichung (1) auf Seite 74 gilt $0 = F(x, y) = y^2 - \frac{x+1}{x^3-5x^2+9x-8} \frac{x+1}{x+4}$. Multipliziert man mit dem Nenner – in faktorisierte Form gleich $a := (x-2)^2(x^2-x+1)$ – so erhält man die polynomiale Gleichung $ay^2 - (x+1) = 0$ und damit $(ay)^2 - a(x+1) = 0$. Man kann also immer $F \in k[x][y]$ erzielen.

Das Problem einer normierten Darstellung kann nun dadurch gelöst werden, daß man sich erinnert, daß im Falle von $n := \deg_y F$ der $k(x)$ -Vektorraum $k(x, y) = k(x)(y) = k(x)[y]$ die Dimension n hat, da sich y^n und damit auch alle höheren Potenzen mittels F reduzieren lassen. Wir können daher die kanonische Basis $(1, y, y^2, \dots, y^{n-1})$ benutzen und jedes f eindeutig als

$$\sum_{i=0}^{n-1} \frac{a_i(x)}{d_i(x)} y^i$$

mit $a_i, d_i \in k[x]$ darstellen. Zusätzlich kann man das kleinste gemeinsame Vielfache d der Nennerpolynome bilden und erhält dann die normierte Darstellung

$$f(x, y) = \frac{\sum_{i=0}^{n-1} a_i(x) y^i}{d} \quad (22)$$

mit $a_i, d \in k[x]$ und $\text{ggT}(d, a_1, \dots, a_n) = 1$. Betrachten wir das Beispiel

$$F(x, y) = y^4 - x - 1$$

und

$$f(x, y) := \frac{y^2 - x y}{y^2 + (x-1)y - x}$$

Der Ansatz $f - a - by - cy^2 - dy^3 = 0$ liefert ein Gleichungssystem das leicht¹⁵ nach den Koeffizienten a, b, c, d gelöst werden kann und wir erhalten als normierte Darstellung die Form

$$\frac{(x^4 - 2x^3 - x + 1)y^3 + (x^4 - x + 1)y^2 + (-x^4 - x + 1)y - x^5 + x^4 - x^2 - 1}{x^5 - x^2 - x}$$

6.2 Ganzheitsbasen

Was aber passiert mit den endlichen Polstellen? Diejenigen der Ordnung 1 sind wie wir wissen, lebenswichtig für die Logarithmen des Integrals. Es muß also hier sichergestellt werden, daß wir mit diesen Transformationen nicht wie im obigen Beispiel scheinbar neue Pole generieren! Genauer gesagt, wir hätten gern die Situation wie im rationalen Fall, daß die Nullstellen des Nennerpolynoms genau die Polstellen des Integranden sind! Dieses Problem hat B.M. TRAGER gelöst indem er nicht die Basis $(1, y, \dots, y^{n-1})$ benutzt, sondern eine Basis $w := (w_0, w_1, \dots, w_{n-1}) \in k(x, y)^n$ des $k[x]$ -Moduls¹⁶ der über $k[x]$ ganzen Elemente in $k(x, y)$ bestimmt. Das sind diejenigen rationalen Funktionen in x und y , die ebenso wie y mit F eine polynomiale Ganzheitsgleichung mit Koeffizienten aus $k[x]$ erfüllen. Mit Sicherheit sind alle $k[x]$ -Linearkombinationen von $(1, y, \dots, y^{n-1})$, also die Elemente aus $k[x](y)$ ganz. Ist die Diskriminante von F bzgl. y quadratfrei, so liegt bereits die gesuchte *Ganzheitsbasis* in $(y^i)_{0 \leq i < n}$ vor.

Solche Basen haben nun die hervorragende Eigenschaft, daß für jede polynomiale Linearkombination $\sum_{i=0}^{n-1} h_i w_i$, – genau wie im Falle der rationalen Funktionen durch die Nullstellen des Nenners, – die endlichen Pole durch die Nullstellen der Nenner der Summanden $h_i w_i$ bzw. die Nullstellen des Nenners d des kleinsten gemeinsamen Vielfaches in der normierten Form $f = \frac{\sum_{i=0}^{n-1} a_i w_i}{d}$ bestimmt sind. Durch Summation können sich also keine Polstellen eliminieren.

6.3 Konstruktion von Ganzheitsbasen

Die Diskriminante mißt in gewisser Weise, wie weit die Basis $(1, y, \dots, y^{n-1})$ noch vom Ziel einer Ganzheitsbasis entfernt ist. So ist beispielsweise das Produkt jedes ganzen Elementes mit der Diskriminante in $k[x](y)$! Mit der Berechnung der Determinante der Matrix $(\text{sp}(y^i y^j))_{0 \leq i, j < n}$ der Spuren der Multiplikationshomomorphismen der Produkte $y^i y^j$ ergibt sich in diesem Fall (F irreduzibel) eine alternative Berechnung der Diskriminante $\text{disk}_y(F)$. Statt $(y^i)_{0 \leq i < n}$ kann man nun eine beliebige Basis $(b_i)_{0 \leq i < n}$ in diese Konstruktion einsetzen und man erhält die *Diskriminante* dieser Basis.

Ausgehend von der Basis $(1, y, \dots, y^{n-1})$ bestimmt man eine $k[x]$ -Basis des größten Ringes (*Idealisator*) in $k(x, y)$, indem das Radikal¹⁷ des von der Diskriminante in $k[x](y)$ erzeugten Ideals noch ein Ideal ist und iteriert dieses Verfahren bis die Diskriminante quadratfrei ist.

In Beispiel oben ist $(1, y, y^2, y^3)$ bereits eine Ganzheitsbasis. Im Falle von $F(x, y) = y^3 - xy^2 + x^2y = 0$ liefert der Algorithmus von TRAGER die Ganzheitsbasis $w = (1, \frac{y}{x}, \frac{y^2}{x^2})$. Dazu wird im Algorithmus zunächst die Diskriminante $d := \text{disk}_y(F) = -3x^6$ bestimmt. Eine Basis des Radikals des Diskriminantenideals in $V_1 := \langle 1, y, y^2 \rangle_{\mathbb{Q}[x]} \leq \mathbb{Q}(x, y) = \mathbb{Q}(x)(y)$ kann in diesem Fall (nur ein Primteiler x , der allgemeine Fall ist auf Seite 24 in [67] behandelt) als homogener Lösungsraum modulo (x) des durch

¹⁵Natürlich mit einem Computeralgebra-System!

¹⁶Wie ein Vektorraum, aber die Skalare sind nicht aus einem Körper, sondern nur aus einem Ring.

¹⁷Das Radikal eines Ideals I sind alle Elemente des Rings, die man durch Wurzelziehen erhält, also die Menge $\{r \in R \mid \exists n : r^n \in I\}$.

die Matrix $(\text{sp}(y^{i+j}))_{0 \leq i, j \leq 2}$ gegebenen linearen Gleichungssystems bestimmt werden. Die Spur eines Elementes des algebraischen Funktionenkörpers ergibt sich durch die übliche Spur (Summe der Elemente auf der Hauptdiagonalen) der Matrix, die die Multiplikation mit diesem Element bzgl. der gegebenen Basis $(1, y, y^2)$ – etwa in Spaltenschreibweise – beschreibt. Hier ergibt sich (x, y, y^2) als $\mathbb{Q}[x]$ -Basis des Radikals I_1 . Die Bestimmung des größten Ringes über V_1 indem I_1 noch Ideal ist, d.h. allen Elementen u aus $\langle 1, y, y^2 \rangle_{\mathbb{Q}(x)}$, die $uI_1 \subseteq I_1$ erfüllen, kann nun durch Formulierung von Bedingungen auf der $\mathbb{Q}[x]$ -Basis (x, y, y^2) von I_1 getestet werden, d.h.

$$xu = r_{1,1}x + r_{1,2}y + r_{1,3}y^2, yu = r_{2,1}x + r_{2,2}y + r_{2,3}y^2, y^2u = r_{3,1}x + r_{3,2}y + r_{3,3}y^2$$

mit $r_{i,j} \in k[x]$. Werden die darstellenden Matrizen der Multiplikation mit x, y und y^2 und zwar bzgl. der Basis $(1, y, y^2)$ im Quellbereich, und bzgl. der Basis (x, y, y^2) im Zielbereich(!) bestimmt, so können die neun Bedingungen so formuliert werden. Die darstellenden (3×3) -Matrizen der Multiplikationshomomorphismen der Basisvektoren von I_1 werden zu einer (9×3) -Matrix konkateniert und durch Zeilenoperation auf ihre hermitesche Normalform gebracht. Hier ergibt das nur noch drei dazu äquivalente

Bedingungen:¹⁸ $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & x \end{pmatrix}$, umgeformt. Die Spalten der Inversen $\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{x} \end{pmatrix}$ die-

ser Matrix bestimmen eine $\mathbb{Q}[x]$ -Basis des Idealisators, hier also $(1, y, \frac{y^2}{x})$. Damit hat sich die Diskriminante $-3x^4$ um x^2 verkleinert. Nun wird das Verfahren iteriert bis die Diskriminante keine nicht-trivialen Bestandteile hat. In unserem Beispiel genügt ein weiterer Schritt, der als Basis des Radikals $(x, y, \frac{y^2}{x})$ und als Transformationsmatrix

$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{x} & 0 \\ 0 & 0 & \frac{1}{x} \end{pmatrix}$ liefert. Damit ist $(1, \frac{y}{x}, \frac{y^2}{x^2})$ die gesuchte Basis des ganzen Abschlusses.

Die minimalen ganzen Gleichungen für $\frac{y}{x}$ und $\frac{y^2}{x^2}$ sind übrigens $z^3 - z^2 - z = 0$ und $z^3 + z^2 + z = 0$.¹⁹

6.4 Ganzheitsbasen für einfache Radikalerweiterungen

Besonders einfach ist der Spezialfall $F(x, y) = y^n - p(x)$, siehe Abschnitt 2.5. in [67]. Man kann annehmen, daß die Vielfachheit einer Nullstelle von p höchstens $n - 1$ ist. Ist also $p = \prod_{i=0}^{n-1} p_i^i$ die quadratfreie Zerlegung, dann kann man $g_i := \prod_{j=0}^{n-1} p_j^{\lfloor \frac{ij}{n} \rfloor}$ setzen²⁰ und erhält die Ganzheitsbasis

$$\left(\frac{y^i}{g_i} \right)_{0 \leq i \leq n-1}.$$

Im Beispiel $F(x, y) = y^3 - x^3 + 2x^2 - x$ ergibt sich damit $(1, y, \frac{y^2}{x(x-1)^2})$. Der Spezialfall eines quadratfreien p liefert natürlich die Ganzheitsbasis $(1, y, \dots, y^{n-1})$. Um in dem von $y := f_+$ aus (1) erzeugten Funktionenkörper eine Ganzheitsbasis zu bestimmen, müssen wir die in Abschnitt 6.1 für ay gewonnene Gleichung $(ay)^2 - a(x+1) = 0$ mit $a = (x-2)^2(x^2 - x + 1)y$ noch modifizieren, da die quadratfreie Faktorisierung des Polynoms $p := a(x+1)$ noch Potenzen $n = 2$ enthält. Dazu dividieren wir durch $(x-2)^2$ und betrachten nun $(x-2)(x^2 - x + 1)y$ mit Gleichung $((x-2)(x^2 - x +$

¹⁸In AXIOM mit completeHermite.

¹⁹Ein schnellerer Weg in diesem Beispiel wäre die Erkenntnis gewesen, daß F homogen vom Grad 3 ist. Man hätte also durch x sofort dividieren können und eine Ganzheitsgleichung von $\frac{y}{x}$ erhalten.

²⁰ $\lfloor \]$ bezeichnet die untere Gaußklammer, die die größte ganze Zahl bestimmt, die kleiner oder gleich der eingeschlossenen Zahl ist.

$1)y)^2 - (x^2 - x + 1)(x + 1) = ((x^3 - 3x^2 + 3x - 2)y)^2 - (x^3 + 1)$. Wir haben aus didaktischen Gründen diesen Umweg gewählt, man sieht natürlich auch sofort in f_+ , daß man eine durch $\sqrt{x^3 + 1}$ definierte algebraische Erweiterung von $\mathbb{Q}(x)$ benötigt. Nach der Formel für die Ganzheitsbasis sehen wir sofort, daß $(w_0, w_1) = (1, (x-2)(x^2 - x + 1)y)$ eine Ganzheitsbasis der durch f_+ erzeugten einfachen Radikalerweiterung ist. Die kanonische Darstellung von f_+ ist damit $\frac{w_1}{(x-2)(x^2-x+1)}$.

6.5 Große Grade der polynomialen Zählerkoeffizienten – Polstelle im Unendlichen

Mit den bisherigen Überlegungen sind die endlichen Pole unter Kontrolle. Um nun analog dem rationalen Fall zusätzlich zu erreichen, daß die Grade der Zählerpolynome a_i in Gleichung (22) kleiner als die des Nennerpolynoms d sind, transformiert man vorab mit einer ganzen Zahl j , die weder Nullstelle von d noch von der Diskriminante²¹ von $k(x, y)$ ist, das Problem mittels $z := \frac{1}{x-j}$ bzw. $x = \frac{1}{z} + j$ und der Substitutionsregel auf die Integration von $\frac{\sum_{i=0}^{n-1} -a_i(j+\frac{1}{z})y^i}{z^2 d(j+\frac{1}{z})}$, bestimmt dann die Ganzheitsbasis und transformiert am Ende das Ergebnis zurück.

6.6 Simulation von Hermite

Nach diesen Vorbereitungen kann man nun mit Hilfe von Hermite wie im rationalen Fall die Exponenten der quadratfreien Zerlegung von d auf 1 reduziert werden.

Für festes $j \geq 2$ und $d_j \neq 1$ schreiben wir wie in Gleichung (7) auf Seite 79 $d = \prod_{k=1}^m d_k^k = uv^j$ mit $v := d_j$ und $u := \frac{d}{v^j} = \prod_{k \neq j} d_k^k$ und setzen Polynome b_i und c_i an, so daß

$$\sum_{i=0}^{n-1} \frac{a_i w_i}{uv^j} = \sum_{i=0}^{n-1} D \left(\frac{b_i w_i}{v^{j-1}} \right) + \sum_{i=0}^{n-1} \frac{c_i w_i}{uv^{j-1}} \quad (23)$$

gilt. Durch Betrachtung der Nenner von Dw_i muß man hier allerdings noch fordern, daß der größte gemeinsame Teiler e dieser Nenner, – der immer quadratfrei ist, – von Anfang zusätzlich uv teilt, d.h. gegenfalls muß entsprechend erweitert werden. Damit gibt es Polynome m_{ik} in x mit $eDw_i = \sum_{k=0}^{n-1} m_{ik} w_k$ und ein Polynom s mit $uv = se$. Nach Ableiten, Multiplikation mit uv^j und Betrachten der Gleichung modulo v erhält man

$$\sum_{i=0}^{n-1} a_i w_i = \sum_{i=0}^{n-1} b_i uv^j D \left(\frac{w_i}{v^j} \right).$$

B.M. TRAGER hat nun in Abschnitt 4.2 in [67] gezeigt, daß $(b_i uv^j D(\frac{w_i}{v^j}))_{0 \leq i < n}$ eine Ganzheitsbasis des $k[x]_v$ -Moduls $k(x, y)$ ist und damit ist garantiert, daß diese Gleichung immer eindeutig gelöst werden kann. Dies geschieht durch Lösen des linearen Gleichungssystems in den b_i

$$a_i = (1-j)u D v b_i + t \sum_{k=0}^{n-1} b_k m_{ki}$$

modulo v .

Durch Iteration des Verfahrens kann man erreichen, daß $f = Dg + h$ gilt, wobei die algebraische Funktion bzgl. der Basis w dargestellt und in normierter Form einen quadratfreien Nenner besitzt. Nun kann es allerdings passieren, daß h eine Polstelle im Unendlichen besitzt. Man erinnere sich, daß wir in Abschnitt 6.5 f so modifiziert

²¹Die Diskriminante bestimmt die Verzweigungspunkte!

hatten, daß für f das nicht gilt. Falls das passiert, ist f nicht elementar integrierbar, siehe Abschnitt 4.3 in [67].

6.7 Der logarithmische Anteil

Andernfalls ist $\int h$ zu bestimmen, wobei $h = \frac{\sum_{i=0}^{n-1} a_i w_i}{d}$ ist mit quadratfreiem d und die Pole von h durch die Nullstellen von d bestimmt. Wir können die Nenner der Basis w zu $e \in k[x]$ zusammenfassen und den Zähler $\sum_{i=0}^{n-1} a_i w_i = \frac{b}{e}$ mit $b \in k[x, y]$ schreiben. Wie im Fall rationalen Funktionen sind nun die Residuen der Polstellen zu bestimmen. Das zur Stelle (x', y') gehörende Residuum c der Funktion $h := \frac{b}{de}$ mit ausschließlich durch d bestimmten einfachen Polstellen ist bis auf einen durch das Verzweigungsverhalten bestimmten ganzzahligen Faktor gleich dem Wert von $\frac{b}{eDd}$ an der Stelle (x', y') , siehe dazu das Theorem auf Seite 67 in [67]. Es wird dort auch gezeigt, daß ohne Einschränkung angenommen werden kann, daß keine Verzweigung an Polstellen auftreten. Das Residuum c einer Polstelle (x', y') ist also eine Nullstelle des Polynoms $b(x', y') - ze(x')Dd(x') \in k[z]$. Zusätzlich muß natürlich immer $F(x', y') = 0$ gelten. Das heißt, die Residuen c sind Nullstellen der Resultante $\text{res}_y(b - zeDd, F)$, die wir anschließend noch durch den größten gemeinsamen Teiler der polynomialen Koeffizienten in x teilen: prim_z . Da die Polstellen von h durch die Nullstellen von d bestimmt sind, ergibt sich schließlich die ROTHSTEIN-TRAGER-Resultante

$$r := \text{res}_x(\text{prim}_z(\text{res}_y(b - zeDd, F)), d),$$

deren Nullstellen die Residuen sind. Im Zerfällungskörper sind die Nullstellen α_i von r zu bestimmen und dann eine Basis (q_1, \dots, q_s) des \mathbb{Z} -Moduls, den sie erzeugen. Dann gibt es ganze Zahlen α_{ij} mit

$$\alpha_i = \sum_{j=1}^s \alpha_{ij} q_j.$$

Sei P_i die Stelle mit Residuum α_i , dann kann damit für jedes j der Divisor

$$D_j = \sum_i \alpha_{ij} P_i$$

definiert werden, zu dem – der schwierigste Teil! – eine algebraische Funktion mit den dadurch bestimmten Ordnungen α_{ij} der Pol- und der Nullstellen gefunden werden muß. Genauer, es genügt eine rationale Funktion u_j in x und y mit Koeffizienten in einer algebraische Erweiterung von k in diesem Sinne für ein ganzzahliges Vielfaches $n_j D_j$ zu finden. Gelingt das für alle j , dann ist

$$\int h = \sum_{j=1}^s \frac{q_j}{n_j} \log(u_j).$$

Andernfalls ist h und damit f nicht elementar integrierbar. Die Beschränkung dieser Vielfachen gelingt mit der Methode von B.M. TRAGER durch Reduktionen des Problems auf endliche Körper!

Betrachten wir nun die Beispiele f_{\pm} aus Abschnitt 2.2. Wir haben $F(x, y) = y^2 - x^3 - 1$ und die Ganzheitsbasis $(1, y)$. Die Normalform von f_+ ist demnach $\frac{y}{(x-2)(x^2-x+1)}$ mit quadratfreiem Nenner d , dessen Ableitung gleich $3x^2 - 6x + 3$ ist. In der Notation von oben ist $b = y$ und $e = 1$. Die x -Werte der Polstellen von f_+ sind also $\left[\frac{-\sqrt{-3}+1}{2}, \frac{\sqrt{-3}+1}{2}, 2 \right]$, die zugehörigen y -Werte sind $[0, 0, \pm 3]$. Die Resultante

$$ry_+ := \text{res}_y(b - zeDd, F) = \text{res}_y(y - z(3x^2 - 6x + 3), y^2 - x^3 - 1) =$$

$$(9x^4 - 36x^3 + 54x^2 - 36x + 9)z^2 - x^3 - 1$$

hat teilerfremde Koeffizienten. Damit sind die Residuen die Nullstellen von $r_+ := \text{res}_x(ry_+, d) = 729z^4(z^2 - 1)$. Eine \mathbb{Z} -Basis des von den Nullstellen $\alpha = (0, 1, -1)$ von r_+ erzeugten \mathbb{Z} -Moduls ist $q := (1)$. Damit ist der einzige zu betrachtende Divisor $D_1 = P_1 - P_2$, falls mit P_1 die durch $(2, 3)$ und mit P_2 die durch $(2, -3)$ bestimmten Polstellen mit Residuum 1 bzw. -1 bezeichnet sind. Man muß sich nun überlegen, daß es keine Funktionen u in $\mathbb{Q}(x, y)$ geben kann, die als Divisoren $(P_1 - P_2)$ oder $2(P_1 - P_2)$ oder besitzen. Erst der Divisor $3(P_1 - P_2)$ hat mit $u := \frac{x^3 - 6x^2 + 12x - 8}{(6x+6)y + x^3 + 12x^2 - 6x + 10}$ eine Realisierung als Funktion, die genau eine dreifache Nullstelle P_1 und genau eine dreifache Polstelle P_2 besitzt. Also ist $u^{\frac{1}{3}}$ eine geeignete Funktion und damit ist das Integral $\int f_+ dx = \frac{1}{3u} \log(u)$ bestimmt.

7 Ausblick

7.1 Übersichtsarbeiten

Weitere Arbeiten mit Übersichtscharakter oder mit didaktischer Aufbereitung des Stoffes sind und [35] von T. KASPER, der einen historischer Abriss der Entwicklungen bis Ende der 1970er Jahre gibt, [4] von M. BRONSTEIN sowie [5] von M. BRONSTEIN, J.H. DAVENPORT, B.M. TRAGER. Eine Aufbereitung des Stoffes für die Leser des Spektrums der Wissenschaft hat R. KRAUME in [42] durchgeführt.

7.2 Spezielle Funktionen

Hier kann sind zwei verschiedene Dinge zu unterscheiden. Zum einen kann man eine Erweiterung der Klasse \mathcal{F} der zu integrierende Funktionen durch $\text{er}, \text{li}, \text{ei}, l_2, \int \frac{y}{x}$ etc. zulassen, zum anderen natürlich eine Erweiterung der Klasse \mathcal{G} der Funktionen, die für die Darstellung der Integrale zugelassen sind. Zu den Resultaten in dieser Hinsicht zitieren wir aus der Kurzübersicht von M. BRONSTEIN in [13]:

Erweiterungen des Risch-Algorithmus mit speziellen Funktionen wurden zuerst von Cherry [14, 15] betrachtet. In diesen Arbeiten dürfen sie im Ergebnis auftreten, bei Knowles [38, 39] auch im Integrand. Obgleich hier spezielle Funktionen erlaubt sind, gestattet weder Cherrys noch Knowles' Algorithmus *gemischte* Funktionen, d.h. zugleich algebraische und transzendente Funktionen im Integrand. Bronstein [9] erweiterte Tragers Algorithmus für den Fall gemischter elementarer Funktionen und Weileder [69] gab einen Algorithmus an, der gemischte elementare Funktionen behandelt und noch spezielle Funktionen im Ergebnis zuläßt. Während der Fall gemischter Funktionen des Algorithmus von Bronstein teilweise in AXIOM implementiert ist, ist das allgemeine Problem der effizienten Behandlung gemischter Integranden offen.

Jüngste Arbeiten zu diesem Themenkreis – auch mit Beiträgen zur weiteren Verbesserung der Algorithmen – sind die Doktorarbeiten von [1] von J. BADDOURA und [7] von L. BERTRAND.

7.3 Definite Integration

Wie wir schon im Abschnitt 3.10 gesehen haben, treten hier noch schwierigere Probleme auf. Schon bei der definiten Integration von rationalen Funktionen muß man sehr aufpassen, damit man keine unstetigen Stammfunktionen auswählt für die die Voraussetzungen des Hauptsatzes der Differential- und Integralrechnung natürlich nicht

gegeben sind. Es gibt keine geschlossene Theorie zur definiten Integration. Aus diesem Grund läßt AXIOM als Integrationsbereich den Parameter $x=a..b$ im wesentlichen nur bei rationalen Funktionen zu und warnt, falls a oder b keine konkreten numerischen Werte sind, mit `potentialPole`, so daß man als drittes Argument eigenverantwortlich `noPole` angeben muß! In einzelnen Systemen wie MAPLE oder MACSYMA sind Spezialalgorithmen und Heuristiken für einige Teilklassen von Funktionen implementiert ([27]).

Eine gewisse Hoffnung zu weiteren Fortschritten bei der symbolischen Lösung von definiten Integralen bieten die Ideen der neueren Methoden von D. ZEILBERGER für das diskrete Analogon der definiten Summation. Ein guter Übersichtsartikel zu diesem Themenkreis von P. PAULE und V. STREHL ist in [51].

7.4 Symbolisches Lösen von Differentialgleichungen

Diese Fragestellung ist natürlich der nächste logische Schritt. Nachdem die einfachste Differentialgleichung, das Bestimmen eines Integrals, schon solche Probleme bereitet, ist es sicher nur allzu verständlich, daß auch dies nicht einfach sein wird. Allerdings läßt man hier sinnvollerweise sogenannte LIOUVILLESche Funktionen als Lösungen zu. M.F. SINGER hat hier wichtige Beiträge geliefert, insbesondere eine prinzipielle Lösung für lineare gewöhnliche Differentialgleichungen – ähnlich wie die Lösung von RISCH für die symbolische Integration, siehe etwa [61], [62] oder [63]. Derzeit ist man dabei für kleine Ordnungen – für die Ordnung 2 hat J.J. KOVACIC in [41] einen Algorithmus angegeben – effektive Methoden zu entwickeln. Methoden der Gruppentheorie spielen hier eine wichtige Rolle. Bei partiellen Differentialgleichungen versucht man mit der Symmetrieanalyse, die auf S. LIE zurückgeht, die Gleichungen zu behandeln. Eine erste Orientierung zu diesen Themen kann man sich im Beitrag auf den Seiten 61–69 von F. OLLIVIER et al. in [13] verschaffen. In der schon oben erwähnten Ausgabe des Spektrum der Wissenschaft gibt es auch dazu Beiträge: [26] und [60].

8 Dank

Der Autor dankt M. Bronstein für die Überlassung einer Vorversion seiner Monographie [10], die er für seine Vorlesung im Wintersemester 93/94 an der Universität Heidelberg nutzen konnte und die auch diese Übersichtsarbeit stark beeinflusst hat. Zugleich sind viele Ideen des Hauptvortrages *Recent Advances in Symbolic Indefinite Integration*, den M. Bronstein in der Sektion Computeralgebra auf der DMV-Jahrestagung 1995 in Ulm gehalten hat, eingegangen.

Literatur

- [1] J. Baddoura. *Integration in Finite Terms with Elementary Functions and Dilogarithms*. Ph.D. Thesis, MIT, Mathematics, 1994.
- [2] J. Baddoura. A Conjecture on Integration in Finite Terms with Elementary Functions and Dilogarithms. Proceedings of ISSAC'94, ACM Press, 158–162.
- [3] Th. Becker, V. Weispfenning; in cooperation with H. Kredel. *Gröbner Bases, A Computational Approach to Commutative Algebra*, volume 141 of Graduate Texts in Mathematics. New York-Berlin-Heidelberg, Springer-Verlag, 1993.
- [4] M. Bronstein. Symbolic Integration: towards Practical Algorithms. In E. Tournier (Hrsg.) *Computer Algebra and differential equations.*, Acad. Press, 59–85, 1989.

- [5] M. Bronstein, J.H. Davenport, B.M. Trager. Symbolic Integration ist Algorithmic! Mini Course Lecture Notes at the conference Computers & Mathematics, M.I.T., Boston, 1989.
- [6] L. Bertrand. Computing a Hyperelliptic Integral using Arithmetic in the Jacobian of the Curve. *Applicable Algebra in Engineering, Communication and Computing*, **6**, 275–298, 1995.
- [7] L. Bertrand. *Calcul Symbolique des Intégrales Hyperelliptiques*. Thèse de Doctorat de l'Université de Limoges, Mathématiques, 1994.
- [8] M. Bronstein. The transcendental Risch differential equation. *Journal of Symb. Comp.*, **9**, 49–60, 1990.
- [9] M. Bronstein. On the integration of elementary functions. *Journal of Symb. Comp.*, **9**, 117–173, 1990.
- [10] M. Bronstein. *Symbolic Integration, Part I Transcendental Functions*. Springer-Verlag, Heidelberg - Wien - New York, 1996
- [11] B. Buchberger, G.E. Collins, and R. Loos (Ed.). *Computer Algebra, Symbolic and Algebraic Computation*. Springer-Verlag, Wien - New York, second edition, 1983.
- [12] B.F. Caviness, Integration in Finite Terms. Vortragsfolien Computeralgebra-Sommerschule RISC Linz 1993.
- [13] Fachgruppe Computeralgebra der GI, DMV, GAMM, V. Weispfenning, J. Grabmeier (Hrsg.). *Computeralgebra in Deutschland – Bestandsaufnahme, Möglichkeiten, Perspektiven*. Passau und Heidelberg, 1993.
- [14] G.W. Cherry. Integration in finite terms with special functions: the error function. *Journal of Symbolic Computation*, **1**(3):283–302, 1985.
- [15] G.W. Cherry. Integration in finite terms with special functions: the logarithmic integral. *SIAM Jour. Comp.*, **15**:1–21, 1986.
- [16] J.H. Davenport. On the parallel Risch algorithm (iii): Use of tangents. *ACM SIGSAM Bulletin*, **16**(3):3–6, 1982.
- [17] J.H. Davenport. *On the Integration of Algebraic Functions*. Lectures Notes in Computer Science No. 102. Springer-Verlag, Berlin-Heidelberg-New York, 1981.
- [18] J.H. Davenport. On the Parallel Risch Algorithm (I). Proc. EUROCAL'83, Springer Lect. Notes in Comp. Sc. 162, pp. 1–12, Springer-Verlag, Berlin-Heidelberg-New York, 1983.
- [19] J.H. Davenport, B.M. Trager On the Parallel Risch Algorithm (II). *ACM Trans. Math. Software* **11** (1985) pp. 356–362.
- [20] J.H. Davenport. The Risch Differential Equation Problem. *SIAM J. Comp.* **15**, (1986), No. 4, 903–916.
- [21] J.H. Davenport. Integration in Closed Form. preprint.
- [22] J.H. Davenport. Vortrag 1987 an der Universität Heidelberg.
- [23] J.H. Davenport, Y. Siret, E. Tournier. *Computer Algebra: Systems and Algorithms for Algebraic Computation*. Academic Press, London, 1989.
- [24] K.O. Geddes, S.R. Czapora, and G. Labahn. *Algorithms for Computer Algebra*. Kluwer Academic Publishers, Boston, 1992.
- [25] J. Fleischer, J. Grabmeier, F. Hehl, W. Küchlin, W. (Hrsg.). *Computer Algebra in Science and Engineering, Conf. Proc. ZiF Bielefeld 1994*. Singapore: World Scientific, 1995.

- [26] B. Fuchssteiner. Symmetrien bei partiellen Differentialgleichungen – ein Anwendungsfeld der Computeralgebra. *Spektrum der Wissenschaft* **3** (1996), S. 102–104.
- [27] K.O. Geddes, T.C. Scott. Recipes for Classes of Definite Integrals Involving Exponentials and Logarithms. In E. Kaltofen, S.M. Watt (Ed.) *Computer and Mathematics*. Springer-Verlag, New York-Heidelberg-Berlin, 192–202, 1989.
- [28] J. Grabmeier. Computeralgebra – eine Säule des Wissenschaftlichen Rechnens. *it+ti*, **6** (1995), S. 5–20.
- [29] G.H. Hardy. *The Integration of Functions of a Single Variable*. Cambridge Tracts in Mathematics and Mathematical Physics. No. 2, Cambridge University Press, 1905.
- [30] G.H. Hardy. *The Integration of Functions of a Single Variable, Second Edition*. Cambridge Tracts in Mathematics and Mathematical Physics. No. 2, Cambridge University Press, 1916.
- [31] E. Hermite. Sur l'integration des fractions rationnelles. *Nouvelles Annales de Mathématiques*, 2^{eme} Série 11, (1872), 145–148.
- [32] E. Horowitz. Algorithms for partial fraction decomposition and rational function integration. in S.R. Petrick (Ed.) *Proc. SYMSAM '71*, ACM Press, 441–457, 1971.
- [33] E. Kaltofen. A Note on the Risch Differential Equation. 359-366 in *Proc. EURO-SAM'84*, Lect. Notes in Computer Science 174, Hrsg. J. Fitch, Springer-Verlag, Springer-Verlag, Berlin-Heidelberg-New York, 1984.
- [34] I. Kaplansky *An Introduction to Differential Algebra*. Publ. de l'inst. de math. de l'univ. de Nancago, Hermann, Paris, 1957.
- [35] T. Kasper *Integration in Finite Terms: The Liouville Theory*. *Mathematics Magazine*, Vol. 53, No. 4, 1980. The Mathematical Association of America.
- [36] M. Klerer, F. Grossman. Error Rates in Tables of Indefinite Integrals. *Industrial Math.* 18(1968) pp. 31-62.
- [37] M. Klerer, F. Grossman. *A new Table of Indefinite Integrals Computer Processed*. Dover, 1971.
- [38] P.H. Knowles. Integration of a class of transcendental Liouvillian functions with error-functions, Part I. *Journal of Symbolic Computation*, 13(5):525–543, 1992.
- [39] P.H. Knowles. Integration of a class of transcendental Liouvillian functions with error-functions, Part II. *Journal of Symbolic Computation*, **16**, 227–241, 1993.
- [40] E.R. Kolchin. *Differentialalgebra and algebraic groups*. 1973.
- [41] J.J. Kovacic. An algorithm for solving second order linear differential equations. *Journal of Symb. Comp.*, 2(3):3–43, 1986.
- [42] R. Kraume. Symbolische Integration. *Spektrum der Wissenschaft* **3** (1996), S. 95–98.
- [43] D. Lazard und R. Rioboo, Integration of Rational Functions: Rational Computation of the Logarithmic Part. *Journal of Symbolic Computation*, **9**, 113–116, 1990.
- [44] J. Lützen. *Joseph Liouville. 1809-1882: master of pure and applied mathematics.*, Springer-Verlag New York, 1990.
- [45] A. Heck. *Introduction to Maple*. Abschnitt 10 über Integration, pp. 193-226, Springer-Verlag, New York-Berlin-Heidelberg, 1993.
- [46] D. Mack. *On Rational Integration*, UCP-38, Dpt. of Computer Science, University of Utah, 1975.

- [47] J. Moses. Symbolic Integration: The Stormy Decade. Comm. of the ACM, August 1971, Vol.14, N. 8, 548–560. 881-892(1991).
- [48] A.C.Norman, P.M.A. Moore. *Implementing the New Risch Integration Algorithm*. Proc. Int. Symp. on Advanced Computing Methods in Theoretical Physics, Saint-Maximin, 1977.
- [49] M.W. Ostrogradsky. *De l'intégration des fractions rationnelles*, Bulletin de l'Académie Impériale des Sciences de St. Pétersbourg, class phys-math. 4, 145–167, 286–300, 1846.
- [50] A.M. Ostrowski. Sur l'intégrabilité élémentaire de quelques classes d'expressions. Commentarii Mathematici Helvetici 18, 283–308, 1946.
- [51] P. Paule, V. Strehl. Symbolic summation – some recent developments. In: [25], S. 138–162, 1995.
- [52] D. Richardson. *Some Unsolvable Problems Involving Elementary Functions of a Real Variable*. J. Symbolic Logic 33 (1968), pp. 514–520.
- [53] R.H. Risch. The Problem of Integration in Finite Terms. Transactions of the American Mathematical Society 139, (1969), 167–189.
- [54] R.H. Risch. The Solution of the Problem of Integration in Finite Terms. Bulletin of the American Mathematical Society 76, 605–608.
- [55] R. Rioboo. *Quelques aspects du calcul exact des nombres réels*. Thèse de Doctorat de l'Université Paris 6, spécialité Informatique. Paris, 1991.
- [56] J.F. Ritt. *Integration in Finite Terms – Liouville's Theory of Elementary Methods*. Columbia University Press, New York, 1948.
- [57] M. Rosenlicht. Liouville's Theorem on functions with elementary integrals, Pacific J. Math. 24 (1968), 153–161.
- [58] M. Rosenlicht. Integration in finite terms. American Mathematical Monthly 79, 963–972, 1972.
- [59] M. Rothstein. A new algorithm for the integration of exponential and logarithmic functions. In *Proceedings of the 1977 MACSYMA Users' Conference, held at Berkeley, Calif., July 27–29, 1977*, number NASA CP 2012, pages 263–274. NASA Scientific and Technical Information Service, Washington, D.C., 1977.
- [60] F. Schwarz. Differentialgleichungen lösen mit Computeralgebra. Spektrum der Wissenschaft 3 (1996), S. 98–102.
- [61] M.F. Singer. Liouvillian first integrals of differential equations. Transactions of the American Mathematical Society. 333, 673–687, 1977.
- [62] M.F. Singer. Formal solutions of differential equations. Journal of Symb. Comp., 10(1):59–95, 1990.
- [63] M.F. Singer. An outline of differential Galois theory. In E. Tournier (Hrsg.) *Computer Algebra and differential equations*, Acad. Press, 1989.
- [64] M. Singer, B. Saunders, and B.F. Caviness. An Extension of Liouville's Theorem on Integration in Finite Terms. SIAM Journal of Computing, 966–990, 1985.
- [65] J. Slagle. *A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus*. Ph.D. Dissertation, Harvard U., Cambridge, Mass. May 1961.
- [66] B.M. Trager. Algebraic Factoring and Rational Function Integration. in R.D. Jenks (Ed.) Proc. SYMSAC '76, 219–226.
- [67] B.M. Trager. *Integration of Algebraic Functions*. Ph.D.-thesis. M.I.T., Boston, 1984.

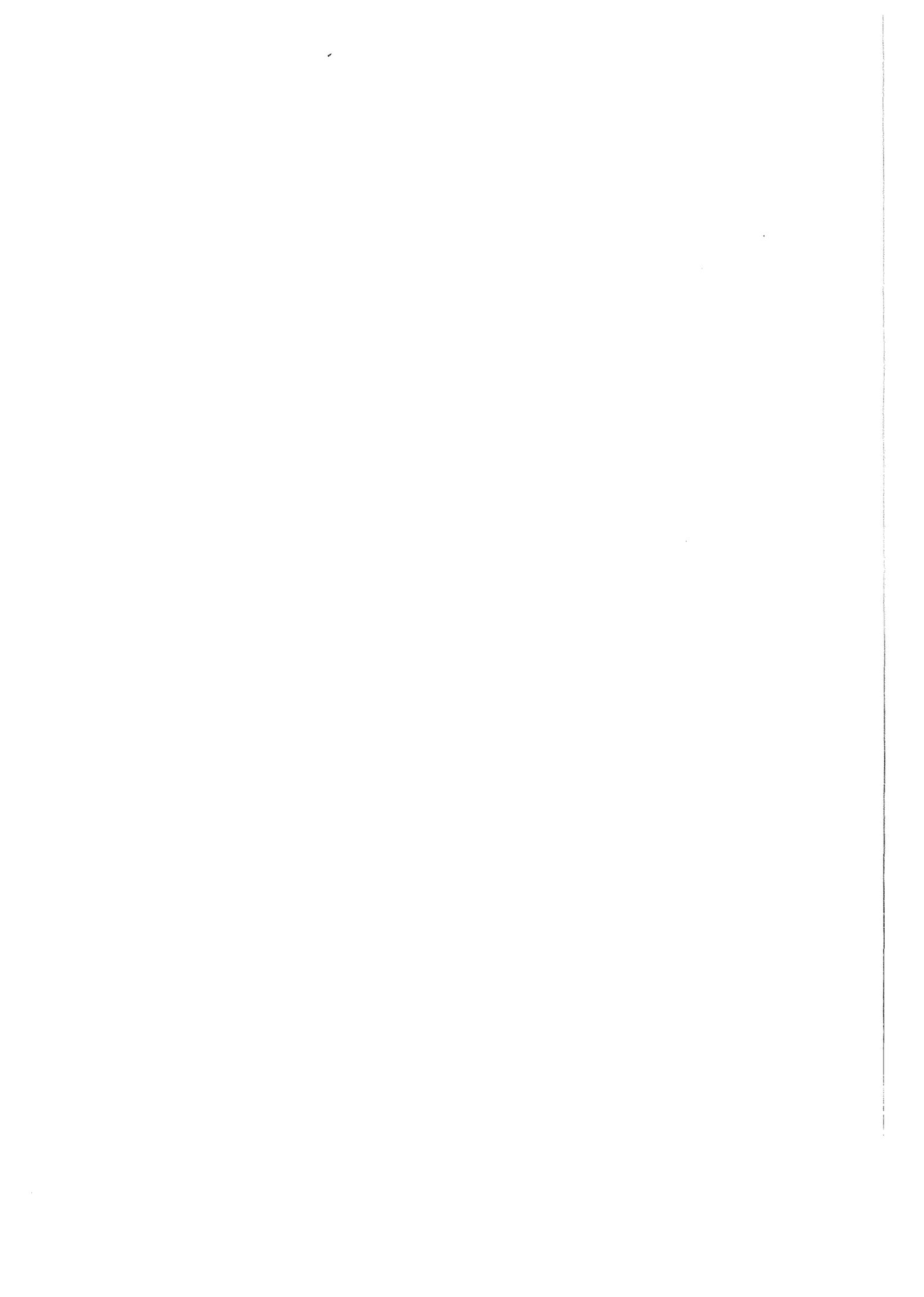
- [68] J.A. Weil. *Constantes et polynômes de Darboux en algèbre différentielle: application aux systèmes différentiels linéaires*. Thèse de Mathématiques, Ecole Polytechnique, 1996.
- [69] M. Weileder. *Integration elementarer Funktionen durch elementare Funktionen, Errorfunktionen und logarithmische Integrale*. Dissertation, Fachbereich Mathematik der Ludwig-Maximilians-Universität München, München, 1990.
- [70] V. Weispfenning. *Differentialalgebra und Symbolische Integration*. Vorlesungsmanuscript Univ. Passau, 1992.
- [71] V. Weispfenning. *Solving parametric polynomials equations and inequalities by symbolic algorithms*. In: [25], S. 163–182, 1995.





Prof. Dr. Hartmut Gemmeke

Prof. Dr. Hartmut Gemmeke studiert von 1964 bis 1969 Physik an den Universitäten Göttingen und Heidelberg. 1971 promovierte er und war anschließend bis 1979 wissenschaftlicher Assistent an der Universität Heidelberg. Dort habilitierte er auch mit der Arbeit „Projektilaufbruch an schweren Kernen“. 1980 wechselte er an die Universität Marburg, von wo aus er die Neutronenemission bei Schwerionen Reaktionen und „Hochangeregte Symmetriezustände in 8Be “ am MPI Heidelberg, der GSI und dem HMI untersuchte. 1982 kam er ins Forschungszentrum Karlsruhe und wurde am Institut für Kernphysik I mit der Entwicklung der physikalischen Meßtechnik eines 56 t Szintillations-Kalorimeters betraut. Weiter gehörte die Simulation und Inbetriebnahme des Neutrino-Experimentes KARMEN am Rutherford Appleton Laboratory (1987-89) zu seinen Aufgaben. 1991 wurde er zum Professor an der Universität Heidelberg berufen, seit 1993 ist er Leiter der Hauptabteilung für Prozeßdatenverarbeitung und Elektronik im Forschungszentrum Karlsruhe. Mit den Themen Fuzzy Logik und Neuronale Netze befaßt er sich in Praxis und Vorlesungen seit 1991.



Fuzzy Logik und ihre Anwendung

Hartmut Gemmeke

Hauptabteilung Prozeßdatenverarbeitung und Elektronik, Forschungszentrum Karlsruhe

Zusammenfassung

Es wird in die Grundlagen der Fuzzy Logik und ihre Anwendungen eingeführt. Die entscheidenden theoretischen Grundlagen wie unscharfe Mengen und ihre Repräsentation durch linguistische Variablen, sowie die unscharfe Logik mit plausiblen Schluffweisen (fuzzy Inferenz) werden erläutert. Die Hauptanwendungsgebiete wie Fuzzy Regelung, Expertensysteme und Mustererkennung werden anhand von Beispielen dargestellt.

Einleitung

Was ist fuzzy? Mein Wörterbuch interpretiert "fuzzy" als unscharf, verschwommen, verwischt oder auch fusselig. Wozu benötigen wir Unschärfe in der Beschreibung von Modellen, Daten, Begriffen und in der Logik? In unserer modernen abendländischen Kultur wird ein unsicherer Erkenntnisprozess oder eine verschwommene Beschreibung als unlogisch oder unwissenschaftlich abgetan oder sogar verunglimpft. Schaut man dem Wissenschaftler oder Ingenieur aber aufs Maul, so hört man nicht selten im Zusammenhang mit Messungen und Daten von Fehlern, Begrenzungen durch die Auflösungsfunktion des Detektors und unvollständigen Messungen. Bei der Modell-Bildung redet man von vagen Konzepten, heuristischen Annahmen bis hin zu groben Verallgemeinerungen. Die Auswertung und der Erkenntnisprozess wird von intuitiven Schlüssen und Vermutungen sowie groben Abschätzungen begleitet. Wir werden darin von Albert Einstein bestärkt: "Keep it simple - but only simple enough to work".

Woher kommt nun diese Unschärfe? Bei den Messungen ist die Unschärfe bedingt aus statistischen und systematischen Fehlern, den Grenzen der Auflösung des physikalischen Nachweisgerätes und der Unvollständigkeit vieler Messungen. Im allg. wird z.B. aus der Zahl der angezeigten Stellen auf elektrischen Meßinstrumenten eine überhöhte Erwartung an die mögliche Meßgenauigkeit gestellt. Unschärfe ist auch in der physikalischen Meßtechnik etwas sehr Natürliches. Zudem können die Daten oder Ereignisse zufällig verteilt sein, d.h. einer stochastisch gegebenen Verteilung folgen. Schließlich werden die Messungen, Erkenntnisse und Beschreibungen von Menschen gemacht und verfaßt. Sie sind oft unvollständig, idealisierend oder stark vereinfachend. Daraus ergibt sich noch eine linguistische Unschärfe. Insgesamt haben wir damit drei Unschärfen: die Daten- oder informationelle, die zufällige und die linguistische Unschärfe. Trotz all dieser Unsicherheiten können wir deterministische Modelle bilden, aber auch stochastische, heuristische oder fuzzy Modelle. Was gerade die unscharfe (fuzzy) Modellbildung uns einbringen kann, soll mit diesem Vortrag näher gebracht werden. Brauchbare Modelle wird man danach beurteilen, ob sie mathematisch vielfältig genug sind, physikalisch möglichst einfach die Naturgesetze wiedergeben und mit einer Rechnung z.B. im Computer leicht nachvollziehbar sind.

Theorie unscharfer Mengen (Fuzzy Sets)

Die Theorie unscharfer Mengen oder Begriffe hat im wesentlichen zwei Komponenten, die ihre Unschärfe verursachen. Zum einen die Unschärfe der natürlichen Sprache oder der Linguistik. Zum anderen sind die Regeln, mit denen die Eingangs- und Ausgangsvariablen verknüpft werden, verschwommen und/oder mehrdeutig. Das läßt sich leicht an ein paar Beispielen illustrieren.

Beispiel 1.1

- Der klinische Schweregrad III beim Bluthochdruck kann aus medizinischer Sicht folgendermaßen beschrieben werden: "Systolischer Blutdruck höher als 200 Torr oder diastolischer Blutdruck zwischen 120 und 130 Torr. *Meistens* stabiles Blutdruckniveau. *Sehr oft* Anzeichen kardialer Insuffizienz, Augenhintergrundveränderungen, *stärkere* Beschwerden, *teilweise* zerebrale Störungen" (siehe H. Bothe 1993). Die kursiv hervorgehobenen Textab-

schnitte weisen auf die unscharfen (fuzzy) Bewertungen hin, die eine flexible, intelligente Reaktion des Arztes ermöglichen und erfordern.

Ähnlich werden wir die Hilfe eines Außenstehenden beim Einparken eher mit unscharfen Argumenten akzeptieren:

Beispiel 1.2:

unscharfe

das Lenkrad *stark* nach rechts ein schlagen,
das Auto *ca.* 1 m zurück setzen,
das Lenkrad *stark* nach links ein schlagen
Wenn wir uns dem geparkten Auto *nähern*,
das Auto *langsam* abbremsen

exakte Anweisungen beim Einparken

das Lenkrad 30° 10' nach rechts
76,3 cm zurückfahren
das Lenkrad 40° 20' nach links
mit der Bremsbeschleunigung = 3,0 m/s² nach
1,76 s abbremsen

Mit den exakten Anweisungen wird man nicht viel anfangen können, da die relative Genauigkeit unserer menschlichen Datenerfassung nicht ausreicht. Ebenso formulieren wir die Regeln für die Verknüpfung (kursiv hervorgehoben), siehe Beispiel 1.2, mit unscharfen Argumenten. Trotzdem gelingt es uns im allgemeinen, mit unpräzisen, vagen Daten und Regeln reale komplizierte Aktionen wie das Einparken eines Autos in eine Parklücke zu bewältigen. Die gleiche Aufgabe läßt sich mit einem streng deterministischen Programm in einem Computer nicht so leicht realisieren, und noch weniger leicht programmieren. Damit erweist sich das neuronale Netzwerk in unserem Gehirn als ein effektives Werkzeug, um die unpräzisen Anweisungen in einem funktionierenden dynamischen Ablauf umzusetzen - ohne daß wir erst ein Modell für den Prozeß entwerfen müssen. Unser Gehirn lernt mit "trial and error" oder einem Lehrer, der uns anleitet und ist in der Verarbeitung von Fuzzy Regeln dem größten Computer überlegen.

Indem wir unscharfe Mengenbeschreibungen und Schlußweisen benutzen, um reale Aktionen auszulösen, können wir quasi modellfrei komplexe dynamische Prozesse kontrollieren oder Situationen klassifizieren. Dadurch wird die Kopplung zwischen Mensch und Computer auch in komplizierten Situationen einfacher und für den Benutzer transparenter. Die Ziele für die Anwendung von fuzzy Mengen und Regeln lassen sich in drei Hauptgruppen einteilen:

1. Regelungsaufgaben, insbesondere nichtlineare Regelung, wo das Modell des zu regelnden Prozesses nicht bekannt oder sehr kompliziert ist, bzw. sich zeitlich ändern kann und eine adaptive Regelung erfordert.
2. Mustererkennung und/oder Datenfilter, d.h. es geht um die Merkmalsextraktion aus zumeist verrauschten Meßdaten und die Clusterbildung und Klassifikation mit fuzzy Regeln. Typische Anwendungen sind z.B. die Kontrastverstärkung von Ultraschallaufnahmen von Föten im Mutterleib oder die Aufarbeitung von komplizierten von der Temperatur und dem Luftdruck abhängigen Sensorsignalen oder zur Identifikation von spezifischen Stickstoffverbindungen bei der Sicherheitskontrolle von Gepäckstücken im Flugverkehr.
3. Unscharfe regelbasierte Expertensysteme, wie z.B. die Analyse der Kreditwürdigkeit von Bankkunden oder Diagnostik in der Medizin.

Hier sollen die ersten beiden Aufgabenklassen besonders interessieren. Dabei wird der Begriff der Fuzzy Logik nicht im engeren Sinne für die Modellierung der menschlichen Beweis- und Entscheidungsfindung sowie der biologischen neuronalen Prozesse im Gehirn verwendet, sondern es wird mit Fuzzy Logik ein für viele technische und physikalische Probleme neues Handwerkszeug gesucht. Die Bedeutung der Fuzzy Logik liegt darin, daß

- viele natürliche Mengen unscharf begrenzt sind,
- die Gesetze der klassischen Mengenlehre und Logik benutzt werden können, um die Theorie unscharfer Mengen herzuleiten.
- Fuzzy Logik parallel abgearbeitet und damit mit hoher Rechenleistung installiert werden kann.

Im Folgenden sollen die theoretischen Grundlagen der Fuzzy Logik plausibel gemacht werden. Es wird dabei auf die Definition unscharfer Mengen und ihrer Representation durch linguistische Variablen, bzw. Terme erläutert, sowie die unscharfe Logik mit ihrem Kern, dem plausi-

blem Schließen, an Beispielen beschrieben. Schließlich wird auf die drei Hauptanwendungsgebiete Fuzzy Expertensysteme, Regelung und Mustererkennung mit Beispielen eingegangen. Eine Literaturliste ergänzt die Darstellung. Das dort aufgeführte Buch von Herrn Bothe kann ich als gut verständliche Einführung in die Thematik empfehlen.

Literatursammlung zu Fuzzy Logik

Hartmut Gemmeke

Juni 1995

Literatur zu Fuzzy Logik

- H. Bandemer and W. Nather, "Fuzzy Data Analysis", Kluwer Academic Publishers, Dordrecht, 1992
- H. Bothe, "Fuzzy Logik", Springer-Verlag, Berlin, 1993, (E)
- D.I. Brubaker, "Introduction to Fuzzy Logic Systems", The Huntington Group, 883 Santa Cruze Avenue, Suite 27, Menlo Park, CA 94025-4669, 1991, (E)
- D. Driankov, H. Hellendoorn and M. Reinfrank, "An Introduction to FuzzyControl", Springer-Verlag, Berlin, 1993
- D. Driankov et al., Fuzzy Logic and Fuzzy Control, Lecture Notes in AI, Springer, Berlin 1991
- D. Dubois and H. Prade, "Fuzzy Sets and Systems: Theory and Applications", Academic Press, 1980
- D. Dubois, H. Prade and R. Yager (Eds.), "Readings in Fuzzy Sets for Intelligent Systems", Morgan Kaufmann, San Mateo, 1993
- M. Jamshidi, N. Vadiie and T. Ross (Eds.), "Fuzzy Logic and Control", Prentice Hall, Englewood Cliffs, NJ, 1993
- J. Kahlert, H. Frank, "Fuzzy-Logik und Fuzzy-Control, Braunschweig, Vieweg, 1993
- A. Kandel and G. Langholz (Eds.), "Fuzzy Control Systems", CRC Press, Boca Raton, 1994
- E.E. Kerre (ed.), "Introduction to the Basic Principles of Fuzzy Set Theory and Its Applications", Communication and Cognition, Brussels, 1991
- G.J. Klir, T.A. Folger, "Fuzzy Sets, Uncertainty, and Information", Prentice Hall, Englewood Cliffs, NJ, 1992
- R. Kruse (Hrsg), "Fuzzy Systems in Computer Science", F. Vieweg & Sohn, Braunschweig 1994
- W. Pedrycz, "Fuzzy Control and Fuzzy Systems", John Wiley, New York, 1993
- Li-Xin Wang, "Adaptive Fuzzy Systems & Control", Prentice Hall, 1994
- C. Welch, "Fuzzy Logic: From Software to Silicon", Togai Infralogic, 5 Vanderbilt, Irvine, CA 92718, 1992
- R. Yager and L.A. Zadeh (eds.), "An Introduction to Fuzzy Logic Applications in Intelligent Systems", Kluwer Academic Publishers, Dordrecht, 1992
- L.A. Zadeh, "Fuzzy Sets and Applications (selected papers)", John Wiley, New York, 1987
- L.A. Zadeh and J. Kacprzyk (Eds.), "Fuzzy Logic for the Management of Uncertainty", John Wiley & Sons, New York, 1992
- H.J. Zimmermann, "Fuzzy Set Theory and Its Applications", 2nd edition, Kluwer Academic Publishers, Dordrecht, 1990
- H.J. Zimmermann,(Hrsg.) "Fuzzy Technologien, Prinzipien Werkzeuge, Potentiale", VDI-Verlag, Düsseldorf 1993
- H.J. Zimmermann, und C.v. Altröck (Hrsg.) "Fuzzy Logic (Anwendungen)", R. Oldenbourg Verlag, München 1993)

Fuzzy Logik und ihre Anwendung

Gliederung:

- **Was ist Fuzzy?**

Wörterbuch: "fuzzy" = unscharf, verschwommen,
verwischt oder auch "fusselig" ?
warum soll das ein nützliches Konzept sein ?

- **Grundlagen:**

- Fuzzy Mengen und linguistische Variablen
- Fuzzy Logik und plausibles Schließen

- **Anwendungsgebiete mit Beispielen:**

- Fuzzy Expertensystem
- Fuzzy Regelung
- Fuzzy Mustererkennung

Wozu Unschärfe in Modellen, Daten, Definitionen und Logik ?

Beispiel: Einparken mit Helfer

unscharfe

das Lenkrad stark nach links einschlagen,

das Auto knapp 1 m zurücksetzen

Wenn wir uns dem geparkten Auto nähern,

dann das Auto langsam abbremsen

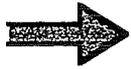
exakte Anweisungen

das Lenkrad 30°10' nach links

76,3 cm zurückfahren

nach 1,76 s abbremsen

Bremsbeschleunigung = 3.0 m/s²



exakte Anweisungen unbrauchbar



1. linguistische Unschärfe (Sprache)
2. Stochastische Verteilungen (Daten)
3. informationelle Unschärfe (*Regeln* & Daten)

Modell: deterministisch, stochastisch, heuristisch, fuzzy ?

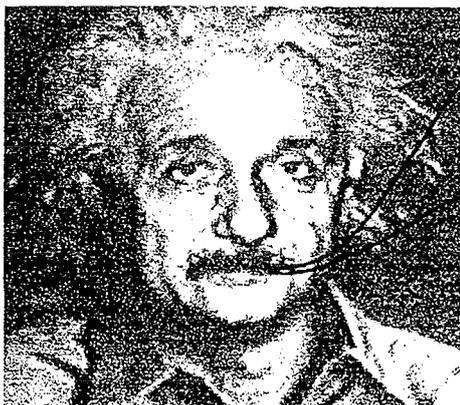
Brauchbarkeit der Modelle ?



Kriterien

mathematisch
physikalisch
rechnerisch

vielfältig
"natürlich"
leicht durchführbar

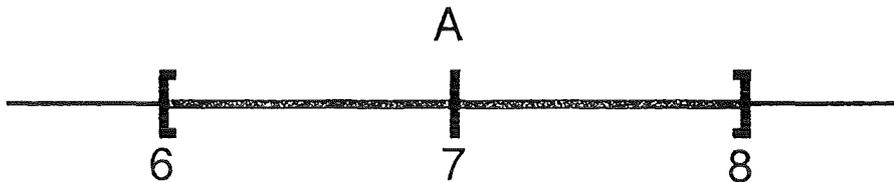


keep it simple -
but only simple
enough to work

Beispiel: scharfe Menge A

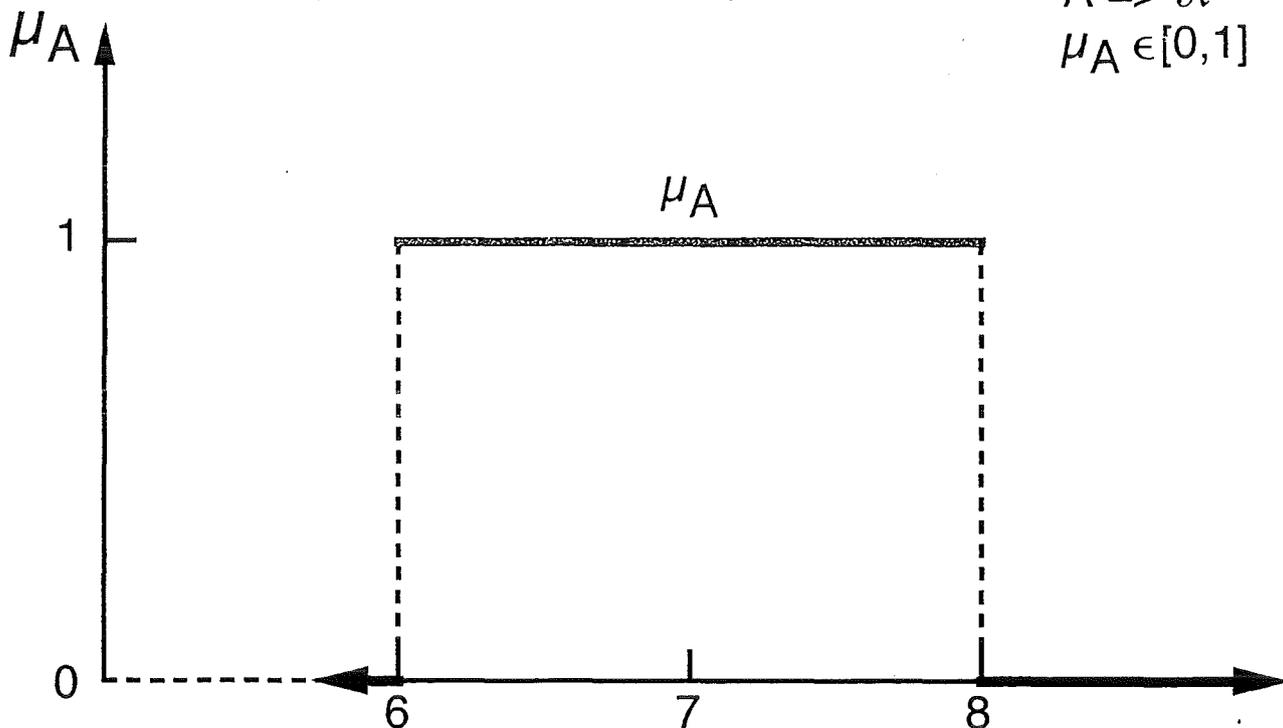
(reelle Zahlen zwischen 6 und 8)

$$A := \{a \mid 6 \leq a \leq 8 \wedge a \in \mathbb{R}\}$$



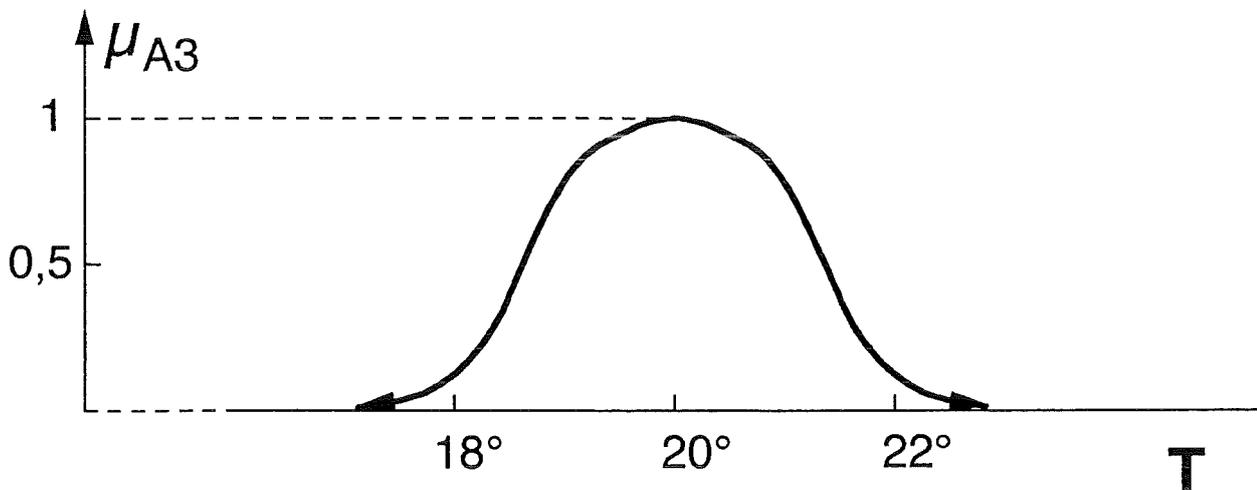
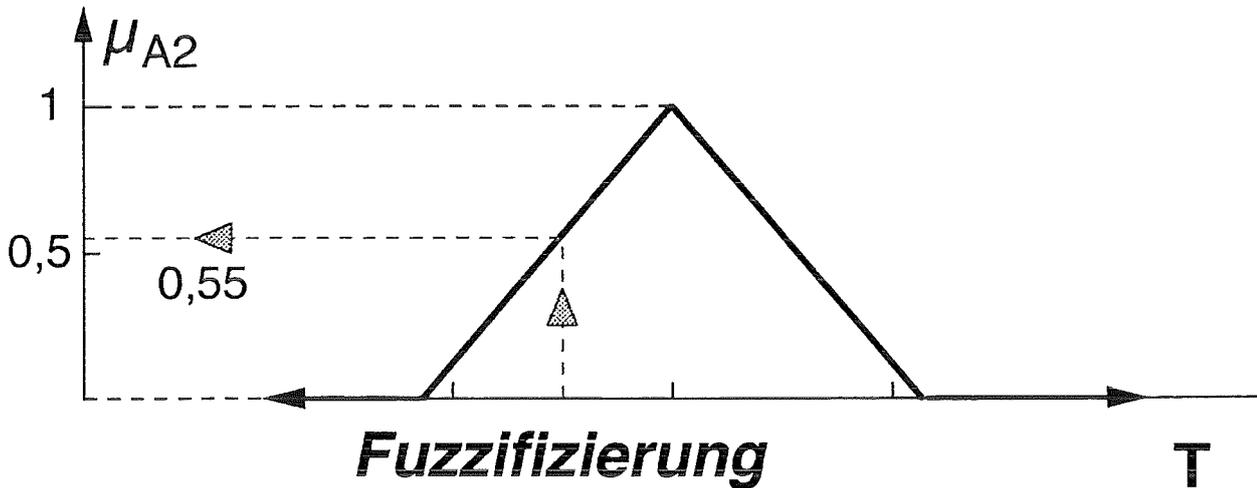
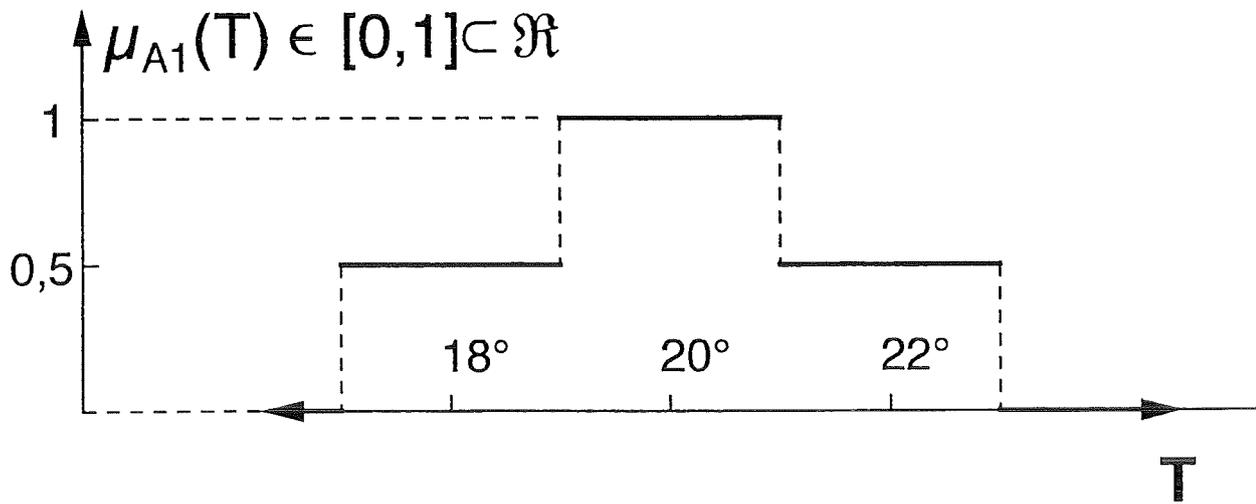
Def.: **Zugehörigkeitsfunktion** $\mu_A = \begin{cases} 1 & x \in A \\ 0 & \text{sonst} \end{cases}$
empirisch (würfeln)

$$\hat{=} A \Rightarrow \mathfrak{R} \\ \mu_A \in [0, 1]$$



Unscharfe Menge: $A = \{ (T, \mu_A) \mid T \text{ nahe } 20^\circ \}$

Zugehörigkeitsfunktion mit der T zur unscharfen Menge A gehört



Was ist diesen Zugehörigkeitsfkt. gemeinsam ?

Zadeh's Operationen auf unscharfen Mengen 1965

$$\mu_A(x), \mu_B(x) \in (0,1) \quad \forall x \in X$$

Gleichheit (=) $A = B$ $\mu_A(x) = \mu_B(x)$

Teilmenge (\subset) $A \subset B$ $\mu_A(x) \leq \mu_B(x)$



Komplement ($\bar{}$) $\mu_{\bar{A}} = 1 - \mu_A$

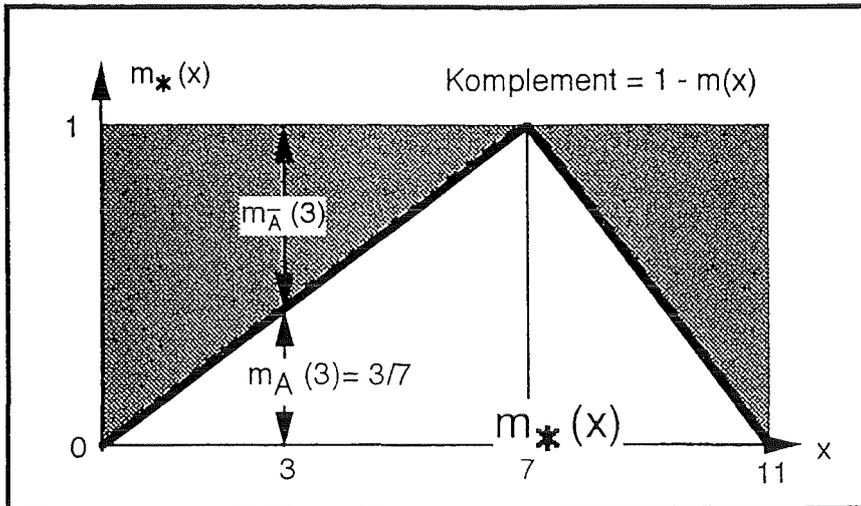
Durchschnitt (\cap) $\mu_{A \cap B} = \min(\mu_A, \mu_B)$ z.B.: $* = \mu_A * \mu_B$

Vereinigung (\cup) $\mu_{A \cup B} = \max(\mu_A, \mu_B)$ $* = \mu_A + \mu_B - \mu_A * \mu_B$

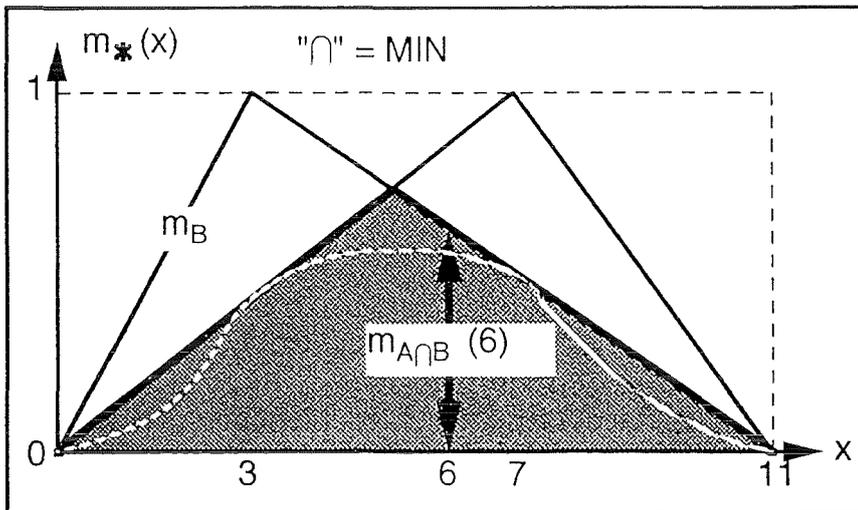
* Es existieren viele andere Operatoren außer min, max für " \cap ", " \cup "

speziell $\mu_{A \cap \bar{A}} = \min(\mu_A, \mu_{\bar{A}}) = \min(\mu_A, 1 - \mu_A) \neq 0$! für $\mu_A \neq 0$

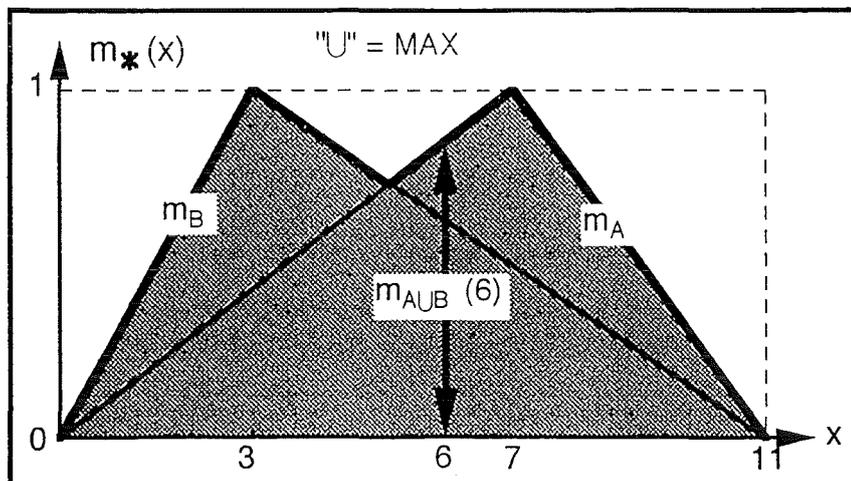
Komplement einer fuzzy Menge und ihre geometrische Interpretation



Fuzzy Durchschnitt von Mengen



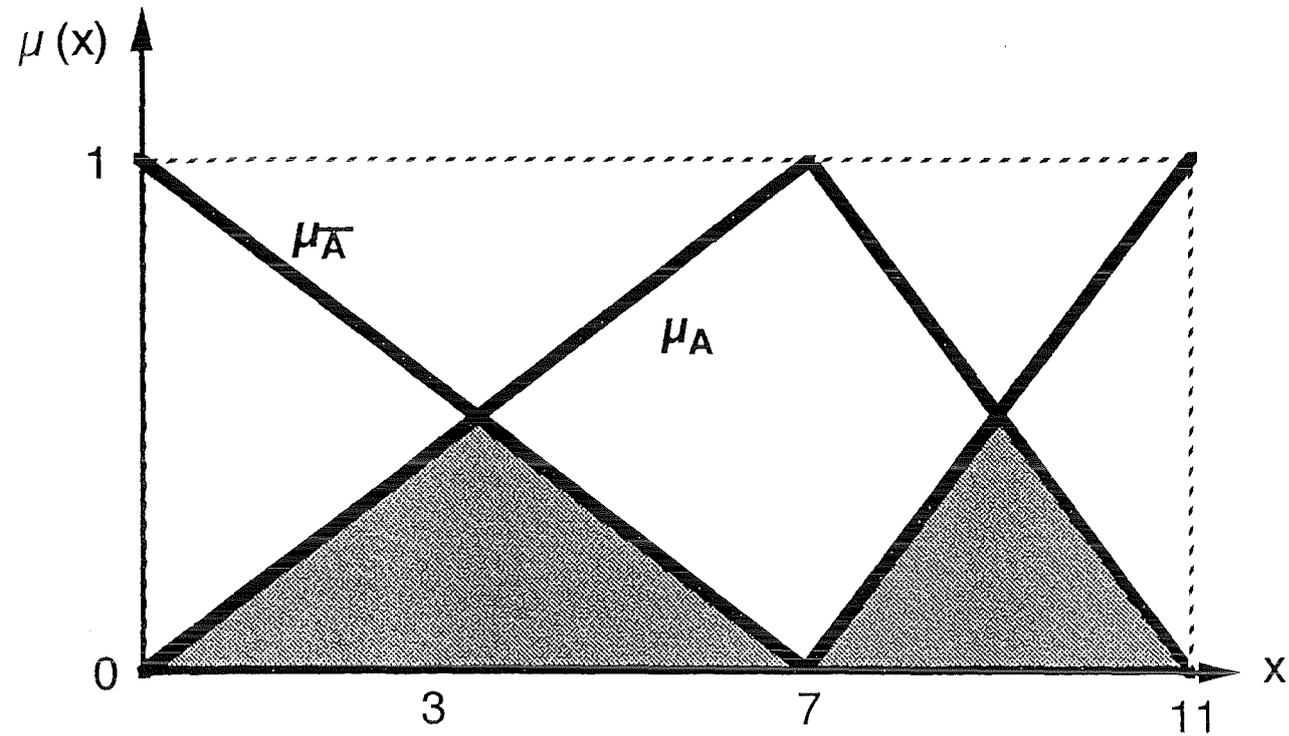
Fuzzy Vereinigungsmenge



Besonderheiten von fuzzy Verknüpfungen

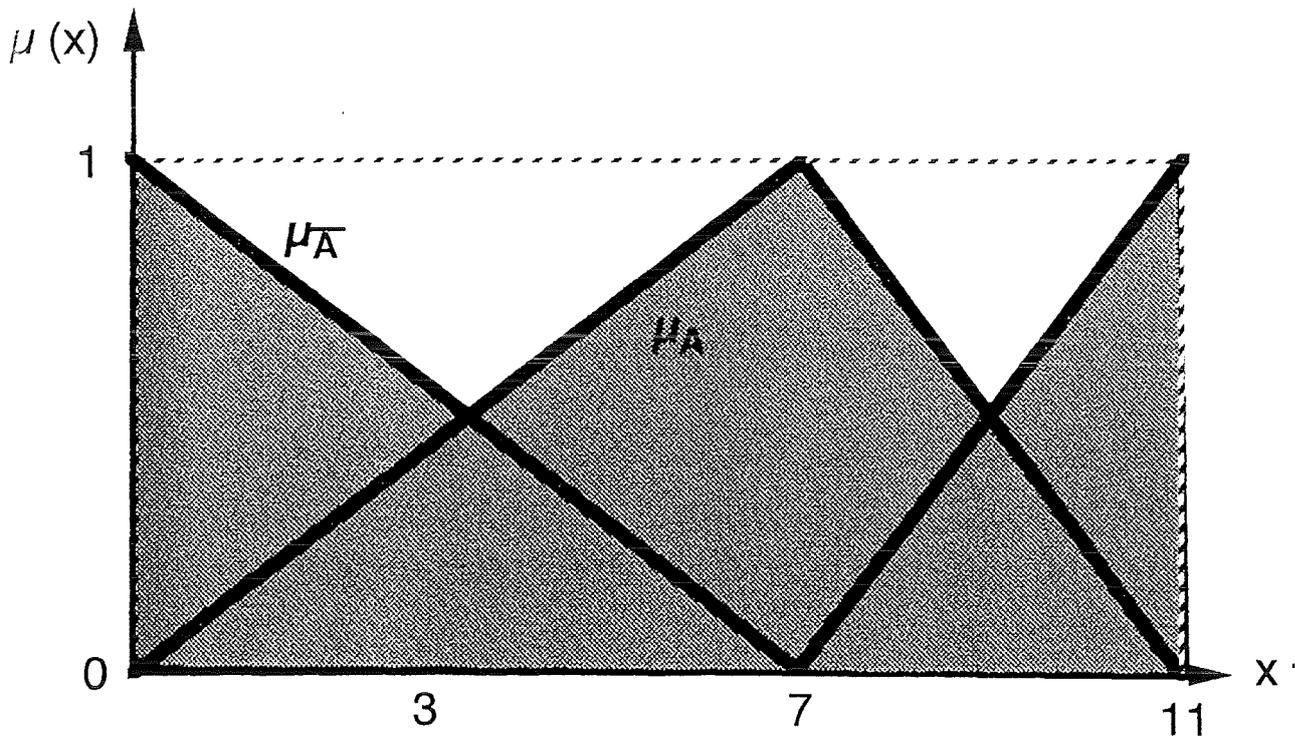
$$A \cap \bar{A} \neq \emptyset$$

$$\mu_{A \cap \bar{A}} = \min(\mu_A, \mu_{\bar{A}})$$



$$A \cup \bar{A} \neq X$$

$$\mu_{A \cup \bar{A}} = \max(\mu_A, \mu_{\bar{A}})$$



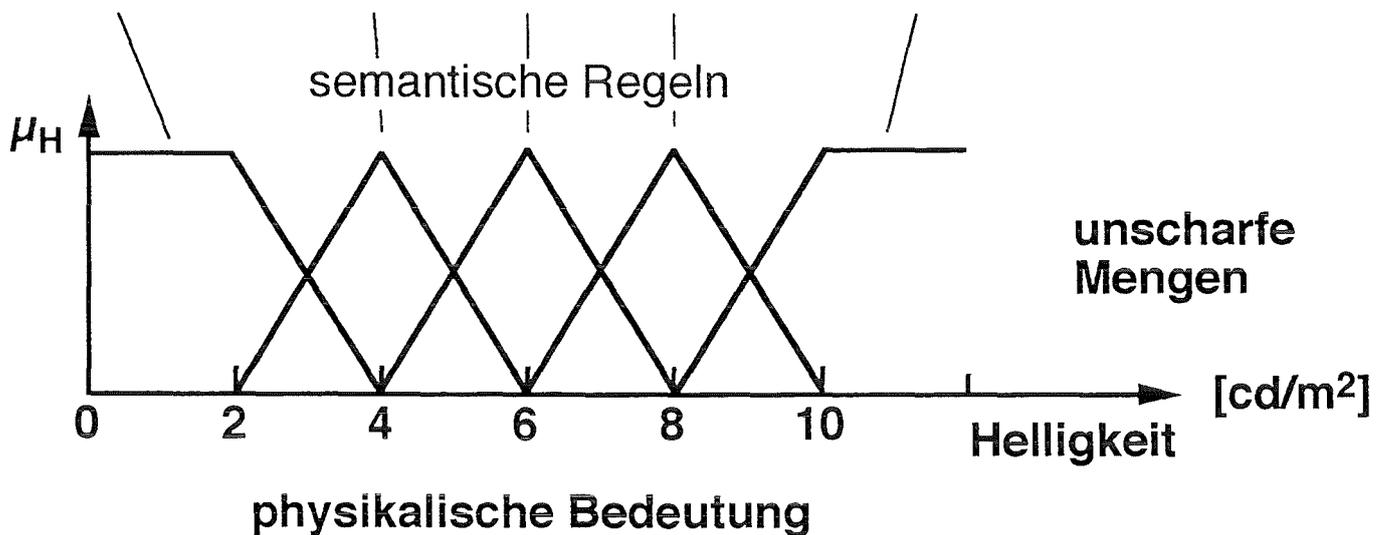
Gemmeke, F, 1189

Linguistische Ausdrücke als symbolische Beschreibung der fuzzy Logik

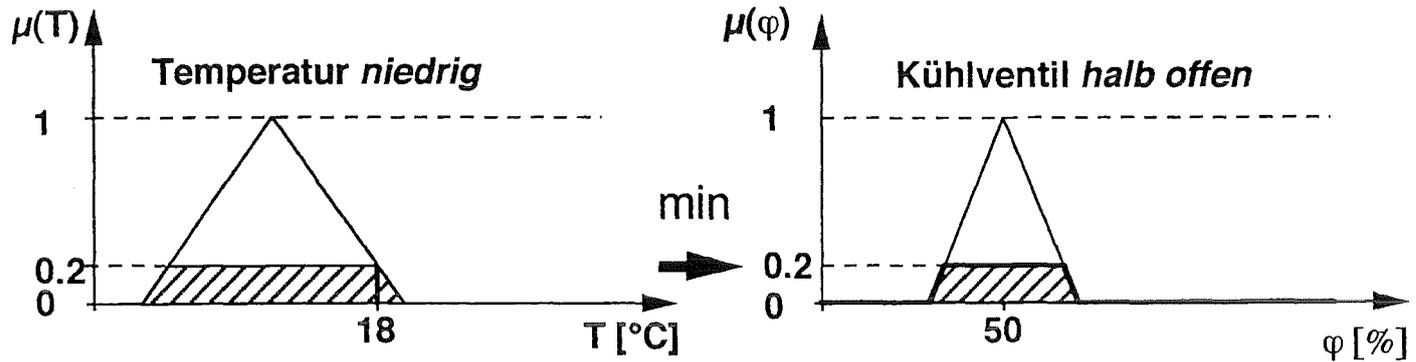
z.B.: Beschreibung der linguistischen Variablen Helligkeit
mit Hilfe von syntaktischen Regeln (Grammatik):

linguistische Terme:

sehr dunkel dunkel mittel hell sehr hell



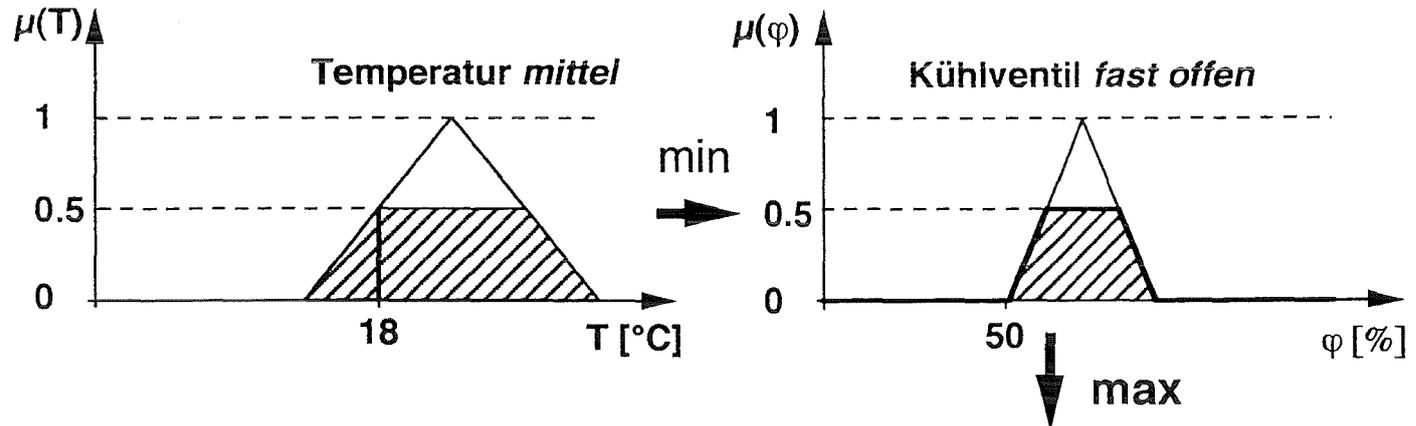
fuzzy Kühlschrank



Regel 1

IF Temperatur = *niedrig* THEN Kühlventil = *halb offen*

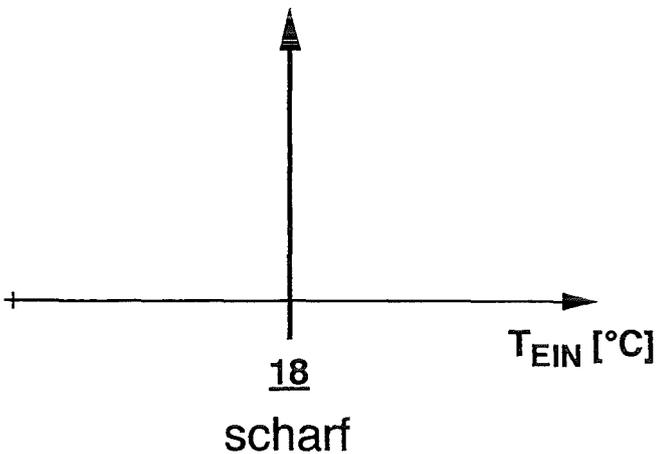
Inferenz
(Gedankenschluß)



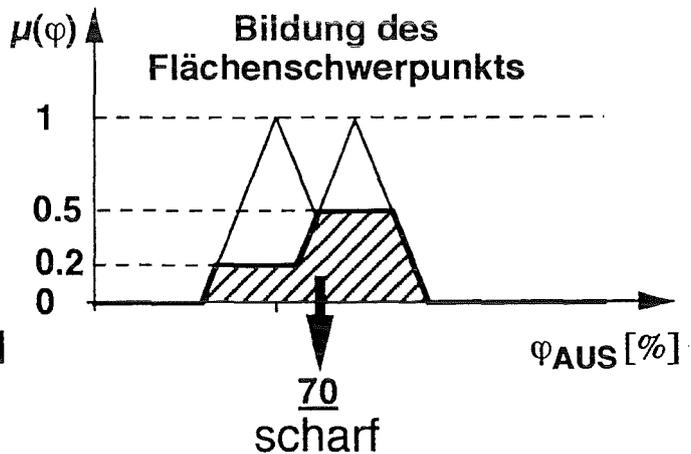
Regel 2

IF Temperatur = *mittel* THEN Kühlventil = *fast offen*

Fuzzifizierung



Defuzzifizierung



Unschärfe als Mehrwertigkeit der Logik

Klassische Logik:

der Wahrheitsgehalt t einer Aussage S

$$t(S) = \left\{ \begin{array}{ll} \text{wahr} & 1 \\ \text{falsch} & 0 \end{array} \right\} \quad \left(\begin{array}{l} 1 \\ -1 \end{array} \right) \left\{ \right.$$

linguistisch boolesch numerisch (Computer)

Parminides ~ 500 a.Chr. : "tertium non datur" $\hat{=} S \cap \bar{S} = \emptyset$

Paradoxon: Lügt der Lügner von Kreta,
wenn er sagt,
alle Kreter sind Lügner ?

Der Satz auf der anderen Seite ist wahr
Der Satz auf der anderen Seite ist nicht wahr

Ist beides falsch oder wahr ?

$$t(S) = t(\bar{S}) = 1 - t(S) \quad ???$$

$$t(S) = \begin{pmatrix} 1 \\ 0 \end{pmatrix} = t(\bar{S}) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \Rightarrow 1 = 0 ?$$

Ausweg: J. Lukasiewicz (1930): mehrwertige Logik

3 wertige Logik: $2 * t(S) = 1 \Rightarrow t(S) = 1/2$

Paradoxon reduziert sich zu einer wortwörtlichen
Halbwahrheit

Schlußfolgen für scharfe Logik (inference)

Implikation = Regel: wenn ... dann...: $(a \Rightarrow b) = \bar{a} \vee b$

d.h. eine Implikation ist nur dann falsch,
wenn a = wahr und b = falsch

modus ponens (Grundregel der Deduktion)

$$(a \wedge (a \Rightarrow b)) \Rightarrow b$$

Prä-
misse } Regel: wenn a = wahr dann b = wahr
Faktum: a = wahr

Schluß: b = wahr

Approximatives Schließen

- unscharfe Wahrheitswerte $a, b \in [0,1] \subset \mathbb{R}$, z.B. $a = b = 0,9$
- linguistische Variablen und unscharfe Quantoren ("meist", "sehr rot", ...)
- unscharfe Logik:
 $a \wedge b = \min(a, b)$ und andere Def.'s
 $a \vee b = \max(a, b)$ Lukasiewicz (1930)
 $\bar{a} = 1 - a$ analog zu den fuzzy Mengen
- aber Implikation nicht $= \max(1 - a, b)$
sondern $\min(a, b)$ (Mamdani)

Nichtidealisierte Realität enthält viele Zwischenstufen
zwischen "wahr" und "falsch" \Rightarrow unscharfe Logik

Gemmeke, Fuzzy13

erweiterter modus ponens "plausibles Schließen" (Zadeh)

u, v = linguistische Variablen mit den linguistischen Werten $A(x), B(y)$

Es existiert Abbildung unsharp Mengen
Linguistik \longrightarrow physikalische Welt

z.B. Expertenaussage: rote Kirschen sind süß
(u =Kirschenfarbe, v =Kirschengeschmack):

Regel: wenn Kirschenfarbe = rot dann Geschmack = süß

Faktum: Kirschenfarbe = sehr rot

Schluß: Geschmack = sehr süß
(mehr als...)

Voraussetzungen: Unschärfe in rot, süß

Monotonie in rot, süß

d.h. i.A. $A'(x)$ ähnlich $A(x)$ und $B'(y)$ ähnlich $B(y)$

in gleicher Richtung monotone Variation der Ausgangsgröße
(z.B. sehr, mehr als, etwas weniger,...)

Regel: wenn $u = A(x)$ dann $v = B(y)$

Faktum: $u = A'(x)$

Schluß: $v = B'(y)$

Defuzzifizierung für

Stellgröße → Regelung

bzw.

Entscheidung

↙ Expertensystem

↘ Klassifikator
(Mustererkennung)

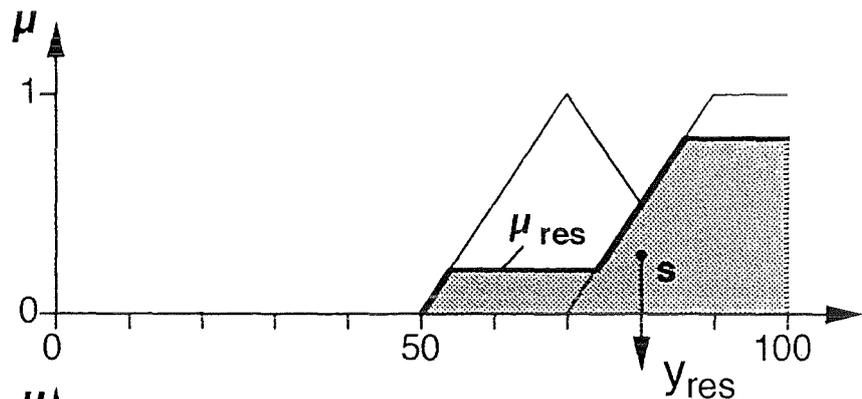
Anforderungen:

- interpolierend (zw. den Regeln)
- alle aktiven Regeln gehen ein
- stetig input ⇒ output
- abhängig vom Erfüllungsgrad der Regeln
- Rechengeschwindigkeit ?
- Hardware - Realisierung ?

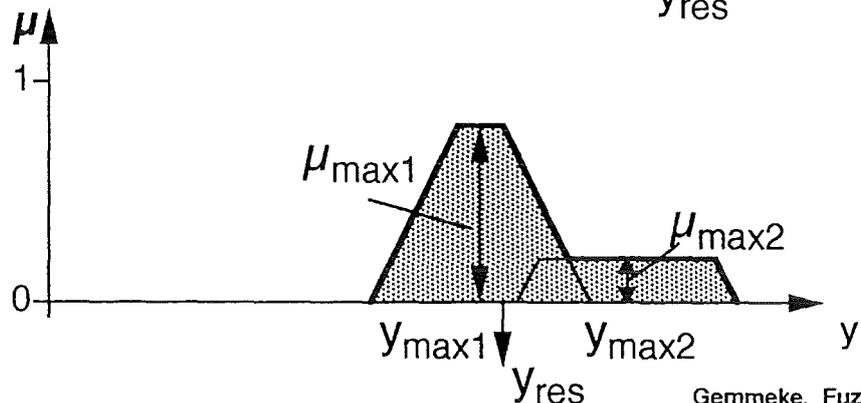
COA: center of gravity (area) = Schwerpunkts - Methode

$$y_{res} = \frac{\int y \cdot \mu_{res}(y) \cdot dy}{\int \mu_{res}(y) \cdot dy}$$

$$\approx \frac{\sum y_i \cdot \mu_{res}(y_i)}{\sum \mu_{res}(y_i)}$$



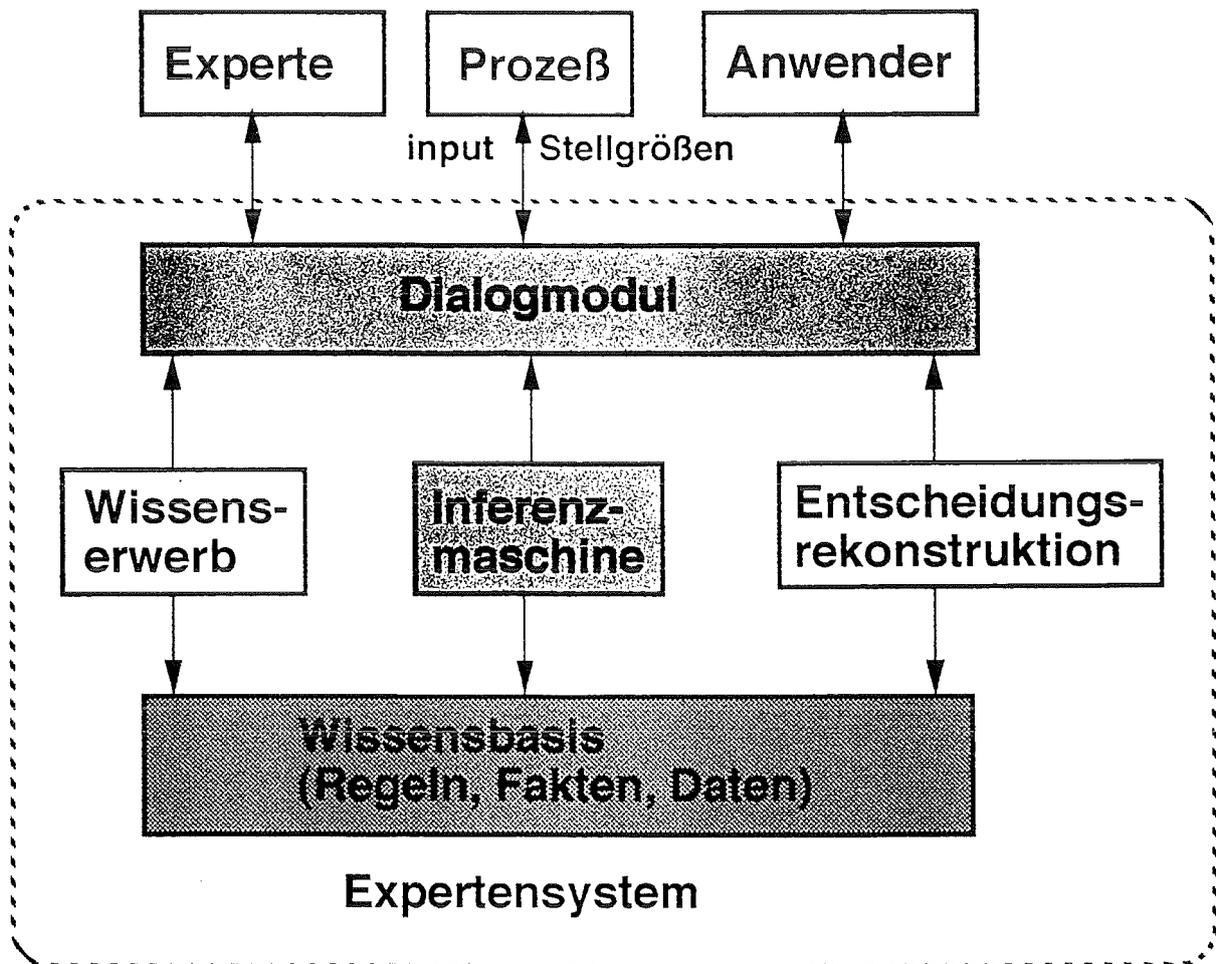
$$\approx \frac{\sum y_{maxi} \cdot \mu_{maxi}}{\sum \mu_{maxi}}$$



Gemmeke, Fuz15

Struktur regelbasierter Fuzzy Expertensysteme

- **Oberfläche (shell) in natürlicher Sprachumgebung**
fuzzy: Verwendung linguistischer Variablen
- **Entscheidungsfällung wie beim Menschen**
fuzzy: Implikation \cong dem menschlichen Empfinden
- **Entscheidungsfällung auch bei unvollständiger Informationslage**
fuzzy: plausible Schlußweise, auch widersprüchliche Regeln, Daten können verarbeitet werden
- **Wissenserweiterung der Wissensbasis durch Lernen (KNN)**



Inferenzmaschine: Fuzzifizierung, Inferenz, Summation aller Regeln, Defuzzifizierung

Anwendung von fuzzy Expertensystemen

- Regel- und Datenbasierte Kontrollsysteme (Leittechnik)
- Regeltechnik (speziell nichtlineare Regelung)
- Mustererkennung
- Finanzsysteme (Kreditwürdigkeit von Kunden, Aktienkursentwicklung)
- Optimierungsprobleme (Logistik, ...)
- Datenanalyse (Fehlerdiagnostik, medizinische Diagnostik, ...)

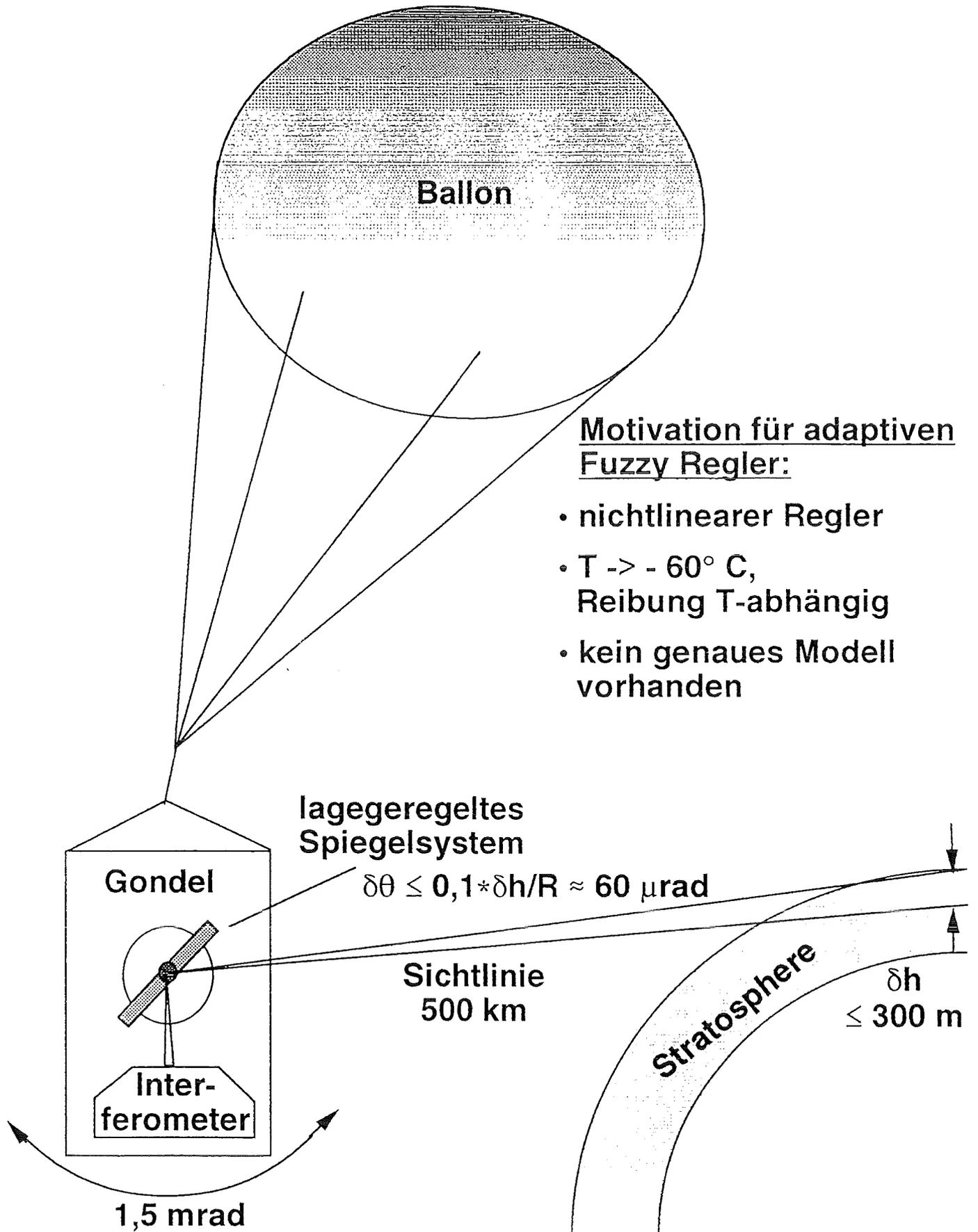
Bekannte fuzzy Expertensysteme

Name	Anwendung
FuzzyCLIPS	<p>Entwicklungssystem für Expertensysteme auf CLIPS-Basis erlaubt freie Mischung von bool'scher und fuzzy Logik verarbeitet auch gewichtete Unsicherheitsfaktoren "then"-Teil kann auch Prozeduren enthalten kann auch Objekte verarbeiten</p> <p>WWW(World Wide Web): http://ai.iit.nrc.ca/fuzzy/fuzzy.html oder über anonymous ftp: ai.iit.nrc.ca:/pub/fzclips/</p>
Fril	<p>fuzzy Entwicklungssystem für Expertensysteme auf PROLOG-Basis</p> <p>Fril Systems Ltd, Maggs House, Bristol, BS8 1QX, UK, fax: +44 272 211594</p>

Gemmeke, F 125

Lageregelung für MIPAS-Ballon-Experiment

(Michelson Interferometer für passive Atmosphärische Sondierung
H. Fischer et al., IMK)

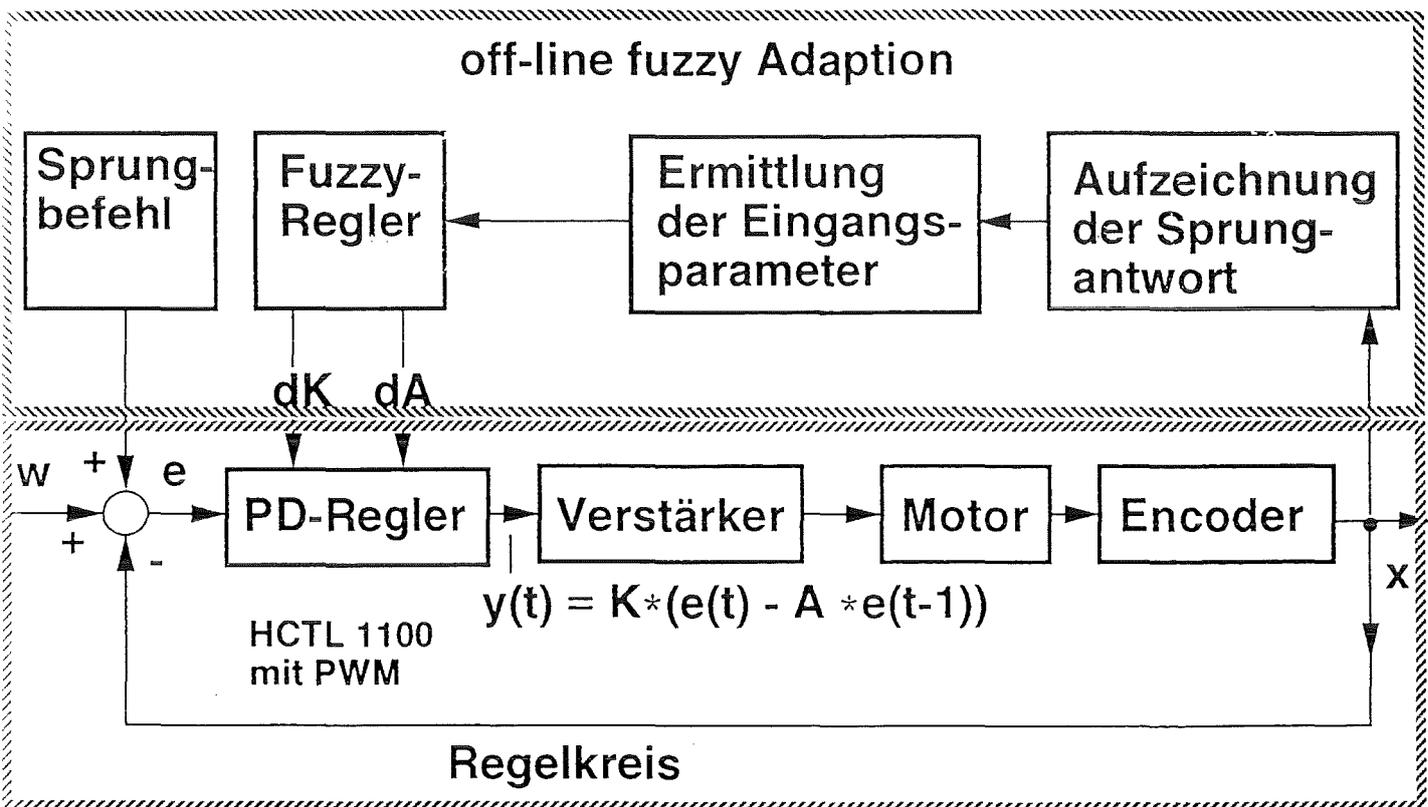


Parameter-adaptive Fuzzy-Controller

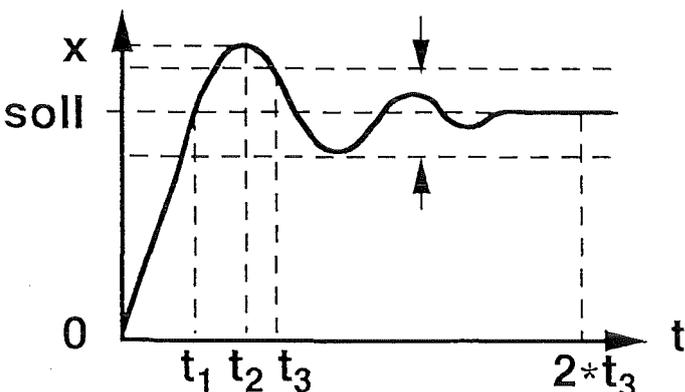
bisher: Fuzzy-Komponenten direkt im Regelkreis

jetzt: im Regelkreis konventioneller Regler, dessen Parameter durch Fuzzy-Logik eingestellt werden:

Beispiel: adaptiver PD-Regler für MIPAS



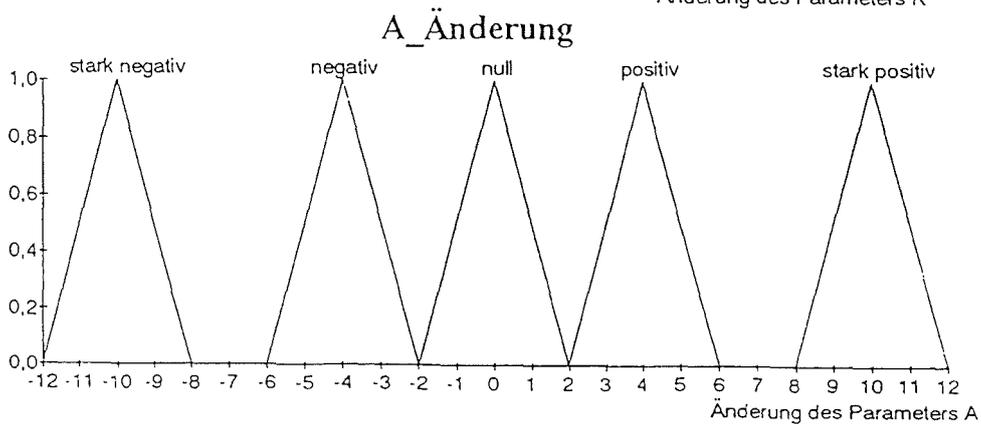
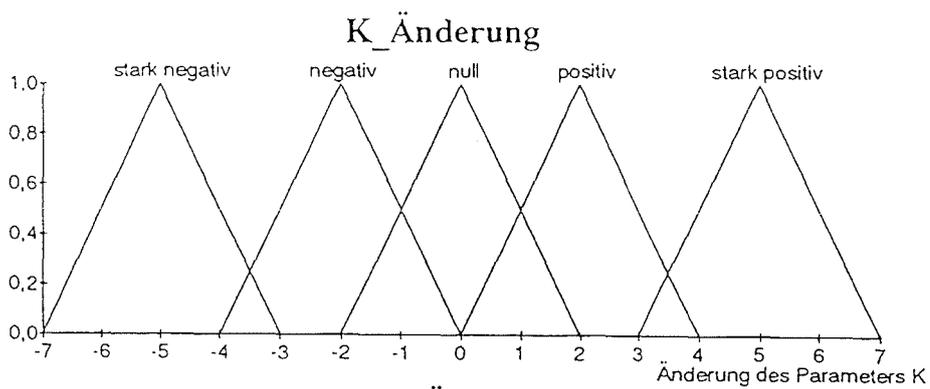
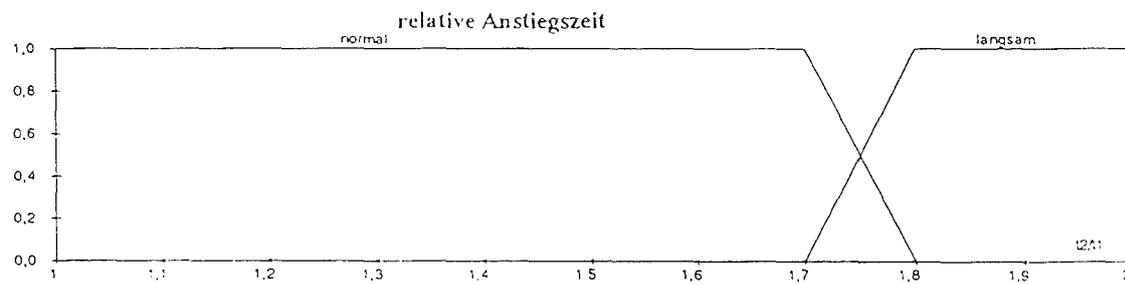
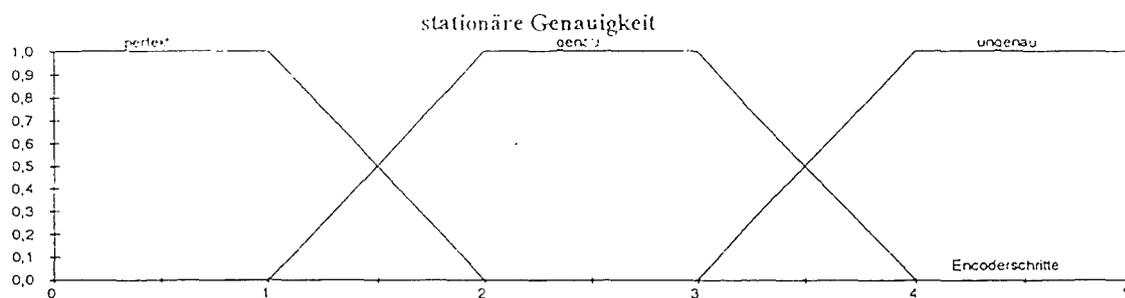
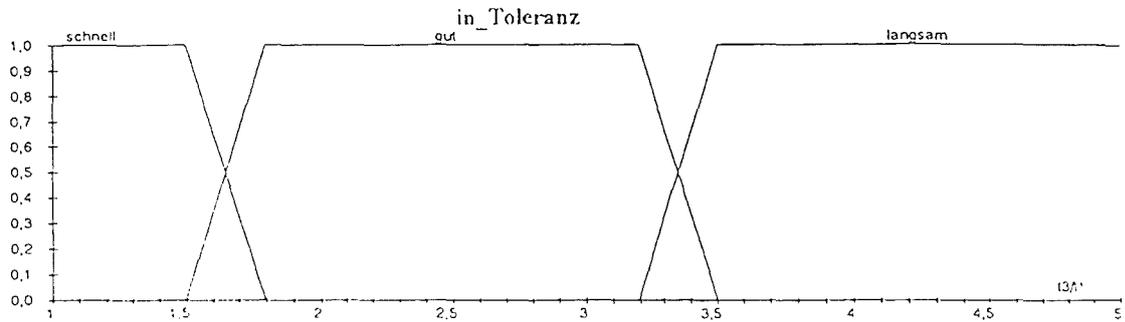
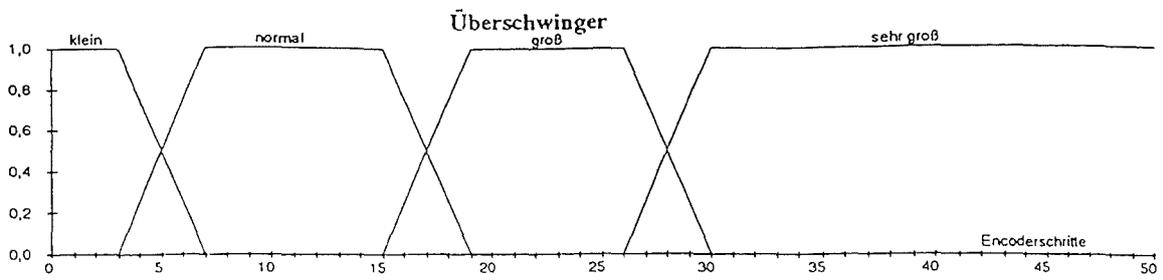
- zunächst Einstellung des Reglers mit Default-Werten
- Umschalten eines Sprungs und Analyse der Sprungantwort
- Bestimmung mehrerer Maßzahlen:



1. Überschwinger $x(t_2)/x_{soll}$
2. in_Toleranz = t_3/t_1
3. Genauigkeit $\sigma(x)$ für $t > 2t_3$
4. relative Anstiegszeit t_2/t_1

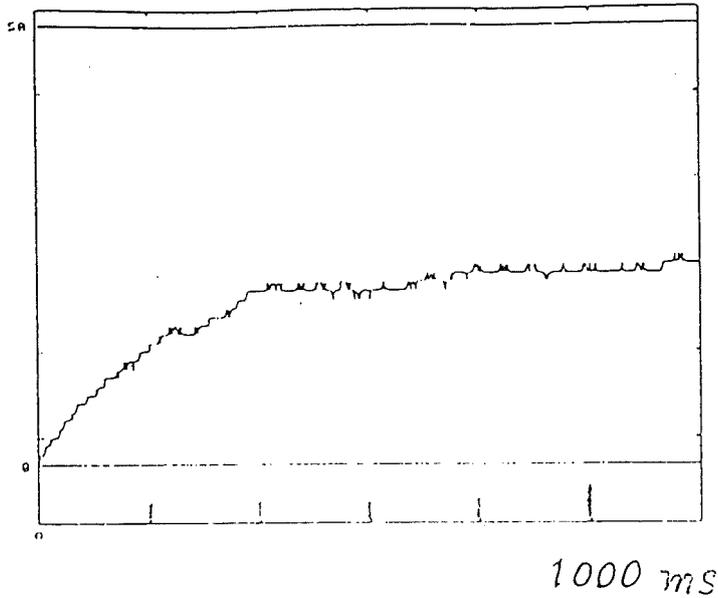
Regeln für den Fuzzy-Controller

		stat. Genauigkeit												
		perfekt	normal	ungenau										
Über- schwinger	klein	-	0	0	--	--	--	++	++	++	++	++	++	in Toleranz schnell
	normal	-	0	0	-	-	-	+	+	+	+	+	+	
	groß	-	+	+	-	-	-	0	0	0	++	++	++	
	sehr groß	0	0	0	-	-	--	-	-	-	++	++	++	
		0	0	0	--	--	--	+	+	+	+	+	+	gut
		0	0	+	-	-	-	0	0	+	0	0	0	
		0	0	+	-	-	-	-	-	+	+	+	+	
		-	-	0	--	--	--	-	-	-	++	++	++	
		+	+	+	0	0	0	0	0	+	+	+	+	langsam
		+	+	+	0	0	0	0	0	+	+	+	+	
		+	+	+	0	0	0	-	-	-	++	++	++	
		++	++	++	-	-	-	--	--	--	++	++	++	
		normal		langsam					normal		langsam			
		relative Anstiegszeit						relative Anstiegszeit						
Reglerparameter A						Reglerparameter K								



Vor der Optimierung der Parameter:

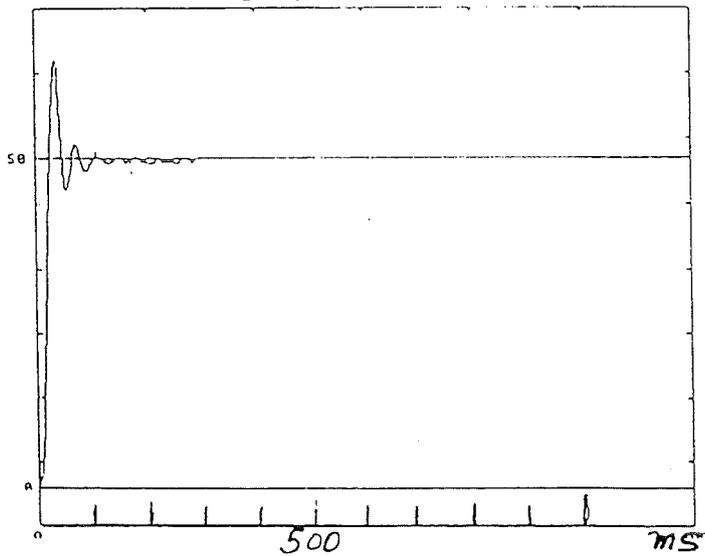
A: 255 K: 255



Sprungantwort mit optimierten Parametern:

A: 206 K: 254

Zahl der Durchgänge: 7

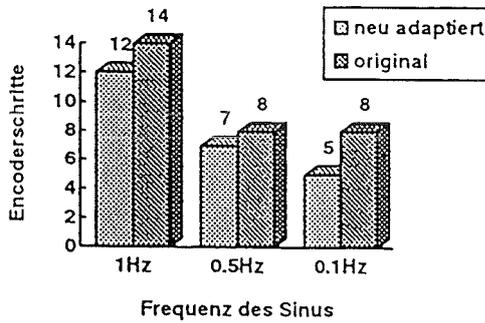


2 Bogenminuten $\hat{=}$

RESULTATE:

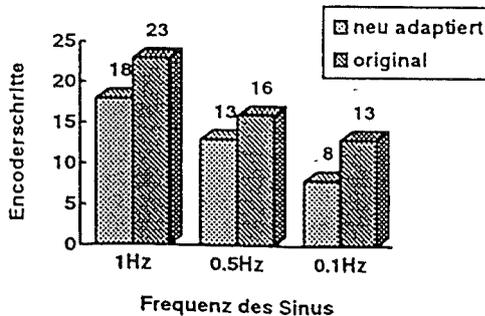
1 ENCODER-Schritt = 2,4"

mit Unwucht:

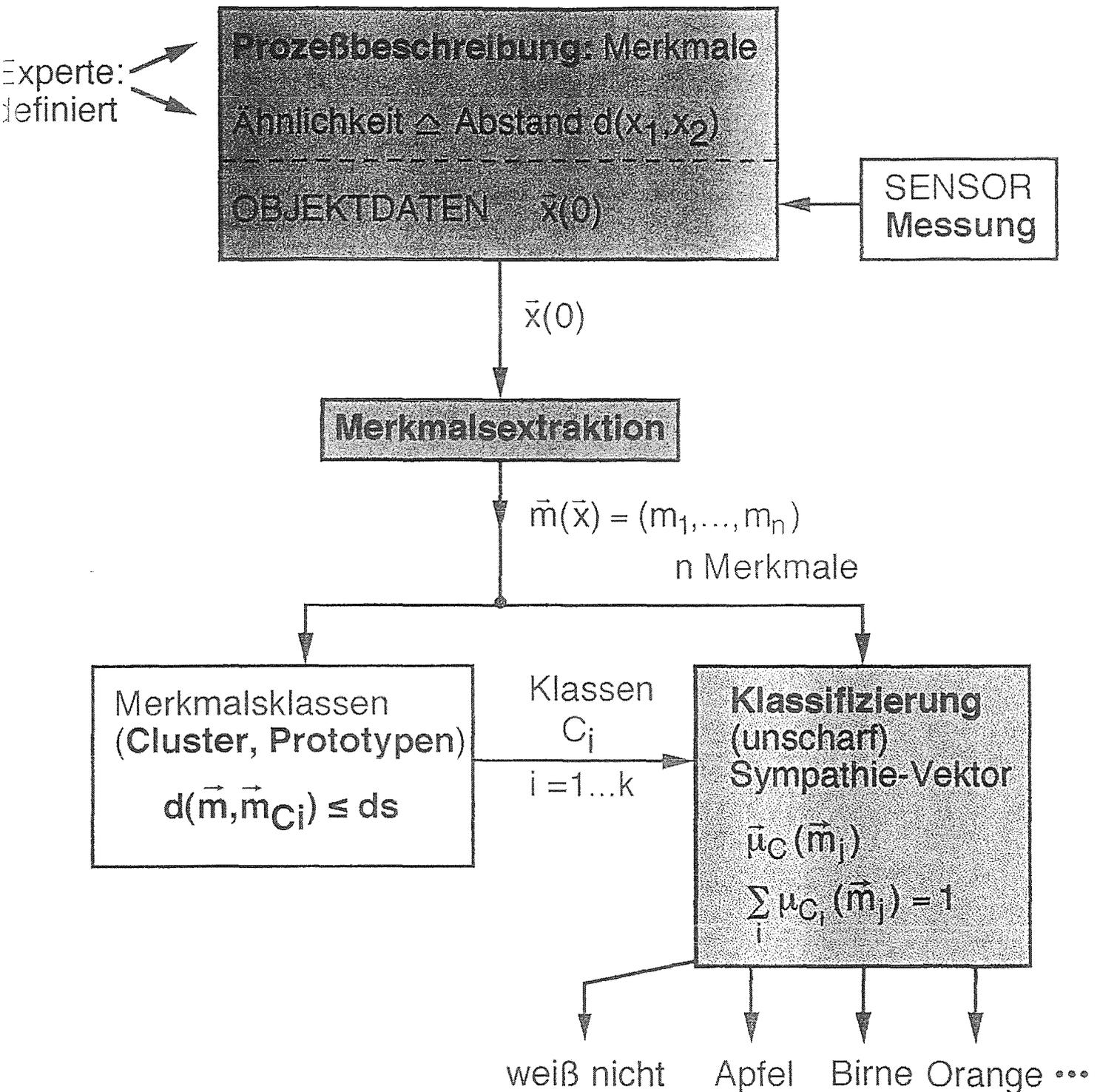


5 ENC-Schritte $\hat{=}$ 60 μ rad

mit Reibung:



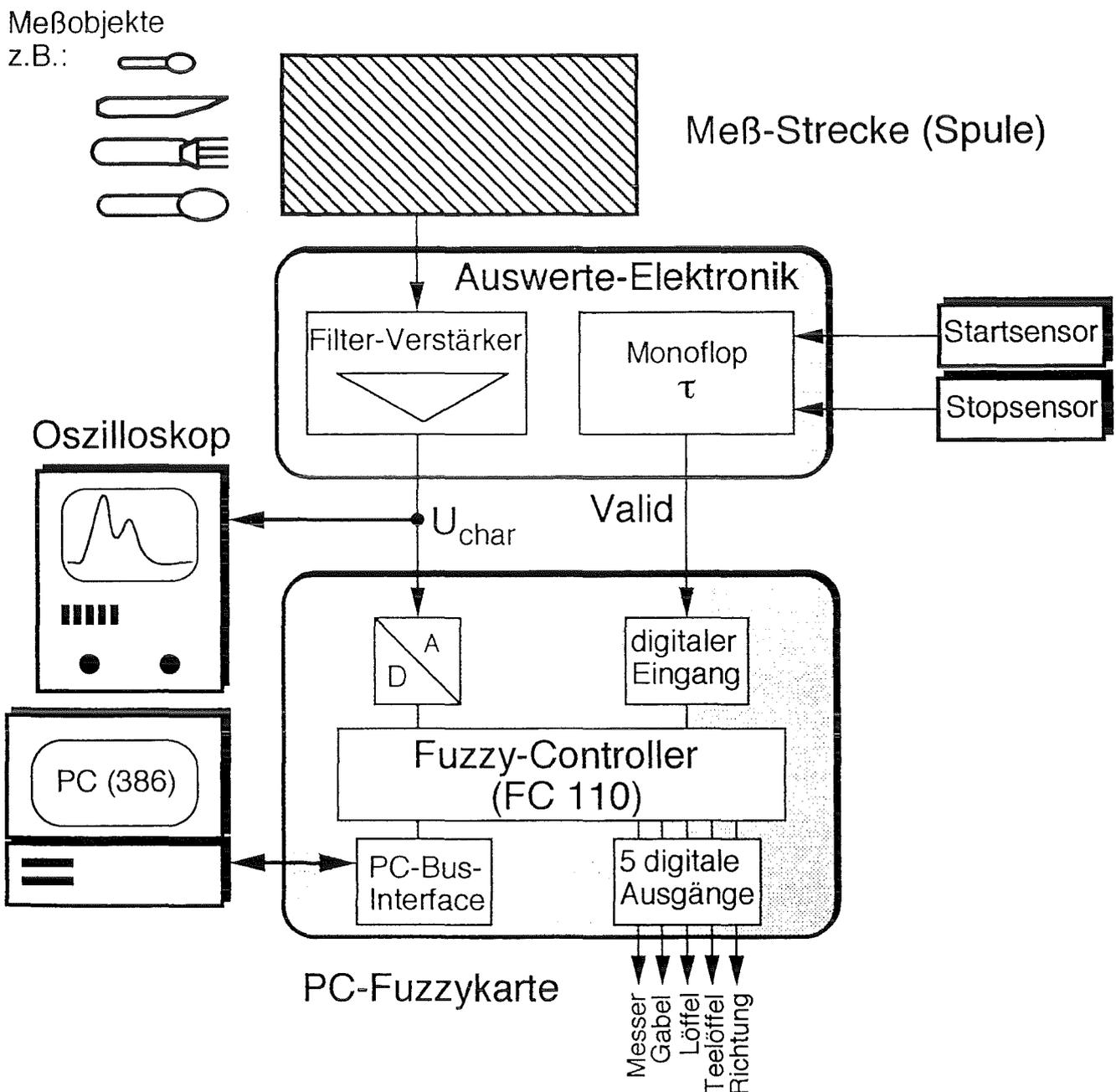
Numerische Mustererkennung



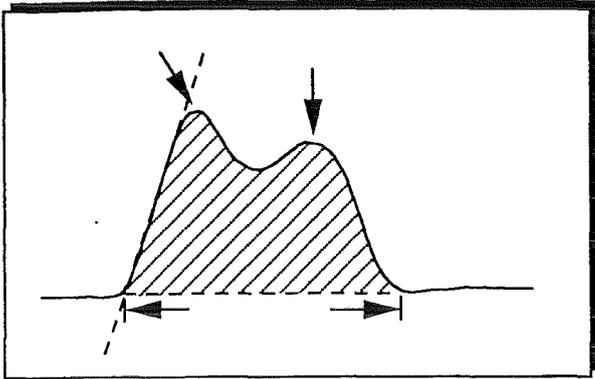
$\mu_{C_i}(\vec{m}_j)$ Zugehörigkeit des Merkmalsvektors \vec{m}_j zur Klasse C_i

Beispiel: **Besteck-Sortieranlage**
(M. Lauzi, Uni. Kaiserslautern)

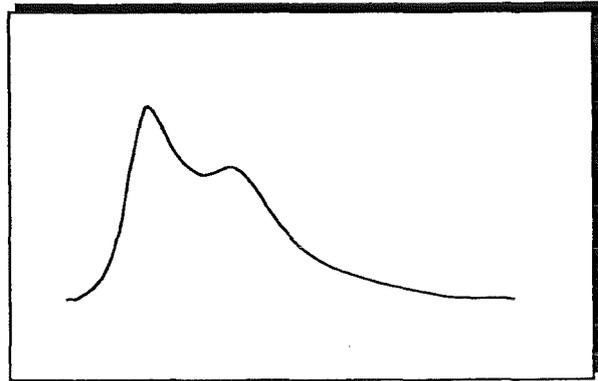
- Besteck läuft an einer Induktionsspule vorbei und erzeugt dabei ein charakteristisches Signal
- Extraktion von Merkmalen aus diesem Signal:
 - Amplitude und Zeitwert des Maximums
 - Differenz zwischen absolutem und lokalem Maximum
 - Fläche und Länge des Gesamtsignals
 - steilster vorkommender Gradient



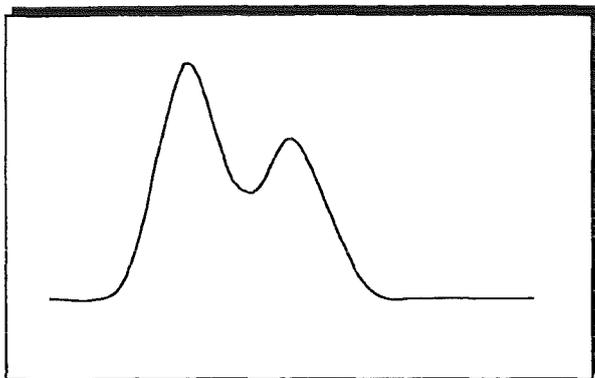
Charakteristische Signalformen des Bestecks



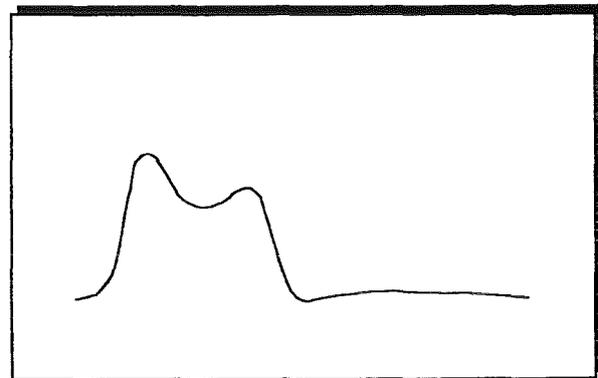
Messer



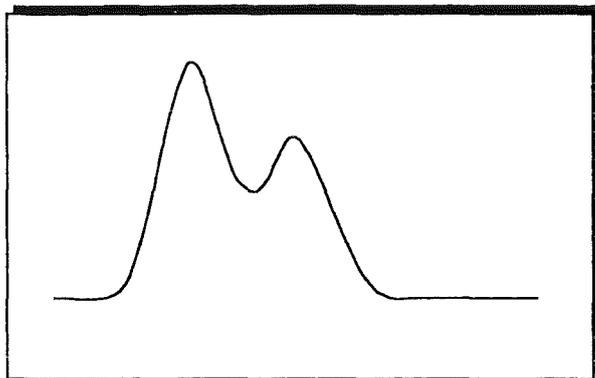
Gabel



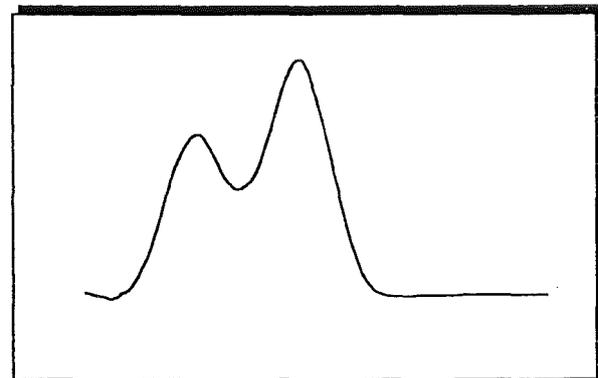
Löffel



Teelöffel



Löffel vorwärts



Löffel rückwärts

Gruppe 5: Richtungserkennung

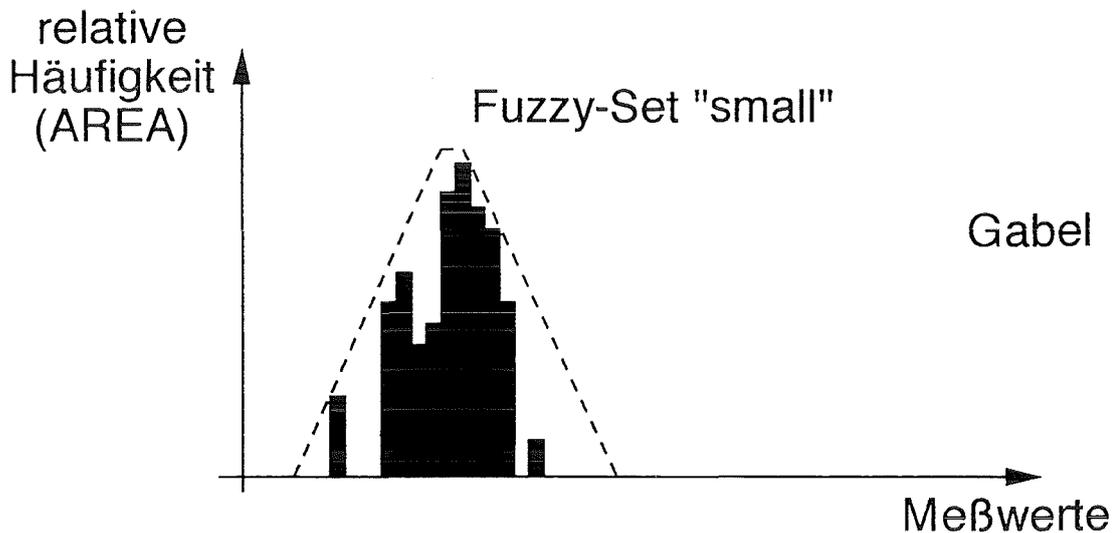
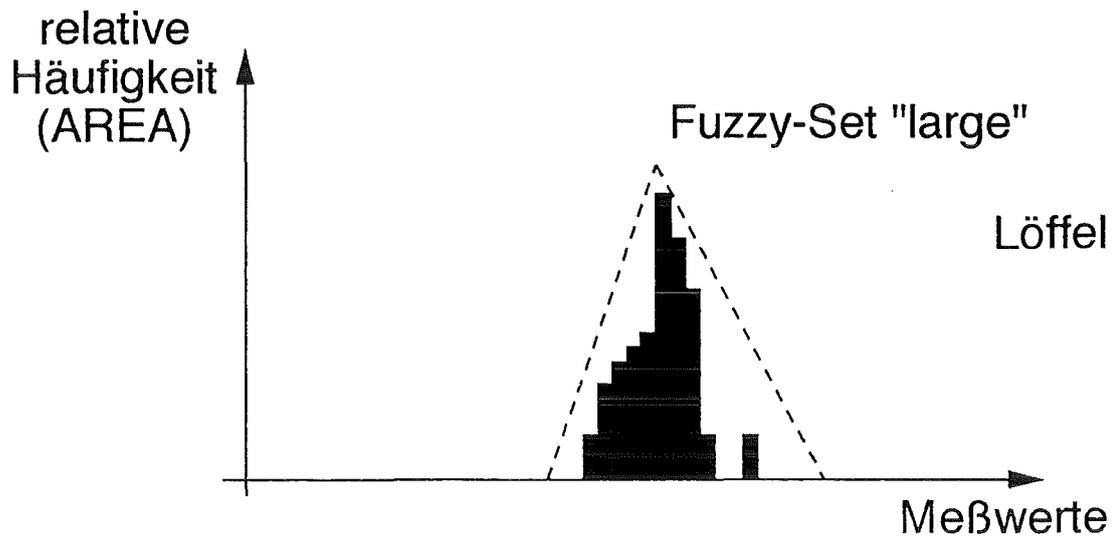
IF (MAX_TIME = early) **AND** (DETECT = Gabel) **THEN** (DIRECTION = forward)

IF (MAX_TIME = medium) **AND** (DETECT = Löffel) **THEN** (DIRECTION = forward)

IF (MAX_TIME = early) **AND** (DETECT = Teelöffel) **THEN** (DIRECTION = forward)

IF (MAX_TIME = medium) **AND** (DETECT = Messer) **THEN** (DIRECTION = forward)

Ergebnisse "Besteck - Identifizierung"



Gruppe 1: Gabel

IF (AREA = medium) **AND** (MAX_DIFF = small) **AND** (TIME = medium)
AND (GRADIENT = medium) **THEN** (DETECT = Gabel)

IF (AREA = medium) **AND** (MAX_DIFF = small) **AND** (TIME = short)
AND (GRADIENT = big) **THEN** (DETECT = Gabel)

IF (AREA = medium) **AND** (MAX_DIFF = medium) **AND** (TIME = medium)
AND (GRADIENT = big) **THEN** (DETECT = Gabel)

IF (AREA = medium) **AND** (MAX_DIFF = medium) **AND** (TIME = medium)
AND (GRADIENT = medium) **THEN** (DETECT = Gabel)

● Ergebnis:

Erkennungsrate > 90 %

Zusammenfassung

Fuzzy Logik hat in seinen Hauptanwendungen, wie Regelungstechnik, Expertensystemen und Mustererkennung einige wesentliche Vorteile:

- **gute Mensch-Maschine-Kopplung** über linguistische Variablen und Operatoren in einer sonst rein mathematischen Theorie
- kann **unvollständige und vage Daten** verarbeiten
- **parallele Struktur**
- benötigt **kein Modell**, intuitive menschliche Beschreibung oft hinreichend
- **kurze Programmentwicklungszeit** und schneller Anfangserfolg

aber

- **Optimierung** sehr mühsam (viele Parameter)
 - => **Lernen** der Parameter mit **Neuronalen Netzen** oder evolutionären Modellen
- => Aufwand nur für **nichtlineare und komplexe Systeme** lohnend, aber die sind dann **einfacher zu entwickeln**

Bisher erschienene Berichte zur Seminarreihe “Aktuelle Forschungsgebiete in der Mathematik“

KfK-Seminarreihe “Aktuelle Forschungsgebiete in der Mathematik“ Seminarbeiträge 1993, R. Seifert, T. Westermann (Hrsg.), KfK-Bericht 5276, März 1994:

“Was ist schlecht an schlecht gestellten Problemen?“

H.J. Dobner, Universität Karlsruhe

“Mathematische Verfahren für technische Feldprobleme“

E. Halter, Fachhochschule Karlsruhe

“Material Accountancy: A Game Theoretical Analysis“

S. Zamir, Hebrew University of Jerusalem

“Statistical Methods for the Carcinogenic Risk Assessment“

A. Yakovlev, St. Petersburg University“

“Numerische Lösung partieller Differentialgleichungen mittels finiter Differenzen“

C.P. Hugelmann, Kernforschungszentrum Karlsruhe

KfK-Seminarreihe “Aktuelle Forschungsgebiete in der Mathematik“ Seminarbeiträge 1994, C.P. Hugelmann, R. Seifert, T. Westermann (Hrsg.), FZKA-Bericht 5560, April 1995:

“Globale Optimierung mit Ergebnisverifikation“

D. Ratz, Universität Karlsruhe

“Ein Überblick zur Geschichte der Zahl Null“

P. Mäder, Staatl. Seminar Freiburg

“Die Vorhersage und das Entdecken von Trendänderungen bei Finanzdaten“

H.R. Lerche, Universität Freiburg

“Nonparametric Estimation of a Survivor Function from Incomplete Data“

A. Tsodikov, St. Petersburg University

“Rechnen in komplexen Geometrien“

J. Weidner, IBM Heidelberg

“Computeralgebra und Ingenieurmathematik - Beispiele mit Maple“

H. Bauer, FHT Reutlingen