

Entwicklung einer adaptiven
Neuro-Fuzzy-Regelung für ein
leistungskompensiertes Reaktionskalorimeter

Von der Gemeinsamen Naturwissenschaftlichen Fakultät
der Technischen Universität Carolo-Wilhelmina
zu Braunschweig

zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

genehmigte
D i s s e r t a t i o n

von Andre Lemke
aus Gifhorn

1. Referent: Prof. Dr. A. Löwe
2. Referent: Prof. Dr. A. Schumpe

eingereicht am: 09.08.01

mündliche Prüfung (Disputation) am: 06.11.01

Druckjahr: 2001

Meinen Eltern

„Es muß hier enden.“

Arnold Schwarzenegger als T-101
in Terminator 2 - Judgement Day
von James Cameron, 1991

Danksagung:

Die vorliegende Arbeit wurde in der Zeit von Januar 1999 bis Juli 2001 am Institut für Technische Chemie, Abteilung Reaktionstechnik, der Technischen Universität Carolo-Wilhelmina zu Braunschweig angefertigt.

Allen, die zum Gelingen dieser Arbeit beigetragen haben, möchte ich herzlich danken, insbesondere

meinem Doktorvater, Herrn Univ.-Prof. Dr. A. Löwe, für die Anregung zu dieser Arbeit und seine Betreuung,

Herrn Univ.-Prof. Dr. A. Schumpe für die Übernahme des Korreferats,

allen Mitarbeitern des Instituts für Technische Chemie für ihre ständige Diskussions- und Hilfsbereitschaft,

dem Leiter der Institutswerkstatt, Herrn Rolf Harrendorf, für die zügigen Umbauten an der Versuchsanlage,

und Herrn Prof. Dr. Ph. A. Bopp (Université Bordeaux I, LPCM) vor allem für die stete Diskussionsbereitschaft der Trainingsalgorithmen.

Ich danke ferner der Vereinigten Stiftung der Technischen Universität Braunschweig für die finanzielle Unterstützung durch das Hilpert-Stipendium.

Zusammenfassung

Die vorliegende Arbeit basiert auf dem apparativen Aufbau eines leistungskompensierten Reaktionskalorimeters, das HORN in seiner Arbeit [51] vorstellt. Das Meßprinzip der leistungskompensierten Reaktionskalorimetrie besteht darin, die Reaktortemperatur mittels einer Kompensationsheizung auf einem konstanten Sollwert über der Temperatur des umgebenden Kühlmantels zu halten. Über die schnell ansprechende Heizung läßt sich die Reaktortemperatur während einer chemischen Reaktion besser regeln, als dies über eine Temperierung der Manteltemperatur möglich wäre.

Aufgrund der besonderen Konstruktion des Reaktors, wobei zwei Heizungen im Reaktorinnenraum als Strömungsbrecher dienen, wird in Mischzeit- und Wärmedurchgangsmessungen bestimmt, welches Rührsystem am ehesten das Verhalten eines idealen Rührkessels gewährleistet. Als Rührerarten werden ein Ankerrührer, ein Schrägblattrührer und eine Kombination aus beiden Rührerformen untersucht.

Durch die rechnergestützte Meßdatenerfassung und Regelung der Reaktortemperatur wird die Möglichkeit geschaffen, aufwendigere Regelalgorithmen als einen Standard-PID-Regler einzusetzen. Daher wird in dieser Arbeit ein Fuzzy-Logik-Regler entworfen und auf die Bedürfnisse der Reaktionskalorimetrie optimiert.

Hierzu wird neben der aufwendigen, experimentellen Optimierung an der Anlage auch die Optimierung anhand eines Ein-/Ausgangsmodells beschrieben. Das Modell soll in der Lage sein, unterschiedliches Streckenverhalten (Totzeit, Ordnung, Streckenverstärkung) zu beschreiben, weshalb auf die Möglichkeiten der experimentellen Modellbildung mit künstlichen neuronalen Netzen zugegriffen wird.

Als robuster Regler ist ein Fuzzy-Logik-Regler zwar fähig, Streckenparameteränderungen in gewissen Grenzen zu tolerieren. Bei einer Reaktion im Satzbetrieb besteht jedoch die Möglichkeit das diese Grenzen verlassen werden. Die Übersetzung des Fuzzy-Logik-Reglers in einen anpassungsfähigen Neuro-Fuzzy-Regler ist die logische Konsequenz dieser Anforderung. Der Neuro-Fuzzy-Regler wird mit einem Adaptionsalgorithmus ausgestattet, so daß ein robustes Regelkonzept um ein adaptives Verfahren ergänzt wird.

Die Leistungsfähigkeit des Reaktionskalorimeters und der modelladaptiven Regelung wird durch die Lösungspolymerisation von Methylmethacrylat in Toluol bei unterschiedlichen Temperaturen unter Beweis gestellt.

Bei den in dieser Arbeit eingesetzten Verfahren zur Regelung (Fuzzy-Logik- bzw. Neuro-Fuzzy-Regler), zur Systemidentifikation (künstliche neuronale Netze) und zur Optimierung (genetische Algorithmen, Toleranzschwellenverfahren) handelt es sich um relativ moderne Verfahren, so daß auch die Literatur zu diesen Themen noch eine Vielzahl unterschiedlicher Ansätze beschreibt. Um einen Einstieg in diese Vielfalt zu erleichtern, werden die hier angewendeten Methoden im Anhang dieser Arbeit ausführlich aufgezeigt.

Inhaltsverzeichnis

1	Einleitung	1
2	Leistungskompensierte Reaktionskalorimetrie	3
2.1	Reaktionstechnische Grundlagen der leistungskompensierten Reaktionskalorimetrie	3
2.2	Apparativer Aufbau des Reaktionskalorimeters.....	9
2.2.1	Aufbau der Anlage	12
2.2.2	Rechnergestützte Meßdatenerfassung.....	15
2.3	Homogenisieren des Reaktionsmediums	18
2.4	Wärmedurchgang durch die Behälterwand	21
3	Regelung der Reaktortemperatur	25
3.1	Theoretische Grundlagen.....	25
3.1.1	Fuzzy-Logik-Regler	26
3.1.2	Künstliche Neuronale Netze zur Systemidentifikation	34
3.1.3	Adaptive Regelung.....	41
3.2	Implementierung am Reaktionskalorimeter.....	50
3.2.1	Grundentwurf eines Fuzzy-Logik-Reglers	50
3.2.2	Optimierung der Skalierungsfaktoren des Fuzzy-Logik-Reglers mittels eines Genetischen Algorithmus.....	57
3.2.3	Identifikation der Regelstrecke	67
3.2.4	Modellgestützte Validierung der Skalierungsfaktoren und Parameteroptimierung des Fuzzy-Logik-Regler	80
3.2.5	Übersetzung des Fuzzy-Logik-Reglers in ein Neuro-Fuzzy-Regler	88
3.2.6	Das modelladaptive Regelungskonzept.....	92

4	Messungen und Diskussion	99
4.1	Mischzeitexperimente.....	99
4.2	Untersuchung des Wärmeübergangs mit der Wilson-Methode	103
4.3	Kinetische Untersuchung der Lösungspolymerisation von Methyl- methacrylat.....	111
5	Ausblick	124
6	Literaturverzeichnis	126
7	Symbol- und Abkürzungsverzeichnis.....	130
8	Anhänge	135
	Anhang A: Fuzzy-Logik-Systeme.....	136
	Anhang B: Künstliche Neuronale Netze.....	151
	Anhang C: Neuro-Fuzzy-Systeme.....	176
	Anhang D: Genetische Algorithmen	197

Abbildungsverzeichnis

Abb. 1	Schematische Darstellung der Wärmebilanz eines Rührkessels.....	6
Abb. 2	Schematische Darstellung der beiden Regelkreise a und b in einem leistungskompensierten Wärmebilanzkalorimeter	10
Abb. 3	Regelkreis aus Fuzzy-Logik-Regler und Regelstrecke	28
Abb. 4	Seriell-parallele Systemidentifikation mit einem dreischichtigen Perzeptron.....	37
Abb. 5	Regelkreis mit paralleler Prozeßidentifikation und adaptivem Neuro-Fuzzy-Regler	47
Abb. 6	Erweiterter Adaptionsalgorithmus	49
Abb. 7	Auftragung der Terme A_i (<i>Temperaturabweichung</i>).....	52
Abb. 8	Auftragung der Terme B_j (<i>Temperaturänderung</i>)	53
Abb. 9:	Auftragung der Terme C_k (<i>Heizleistungsänderung</i>)	54
Abb. 10	Temperaturverlauf bei Aufgabe einer rechteckförmigen Kalibrierheizleistung.....	56
Abb. 11	Zeitlicher Verlauf der Kalibrierheizleistung q_{cal}	57
Abb. 12	Parametersätze der Startpopulation bei Wasser	61
Abb. 13	Alle beurteilten Parametersätze bei Wasser	62
Abb. 14	Gütekriterium J des besten Parametersatzes als Funktion der Generationen.....	62
Abb. 15	Parametersätze der Startpopulation bei 6%iger CMC-Lösung.....	64
Abb. 16	Alle beurteilten Parametersätze bei 6%iger CMC-Lösung.....	65
Abb. 17	Gütekriterium J des besten Parametersatzes als Funktion der Generationen.....	66
Abb. 18	Führungsgrößen- und Störgrößenprogramm bei einer Reaktorbefüllung mit Wasser	68
Abb. 19	Führungsgrößen- und Störgrößenprogramm bei einer Reaktorbefüllung mit 6%iger CMC-Lösung.....	69
Abb. 20	Durchschnittlicher <i>rms</i> -Fehler der Modellbildung bei Wasser als Reaktorbefüllung, Fehlerrückführungsalgorithmus	71
Abb. 21	Durchschnittlicher <i>rms</i> -Fehler der Modellbildung bei 6%iger CMC-Lösung als Reaktorbefüllung, Fehlerrückführungsalgorithmus	72
Abb. 22	Durchschnittlicher <i>rms</i> -Fehler (Ausschnitt) der Modellbildung bei 6%iger CMC-Lösung als Reaktorbefüllung.....	72
Abb. 23	Differenz der <i>rms</i> -Fehler des Fehlerrückführungsalgorithmus und des mit Simulated Annealing ergänzten Verfahrens (Wasser)	74

Abb. 24	Differenz der <i>rms</i> -Fehler des Fehlerrückführalgorithmus und des mit Simulated Annealing erg. Verfahrens (6%ige CMC-Lösung) ..	74
Abb. 25	Durchschnittlicher <i>rms</i> -Fehler nach 500 Epochen bei unterschiedl. Lernraten und Momentumterm-Konstanten (Wasser)	76
Abb. 26	Durchschnittlicher <i>rms</i> -Fehler nach 500 Epochen bei unterschiedl. Lernraten und Momentumterm-Konstanten (6%ige CMC-Lösung)	76
Abb. 27	Gemessene und simulierte Temperaturwerte (Wasser) bei einer Modellierung mit einem [14-5-1] Mehrschicht-Perzeptron	77
Abb. 28	Absolute Häufigkeit der Abweichung zwischen gemessener und simulierter Temperatur (Wasser)	78
Abb. 29	Gemessene und simulierte Temperaturwerte (6% CMC) bei einer Modellierung mit einem [14-5-1] Mehrschicht-Perzeptron	79
Abb. 30	Absolute Häufigkeit der Abweichung zwischen gemessener und simulierter Temperatur (6% CMC)	79
Abb. 31	Gütekriterium <i>J</i> in der Umgebung der optimalen Skalierungsfaktoren, Simulation der Reaktorbefüllung mit Wasser	81
Abb. 32	Gütekriterium <i>J</i> in der Umgebung der optimalen Skalierungsfaktoren, Simulation der Reaktorbefüllung mit 6% CMC.....	82
Abb. 33	Erweitertes Störgrößenprogramm für die Parameteroptimierung des Fuzzy-Logik-Reglers.....	83
Abb. 34	Auftragung der Terme <i>A_i</i> und <i>B_j</i> der linguistischen Variablen <i>Temperaturabweichung</i> und <i>Temperaturänderung</i>	86
Abb. 35	Kennfeld des in Kapitel 3.2.4 optimierten Fuzzy-Logik-Reglers	89
Abb. 36	Kennfeld des anhand des Kennfelds in Abb. 35 trainierten Neuro-Fuzzy-Reglers.....	90
Abb. 37	Auftragung der Terme <i>A_i</i> und <i>B_j</i> der linguistischen Variablen <i>Temperaturabweichung</i> und <i>Temperaturänderung</i>	91
Abb. 38	Berechnete Änderung der Heizleistungsänderung $\Delta\Delta q_{comp}$	95
Abb. 39	Beispiel für einen akzeptierten und einen zurückgewiesenen Adaptionsschritt.....	98
Abb. 40	Mischzeitcharakteristiken bei Hauptkomponente Glycerin.....	102
Abb. 41	Mischzeitcharakteristiken bei Hauptkomponente 85%ige Glycerinlösung.....	102
Abb. 42	Wilson-Plots bei unterschiedlichen Rührerformen ($T_r = 80\text{ °C}$, $V_r = 450\text{ mL}$, Toluol).....	105
Abb. 43	Wilson-Plot: Ankerrührer mit aufgesetztem Schrägblattrührer $T_r = 80\text{ °C}$ (+), $T_r = 90\text{ °C}$ (-)	106

Abb. 44	Wilson-Plot: Ankerrührer mit aufgesetztem Schrägblattrührer $T_r = 80\text{ °C}$ (+), $T_r = 90\text{ °C}$ (·)	107
Abb. 45	Vergleich von berechneten und gemessenen Wärmeübergangskoeffizienten a_r bei unterschiedlichen Drehzahlen N	109
Abb. 46	Kalibrierung des Reaktionskalorimeters anhand einer Rechteckfunktion der Kalibrierheizleistung q_{cal}	116
Abb. 47	Kalibrierung des Reaktionskalorimeters und Test des Neuro-Fuzzy-Reglers anhand einer Rampenfunktion.....	117
Abb. 48	Verlauf der Reaktortemperatur T_r bei Kalibrierung anhand der Rampenfunktion.....	117
Abb. 49	Verlauf der Reaktortemperatur T_r bei der Lösungspolymerisation von Methylmethacrylat	119
Abb. 50	Anfängliche Wärmeproduktionsraten q_r der MMA-Polymerisationen bei den verschiedenen Reaktortemperaturen.....	120
Abb. 51	Zeitlicher Verlauf des reaktionsseitigen Wärmeübergangskoeffizienten a_r während der MMA-Polymerisation bei $T_r = 80\text{ °C}$	122
Abb. 52	Logarithmierte Wärmeproduktionsrate $\ln(q_r)$ der MMA-Polymerisation bei $T_r = 85\text{ °C}$	123
Abb. 53	Arrhenius-Auftragung von $\ln(k_{br})$ gegen $1/T_r$ für die Lösungspolymerisation von Methylmethacrylat.....	123
Abb. 54	Funktionsweise eines Fuzzy-Logik-Systems	143
Abb. 55	Dreischichtiges neuronales Netz (Mehrschicht-Perzeptron)	152
Abb. 56	Dreischichtiges neuronales Netz mit nur einem Ausgabeneuron.	157
Abb. 57	Simuliertes Abkühlen.....	173
Abb. 58	Vergleich von dreieckförmiger Funktion und Gauß-Funktion mit gleichem Mittenwert x_m und gleicher Breite a	179
Abb. 59	Vergleich von Gauß-Trapez-Funktion und s-förmiger Funktion mit gleichen Parametern	180
Abb. 60	Verarbeitungsschritte in einem Fuzzifizierungsneuron.....	182
Abb. 61	Grafische Darstellung eines Neuro-Fuzzy-Systems mit zwei Eingängen und einem Ausgang.....	183
Abb. 62	Genetischer Algorithmus.....	198

Tabellenverzeichnis

Tab. 1: Adaptive Regelstrategien	44
Tab. 2: Übersicht der Eingangs- und Ausgangsgrößen.....	50
Tab. 3: Linguistische Variable: <i>Temperaturabweichung</i> (Grundentwurf)....	52
Tab. 4: Linguistische Variable: <i>Temperaturänderung</i> (Grundentwurf)	53
Tab. 5: Linguistische Variable: <i>Heizleistungsänderung</i> (Grundentwurf)	54
Tab. 6: Matrix der vollständigen und konsistenten Regelbasis	55
Tab. 7: Zeiten des Störgrößenverlaufs von q_{cal} bei Reaktorbefüllung mit Wasser	60
Tab. 8: Anfänglicher Wertebereich der Skalierungsfaktoren (Wasser).....	60
Tab. 9: Zeiten des Störgrößenverlaufs von q_{cal} bei Reaktorbefüllung mit 6%iger CMC-Lösung.....	63
Tab. 10: Anfänglicher Wertebereich der Skalierungsfaktoren (6% CMC)	64
Tab. 11: Optimierte Skalierungsfaktoren.....	80
Tab. 12: Wertebereiche für die Terme der Startpopulation.....	85
Tab. 13: Linguistische Variable: <i>Heizleistungsänderung</i> (optimaler FLR) ..	86
Tab. 14: Regelbasis des optimalen Fuzzy-Logik-Reglers.....	87
Tab. 15: Wertebereiche des Reglerkennfelds	88
Tab. 16: Lernraten für die Anpassung des Neuro-Fuzzy-Reglers	90
Tab. 17: Linguistische Variable: <i>Heizleistungsänderung</i> (optimaler NFR) ..	91
Tab. 18: Anzahl Epochen bei unterschiedlichem Umfang des Trainings- datensatzes.....	93
Tab. 19: Lernraten für die <i>Online</i> -Anpassung des Neuro-Fuzzy-Reglers.....	97
Tab. 20: Dichte und dynamische Viskosität der Hauptkomponente und Mischung.....	99
Tab. 21: Gemessene Mischzeiten.....	100
Tab. 22: Parameter der Wilson-Plots.....	107
Tab. 23: Berechnete und gemessene Wärmeübergangskoeffizienten für Toluol	108
Tab. 24: Totzeiten bei Verwendung unterschiedlicher Rührer	110
Tab. 25: Reaktionsenthalpie $\Delta_r H$ bei verschiedenen Temperaturen.....	121
Tab. 26: Geschwindigkeitskonstante bei verschiedenen Temperaturen	122

1 Einleitung

Die Reaktionskalorimetrie ist eine Methode zur Erfassung einer chemischen Reaktion anhand ihrer zeitlichen Wärmeentwicklung. Sie ermöglicht es, aufgrund eines Reaktionsvolumens über 0.2 L technisch interessante Effekte, wie das Rühren oder die Dosierung von Reaktionskomponenten, in die Reaktionsführung zu integrieren. Solche Effekte müssen bei den thermoanalytischen Mikromethoden, wie der Differenz-Thermo-Analyse (DTA) oder der Dynamischen Wärmestrom-Differenz-Kalorimetrie (DSC), unberücksichtigt bleiben. Aufgrund der geometrischen Ähnlichkeit mit technischen Reaktoren liefert die Untersuchung im Reaktionskalorimeter die kinetischen und kalorischen Daten, die für die Auslegung einer Anlage in größerem Maßstab (Scale-Up) notwendig sind.

Am häufigsten eingesetzt werden isotherm arbeitende Reaktionskalorimeter. Sie liefern als primäre Meßgröße die Wärmeproduktionsrate einer Reaktion. Wenn keine weiteren thermischen Effekte (Kristallisation, Verdampfung) berücksichtigt werden müssen, kann über eine Integration die insgesamt freigesetzte Wärmemenge erhalten werden. Findet im Reaktionsmedium nur eine Reaktion statt oder ist nur eine Reaktion in ihrer Wärmeentwicklung dominierend, so kann die Anpassung an einen reaktionskinetischen Ansatz erfolgen.

Die leistungskompensierte Reaktionskalorimetrie regelt die Temperatur der Reaktionsmasse über eine schnell ansprechende Kompensationsheizung im Reaktorinnenraum. Auf diese Weise ist sie in der Lage, ein streng-isothermes Verhalten zu realisieren, was bei einer Temperierung über die Manteltemperatur nicht der Fall ist. Aufgrund der Besonderheit der Einflußnahme auf die Temperatur ist verständlich, daß mit einer verbesserten Regelung ein konstanter Temperaturverlauf auch unter dem Einwirken erheblicher Störgrößen gewährleistet werden kann. Dann kann die Wärmeakkumulation im Reaktionsmedium vernachlässigt werden. In dieser Arbeit wird daher der Bereich der Regelung mit einem implementierten PI-Regler verlassen und das Regelkonzept eines Fuzzy-Logik-Reglers (Kennfeldregler) als erfolgversprechender weiterverfolgt.

Je nach Reaktorbefüllung oder Reaktionsfortschritt ändern sich die Durchmischungs- und Wärmeübergangsverhältnisse, so daß sich auch die Parameter der Regelstrecke (Heizung – Reaktionsmedium – Temperatursensor) verändern. Aus diesem Grund wird ein zweiter Schwerpunkt auf

eine experimentelle Modellbildung der Regelstrecke gelegt. Hier wird auf die Möglichkeit einer Systemidentifikation mit künstlichen neuronalen Netzen zurückgegriffen, die als *black box*-Modelle die geringsten Anforderungen an eine Streckenparameterschätzung stellen.

Mit einem Modell der Regelstrecke kann zunächst eine Optimierung des Fuzzy-Logik-Reglers *offline* vorgenommen werden, so daß er bestens für die Regelung eines statischen Systems ausgelegt ist. Die Temperaturregelstrecke eines satzweise betriebenen Reaktors unterliegt – einhergehend mit der Reaktion – meist Parameteränderungen. Die Verwendung eines Neuro-Fuzzy-Reglers, ergänzt um einen Adaptionalgorithmus, schafft ein robustes und adaptives Regelkonzept, welches dem Meßprinzip der leistungskompensierten Reaktionskalorimetrie einen weiteren Vorteil gegenüber anderen isothermen Reaktionskalorimetern verleiht.

2 Leistungskompensierte Reaktionskalorimetrie

Die leistungskompensierte Reaktionskalorimetrie mit Bilanzierung des Kühlmantels hat sich aus der Wärmebilanzkalorimetrie mit Regelung über die Manteltemperatur entwickelt. Durch Einführung einer Differentialkühlung [52] ist sie bezüglich des Meßprinzips und notwendig zu machender Vereinfachungen zur vorteilhaftesten Betriebsweise eines Reaktionskalorimeters geworden. In Kapitel 2.1 werden zunächst die reaktionstechnischen Grundlagen der leistungskompensierten Reaktionskalorimetrie dargelegt, um anschließend in Kapitel 2.2 auf die technische Realisierung dieses Meßprinzips einzugehen. Der apparative Aufbau basiert auf dem von HORN [51] erstellten Kalorimeter, das die Weiterentwicklung des von SCHLEGEL [52] vorgestellten Typs ist. Reaktionskalorimetrische Messungen verlangen, daß sich der Reaktor im Satz- oder Teilfließbetrieb wie ein idealer Rührkessel verhält, weshalb in Kapitel 2.3 auf die Grundlagen des Homogenisierens der Reaktionsmischung (Rühren und Mischen) eingegangen wird. Die Erfassung von Wärmeströmen ist Grundvoraussetzung der Kalorimetrie. Kapitel 2.4 ist daher der Intensivierung des Wärmetransports sowohl von der Kompensationsheizung zum Reaktionsmedium als auch vom Reaktionsmedium auf die Behälterwand gewidmet.

2.1 Reaktionstechnische Grundlagen der leistungskompensierten Reaktionskalorimetrie

Um die Wärmeproduktionsrate q_r als die interessierende Größe einer kalorimetrischen Messung zu erhalten, ist eine Bilanzierung der Masse- und Energieflüsse notwendig. Als Reaktor des Reaktionskalorimeters dient ein doppelwandiger Rührkesselreaktor mit Rühr- und Dosiervorrichtung. Für die Aufstellung der Bilanzgleichungen wird davon ausgegangen, daß sich der Reaktor wie ein idealer Rührkessel verhält, in dem aufgrund der durch den Rührer verursachten turbulenten Strömung weder Konzentrations- noch Temperaturgradienten vorhanden sind. Als Bilanzvolumen wird das Reaktionsvolumen V_r gewählt.

Der Rührkesselreaktor des Reaktionskalorimeters kann im Satz- oder im Teilfließbetrieb betrieben werden. Die Möglichkeit, den Reaktor im Fließbetrieb zu verwenden, ist nicht vorgesehen.

Die Stoffbilanz der Komponente i für einen idealen Rührkessel im Satzbetrieb ergibt sich als zeitliche Änderung der Stoffmenge im Bilanzraum. Die Stoffmengenänderung ist wegen des Satzbetriebs lediglich auf eine Änderung durch Reaktion beschränkt.

$$\frac{dn_i}{dt} = V_r R_i \quad (1)$$

Darin ist R_i die Stoffmengenänderungsgeschwindigkeit, die alle im Reaktionsvolumen ablaufenden Reaktionen j erfaßt, an denen die Komponente i mit dem stöchiometrischen Faktor n_{ij} beteiligt ist.

$$R_i = \sum_j n_{ij} r_j \quad (2)$$

Für den Teilfließbetrieb wird die Stoffbilanz um einen weiteren Term ergänzt, um die Stoffzufuhr oder -abfuhr der Komponente i im Verlauf der Reaktion zu berücksichtigen.

$$\frac{dn_i}{dt} = \dot{n}_i^0 + V_r R_i \quad (3)$$

Bei der Stoffzufuhr oder -abfuhr tritt ein an den Massestrom gekoppelter Energiestrom auf. Für die folgende Wärmebilanzierung müssen daher von der Komponente i die physikalischen Eigenschaften, wie z.B. die Wärmekapazität, bekannt sein. Dies ist bei einer Stoffzufuhr, bei der man die dosierte Komponente kennt, eher gewährleistet als bei einer Stoffaustragung von Komponenten aus dem Reaktionsvolumen. Die Stoffzufuhr ist der relevantere Fall in der Reaktionskalorimetrie.

Läuft im Bilanzvolumen nur eine einzige Reaktion ab, oder ist nur eine Reaktion in ihrer Wärmeentwicklung dominierend, so können aus reaktionskalorimetrischen Messungen auch kinetische Daten für die Reaktion ermittelt werden. Eine Zuordnung der gemessenen Wärmeproduktionsrate q_r zu mehreren gleichzeitig stattfindenden Reaktionen ist nur mit zusätzlicher *Online*-Analytik möglich, wobei die kinetischen Daten dann auch aus den zusätzlich ermittelten Meßwerten bestimmt werden können.

Neben der Stoffbilanz wird für eine reaktionskalorimetrische Auswertung eine Wärmebilanz aufgestellt. Diese Bilanz ist das Ergebnis der Enthalpiebilanz der im Rührkessel des Reaktionskalorimeters ablaufenden Prozesse:

$$q_{accu} = q_r + q_{dos} + q_{comp} + q_{cal} - q_{fr} + q_{sec} \quad (4)$$

Die Wärmeströme sind mit ihrer Richtung in Abbildung 1 dargestellt. Auf sie soll im einzelnen eingegangen werden, wobei sich im Rahmen der leistungskompensierten Reaktionskalorimetrie einige Vereinfachungen ergeben.

Der Akkumulationsterm q_{accu} erfaßt die Wärmeaufnahme oder -abgabe der Reaktionsmasse, welche durch eine Temperaturerhöhung oder -erniedrigung von T_r verursacht wird. Das hierbei auftretende Produkt aus spezifischer Wärmekapazität und Masse $(m \cdot c_p)_r$ enthält neben den Wärmekapazitäten der Reaktionsedukte und -produkte auch die Wärmekapazitäten der Reaktoreinbauten, wie Strömungsbrecher und Rührer.

$$q_{accu} = (m \cdot c_p)_r \frac{dT_r}{dt} \quad (5)$$

Die Zielgröße einer kalorimetrischen Messung ist die Wärmeproduktionsrate q_r . Sie ist das Produkt aus Reaktionsvolumen V_r , der volumenbezogenen Reaktionsgeschwindigkeit r und der Enthalpie der Reaktion $\Delta_r H$.

$$q_r = V_r \cdot r \cdot (-\Delta_r H) \quad (6)$$

Dieser direkte Zusammenhang zwischen gemessener Wärmeproduktionsrate q_r und der Reaktionsgeschwindigkeit r gilt nur, wenn keine weiteren thermischen Effekte (Mischungs-, Lösungs- oder Kristallisationswärmen) berücksichtigt werden müssen.

In q_{dos} wird der Wärmefluß berücksichtigt, der durch die Dosierung von Reaktionskomponenten mit der Temperatur T_{dos} verursacht wird. Der Mittelwert von c_p resultiert einerseits aus unterschiedlichen spezifischen Wärmekapazitäten einer Komponente bei den Temperaturen T_{dos} und T_r und andererseits aus der Mittelwertbildung über die Wärmekapazitäten verschiedener Reaktionskomponenten.

$$q_{dos} = (\dot{m} \cdot \bar{c}_p)_{dos} \cdot (T_{dos} - T_r) \quad (7)$$

Im Reaktionsraum sind eine Kompensations- und eine Kalibrierheizung eingebaut, deren Leistungseinträge durch q_{comp} und q_{cal} in der Wärmebilanz

berücksichtigt sind. Unter Berücksichtigung des Akkumulationsterms für das Heizungsmaterial ergibt sich:

$$q_c = U \cdot I - (m \cdot c_p)_c \cdot \frac{dT_c}{dt} \quad (8)$$

Kann die Wärmekapazität der Heizung als vernachlässigbar klein angesehen werden, oder ist die Ableitung der Heizungstemperatur nach der Zeit ausreichend klein, was im stationären Zustand der Fall ist, so kann der zweite Term vernachlässigt werden, und die Gleichung vereinfacht sich zu:

$$q_c = U \cdot I \quad (9)$$

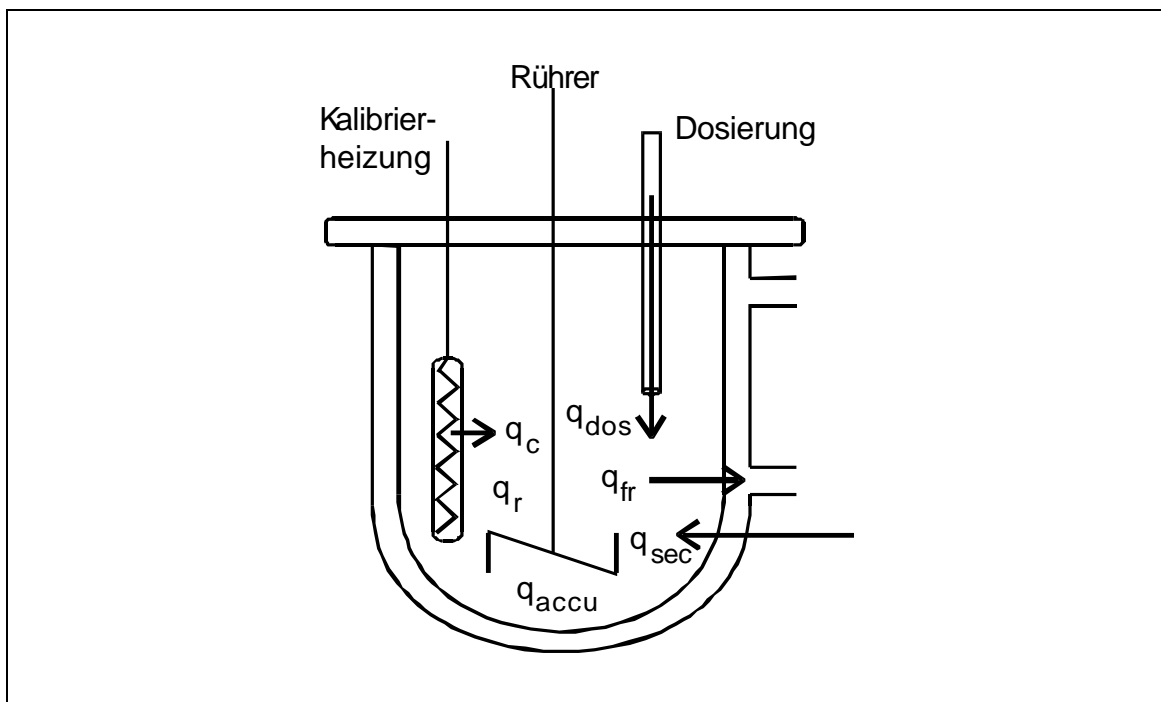


Abb. 1 Schematische Darstellung der Wärmebilanz eines Rührkessels

Der Ausdruck q_{fr} beschreibt den Wärmestrom vom Reaktionsmedium auf die Reaktorwand:

$$q_{fr} = a_r \cdot A \cdot (T_r - T_{wr}) \quad (10)$$

mit T_{wr} : Temperatur der Wandinnenseite, a_r : reaktionsseitiger Wärmeübergangskoeffizient, A : Wärmeaustauschfläche.

Die Wärmebilanz der Reaktorwand im instationären Zustand wird durch die Fouriersche Wärmeleitungsgleichung beschrieben.

$$\frac{\partial T_w}{\partial t} = \left(\frac{l}{r \cdot c_p} \right)_w \cdot \frac{\partial^2 T_w}{\partial x^2} = a \cdot \frac{\partial^2 T_w}{\partial x^2} \quad (11)$$

Die gültigen Anfangs- und Randbedingungen für eine Reaktorwand der Dicke d_w lauten:

$$\begin{aligned} t = 0 : T_w &= T_w(0) \\ x = 0 : a_r (T_r - T_{w,x}) &= -l_w \cdot \frac{\partial T_{w,x}}{\partial x} \\ x = d_w : a_k (T_{w,x} - T_k) &= -l_w \cdot \frac{\partial T_{w,x}}{\partial x} \end{aligned} \quad (12)$$

Dieses Gleichungssystem beschreibt die Reaktorwand, die zu Anfang eine Temperatur T_w besitzt und anschließend sowohl reaktionsseitig als auch mantelseitig einem Wärmeübergang ausgesetzt wird, wodurch sich ein zeitlich veränderliches Temperaturprofil ergibt.

Hat sich bei konstanter Reaktortemperatur T_r , konstanter Manteltemperatur T_j und einem konstanten Wärmeübergangskoeffizienten α_r ein stationärer Zustand eingestellt, ist der Wärmestrom vom Reaktor auf die Reaktorwand gleich dem Wärmedurchgang vom Reaktor auf das Kühlmedium des Reaktormantels:

$$q_{fr} = q_j = h \cdot A \cdot (T_r - T_j) \quad (13)$$

Darin enthalten ist der Wärmedurchgangskoeffizient h , dessen Kehrwert sich additiv aus den Kehrwerten der reaktions- und mantelseitigen Wärmeübergangskoeffizienten a_r und a_j sowie dem Quotienten aus Reaktorwanddicke d_w und Wärmeleitfähigkeit l zusammensetzt.

$$\frac{1}{h} = \frac{1}{a_r} + \frac{d_w}{l_w} + \frac{1}{a_j} \quad (14)$$

In (13) tritt die Manteltemperatur T_j auf, welche repräsentativ für die Temperatur im gesamten Mantel sein muß. Bei einem doppelwandigen Reaktor kann diese Anforderung entweder dadurch erfüllt werden, daß die Strömungsgeschwindigkeit im Kühlmantel so hoch gewählt wird, daß kein nennenswerter Temperaturgradient zwischen Manteleingang und -ausgang

besteht, oder durch ein Verweilzeitverhalten des Kühlmediums im Mantel, das dem Verweilzeitverhalten eines idealen kontinuierlichen Rührkesselreaktors entspricht.

Nachteil der Bestimmung des Wärmestroms nach (13) ist auch, daß neben der Wärmeaustauschfläche A auch der Wärmedurchgangskoeffizient h bekannt sein oder durch Kalibriermessungen bestimmt werden muß. Gelöst wird dieses Problem durch die adiabatische Isolierung des Kühlmantels gegenüber der Umgebung, so daß dann eine Bilanzierung des Mantels selbst möglich wird:

$$q_j = (\dot{m} \cdot c_p)_j \cdot (T_j - T_{j,0}) \quad (15)$$

Dabei müssen für den Betrieb des Kühlkreislaufes zwei entgegengesetzte Anforderungen berücksichtigt werden: Die Strömungsgeschwindigkeit im Kühlmantel sollte für einen guten Wärmeübergang hoch sein, wodurch die Temperaturdifferenz ΔT_j zwischen Manteleingang und -ausgang verringert wird. Gleichzeitig muß die Temperaturdifferenz noch hinreichend groß sein, was durch eine niedrige Strömungsgeschwindigkeit erreicht wird. Auch hier wirkt sich ein Verweilzeitverhalten des Kühlmantels nahe am idealen Rührkessel positiv aus, da selbst bei größeren Temperaturdifferenzen ΔT_j keine Wärmeakkumulation im Mantel berücksichtigt werden muß und T_j für den gesamten Kühlmantel gültig ist.

Aufgrund der adiabatischen Isolierung muß die gesamte im Mantel ermittelte Leistung aus dem Reaktor stammen, so daß (13) und (15) denselben Wert haben. Die Äquivalenz beider Ausdrücke ermöglicht im Fall sich ändernder Wärmeübergangsverhältnisse bei bekannter Wärmeaustauschfläche A die Ermittlung des Wärmedurchgangskoeffizienten h im Verlauf einer chemischen Reaktion.

Der Term q_{sec} erfaßt sekundäre Wärmeeffekte. Darin enthalten sind mögliche Wärmeverluste an die Umgebung und die dissipierte Rührleistung.

Im stationären Zustand ($q_{accu} = 0$) vereinfacht sich die in (4) aufgestellte Bilanz unter Einsetzung der einzelnen Wärmestromterme zu:

$$0 = q_r + (U \cdot I)_{comp} + (U \cdot I)_{cal} - (\dot{m} \cdot c_p) \cdot (T_j - T_{j,0}) + q_{sec} \quad (16)$$

Wird durch die Anwendung moderner Regelungstechnik gewährleistet, daß die Reaktortemperatur T_r konstant den Sollwert $T_{r,soll}$ einhält, so kann der Akkumulationsterm q_{accu} immer gleich Null gesetzt und Gleichung (16) angewendet werden. Damit ist die experimentelle Bestimmung von $(m \cdot c_p)_r$ überflüssig, welche zusätzliche Fehler in die Auswertung einbringt.

An der Bilanzgleichung (16) ist die Vorteilhaftigkeit der Anbringung einer Kalibrierheizung im Reaktionsraum zu erkennen. Durch die Kalibrierheizung läßt sich eine genau definierte Heizleistung q_{cal} mit beliebigem zeitlichen Verlauf in das Reaktionsmedium einbringen. Auf diese Weise kann die Wärmeentwicklung einer Reaktion simuliert und die Regelung der Reaktortemperatur T_r getestet werden.

Darüber hinaus läßt sich über die Kalibrierheizung die zwischen Kompensationsheizleistung q_{comp} und Reaktortemperatur T_r befindliche Regelstrecke frequenzreich anregen, was für die experimentelle Modellbildung wichtig ist. Man ist nicht nur auf eine Systemanregung durch eine Führungsgrößenänderung in $T_{r,soll}$ beschränkt.

2.2 Apparativer Aufbau des Reaktionskalorimeters

Der Aufbau der Anlage entspricht bis auf einige Änderungen dem Aufbau, den HORN in seiner Arbeit vorgestellt hat. Es erfolgt hier nur eine verkürzte Darstellung. Für eine nähere Betrachtung sei auf [51] verwiesen.

Das leistungskompensierte isotherme Reaktionskalorimeter besteht, wie in Abbildung 2 schematisch dargestellt, aus zwei Regelkreisen, da nicht nur die Reaktortemperatur T_r , sondern auch die Manteltemperatur T_j auf einen jeweiligen Sollwert geregelt wird.

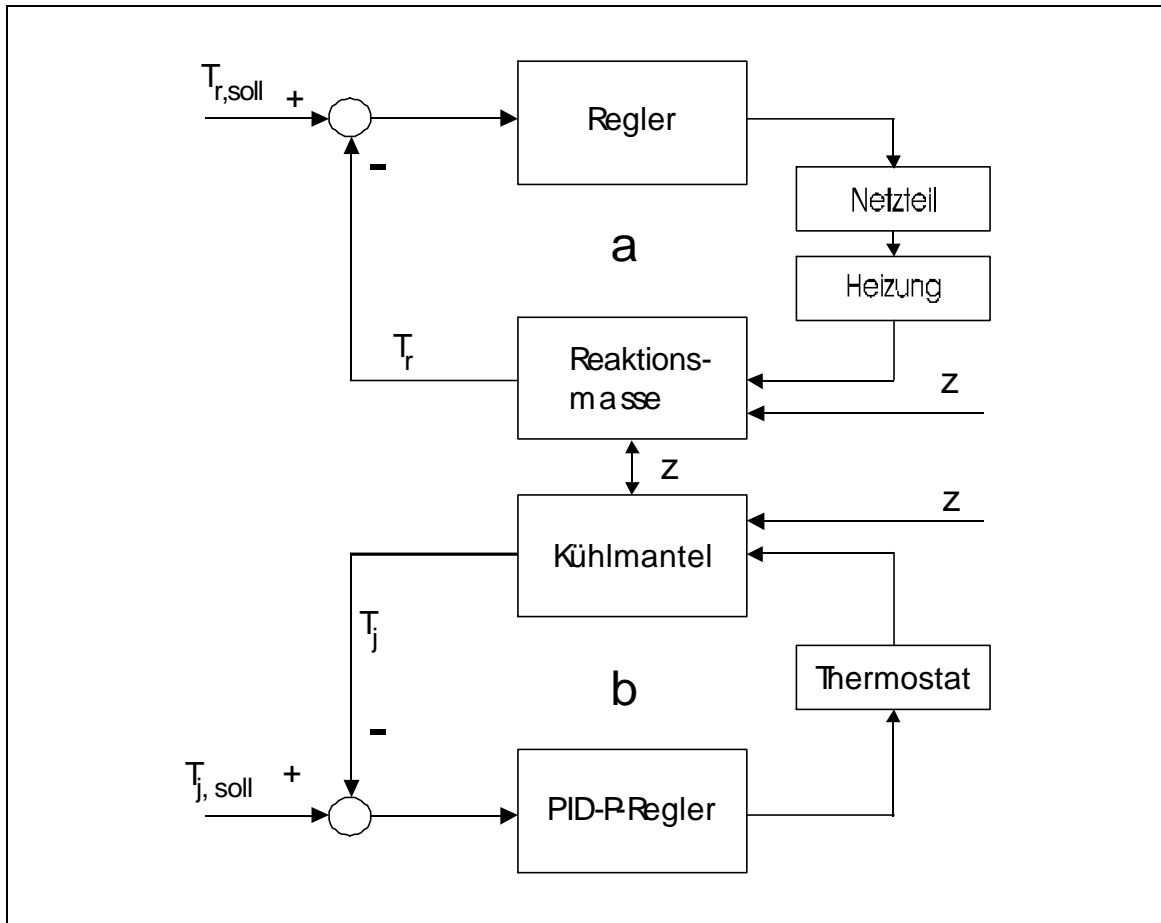


Abb. 2 Schematische Darstellung der beiden Regelkreise a und b in einem leistungskompensierten Wärmebilanzkalorimeter

Die Temperatur im Reaktorinnenraum T_r wird über eine im Reaktionsraum angebrachte Kompensationsheizung (Regelkreis a) und die Kühlmitteltemperatur T_j wird über einen Thermostaten geregelt (Regelkreis b). Als Kühlmedientemperatur wird die Mantelausgangstemperatur gewählt, da diese bei einem idealen Rührkesselverhalten des Mantels als repräsentativ angesehen werden kann. Der Sollwert der Reaktortemperatur $T_{r,soll}$ wird auf einen Wert eingestellt, der einige Kelvin über dem Sollwert der Manteltemperatur $T_{j,soll}$ liegt. So ist ein ständiger Wärmestrom vom Reaktionsraum in den Kühlmantel gewährleistet.

In beiden Regelkreisen greifen unterschiedliche Störgrößen z an. Störgrößen, die die Reaktortemperatur beeinflussen, sind Dosierung, chemische Reaktion, Wärmeströme in den Kühlmantel und an die Umgebung. Für die Kühlmitteltemperatur kommen als Störgrößen veränderliche Wärmeströme aus dem Reaktorinnenraum (aufgrund veränderter reaktionsseitiger Wärmeübergangsverhältnisse) und an die Umgebung in Betracht. Durch eine geeignete Konstruktion des Reaktionskalorimeters wird erreicht, daß

Störgrößen, welche die Auswertbarkeit einer kalorimetrischen Messung behindern, weitgehend ausgeschaltet sind. Wärmeverluste an die Umgebung werden für den Reaktionsraum durch einen temperierten Reaktordeckel unterbunden; der Kühlmantel ist von der Umgebung durch ein temperiertes Luftbad adiabatisch isoliert.

Weicht die Temperatur eines dosierten Reaktanden T_{dos} von der Reaktortemperatur T_r ab, so tritt ein an den Massestrom der Dosierung gekoppelter Wärmestrom q_{dos} auf. Es gibt daher zwei Vorkehrungen, um diesen Temperaturunterschied zu minimieren. Zum einen besteht die Möglichkeit, die Dosierungskomponente in einem auf T_r temperierten Vorratsgefäß vorzuwärmen, zum anderen befindet sich direkt am Reaktordeckel vor dem Eintritt der Komponente in den Reaktor ein Wärmetauscher, der ebenfalls auf T_r temperiert wird. Durch diesen Aufbau soll erreicht werden, daß der dosierte Reaktand bei Eintritt in den Reaktor bereits annähernd die Temperatur T_r hat, so daß die Regelung der Reaktortemperatur nur wenig beansprucht wird. Eine Temperierung des Reaktanden im Vorratsgefäß kann allerdings nur vorgenommen werden, wenn der auf T_r erwärmte Reaktand nicht schon zur Reaktion neigt, z.B. weil er im Dosierungsgefäß in höherer Konzentration vorliegt.

Aus dieser Darstellung wird deutlich, daß die Regelung der Kühlmedientemperatur, auch was zeitlich veränderliche Streckenparameter angeht, das kleinere Problem ist. Größere Anforderungen stellt die Regelung der Reaktortemperatur. Hierbei muß zunächst sichergestellt werden, daß die Rührströmung Konzentrations- und Temperaturgradienten schnell beseitigt, so daß die gemessene Reaktortemperatur T_r wirklich im gesamten Reaktionsvolumen herrscht. Sowohl die Dosierung einer Reaktionskomponente als auch der Leistungseingtrag über die Heizung ist ein auf einen engen Raum lokalisierter Vorgang. In beiden Fällen sorgt die turbulente Rührströmung für den nötigen Konzentrations- und Temperaturausgleich. Ist diese apparateseitige Voraussetzung erfüllt, kann mit neueren Methoden der Regelungstechnik auch bei anspruchsvollen Reaktionsverläufen eine hervorragende Konstanz der Reaktortemperatur gewährleistet werden.

2.2.1 Aufbau der Anlage

Die genaue Erfassung von Temperaturen und Masseflüssen bildet die Grundlage für die Bestimmung und Bilanzierung der Wärmeströme. Alle Temperaturen des Reaktionskalorimeters werden mit hochgenauen, vom Deutschen Kalibrierdienst (DKD) kalibrierten Pt100-Mantel-Widerstandsthermometern nach DIN IEC 751 erfaßt. Der Platin-Widerstandsdraht ist in eine Isolationsschicht aus Magnesium- und Aluminiumoxid eingebettet und zum Schutz vor chemischen und mechanischen Einwirkungen von einem Edelmantel umgeben. Die absolute Genauigkeit der Temperatursensoren liegt entsprechend der DKD-Kalibrierstelle im 0.01 K-Bereich. Die Meßschaltungen für die Pt100-Sensoren sind in Vierleitertechnik ausgelegt, damit weder die Leitungswiderstände noch ihre temperaturbedingten Änderungen in die Messung eingehen. Als passive Bauelemente werden Widerstandsthermometer von einem konstanten Strom von 1 mA durchflossen. Der Meßfehler durch Eigenerwärmung ist bei einer kalorimetrischen Messung vernachlässigbar.

Die Erfassung der Manteltemperaturen am Eingang $T_{j,0}$ und am Ausgang T_j zur Bildung der Temperaturdifferenz ΔT_j nach (15) erfolgt unter Verwendung eines selektierten Pt100-Paares. Die beiden Fühler besitzen sowohl einen ähnlichen Grundwiderstand bei 0 °C als auch einen ähnlichen Temperaturgang und sind in die Ein- und Auslaßstutzen des Kühlmantels eingebaut.

Zur Erfassung der Temperatur der dosierten Flüssigkeit T_{dos} wird ein speziell angefertigter, sehr empfindlicher Pt100-Temperatursensor verwendet. Da der Dosierstrom nur sehr gering ist, steht nur eine kleine Menge Flüssigkeit zur Umströmung des Temperatursensors zur Verfügung. Es wird daher ein Pt100-Mantelwiderstands-Sensor verwendet, der ohne die Isolationsschicht aus Magnesium- und Aluminiumoxid gefertigt wurde. Er zeichnet sich durch einen geringeren Durchmesser, höhere Empfindlichkeit und höhere Dynamik gegenüber den standardmäßig verwendeten Pt100-Sensoren aus.

Die Erfassung und digitale Wandlung aller Temperaturen erfolgt über ein Digitalmultimeter (DMM), das mit einem Meßstellenumschalter (Multiplexer) ausgestattet ist. Alle durchgeführten Messungen werden mit einer Auflösung der Temperaturen von 10^{-3} K durchgeführt.

Zur Ermittlung von q_j (15) und q_{dos} (7) müssen die Masseflüsse \dot{m}_j und \dot{m}_{dos} gemessen werden. Im Falle des strömenden Kühlmediums geschieht dieses

mit einem Masseflußmesser, der nach dem Coriolis-Prinzip arbeitet. Er besitzt den Vorteil, den Massefluß als primäre Meßgröße zu liefern. Die Meßunsicherheit beim verwendeten Coriolis-Masseflußmesser liegt bei weniger als 0.1 % des Meßwertes. Das Funktionsprinzip wird von SCHLEGEL in [52] beschrieben.

Die Erfassung des Masseflusses der Dosierflüssigkeit \dot{m}_{dos} erfolgt über Wiegen des temperierten Vorratsgefäßes, aus dem eine Dosierpumpe die Komponente in den Reaktor transportiert.

Der Glasreaktor besitzt ein Gesamtvolumen von 650 cm³. Er ist mit einem Kühlmantel ausgestattet und kann bei einem Druck von 5 bar und einer Temperatur bis 200 °C betrieben werden. Der Reaktor ist mit einem temperierbaren Edelstahldeckel nach oben abgeschlossen. Er besitzt ein Rühr- und Dosiersystem, ein Pt100-Mantelthermometer zur Erfassung von T_r und zwei Heizungen. Eine durch den Reaktordeckel hindurchgeführte Gaslanze und ein Rückschlagventil erlauben das Spülen des Reaktors mit einem Stickstoffstrom. Auf diese Weise kann der Reaktor getrocknet und eine inerte Atmosphäre über dem Reaktionsmedium geschaffen werden.

Das Rührsystem erfaßt und digitalisiert das Drehmoment des Rührers. Die Drehzahl wird am Gerät eingestellt und konstant geregelt. Die mit Gleitringdichtungen versehene Rührerwelle ist über eine Magnetkupplung mit dem Antriebsystem verbunden, so daß der Reaktionsraum hermetisch abgeschlossen ist. Auf der Rührerwelle lassen sich unterschiedliche Rührertypen anbringen, wobei Schrägblatt- und Ankerrührer näher untersucht werden. Auf zusätzliche Strömungsbrecher wird verzichtet, da die beiden installierten Heizungen und der Pt100-Sensor diese Funktion übernehmen und eine Trombenbildung wirksam verhindern. Zusätzliche Strömungsbrecher hätten das verfügbare Reaktionsvolumen weiter verringert.

Die zu dosierende Flüssigkeit wird in einem Vorratsgefäß, das sich auf einer Waage befindet, mittels eines Thermostaten auf T_r temperiert und mit einer Dosierpumpe zum Reaktor transportiert. Die Dosiervorrichtung ist in einer Öffnung im Deckel des Reaktors verschraubt und besteht aus einem Wärmetauscher und einem Teflonblock mit eingebettetem hochsensiblen PT100-Sensor. Der Dosierstrom fließt dann mit einem Gefälle von ca. 40° durch den Teflonblock und gelangt anschließend in den Reaktionsraum. Durch diese Konstruktion ist eine optimale Benetzung des Temperatur-

sensors mit der fließenden Flüssigkeit und damit eine genaue Messung von T_{dos} möglich.

Die im Reaktor installierten Heizungen (Kompensations- und Kalibrierheizung) bestehen aus je einem mineralisierten Widerstandsdraht mit sogenannten „kalten Enden“. Der Draht ist spiralförmig gewickelt und über eine Halterung am Reaktordeckel befestigt. Die Heizteillänge beträgt 2 m, der Außendurchmesser 1.5 mm. Bei einer rechnerischen Wärmeaustauschfläche von 94.25 cm² ergibt sich bei maximaler Leistungsabgabe von 300 W eine Heizflächenbelastung von 3.2 W/m². Entsprechend sind im Betrieb sehr kurze Ansprechzeiten der Heizungen zu beobachten. Mit dem Auftreten örtlicher Temperaturspitzen (*hot spots*) im Betrieb muß nicht gerechnet werden. Beide Heizungen werden von je einem programmierbaren Gleichspannungsnetzgerät versorgt.

Die Regelung der Reaktortemperatur wird von dem Meßdatenerfassungsprogramm auf dem angeschlossenen Rechner übernommen. Durch die fortlaufende zeitdiskrete Erfassung der Reaktortemperatur T_r wird der implementierte Regelalgorithmus in die Lage versetzt, die nötige Stellgrößenänderung der Heizleistung zu berechnen, um die Reaktortemperatur auf ihren Sollwert T_r zurückzuführen. Die Heizleistung der Kompensationsheizung wird über den IEEE-Bus digital an das Gleichspannungsnetzgerät übermittelt.

Der Kühlkreislauf des Reaktionskalorimeters besteht aus einem Thermostaten, einem Masseflußmesser und dem Kühlmantel des Glasreaktors. Meßtechnisch erfaßt werden der Massefluß des Kühlmediums \dot{m}_j und die Temperaturen am Mantelein- und -ausgang $T_{j,0}$ und T_j .

Um den Kühlmantel nach außen adiabatisch zu isolieren, ist er von einem Luftbad umgeben, das aus einer Umwälzanlage und einem geregelten Heizregister besteht. Im Luftbad ist eine Heizschlange installiert, welche bei einem hohen Leistungsbedarf die Grundlast der Lufterwärmung liefert. Die Zusatzheizung ist an einen weiteren Thermostaten angeschlossen. Zum Aufheizen des Luftbads kann ein Propeller in Betrieb genommen werden, der den Luftstrom gezielt an den Heizschlangen der Zusatzheizung vorbeiführt und so ein schnelles Erreichen der Luftbadsolltemperatur ermöglicht.

Der Kühlmantel besitzt eine Spaltbreite von 7.5 mm und ist außen von einer 5 mm dicken Glaswand umgeben; vom Reaktorinnenraum trennt ihn eine 9 mm dicke Glaswand. Die Ein- und Auslaßstutzen sind tangential und waagrecht im unteren und oberen Bereich des Kühlmantels angebracht.

Die Temperatur des Kühlmantels T_j wird am Ausgang des Kühlmantels erfaßt und durch den Temperierautomaten geregelt. Durch einen speziell konstruierten Einsatz im Zulaufstutzen ist es möglich, eine weitgehende Reduktion der Kurzschlußströmung und der Totzonen zu erreichen und das Verweilzeitverhalten des Kühlmantels dem eines kontinuierlichen idealen Rührkessels anzupassen. Damit kann die Mantelausgangstemperatur T_j als repräsentativ für den gesamten Kühlmantel angenommen werden. Es muß keine Temperaturverteilung im Kühlmantel berücksichtigt werden.

Der Temperierautomat setzt sich aus einem herkömmlichen Umwälzthermostaten und einem darunter montierten Tieftemperaturthermostaten zusammen. Es zeigt sich aber im Betrieb des Reaktionskalorimeters, daß bei einem Wärmestrom von ca. 40 Watt in das Kühlmedium des Kühlkreislaufs die Kühlfunktion des Thermostaten nicht genutzt wird. Vielmehr erfolgt die Regelung der Mantelausgangstemperatur nur über die Heizung des Thermostaten, welche mit einem kaskadierten PID-Regler geregelt wird.

Im Gegensatz zum Anlagenbetrieb nach HORN [51] wird als Kühlmedium nicht mehr Wasser sondern ein Silikonöl verwendet. Dies besitzt eine konstante Wärmekapazität über einen großen Temperaturbereich und ermöglicht den Betrieb oberhalb von 100 °C. Wegen der im Vergleich zu Wasser niedrigeren Wärmekapazität erhöht sich die meßbare Temperaturdifferenz ΔT_j zwischen Manteleingang und -ausgang. Da sich der Kühlmantel in seinem Verweilzeitverhalten wie ein kontinuierlicher idealer Rührkessel verhält, ist diese größere Temperaturdifferenz tolerierbar.

2.2.2 Rechnergestützte Meßdatenerfassung

Die Qualität von reaktionskalorimetrischen Messungen ist abhängig von der Art der Meßdatenerfassung und der anschließenden Auswertung [52]. Dabei muß dafür Sorge getragen werden, daß die Meßwerte im Rahmen ihrer Meßgenauigkeit korrekt sind und daß die für eine Auswertung gemachten Annahmen zutreffen.

Alle Temperaturen, das Drehmoment des Rührers und der Massefluß des Kühlmediums werden über ein Digitalmultimeter mit Meßstellenum-schaltung erfaßt, digital gewandelt und über eine IEC- (IEEE488.2-)

Schnittstelle an den Rechner übermittelt. Die einzelnen Meßwerte werden mittels sogenannter Abtast-Halteschaltungen (*sample and hold*) vom Multimeter gesammelt und als kompletter Datenblock an den Rechner übergeben. Die maximal mögliche Rate beträgt 0.5 Hz. Zudem werden alle Signale bei der Analog-/Digitalwandlung (A/D-Wandlung) zur Vermeidung des „Aliasing“-Effekts mehrfach vorgefiltert, bevor die Übertragung erfolgt. Die beiden programmierbaren Gleichspannungsnetzgeräte, die die Leistung für die Kompensations- und Kalibrierheizung liefern, sind ebenfalls über die IEC-Schnittstelle mit dem Rechner verbunden. Die jeweilig abzugebende Heizleistung wird vom Rechner digital vorgegeben. Im Netzgerät erfolgt eine Messung der tatsächlich abgegebenen Heizleistung, die zurück an den Rechner übermittelt wird. Die Steuerung und Abfrage der Meßgeräte erfolgt universal über einfache, als Zeichenkette formatierte ASCII-Befehle.

Da die Waage zur Erfassung des Masseflusses der Dosierung über keine IEC-Schnittstelle verfügt, werden ihre Meßwerte über die serielle Schnittstelle des Rechners abgefragt.

Die Software zur Steuerung und Abfrage der einzelnen Meßgeräte besteht aus einem Programm, das mit der Programmierumgebung LabVIEW erstellt wurde. LabVIEW stellt Werkzeuge zur Abfrage von diversen Schnittstellen und zur Visualisierung, Bearbeitung und Speicherung von Daten zur Verfügung. Das Hauptprogramm besteht aus der Abfrage der Meßgeräte (Multimeter, Waage und Netzteile), dem Regelalgorithmus zur Regelung der Reaktortemperatur T_r , einem Prozeßidentifikationsalgorithmus, der *online* eine experimentelle Modellbildung der Regelstrecke vornimmt, und einem wahlweise aktivierbaren Adaptionalgorithmus zur permanenten Anpassung des Reglers an sich ändernde Streckenparameter. Die Wahl der Abtastrate, mit der die Meßgrößen zeitdiskret erfaßt werden, stellt einen Kompromiß zwischen apparatetechnischer zeitlicher Auflösung und Informationsverlust dar. Wird sie zu niedrig gewählt, ist die exakte Rekonstruktion eines zeitkontinuierlichen Signals aus den durch periodische Abtastung gewonnenen zeitdiskreten Werten fehlerbehaftet, da das zu messende zeitkontinuierliche Signal durch Abtastung und Rekonstruktion in ein Signal mit niedrigerer Frequenz umgewandelt wird. Dieser Effekt wird als „Aliasing“-Effekt bezeichnet.

In [53] wird für Prozesse der Energie- und Verfahrenstechnik, bei denen die Temperatur die Regelgröße ist, eine Abtastrate von größer als 0.05 Hz empfohlen. An gleicher Stelle wird für Flüsse eine Abtastrate von 1 Hz

genannt. Zur Vereinfachung der Softwareorganisation wird in dieser Arbeit dieselbe Abtastzeit von 3 Sekunden (0.33 Hz) für alle erfaßten Meß- und Regelgrößen gewählt. Dabei konnte ein „Aliasing“-Effekt nicht beobachtet werden.

Ein weiterer Vorteil einer Abtastperiode von 3 Sekunden ist, daß zwischen der Erfassung zweier Meßwertsätze genügend Rechenzeit zur Verfügung steht. Diese Rechenzeit kann für die Abarbeitung eines aufwendigeren Regelalgorithmus als dem eines PID-Reglers verwendet werden (Fuzzy-Logik-Regler). Sie erlaubt die *Online*-Prozeßidentifikation mit einem universellen Funktionsapproximator (Künstliches neuronales Netz) und schließlich die Anwendung eines Adaptionalgorithmus, der die gewonnenen Daten in eine Parameteradaption des Reglers umsetzt.

Die Kombination aus rechnergestützter Meßdatenerfassung, die alle Meßwerte verfügbar macht, und das Vorhandensein von ausreichend Rechnerleistung hebt den Zwang zur Orientierung an der klassischen Regelungstechnik auf und erlaubt den Einsatz neuer Werkzeuge, wie Fuzzy-Logik-Regler (Regelung) und künstlicher neuronaler Netze (Prozeßidentifikation) auf dem Gebiet der Reaktionskalorimetrie.

Die Entwicklung eines adaptiven Konzepts soll darüber hinaus die manuelle Reglerparameterfindung überflüssig machen und eine zufriedenstellende Regelung der Reaktortemperatur über einen weiten Bereich der Betriebsvariablen erlauben.

Die Reaktionsführung in einem Reaktionskalorimeter wird im Satz- oder im Teilfließbetrieb ausgeführt. Bei beiden Betriebsweisen tritt meist im zeitlichen Verlauf eine Änderung der Parameter der Regelstrecke zwischen der Stellgröße Kompensationsheizleistung q_{cal} und der Meßgröße T_r auf. Die Streckenparameteränderungen lassen sich für einen kurzen Zeithorizont am besten durch einen robusten Regler bewältigen, für eine länger andauernde Parameterdrift aber am ehesten durch ein adaptives Konzept handhaben. Diese Anforderungen haben zur Anwendung moderner rechnergestützter Regelungsverfahren geführt, die im Kapitel 3 vorgestellt werden.

2.3 Homogenisieren des Reaktionsmediums [54, 55, 56]

Das Homogenisieren einer Reaktionsmischung soll einen innigen Kontakt der Reaktionsteilnehmer herstellen und dient dem Abbau von Konzentrations- und Temperaturgradienten. Diese Konzentrations- und Temperaturunterschiede können verursacht sein durch die Reaktion (z.B. bei einer Polymerisation in viskoser Medien) oder durch die Dosierung eines Reaktanden in die Flüssigkeit des vorgelegten Reaktionsmediums.

Der Vorgang des Homogenisierens findet in zwei Schritten statt. Zunächst erfolgt eine konvektive, turbulente Makrovermischung, welche Schlieren und turbulente Wirbelballen verursacht. Daran schließt sich die Mikrovermischung durch physikalische Diffusion an, die eine bis in den molekularen Bereich homogene Mischung erzeugt. Die konvektive Makrovermischung bereitet die molekulare Diffusion lediglich durch eine Vergrößerung der Phasengrenzfläche vor. Dabei ist die Makrovermischung der geschwindigkeitsbestimmende Schritt [55].

Um die Aufgabe des Homogenisierens zu erfüllen, stehen bei einem Rührkesselreaktor verschiedene Rührerformen zur Verfügung. Sie sollen zum einen die Umwälzung des Behälterinhalts bewerkstelligen und zum anderen Turbulenzen im flüssigen Reaktionsmedium erzeugen. Eine Einteilung der Rührerformen kann anhand der primär verursachten Strömung erfolgen. Der Schrägblattrührer ist ein axial fördernder Rührer, der darüber hinaus eine radiale Strömungskomponente erzeugt. Für das Homogenisieren ist eine axiale Strömung unabdingbar [55]. Der Ankerrührer hingegen erzeugt primär eine tangentielle Strömung, die zusätzlich eine radiale Komponente besitzt. Er wird vornehmlich für Wärmeaustauschzwecke in viskoser Medien verwendet.

Jeder in einem zylindrischen Reaktionsgefäß zentrisch angebrachte Rührer versetzt die Reaktionsmischung in eine Rotationsströmung, in der sich zwei Reaktionskomponenten mit der gleichen Winkelgeschwindigkeit bewegen und eine turbulente Vermischung nur in geringem Ausmaß stattfindet. Gleichzeitig bildet sich eine Trombe aus, wobei der Flüssigkeitsspiegel in Rührernähe abgesenkt und in Wandnähe angehoben wird. Zur Vermeidung der Trombenbildung und zur Erhöhung der Turbulenz werden Strömungsbrecher in den Reaktor eingebaut. Als Strömungsbrecher können dabei auch zu Registern zusammengefaßte Wärmeaustauscherrohre verwendet werden [54]. Die Vermeidung einer Trombe hat den zusätzlichen Effekt, daß die

Wärmeaustauschfläche rechnerisch bestimmt werden kann und nicht von der Rührergeschwindigkeit oder dem rheologischen Verhalten des Reaktionsmediums abhängt.

Eine molekular homogene Mischung entspricht dem Modell des idealen Rührkessels, welches für die Auswertung reaktionskalorimetrischer Messungen zu Grunde gelegt wird. Es ist daher von seiten des Rührsystems auf eine ausreichend schnelle Homogenisierung des Reaktorinhalts zu achten. Dann kann die Reaktortemperatur T_r als im gesamten Reaktionsmedium gültig angenommen werden, und im Reaktionsmedium herrschen überall die gleichen Reaktandenkonzentrationen und die berechnete Reaktionsgeschwindigkeit r_j .

Reaktionskalorimetrie macht nur in einem idealen Rührkesselreaktor Sinn, da bei nicht-idealem Verhalten eine Temperaturerhöhung in einem Reaktorbereich mit einer Veränderung der Reaktionsgeschwindigkeit einhergehen würde. Dadurch käme es zu örtlich unterschiedlichem Komponentenverbrauch. Auch wären die ermittelten Wärmestromverläufe (welche eine Bestimmung der effektiven Reaktionsgeschwindigkeit im Reaktionsraum r_{eff} zulassen) unbrauchbar, weil ihnen limitierende Wärmeleitungsvorgänge überlagert wären.

Für Reaktionen im Teilfließbetrieb bedeutet dies, daß die Mischzeit q wesentlich kleiner sein sollte als die auf Anfangsbedingungen bezogene Zeitkonstante der Reaktion t_r :

$$t_r = \frac{c_0}{r_0} \quad (17)$$

Die Durchführung von Mischzeitexperimenten ist aufgrund der besonderen Gerätekonstruktion des Reaktionskalorimeters sinnvoll, da die Funktion der Strömungsbrecher lediglich von den im Reaktor angebrachten Heizungen übernommen wird, wobei der Heizdraht in Form einer Heizspirale gewickelt ist. Ein Schrägblattrührer kann wegen der Heizungs-konstruktion nicht wie empfohlen ein Drittel der Flüssigkeitshöhe vom Boden entfernt angebracht werden, sondern nur etwas tiefer im Behälter. Auch erlauben die Abmessungen des Reaktors nicht ein Flüssigkeitshöhe-Reaktordurchmesser-Verhältnis von 1, so daß eine Verlängerung der Mischzeit durch ein ungünstigeres Verhältnis (> 1) zu erwarten ist.

Die Angabe einer Mischzeit q ist nur sinnvoll mit Angabe eines Homogenisierungsgrades. Vollständige Homogenisierung wird angenommen, wenn die

verwendeten Sensoren keine Inhomogenitäten mehr detektieren können. Zur Bestimmung der Mischzeit q stehen verschiedene physikalische und chemische Meßmethoden zur Verfügung: Entfärbungsmethode, Schlierenmethode und Methoden unter Verwendung einer Markierungssubstanz.

Bei der Schlierenmethode werden zwei ineinander lösliche Flüssigkeiten mit unterschiedlichen Brechungsindizes im Reaktionsgefäß übereinander geschichtet, so daß eine Phasengrenze sichtbar wird. Dann wird der Rührer eingeschaltet und die Mischzeit q gemessen, die bis zum Verschwinden der Schlieren verstreicht. Hierbei muß die Anlaufzeit des Motors bis zum Erreichen der Solldrehzahl klein gegenüber der Meßzeit sein. Dem Vorteil dieser Methode, daß der gesamte Reaktionsraum erfaßt wird, stehen die Nachteile möglicher Schwierigkeiten der optischen Endpunkterkennung und die Nicht-Berücksichtigung des Ortes der Dosierung gegenüber.

Als System für die Schlierenmethode eignet sich die Vorlage von Glycerin im Reaktionsgefäß und die Überschichtung mit Wasser, da beide Flüssigkeiten newtonsches Fließverhalten zeigen, und Schlieren aufgrund der unterschiedlichen Brechungsindizes gut auszumachen sind ($n_D = 1.33$ für Wasser, $n_D = 1.47$ für Glycerin bei 20°C). Allerdings weist dieses System einen Dichteunterschied zwischen beiden Komponenten auf, der einer Vermischung entgegenwirkt. Dieser müßte in einer Kennzahl-Beziehung über die Archimedes-Zahl Ar berücksichtigt werden. Für eine qualitative und quantitative Abschätzung kann der Dichteunterschied vernachlässigt werden.

Für den Fall der laminaren Strömung im Rührkesselreaktor ergibt sich dann die Beziehung (18) zwischen der dimensionslosen Mischzeit-Kennzahl $N \cdot q$ und der Reynolds-Zahl Re , welche die Strömung im Rührkessel charakterisiert (19).

$$N \cdot q = f(Re) \quad (18)$$

$$Re = \frac{N \cdot d^2 \cdot \mathbf{r}}{\mathbf{h}} = \frac{N \cdot d^2}{\mathbf{n}} \quad (19)$$

Dabei zeigt sich mit zunehmender Reynolds-Zahl Re eine Abnahme der Mischzeitkennzahl $N \cdot q$. Dieses Verhalten wird durch eine Zunahme der turbulenten Regionen und eine Abnahme der laminaren Regionen im

Rührkessel erklärt. Die Reynolds-Zahl (19) kann mit den Stoffwerten der resultierenden Mischung gebildet werden.

Für den Fall der turbulenten Strömung im Reaktionsgefäß wird die dimensionslose Mischzeit-Kennzahl konstant (20). Dann herrscht in der gesamten Flüssigkeit turbulente Strömung, und die Anzahl der Rührerdrehungen $N \cdot q$ zur Erreichung des Homogenisierungsgrades ist konstant.

$$N \cdot q = \text{const} \quad (20)$$

In [55] wird darauf hingewiesen, daß bei hohen Reynolds-Zahlen ($Re > 10^5$) die molekulare Diffusion geschwindigkeitsbestimmend wird, die sich durch schnelleres Rühren nicht beeinflussen läßt. Dann sollte aber die Mischzeit q und nicht die Mischzeit-Kennzahl $N \cdot q$ konstant werden. Allerdings stößt man in diesem Bereich an die Grenzen der Meßgenauigkeit, und dieser Fall läßt sich bei Laborreaktoren nicht beobachten.

Die Optimierung im Rahmen von Mischzeitexperimenten besteht im Erreichen einer geforderten Mischzeit q (bei gegebenem Grad der Homogenisierung) unter Minimierung der aufgewandten Rührenergie. Von dieser Aufgabe unterscheidet sich die Untersuchung im Rahmen der Reaktionskalorimetrie, da hier die Schaffung eines idealen Rührkesselverhaltens bei gegebenen apparatetechnischen Vorgaben (Heizungen, Temperatursensor) unter Verwendung verschiedener Rührerformen im Vordergrund steht.

2.4 Wärmedurchgang durch die Behälterwand [54, 55, 57]

Die Intensivierung des Wärmedurchgangs durch die Reaktorwand wird selten als die primäre Rühraufgabe betrachtet [55]. In der Reaktionskalorimetrie, deren Ergebnisse von der zuverlässigen Ermittlung der Wärmeproduktionsrate abhängen, ist ein intensiver Wärmetransport eine weitere wichtige Aufgabenstellung neben der Annäherung an das ideale Rührkesselverhalten.

Bei der leistungskompensierten Reaktionskalorimetrie, die ein konstantes Temperaturgefälle zwischen dem Reaktorinnenraum mit der Temperatur T_r und dem Mantel mit der Manteltemperatur T_j aufrechterhält, ist dabei der Wärmeübergang an zwei Orten zu berücksichtigen. Zum einen der Wärmeübergang von der Kompensationsheizung auf das Reaktionsmedium – hier wird die Kompensationsheizleistung q_{comp} eingetragen – und zum

anderen der Wärmeübergang vom Reaktionsmedium auf die Reaktorwand (10), um die Reaktions- und Kompensationsleistung wieder aus dem Reaktorinnenraum abzuführen.

In beiden Fällen ist die Dicke der Prandtlschen Grenzschicht für den Wärmeübergang entscheidend. In dieser wand- oder heizungsnahen Grenzschicht ist die Strömung des Reaktionsmediums laminar, so daß ein Wärmestrom quer zur Strömungsrichtung nur durch Wärmeleitung erfolgt. Durch Erhöhung der Strömungsgeschwindigkeit der Flüssigkeit lassen sich die Dicke der laminaren Grenzschicht verringern und der Wärmeübergang erhöhen.

Für einen verbesserten Wärmeübergang auf die Reaktorwand kommt der wandnah arbeitende Ankerrührer in Betracht. Die primär von ihm erzeugte tangentielle Strömungskomponente, der eine radiale Komponente überlagert ist, sorgt für einen raschen Austausch der Fluidelemente an der Wand und beschränkt auf diese Weise die Dicke der Grenzschicht. Auch bei höherer Viskosität des Reaktionsmediums kommt es zu einer Zwangsförderung an der Wand, so daß der reaktionsseitige Wärmeübergangskoeffizient a_r bei Zunahme der Viskosität groß bleibt. Rührer mit großem Rührerdurchmesser-Reaktordurchmesser-Verhältnis ($d/D < 1$) sind aus energetischer Sicht günstiger für den Wärmetransport, da sie turbulente Strömung dort erzeugen, wo sie für einen Wärmeübergang benötigt wird, in Wandnähe.

Besondere strömungstechnische Anforderungen stellen beim vorgestellten Reaktionskalorimeter die als Strömungsbrecher ausgelegten Heizungen dar. Sie bestehen aus einem in Form einer Spirale gewickelten Heizdraht. Diese Form verlangt eine axiale Strömungskomponente der umgewälzten Flüssigkeit [54, 55], damit der Innenraum der Spirale durchströmt wird. Dafür eignet sich der Schrägblattrührer, der eine axiale Strömung als primäre Strömungskomponente besitzt.

Die durch den Rührer verursachte Strömung nimmt lediglich Einfluß auf den reaktionsseitigen Wärmeübergangskoeffizienten a_r . Dafür existiert die Kennzahlbeziehung (21), die die Nusselt-Zahl Nu als Funktion der Reynoldszahl Re , der Prandtl-Zahl Pr und dem Viskositätsverhältnis ausdrückt. Sie ist analog zur Beschreibung des Wärmeübergangs eines Fluids, das durch ein Rohr strömt.

$$Nu = c \cdot Re^{\frac{2}{3}} \cdot Pr^{\frac{1}{3}} \cdot \left(\frac{h}{h_w} \right)^m \quad (21)$$

Darin enthalten sind die Konstanten $c = 0.240$ und $m = 0.14$. Die Nusselt-Zahl Nu wird gemäß (22), die Prandtl-Zahl Pr gemäß (23) gebildet.

$$Nu = \frac{\mathbf{a}_r \cdot D}{l} \quad (22)$$

$$Pr = \frac{\mathbf{h} \cdot c_p}{l} = \frac{\mathbf{n}}{a} \quad (23)$$

Für den Wärmeübergang vom Reaktionsmedium auf die Reaktorwand wird eine etwas andere dimensionslose Kennzahlbeziehung gefunden, wenn die Flüssigkeit von einem Ankerrührer gerührt wird [57, 58]:

$$Nu = c_1 \cdot \left(Re \cdot Pr^{\frac{1}{2}} + c_2 \right)^{\frac{2}{3}} \cdot \left(\frac{\mathbf{h}}{\mathbf{h}_w} \right)^m \quad (24)$$

In der Literatur [58] werden die Konstanten für das Kühlen mit $c_1 = 0.230$ und $c_2 = 4000$ angegeben, wenn das Viskositätsverhältnis nicht berücksichtigt wird ($m = 0$). Unter Verwendung des Viskositätsverhältnisses werden für das Kühlen die geringfügig veränderten Konstanten $c_1 = 0.238$, $c_2 = 4000$ und $m = 0.02$ genannt. Wird das Viskositätsverhältnis vernachlässigt, so geht die Beziehung (24) für große Reynolds-Zahlen Re in die einfachere Beziehung (21) über.

In Gleichung (14) wird der Kehrwert des Wärmedurchgangskoeffizienten $1/h$ in Anlehnung an das Ohmsche Gesetz als Summe von Einzelwiderständen dargestellt. Bei Variation der Rührerdrehzahl verändert sich nur der reaktionsseitige Wärmeübergangskoeffizient \mathbf{a}_r , so daß der Wärmewiderstand der Wand und der Kehrwert des mantelseitigen Wärmeübergangskoeffizienten in einer Konstanten $\Phi(T_r)$ zusammengefaßt werden können (25).

$$\Phi(T_r) = \frac{d}{l} + \frac{1}{\mathbf{a}_j} \quad (25)$$

Wird die für den reaktionsseitigen Wärmeübergang gültige Kennzahlbeziehung (21) berücksichtigt, so läßt sich der Wärmeübergangskoeffizient \mathbf{a}_r als Funktion der Rührerdrehzahl darstellen:

$$\frac{1}{a_r} = b_r \cdot N^{-\frac{2}{3}} \quad (26)$$

Damit ist der Wärmedurchgangskoeffizient h bezüglich der apparativen Größen eine Funktion der Rührerdrehzahl N (Wilson-Plot).

$$\frac{1}{h} = \Phi(T_r) + b_r \cdot N^{-\frac{2}{3}} \quad (27)$$

Wird der Wärmeübergangskoeffizient h bei mehreren Rührerdrehzahlen N gemessen, und wird eine lineare Regression ($1/h$ gegen $N^{-2/3}$) durchgeführt, so ergibt der Achsenabschnitt die Apparatekonstante $\Phi(T_r)$.

Wegen der Äquivalenz der Gleichungen (13) und (15) kann in der leistungskompensierten Reaktionskalorimetrie der Wärmedurchgangskoeffizient h im Verlauf einer Reaktion verfolgt werden. Da die Gerätekonstante $\Phi(T_r)$ praktisch unverändert bleibt, muß die Veränderung von h aus einem veränderten Wärmeübergang a_r resultieren.

Die eigentliche Optimieraufgabe des Wärmeübergangs besteht im Erreichen eines geforderten Wärmetransport bei Aufwendung minimaler Rührleistung. Der dabei oft betrachtete Fall [55], daß die erreichbare Wärmeabfuhr ein Maximum durchläuft, da die eingebrachte Rührleistung mit Erhöhung der Rührerdrehzahl stärker ansteigt als der Wärmeübergangskoeffizient für die Wärmeabfuhr, gilt nur bei laminarer Strömung im Reaktionsgefäß; dann wird $h \approx a_r$ angenommen. Dieser Fall ist jedoch für die Betriebsweise eines Reaktionskalorimeters ohne Belang.

In der leistungskompensierten Reaktionskalorimetrie kommen die genannten Anforderungen zusammen: Schnelles Homogenisieren des gesamten Behälterinhalts, schnelle Verteilung der von den Heizungen eingebrachten Heizleistung und Erhöhung des Wärmeübergangs von der Reaktionsmischung auf die Behälterwand. Dabei können diese Aufgaben nicht von einer Rührerart allein bewerkstelligt werden [54]. Eine naheliegende Lösung besteht daher in der Kombination zweier Rührerarten. Die Intensivierung des Wärmeübergangs an der Wand wird von einem Ankerrührer gewährleistet, während der Wärmeübergang an den Heizung von einem aufgesetztem Schrägblattrührer übernommen wird. Darüber hinaus soll die axiale Strömungskomponente des Schrägblattrührers für die Umwälzung des Behälterinhalts und das schnelle Homogenisieren sorgen.

3 Regelung der Reaktortemperatur

3.1 Theoretische Grundlagen

Die leistungskompensierte Reaktionskalorimetrie hat als Grundprinzip die Trennung von Reaktortemperatur- und Kühlmitteltemperaturregelung. Der erste Regelkreis umfaßt die Regelung der Reaktortemperatur T_r mittels einer Kompensationsheizung.

Der zweite Regelkreis besteht aus Reaktorkühlmantel und Thermostat. Er wird in [62] näher beschrieben, wo eine Optimierung der Reglerparameter vorgenommen wird. Die dort gefundenen Parameter werden bei den in dieser Arbeit vorgestellten Messungen beibehalten. Bei Betrachtung der möglichen Störgrößen kommt für den von der Umgebung adiabatisch isolierten Kühlmantel nur ein veränderlicher Wärmestrom aus dem Reaktorenraum in Betracht. Daher sind die Anforderungen an die Regelung der Manteltemperatur weitaus geringer als die Anforderungen an die Regelung der Reaktortemperatur, die in diesem Kapitel vorgestellt wird.

Die leistungskompensierte Reaktionskalorimetrie bietet die Möglichkeit, eine Reaktion unter streng-isothermen Bedingungen durchzuführen. Dabei wird die Reaktortemperatur T_r innerhalb eines engen Toleranzbandes um den Sollwert $T_{r,soll}$ (Führungswert) geregelt.

Die Einhaltung des Sollwertes ist in der Reaktionstechnik für die Sicherheit eines Prozesses und für die Gewährleistung der Produktqualität wichtig. Der am häufigsten dafür eingesetzte Regler ist der PID-Regler. Er berechnet anhand eines Proportional-, Integral- und Differentialanteils die Stellgrößenänderung $\Delta u(t)$, die der Regelabweichung entgegenwirkt. Die Erfassung der Regelgröße muß nicht kontinuierlich erfolgen, eine Abtastung mit kurzer Abtastperiode Δt_a ist ausreichend. Die Geschwindigkeitsform des PID-Regelalgorithmus in zeitdiskreter Form lautet:

$$\frac{\Delta u(t)}{\Delta t_a} = \frac{U_h}{X_p} \left[\frac{e(t) - e(t-1)}{\Delta t_a} + \frac{1}{T_n} \cdot e(t) + T_v \frac{\Delta^2 e(t)}{(\Delta t_a)^2} \right] \quad (28)$$

Darin enthalten ist die Regeldifferenz $e(t)$ (29); die negative Regeldifferenz wird als Regelabweichung $x_w(t)$ (30) bezeichnet.

$$e(t) = y_{k,soll} - y_k(t) \quad (29)$$

$$x_w(t) = y_k(t) - y_{k,soll} \quad (30)$$

Über die Parameter Proportionalbereich X_p , Nachstellzeit T_n und Vorhaltezeit T_v läßt sich das Verhalten des Reglers an die Gegebenheiten der Regelstrecke anpassen. Der Stellbereich U_h ist gerätetechnisch vorgegeben. In der Geschwindigkeitsform (28) muß für die Bestimmung des D-Anteils die zweite Ableitung der Regeldifferenz nach der Zeit gebildet werden (31).

$$\frac{\Delta^2 e(t)}{(\Delta t_a)^2} = \frac{e(t) - 2 \cdot e(t-1) + e(t-2)}{(\Delta t_a)^2} \quad (31)$$

Dies ist jedoch nur für Prozesse mit sich langsam ändernder Regelgröße $y_k(t)$ sinnvoll, da die nach (31) gebildete zweite Ableitung streng genommen für den Zeitpunkt $t-1$ und nicht für den Zeitpunkt t gilt.

Für die Regelung reicht meist die Ausführung als PI-Regler aus (Vorhaltezeit $T_v = 0$). Dieser Regler vereint die Vorteile des Proportional- und des Integralanteils. Einerseits kann er mit dem P-Anteil bei Auftreten einer Regeldifferenz $e(t)$ verzögerungsfrei einschreiten, andererseits wird eine bleibende Regeldifferenz aufgrund des I-Anteils vermieden.

Bei ungünstiger Wahl der Reglerparameter (X_p oder T_n zu klein) kann der geschlossene Regelkreis instabil werden, d.h. bei einer Anregung des Regelkreises wird nach endlicher Zeit kein konstanter Beharrungswert der Regelgröße erreicht. Daher gibt es für den PID-Regler empirische Einstellregeln (Verfahren nach Ziegler-Nichols, Verfahren nach Chien-Hrones-Reswick [42, 63, 64]), die zumindest eine brauchbare Abschätzung der Parameter liefern. Aufwendigere Methoden bestehen darin, für den zu regelnden Prozeß im Arbeitspunkt ein lineares Modell zu entwerfen, so daß sich das Zusammenwirken von Regler und Regelstrecke simulieren läßt. Auf diese Weise können die optimalen Reglerparameter für eine gegebene Regelaufgabe gefunden werden.

3.1.1 Fuzzy-Logik-Regler [3, 5, 7, 9, 11, 12]

Die Regelung von mathematisch einfach beschreibbaren linearen Regelstrecken ist die Domäne der klassischen Regelungstechnik [1]. Die Regelstrecken sollten weitgehend bekannten und kalkulierbaren Störungen unterliegen, zeitinvariant sein und nur innerhalb eines engen Arbeitsbereichs betrieben werden.

Auf reale Prozesse treffen diese vereinfachenden Annahmen oft nicht zu, da sie meist nichtlinear, mit Totzeiten versehen oder zeitvariant sind [12]. So sind komplexe industrielle Prozesse, wie chemische Reaktionen in satzweise betriebenen Rührkesseln schwierig automatisch zu regeln, was aus ihrem nichtlinearen, zeitlich veränderlichen Verhalten herrührt [2, 5, 66]. Für die Regelung solcher Prozesse eignet sich ein Fuzzy-Logik-Regler (ein Regler auf Basis der unscharfen Logik), welcher selbst ein nichtlineares System darstellt und daher mit dem nichtlinearen Streckenverhalten besser fertig werden kann als ein klassischer Regler [12]. Aber auch einfache Prozesse sind mit Fuzzy-Logik-Reglern u.U. besser zu regeln, da die mathematische Exaktheit der klassischen Regelungstechnik teilweise ein Mythos ist [2].

Die grundlegende Idee zur Anwendung der unscharfen Logik in der Regelungstechnik besteht darin, die Erfahrung eines menschlichen Prozeßbedieners in das Design des Reglers einfließen zu lassen. So können auch schwierige Prozesse vom Bedienungspersonal einer Anlage geregelt werden, ohne daß das Personal ein mathematisch einwandfreies Prozeßmodell vor Augen hat. Das verständliche und nachvollziehbare Regelkonzept setzt sich dabei zusammen aus der Unterteilung der Prozeßeingangs- und -ausgangsgrößen in umgangssprachliche Ausdrücke (wie GROß, MITTEL, KLEIN) und einem Satz linguistischer Wenn-Dann-Regeln, die die Kontrollstrategie des Bedienungspersonals beschreiben [2, 5, 9]. Vorteil der Konstruktion eines Fuzzy-Logik-Reglers ist, daß kein analytisches Prozeßmodell benötigt wird. Ein solches Modell wird beim Reglerentwurf oder bei der Parameteroptimierung eines PI-Reglers benötigt, wenn man nicht nach empirischen Einstellregeln vorgehen will [5, 6].

Der sprachlichen Interpretation eines Fuzzy-Logik-Reglers steht die Beschreibung als einfacher Kennfeldregler gegenüber [2-4, 7]. Auch wenn bei der Entwicklung eines Reglers auf Basis der unscharfen Logik umgangssprachliche Ausdrücke und Regeln zum Einsatz kommen, so ist das Ergebnis ein statisches Übertragungsglied, das deterministisch den Eingangsgrößen eine Ausgangsgröße zuordnet. Der Wert der Ausgangsgröße wird von einem mathematisch exakten Algorithmus berechnet. Unscharf heißt bei diesen Reglern also nicht ungenau.

Fuzzy-PI-Regler sind eine Verallgemeinerung des klassischen PI-Regelkonzeptes. Sie hat das Ziel, durch die Einführung zusätzlicher Freiheitsgrade eine verbesserte Dynamik verglichen mit dem konventionellen linearen PI-Regler zu erzielen [3]. Die zusätzlichen Freiheitsgrade

resultieren aus der größeren Anzahl Parameter, die bei einem Fuzzy-Logik-Regler festgelegt werden können. Ein Fuzzy-Logik-Regler wird daher auch als überparametrisierter Regler bezeichnet.

Der in Abbildung 3 dargestellte Fuzzy-Logik-Regler basiert auf einem Fuzzy-Logik-System, welches in Anhang A vorgestellt wird.

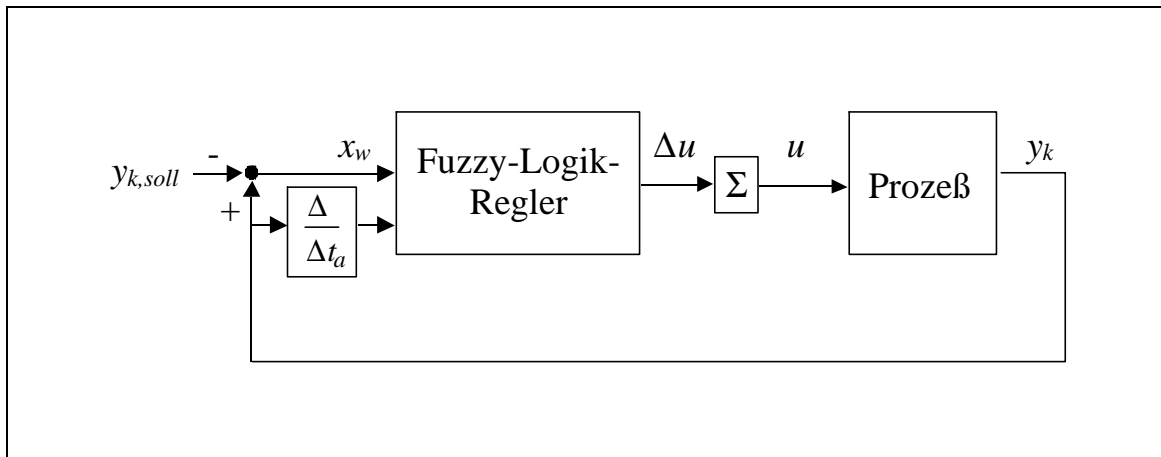


Abb. 3 Regelkreis aus Fuzzy-Logik-Regler (Fuzzy-PI-Regler) und Regelstrecke

Der Regler orientiert sich an dem PI-Regelkonzept. Seine Eingangsgrößen sind die Regelabweichung x_w und die Änderung der Regelgröße mit der Zeit $\Delta y_k / \Delta t_a$. Weil das Fuzzy-Logik-System ein statisches Übertragungsglied ist, muß die zeitliche Änderung $\Delta y_k / \Delta t_a$ außerhalb des Reglers gebildet werden. Die Ausgangsgröße ist die Änderung der Stellgröße Δu , die in der Abtastperiode Δt_a vorgenommen wird. Da der Regler selbst keine Information über den aktuellen Wert der Stellgröße hat, wird dem Fuzzy-Logik-System ein Summierer nachgeschaltet. Dieser ist zusätzlich mit einem Begrenzer versehen, so daß die Stellgröße bei Erreichen eines Maximalwertes (Minimalwertes) nicht weiter zunimmt (abnimmt). Der Begrenzer übernimmt die Funktion einer *Anti-Windup*-Komponente bei einem klassischen PI-Regler.

Der Wertebereich jeder Eingangsgröße (linguistische Variable) wird in mehrere sich überlappende unscharfe Mengen (Terme) unterteilt. Diese unscharfen Mengen beschreiben die umgangssprachlichen Ausdrücke. Wenn von einer „GROSSEN“ Regelabweichung die Rede ist, muß das Fuzzy-Logik-System Informationen darüber besitzen, welche Werte der Regelabweichung als groß bezeichnet werden.

Die unscharfen Mengen werden meist als dreieckförmige oder trapezförmige Funktionen beschrieben, was bei der älteren Literatur auf dem geringeren Rechenaufwand dieser Funktionen beruht. Diese Funktionen besitzen aber den Nachteil, daß sie nicht stetig differenzierbar sind. Zwei benachbarte unscharfe Mengen sollten sich überschneiden oder berühren, damit jedem Wert der Eingangsgröße mindestens ein Term zugeordnet ist und keine Lücken im Regelverhalten des Fuzzy-Logik-Systems entstehen.

Bei der Anzahl der Terme, die für die Beschreibung einer Eingangsgröße verwendet werden, gibt es zwei unterschiedliche Anforderungen. Wird eine große Anzahl gewählt, macht diese eine detaillierte Kategorisierung der Eingangswerte möglich. Andererseits vergrößert sich mit der Anzahl der linguistischen Terme auch der Umfang der Regelbasis (bzw. der Umfang der unscharfen assoziativer Karten). Meist wird eine ungerade Anzahl und eine Anzahl von weniger als sieben unscharfen Mengen empfohlen [6, 7].

Sind die Anzahl, Form und Parameter der unscharfen Eingangsmengen definiert, werden sie in der Regelbasis mit den Termen der Ausgangsgröße verknüpft. Die Regelbasis gliedert sich in mehrere Wenn-Dann-Regeln der Form:

$$\begin{array}{l} \text{WENN die Regelabweichung GROSSPOSITIV} \\ \text{UND die Regelgrößenänderung NEGATIV ist,} \\ \text{DANN VERRINGERE die Stellgröße GERINGFÜGIG} \end{array} \quad (32)$$

Wird jeder Kombination aus Regelabweichung und Regelgrößenänderung eine Stellgrößenänderung zugewiesen, so kann die Regelbasis als Matrix dargestellt werden. Man spricht dann von einer konsistenten Regelbasis, die keine sich widersprechenden Regeln enthält. Die Regelbasis läßt sich auch dahingehend erweitern, daß jeder Kombination aus Regelabweichung und Regelgrößenänderung jeder Term der Stellgrößenänderung mit einem Relevanzfaktor zugewiesen wird (unscharfe assoziative Karten). Wie bei den sich überlappenden unscharfen Mengen, sollten bei der Erstellung der Regelbasis undefinierte Bereiche vermieden werden.

Als Defuzzifizierungsmethode wird in dieser Arbeit die Höhenmethode gewählt; somit sind die Mengen der Ausgangsgröße Singletons, denen lediglich ein Wert der Stellgrößenänderung zugewiesen wird.

Als weiterer Freiheitsgrad wird jede Eingangs- und Ausgangsgröße mit einem Skalierungsfaktor (c_{Rabw} , $c_{Ränd}$, $c_{Uänd}$) versehen. Mit dem zugehörigen Skalierungsfaktor werden alle unscharfen Mengen einer Eingangs- oder

Ausgangsgröße multipliziert, um so die Terme über den Wertebereich verbreitern zu können. Diese Skalierungsfaktoren sind den Reglerparametern Proportionalbereich X_p und Nachstellzeit T_n eines PI-Reglers ähnlich, da sie das Übertragungsverhalten global beeinflussen.

Bei der Erstellung eines Fuzzy-PI-Reglers zur Regelung der Reaktortemperatur T_r sind die beschriebenen Eingangsgrößen die Temperaturabweichung $x_w = T_r - T_{r,soll}$ und die Reaktortemperaturänderung $\Delta T_r / \Delta t_a$. Aus diesen Werten wird der Wert der Ausgangsgröße: die Heizleistungsänderung Δq_{comp} gebildet.

Der für das Reaktionskalorimeter implementierte Fuzzy-Logik-Regler verfügt über eine Erweiterung des Regelalgorithmus, die schon im Rahmen eines klassischen PID-Regler erfolgreich eingesetzt wurde. Um der Totzeit einer Regelstrecke t_D Herr zu werden, hat sich die Anwendung des in [66] vorgeschlagenen Verfahrens bewährt. Dabei wird zwischen der Abtastperiode Δt_a und der Regelperiode Δt_c , die ein ganzzahliges Vielfaches der Abtastperiode ist, unterschieden. Jede Regelstrecke, die mit einem Masse- oder Energiestrom verbunden ist, weist eine Totzeit t_D auf. Die Auswirkung eines Regeleingriffs macht sich deshalb nicht sofort bei der Erfassung der Regelgröße y_k bemerkbar, sondern wird dort mit der Zeit t_D zeitlich versetzt registriert. Es ist daher sinnvoll die Regelperiode Δt_c länger als die Totzeit t_D zu wählen. Auf diese Weise wird der Regler in die Lage versetzt, die Auswirkungen der letzten Stellgrößenänderung Δu zu registrieren und anhand dieser Informationen die neue Stellgrößenänderung vorzunehmen. Wird die Abtastperiode Δt_a recht klein gewählt, ist gewährleistet, daß der Regler immer mit aktuellen Eingangswerten arbeitet. Auch wird vermieden, daß bei der Berechnung von $\Delta T_r / \Delta t_a$ die Mittelung über ein zu großes Zeitintervall erfolgt.

In der klassischen Regelungstechnik lassen sich die linearen PID-Regler zusammen mit für einen Arbeitspunkt linearisierten Modellen stabilitätstheoretisch erfassen. In der regelungstechnischen Praxis ist der Stabilitätsnachweis jedoch oft von untergeordneter Bedeutung, da die benötigte Güte des mathematischen Modells nicht zur Verfügung steht [7]. Davon hängt aber die Güte der Stabilitätsanalyse ab, denn diese kann nur so gut sein wie die Güte des zugrundeliegenden Modells [9].

Bei Fuzzy-Logik-Reglern, die selbst ein nichtlineares System [3] darstellen, kann die Stabilitätsanalyse nur numerisch durch Simulation erfolgen [7]. In

diesem Fall spielen das lineare Verhalten der Regelstrecke bzw. der Zwang zu einem linearisierten Modell keine Rolle.

Instabilitäten bei Verwendung eines PID-Reglers können bei falscher Wahl der Reglerparameter auftreten (X_p oder T_n zu klein). Durch die Partitionierung des Eingangsraumes weist der Fuzzy-Logik-Regler für unterschiedliche Regeldifferenzen x_w und Änderungen der Regelgröße $\Delta y_k/\Delta t_a$ auch unterschiedliche Reglerverstärkungen auf. So kann er auf gravierende Abweichungen viel aktiver reagieren als auf kleine Abweichung, dabei aber trotzdem zusammen mit der Regelstrecke ein stabiles System bilden.

In [2] werden Fuzzy-Logik-Regler wegen ihres nichtlinearen Verhaltens als stabiler und robuster im Vergleich zu klassischen PID-Reglern bezeichnet. Robustheit bezieht sich in diesem Zusammenhang auf die Anfälligkeit des geschlossenen Regelkreises auf Effekte, die bei der Reglerparameterwahl nicht berücksichtigt werden. Diese Effekte können Störgrößen unbekanntem Ausmaßes, Meßrauschen oder sich ändernde Prozeßparameter sein [67]. Veränderungen der Streckenparameter und Veränderungen der Totzeit führen bei Fuzzy-Logik-Regler nur zu einer geringen Verschlechterung der Regelgüte [18].

Diese Überlegungen haben zur Wahl eines Fuzzy-Logik-Regles für die Regelung der Reaktortemperatur geführt. Bei dieser Regelung geht es im wesentlichen um das Einhalten eines Toleranzbandes um den Sollwert. Lediglich bei Beginn und Ende einer Dosierung oder beim Start einer Reaktion treten erhebliche Regelabweichungen auf, die kompensiert werden müssen. Hierfür ist der Fuzzy-Logik-Regler mit einer Partitionierung des Eingangsraumes und entsprechend unterschiedlichen Reglerverstärkungen bestens geeignet. Der direkte Vergleich der Regelgüte eines PID-Reglers und eines Fuzzy-Logik-Reglers verbietet sich aber, da der rechnerische Aufwand eines Fuzzy-Logik-Reglers um ein vielfaches höher ist.

Anpassung eines Fuzzy-Logik-Reglers mittels eines genetischen Algorithmus [4, 17, 24, 26, 29, 65]

Da gradientenbasierte Verfahren wegen der fehlenden Differenzierbarkeit eines Fuzzy-Logik-Systems (mit dreieckförmigen Zugehörigkeitsfunktionen oder nichtdifferenzierbaren Operatoren) nicht zur Optimierung der Parameter verwendet werden können, wird oft ein genetischer Algorithmus für diese Aufgabe herangezogen [17, 26, 65].

Genetische Algorithmen sind stochastische Optimierungsverfahren, welche den Parametersatz eines Fuzzy-Logik-Reglers als Individuum auffassen und anhand vieler solcher Individuen eine Güteverbesserung des Regelverhaltens vornehmen. Die Arbeitsweise von genetischen Algorithmen ist in Anhang D beschrieben.

Zur Optimierung eines Reglers wird die Beschreibung der Regelgüte durch ein Gütekriterium benötigt. Als Gütekriterien eignen sich z.B. die quadratische Regelfläche *ISE* (*integral of squared error*) (33) oder die zeitgewichtete quadratische Regelfläche *ITSE* (*integral of time multiplied squared error*) (34). Diese werden aus dem zeitlichen Verlauf der Regeldifferenz nach einem Führungsgrößen- oder Störgrößensprung zum Zeitpunkt $t=0$ ermittelt.

$$ISE = \int_0^{\infty} e(t)^2 dt \quad (33)$$

$$ITSE = \int_0^{\infty} t \cdot e(t)^2 dt \quad (34)$$

In der klassischen Regelungstechnik werden diese Gütekriterien oft zur Bestimmung der optimalen Reglerparameter verwendet, da bei bekannten Regelstreckenparametern ein analytisch lösbarer Zusammenhang zwischen den Reglerparametern und dem Gütekriterium besteht.

Ein genetischer Algorithmus benötigt zur Optimierung der Reglerparameter eines Fuzzy-Logik-Reglers ebenfalls eine Beurteilung der Regelgüte anhand eines Gütekriteriums. Allerdings ist er nicht auf einen analytischen Zusammenhang zwischen Reglerparameter und Gütekriterium angewiesen. Im übrigen kann dieser Zusammenhang für einen Fuzzy-Logik-Regler nicht hergestellt werden.

Bei Verwendung eines genetischen Algorithmus können aber andere Kriterien als die oben genannten zur Optimierung eines Fuzzy-Logik-Reglers herangezogen werden. Neben Kriterien wie Überschwingweite, An- und Ausregelzeit [12] werden dazu auch rein qualitative Bewertungen verwendet [29]. In [4] wird z.B. der gewünschte zeitliche Verlauf der Regelgröße nach einem Führungsgrößensprung vorgegeben und eine Anpassung der Parameter so vorgenommen, daß der gewünschte Verlauf möglichst erreicht wird. Darüber hinaus eignet sich ein genetischer Algorithmus zur Optimierung mehrere Kriterien gleichzeitig [24].

Bei den in der Literatur vorgestellten Beispielen erfolgt die Reglerparameteroptimierung allerdings immer anhand eines Prozeßmodells. Der Vorteil des Fuzzy-Logik-Reglers, eben kein Modell des Prozesses für den Reglerentwurf zu benötigen und mit qualitativem Bedienerwissen auszukommen, ist damit wieder dahin. Ein Vorteil des Fuzzy-Logik-Reglers bleibt seine größere Robustheit, so daß die Anforderungen, die an die Qualität des Prozeßmodells gestellt werden, geringer sind als die für den Reglerentwurf eines klassischen PI-Reglers.

Die Anwendung des genetischen Algorithmus ist nicht an die Wahl eines Fuzzy-Logik-Reglers gekoppelt. Er läßt sich auch auf die Reglerparameterfindung eines PID-Reglers anwenden [29].

Ein Fuzzy-Logik-Regler wird als selbsteinstellender Regler (*self-tuning*) bezeichnet, wenn bei der Optimierung die Parameter der Zugehörigkeitsfunktionen (Modalwerte, Breiten) verändert werden. Als selbstorganisierend (*self-organizing*) bezeichnet man den Regler, wenn die einzelnen Regeln (also die Einträge in die Regelbasismatrix) oder deren Relevanzfaktoren optimiert werden [3, 12, 67].

Abhängig von den zu optimierenden Parametern wird das Übertragungsverhalten eines Fuzzy-Logik-Reglers eher global oder lokal verändert [3, 5]. Werden die Skalierungsfaktoren angepaßt, so ändert sich das Übertragungsverhalten global, da sich ein Skalierungsfaktor auf alle Terme einer Ein- oder Ausgangsgröße bezieht. Wird nur eine Regel ausgetauscht (oder ein Relevanzfaktor in den unscharfen assoziativen Karten modifiziert), so ist dies nur eine geringfügige lokale Änderung im Übertragungsverhalten. Die Modifikation der Zugehörigkeitsfunktionen der unscharfen Eingangsmengen oder der Positionen der Ausgangsmengen stellen in ihrem Ausmaß eine Änderung zwischen diesen beiden Extremen dar. Bei der Veränderung der Reglerparameter X_p und T_n eines PI-Reglers hingegen wird immer eine globale Veränderung des Übertragungsverhaltens vorgenommen.

Ein selbstorganisierender Regler, der die Regelbasis anhand des ITAE-Kriteriums optimiert, wird in [4] aufgezeigt. In [24] wird ein Regelkonzept für die Regelung einer exothermen chemischen Reaktion in einem kontinuierlichen idealen Rührkessel aufgezeigt, das aus einem selbsteinstellenden Fuzzy-Logik-Regler besteht, dessen Zugehörigkeitsfunktionen über einen genetischen Algorithmus angepaßt werden. Dieses Konzept wird als robuster und adaptiver Regler bezeichnet.

3.1.2 Künstliche Neuronale Netze zur Systemidentifikation

Bei der Identifikation wird das mathematische Modell eines Prozesses oder Systems aus vorliegenden Meßdaten ermittelt. Dabei werden Ein- und Ausgangssignale des Prozesses mittels Identifikationsverfahren so ausgewertet, daß ihre Zusammenhänge in einem mathematischen Modell ausgedrückt werden können [40]. Eine einfache Art der experimentellen Modellbildung ist das Ein-/Ausgangsmodell (*input-output-model*). Es versucht, die Ausgangsgröße des Systems $y_k(t+1)$ für den nächsten Zeitschritt $t+1$ aus vergangenen Werten der Ein- und Ausgangsgrößen ($u(t-f), y_k(t-f)$ mit $f=0..(N_V-1)$) vorherzusagen. Das mathematische Modell für das dynamische Verhalten des Systems ist dabei eine Funktion zwischen den Ein- und Ausgangsgrößen. Interessiert darüber hinaus der innere Zustand des Systems (interne Meßgrößen), so wird übergegangen zur Zustandsgrößendarstellung (*state-space-model*). Auf die Zustandsgrößendarstellung soll hier nicht eingegangen werden.

Im Rahmen eines modelladaptiven Regelungskonzepts handelt es sich bei dem zu identifizierenden System um die Regelstrecke. Um Ein-/Ausgangsdaten einer Regelstrecke zu gewinnen, kann ihr Verhalten im geöffneten oder im geschlossenen Regelkreis beobachtet werden [18]. Bei geöffnetem Regelkreis wird die Antwortfunktion der Strecke auf eine charakteristische Eingangsfunktion (z.B. Sprung- oder Impulsfunktion) aufgezeichnet. Bei geschlossenem Regelkreis werden Zufallssignale zur Charakterisierung der Regelstrecke herangezogen: Entweder wird der Sollwert der Regelstrecke zufällig variiert, oder man läßt eine genau meßbare Störgröße mit zufälligen Werten auf die Regelstrecke einwirken [41]. Besondere Bedeutung besitzt in den meisten Fällen die Sprungfunktion [42], denn jeder Sprung läßt sich als Summe von unendlich vielen Sinusfunktionen mit unterschiedlicher Frequenz deuten.

Für die Identifikation linearer Systeme und Prozesse verwendet man meist binäre Rauschsignale (PRBS, *pseudo random binary signals*). Diese sind für die Identifikation nichtlinearer Systeme u.U. ungünstig. Hier werden dann gaußverteilte oder normalverteilte Testsignale verwendet. Bei diesen Signalen wird jeder Wert ausreichend lange gehalten, um eine gute Systemanregung zu gewährleisten [43]. Die Abtastzeit, mit der die Meßwerte ($u(t), y_k(t)$) aufgenommen werden, muß im Vergleich zu den Systemzeitkonstanten kurz sein [40, 43].

Beim Ein-/Ausgangsmodell besteht die Aufgabe in einer Prädiktion $o_k(t+1)$ der Ausgangsgröße $y_k(t+1)$ aus zurückliegenden externen Eingangsgrößen $u(t-f)$ und zurückliegenden Ausgangsgrößen $y_k(t-f)$ ($f=0..(N_V-1)$). Werden die Ein- bzw. Ausgangsgrößen $u(t-f)$ und $y_k(t-f)$ zum Vektor $x_h(t)$ zusammengefaßt (35), so besteht die experimentelle Modellbildung aus dem Auffinden einer geeigneten Funktion G (Abbildungsvorschrift) (36).

$$x_h(t) = \begin{pmatrix} u(t), u(t-1), \dots, u(t-N_V+1), \\ y_k(t), y_k(t-1), \dots, y_k(t-N_V+1) \end{pmatrix} \quad (35)$$

$$o_k(t+1) = G(x_h(t), \Theta) \quad (36)$$

Diese Funktion G soll die Meßwerte $y_k(t)$ der Ausgangsgröße des Systems möglichst genau wiedergeben. Es gilt, die Abweichung (37) (Residuum)

$$e_i(t+1) = y_k(t+1) - o_k(t+1) \quad (37)$$

zwischen Messung und Prädiktion zu minimieren [18, 43]. Alle Modellparameter werden in einem Vektor Θ zusammengefaßt. Die Bestimmung von Θ erfolgt durch Auswertung von Ein-/Ausgangsdaten ($u(t), y_k(t)$) des betreffenden Prozesses, die vorliegen oder durch ein Experiment zu generieren sind. Die Modelloptimierung und -beurteilung erfolgt im allgemeinen an Hand des mittleren quadratischen Fehlers $E(\Theta)$ [43]

$$E(\Theta) = \frac{1}{r} \sum_{i=1}^r e_i^2(\Theta) \quad (38)$$

Die Theorie der Identifikation linearer Modelle liefert eine Vielzahl an möglichen Ein-/Ausgangsmodellformen, wie das lineare Autoregressionsmodell (AR), das Modell des gleitenden Durchschnitts (MA) oder die Kombination aus beiden (39) (ARMA, *auto regressive moving average*) [17, 34, 43].

$$o_k(t+1) = \sum_{f=0}^{N_V-1} a_f y_k(t-f) + \sum_{f=0}^{N_V-1} b_f u(t-f) \quad (39)$$

Autoregressiv bezieht sich hierbei auf die y -Terme. Die Summation muß bei Eingangsgröße u und Ausgangsgröße y_k nicht über die gleiche Anzahl N_V der Vergangenheitswerte laufen. Dies sei hier nur zur einfacheren Darstellung wegen angenommen. Ist die Anzahl der benötigten Vergangenheitswerte

unterschiedlich, so kann dies durch das Setzen der entsprechenden Koeffizienten a_f oder b_f auf Null ausgedrückt werden. Die lineare Gleichung (39) gilt nur für die Betrachtung in der näheren Umgebung eines Arbeitspunktes [43]. Soll ein größerer Bereich um den Arbeitspunkt im Modell erfaßt werden, oder verhält sich der Prozeß nichtlinear, so kann ein Übergang zu nichtlinearen Modellen durch die Aufnahme von nichtlinearen Funktionen v , w in (39) erfolgen [17, 18, 34]:

$$o_k(t+1) = \sum_{f=0}^{N_v-1} v_f(a_f y_k(t-f)) + \sum_{f=0}^{N_w-1} w_f(b_f u(t-f)) \quad (40)$$

Anstelle eines mathematischen Modells wird in dieser Arbeit ein künstliches neuronales Netz verwendet werden, um das Übertragungsverhalten der Regelstrecke zu modellieren [18]. Auch bei einer Modellbildung mit neuronalen Netzen ist die möglichst exakte Nachbildung des dynamischen und statischen Verhaltens der Regelstrecke erwünscht [38]. Im Gegensatz zu dem linearen ARMA-Modell (39) besitzt ein neuronales Netz mit s-förmigen Transferfunktionen jedoch die inhärente Fähigkeit, nicht-lineares Prozeßverhalten darstellen zu können.

Die im folgenden beschriebene experimentelle Modellbildung mit künstlichen neuronalen Netzen basiert auf dem in Anhang B beschriebenen vorwärtsgekoppelten Mehrschicht-Perzeptron. Wenn hier von neuronalen Netzen die Rede ist, ist damit immer das Mehrschicht-Perzeptron gemeint. Andere Netzwerkarten finden hier keine Berücksichtigung.

Bei der Systemidentifikation mit einem Mehrschicht-Perzeptron ist der in (53) gebildete Vektor x_i der Eingabevektor des neuronalen Netzes. Dieses Verfahren wird auch als *tapped-delay-line*-Methode bezeichnet [17]. Würde man ein neuronales Netz aus der Klasse der rekurrenten neuronalen Netze verwenden, so wäre die Konstruktion eines Eingabevektors überflüssig. Bei dieser Art von Netzen reicht als Eingabe die Eingangsgröße $u(t)$ des Systems aus, um den Ausgangswert $y_k(t+1)$ vorherzusagen [17]. Rekurrente neuronale Netze führen die Vergangenheitswerte der Ein- und Ausgangsgröße intern zurück und verfügen so über ein Gedächtnis. Mehrschicht-Perzeptrons als vorwärtsgekoppelte neuronale Netze hingegen haben kein Gedächtnis. Durch die *tapped-delay-line*-Methode wird das gedächtnislose Mehrschicht-Perzeptron quasi um ein Gedächtnis erweitert, um so das dynamische System nachbilden zu können [26, 44]. Dies ermöglicht ferner ein einfacheres Training des Mehrschicht-Perzeptrons im Gegensatz zu rekurrenten Netzen.

In der folgenden Abbildung 4 ist die Systemidentifikation mit einem Mehrschicht-Perzeptron in der beschriebenen seriell-parallelen Form (*series-parallel model*) dargestellt [17, 33]. Der Operator z^{-1} ist eine Zeitverschiebungsfunktion $z^{-1}(u(t)) = u(t-1)$, der zum Ausdruck bringt, daß der Eingabevektor $x_h(t)$ aus aktuellen oder zeitlich zurückliegenden Werten gebildet wird.

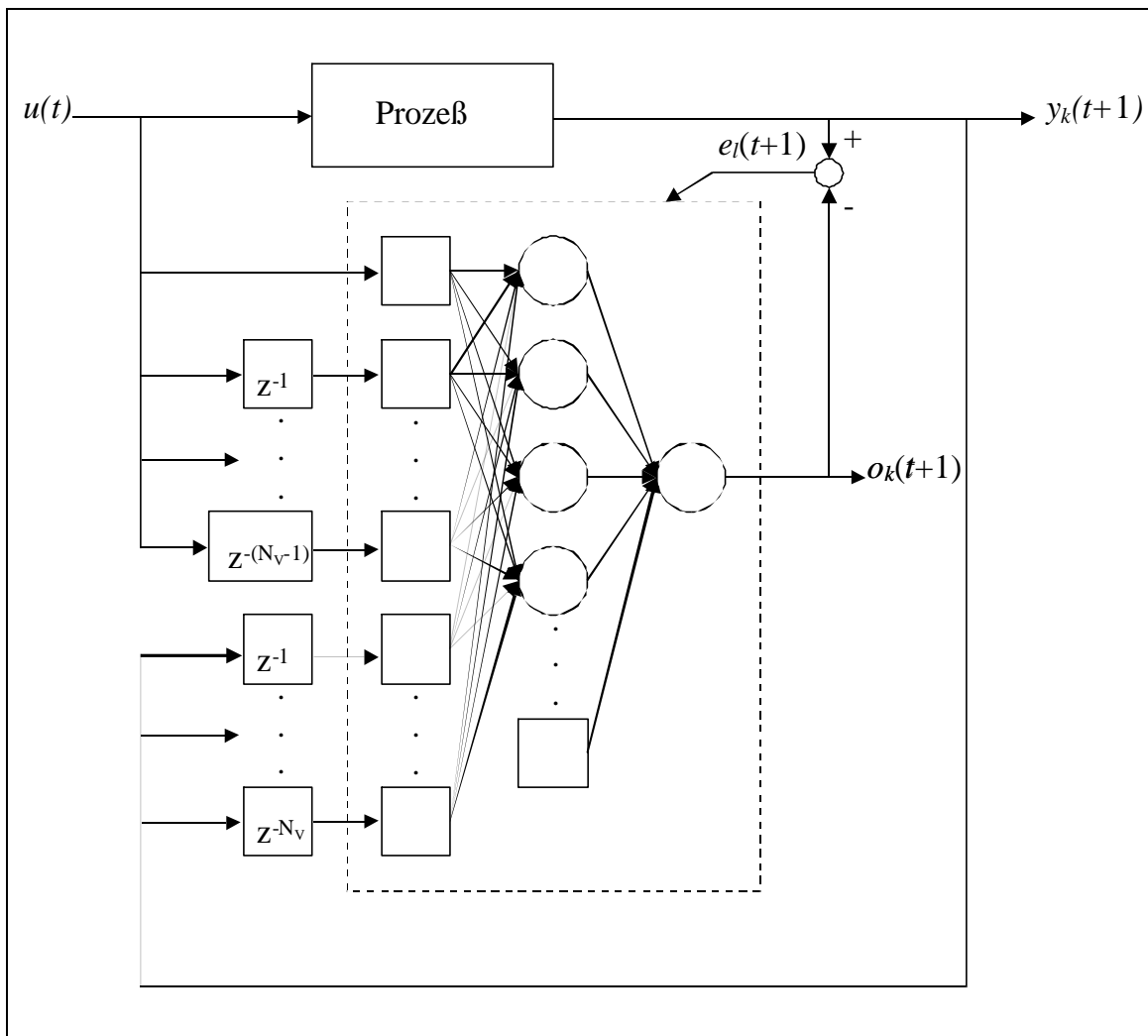


Abb. 4 Seriell-parallele Systemidentifikation mit einem dreischichtigen Perzeptron

In einer weiteren Form der Modellbildung kann ein neuronales Netz auch als paralleles Modell (*parallel model*) benutzt werden. Dann werden bei der Bildung des Eingabevektors $x_h(t)$ (35) nicht die Meßdaten des Prozesses $y_k(t-f)$ verwendet, sondern die vorhergehenden Ausgaben des Modells selbst $o_k(t-f)$ (mit $f=0..(N_V-1)$). In gewisser Weise verhält sich dieses Modell wie ein

rekurrentes neuronales Netz. Die Betriebsweise als paralleles Modell zeigt aber in der Lernphase eine schlechtere Konvergenz und wird hier nicht weiter benutzt.

Als ein Hauptvorteil von künstlichen neuronalen Netzen gegenüber klassischen Identifikationsmethoden werden die Einfachheit der Implementierung und die guten (universellen) Approximationsfähigkeiten genannt [17, 68]. Die Aufgabe der Systemidentifikation besteht aus dem Auffinden einer Funktion zwischen den vergangenen Ein- und Ausgangswerten des Systems (angeordnet als Vektor $x_h(t)$) auf der einen Seite und dem zukünftigen Ausgangswert ($y_k(t+1)$) auf der anderen Seite. Dabei geht die klassische Modellbildung von einem vorher festgelegten funktionalen Zusammenhang aus und paßt die Parameter des Modells an. Dazu müssen die Ordnung und die Totzeit des Systems bekannt sein [40]. Bei der Systemidentifikation mit einem neuronalen Netz müssen weder der funktionale Zusammenhang noch die Systemparameter bekannt sein, um es dem neuronalen Netz zu ermöglichen seine Parameter (die Netzgewichte) an das Ein-/Ausgangsverhalten anzupassen [17]. Dieser Vorteil, daß kein a-priori-Wissen über das System verlangt wird [18], wird aber erkauft durch den Nachteil der fehlenden Interpretierbarkeit der Netzgewichte [15, 24], die keine Auskunft über physikalische Variablen geben.

Neuronale Netze besitzen die Fähigkeit, jede nichtlineare Funktionen von praktischem Interesse [23] beliebig genau zu approximieren [18, 38], wenn ein dreischichtiges Netz eingesetzt [20, 32] und eine ausreichend hohe Anzahl an operativen Neuronen in der verborgenen Schicht eingesetzt wird. Dabei ist ein neuronales Netz in der Lage, jede Funktion mit einer endlichen Anzahl an Diskontinuitäten darzustellen, wenn seine Neuronen über einen Bias-Eingang verfügen, die Neuronen der verborgenen Schicht eine s-förmige Transferfunktion t_j haben [17] und das Ausgabeneuron eine lineare Transferfunktion t_k besitzt [16].

Besitzt das Ausgabeneuron k eine s-förmige Transferfunktion t_k , so ist der Bereich des Ausgangswertes $o_k(t+1)$ auf den schmalen Bereich der Transferfunktion beschränkt. Es ist daher sinnvoll, die Identität (B.4) als Transferfunktion des Ausgabeneurons zu wählen [16]. Eine andere Möglichkeit besteht darin, alle Ausgabewerte $y_k(t+1)$ des Trainingsdatensatzes auf einen Bereich von 0.1 bis 0.9 bei der logistischen Funktion bzw. von -0.9 bis 0.9 bei der Hyperbeltangens-Funktion zu skalieren [15, 18]. Nachteil dieser Methode ist die Notwendigkeit zu einer erneuten Skalierung und

eines längeren Trainings, wenn neue Eingabe-Ausgabe-Paare dem Trainingsdatensatz hinzugefügt werden, deren Ausgabewerte $y_k(t+1)$ außerhalb des Wertebereichs der Transferfunktion t_k liegen.

Prinzipiell sind Neuronale Netze nicht in der Lage zu extrapolieren [13, 20]. Unter dem Extrapolationsbereich versteht man dabei den Eingangsbereich des Eingabevektors x_h , der außerhalb des oberflächenminimalen $(n-1)$ -dimensionalen Hyper-Volumenkörpers liegt, welcher die Eingabevektoren aller r Eingabe-Ausgabe-Paare beinhaltet [37]. Aufgrund der fehlenden Extrapolationsfähigkeit sollte Sorge dafür getragen werden, daß der Trainingsdatensatz den gesamten möglichen Eingangsraum des Eingabevektors x_h umfassend abdeckt [20]. Es kann jedoch nicht immer ausgeschlossen werden, daß bei Verwendung des neuronalen Netzes der Approximationsbereich verlassen wird und man in den Extrapolationsbereich gerät. Daher sollte die Approximationsfunktion auch im Extrapolationsbereich sinnvolle Werte liefern, obwohl in diesem Bereich definitionsgemäß kein durch Stützwerte repräsentiertes Wissen vorhanden ist [37]. Diese Bedingung ist bei mathematischen Modellen eher erfüllt als bei neuronalen Netzen.

Neben der universellen Funktionsapproximation ist die Lernfähigkeit eine weitere herausragende Eigenschaft künstlicher neuronaler Netze [38], die es erlaubt, das neuronale Netz parallel zu einem laufenden Prozess mit sich ändernden Prozeßparametern zu trainieren. Dabei sollte das neuronale Netz bereits anhand von vorher gewonnenen, repräsentativen Ein-/Ausgangsdaten des zu modellierenden Prozesses vortrainiert werden, um eine schnellere Konvergenz zu gewährleisten, wenn das Netz schließlich parallel zu dem Prozess trainiert wird [18]. Es wird aber angemerkt, daß das Training des Netzes idealerweise nur parallel zu dem Prozess stattfinden sollte [18]. Das Vortraining kann als das Aufsuchen von vorteilhaften Startgewichten für das neuronale Netz verstanden werden.

Ein Nachteil, der mit der *tapped-delay-line*-Methode für Mehrschicht-Perzeptrons verbunden ist, ist eine längere Rechenzeit, die durch eine zu hohe Anzahl ($h = 2 \cdot N_V$) der Elemente des Eingabevektors x_h verursacht wird. Ist die Ordnung oder die Totzeit des zu identifizierenden Systems unbekannt, so muß die Anzahl h größer als eigentlich notwendig gewählt werden [17, 33]. Darüber hinaus besteht die Gefahr, daß eine hohe Anzahl an Elementen des Eingabevektors x_h es dem neuronalen Netz ermöglicht, das Meßrauschen in den Eingabe-Ausgabe-Paaren mitzulernen [17].

Eine einfache Anwendung des trainierten neuronalen Netzes ist die Approximation [18] der Ableitung der Regelgröße $y(t)$ nach der Stellgröße $u(t)$.

$$y'_k(u) \approx \frac{\Delta y_k(u)}{\Delta u} \approx \frac{\Delta o_k(u)}{\Delta u} \quad (41)$$

$$\text{mit } \Delta y_k(u) = y_k(u(t-1)) - y_k(u(t-2))$$

$$\text{und } \Delta u = u(t-1) - u(t-2)$$

Wie jedes Modell kann das neuronale Netz aber auch als Prozeßmodell in einer Simulation dienen und zu den Aufgaben des Reglerentwurfs oder der Regleroptimierung herangezogen werden. Die Lernfähigkeit neuronaler Netze ermöglicht es, sie als Modell *online*, parallel zu dem identifizierten Prozeß einzusetzen. Über einen Adaptionsalgorithmus können dann die Parameter eines Reglers optimal an das Modell angepaßt werden. Bei ausreichender Güte des Modells oder Robustheit des Reglers ist dann für den Regelkreis, bestehend aus Regler und Prozeß, ein optimales Regelverhalten zu erwarten.

In dieser Arbeit werden künstliche neuronale Netze wegen ihrer einfachen Implementierung und ihrer universellen Approximationseigenschaften zur experimentellen Modellbildung verwendet. Aus Vorversuchen ist bekannt, daß mit verschiedenen Reaktionsmedien und Durchmischungsverhältnissen die Totzeiten der Regelstrecke erheblich variieren. Für eine aussagekräftige Parameterschätzung bei einem mathematischen Modell müßte diese Totzeit vorher abgeschätzt werden. Mit der Anwendung eines Fuzzy-Logik-Reglers wird ein neues leistungsfähiges Werkzeug der Regelungstechnik verwendet, und es ist nicht sinnvoll, sich dann bei der Systemidentifikation nur auf lineare Modell zu beschränken, wenn für diese Aufgabe neuronale Netze zur Verfügung stehen.

3.1.3 Adaptive Regelung

Die Verwendung eines adaptiven Reglers ist immer dann angebracht, wenn die Regelstrecke gänzlich unbekannt oder sich zeitlich ändernden Streckenparametern unterworfen ist [66, 67, 69]. Die sich ändernden Parameter umfassen die Totzeit, die Ordnung und die Verstärkung (Kapazität) einer Regelstrecke. Diese können sich ändern, wenn das System in verschiedenen Arbeitspunkten betrieben wird, oder wenn sich während eines ablaufenden Prozesses die Eigenschaften der Regelstrecke zeitlich ändern [12]. Es kann bei unveränderten Reglerparametern eines PID-Reglers zu einem instabilen Verhalten des geschlossenen Regelkreises kommen, wenn die Totzeit oder die Streckenverstärkung zunimmt [66]. Da es sich meist um eine fortlaufende Variation der Regelstreckenparameter handelt, ist auch eine kontinuierliche Anpassung des Reglers nötig [41], um ein zufriedenstellendes Regelverhalten beibehalten zu können.

Beim hier vorgestellten Reaktionskalorimeter wird die Ergänzung des Fuzzy-Logik-Reglers um einen Adaptionalgorithmus aus zwei Gründen in Erwägung gezogen. Zum einen sind die Streckenparameter der Regelstrecke bei der Befüllung des Reaktors mit unterschiedlichen anfänglichen Reaktionsmischungen tatsächlich unbekannt, zum anderen treten im Laufe einer Reaktion Änderungen der Streckenparameter auf, die eine Anpassung der Reglerparameter notwendig machen. So ändert sich durch Dosierung einer Reaktionskomponente das Volumen V_r , und die Wärmekapazität der Reaktionsmischung und die Wärmeaustauschfläche nehmen zu. Bei sich ändernder Viskosität oder Dichte des Reaktionsmediums oder variiertes Drehzahl des Rührers kommt es zu veränderter Durchmischung im Reaktionsraum. Die Viskosität hat ferner Einfluß auf die Dicke (und damit die Wärmekapazität) der Prandtlischen Grenzschicht und auf die Wärmeübergangsverhältnisse. Gerade die Betriebsweise als satzweise betriebener Rührkessel geht mit den beschriebenen Streckenparametervariationen einher [2, 5, 66].

Voraussetzung jeder adaptiven Regelung ist das Erkennen von Prozeßparameteränderungen der Regelstrecke, um sie dann durch geeignete Modifikationen der Reglerparameter optimal kompensieren zu können [3, 66].

Der Adaptionsalgorithmus gliedert sich in drei Schritte zur Anpassung der Reglerparameter [41, 66, 69]:

1. Identifikation der Regelstreckenparameter bzw. deren Änderung
2. Beurteilung des Reglers anhand eines Gütekriteriums
3. Modifikation des Reglers (Reglerparameteränderung)

Dazu wird eine zweite Regelschleife zusätzlich zu dem primären Regelkreis bestehend aus Regler und Regelstrecke benötigt [41, 66, 69]. Es handelt sich hier um ein hierarchisches System aus zwei Schleifen mit negativer Rückkopplung. Zunächst einmal ist der Regler dafür zuständig, die Regelgröße y_k mittels des Regelalgorithmus auf den Führungswert $y_{k,soll}$ einzustellen. Dieser Regelkreis bleibt geschlossen und funktionstüchtig. Ihm überlagert ist der zweite Regelkreis, bestehend aus Parameteridentifikation, Reglerbeurteilung und -modifikation. Während der erste Regelkreis den Einfluß von Störgrößen auf die Regelgröße minimiert, verringert der zweite Regelkreis den Einfluß von Streckenparameteränderungen auf die Regelgüte [41]. Diese Form der Reglerparameteranpassung auf Basis der negativen Rückkopplung hat das gleiche Problem wie die Regelgrößenanpassung mittels *Feedback*-Regler: Eine schlechte Regelgüte muß überhaupt erst auftreten und registriert werden, bevor mit einer Reglerparameteranpassung darauf reagiert werden kann [66].

Eine adaptive Regelung in der beschriebenen Art ist grundsätzlich ein nichtlineares System. Durch die zusätzlichen Entwurfsschritte, die bei der Gestaltung gegeben sind, verfügt dieses System auch über zusätzliche Freiheitsgrade, verglichen mit einem nicht-adaptiven Reglerentwurf [41]. Daher existiert in der Literatur auch eine Vielzahl möglicher Adaptionsverfahren.

Prinzipiell gibt es für die Konstruktion eines adaptiven Systems zwei Herangehensweisen [3, 41, 66, 67]. Bei der direkten adaptiven Regelung (*direct adaptive control, model reference adaptive control*) wird eine Regelstrecke in mehreren bekannten Arbeitspunkten betrieben, wobei für jeden Arbeitspunkt optimale Reglerparameter bekannt sind. Aufgabe der adaptiven Regelung ist es, den aktuellen Arbeitspunkt der Regelstrecke anhand von Regel- und Stellgröße sowie Informationen über die Umgebungsbedingungen zu identifizieren und die Reglerparameter dafür einzustellen. Für die adaptive Regelung eines Reaktionskalorimeters ist diese Form eher ungeeignet. Für diese Aufgabe eignet sich das zweite adaptive Konzept, die indirekte adaptive Regelung (*indirect adaptive control*). Hierbei wird eine

Systemidentifikation der Regelstrecke durchgeführt und die Reglerparameter optimal an das Modell der Regelstrecke angepaßt (daher die Bezeichnung indirekt). Dabei wird angenommen, daß das Modell exakt mit dem Prozeß übereinstimmt. Für dieses Konzept sind keine weiteren Vorkenntnisse über das Verhalten der Regelstrecke nötig. Prinzipiell kann jedes Systemidentifikationsverfahren mit jedem Reglerentwurfverfahren kombiniert werden. Für den Reglerentwurf für lineare Strecken existieren systematische Methoden [41, 67, 69].

Als Alternative zum Reglerentwurf durch Polvorgabe [41] ist die Reglerparameterbestimmung durch Minimierung eines Gütekriteriums (*performance index*) möglich [70]. Dieses Verfahren ist unter dem Begriff optimaler Reglerentwurf bekannt. Als Gütekriterien eignen sich die in Kapitel 3.1.1 genannten Integralkriterien (33) und (34) oder Kriterien wie An- und Ausregelzeit oder Überschwingweite [9, 64]. Für beliebige Gütekriterien kann keine analytische Lösung erwartet werden, weshalb man sich für lineare Systeme meist auf quadratische Funktionen beschränkt [70].

Wird die quadratische Regelfläche (33) als Gütekriterium für die Optimierung der Reglerparameter herangezogen, so ist das Ergebnis ein Minimum Varianz Regler [69]. Bei einer Optimierung nur anhand des *ISE* kann der Adaptionalgorithmus Stellgrößen fordern, die so groß sind, daß sie vom Stellglied nicht realisiert werden können [38]. In diesem Fall ist auch die Stabilität des Regelkreises nicht gewährleistet. Die quadratische Regelfläche (42) wird daher um einen zweiten Term ergänzt:

$$J = \int \left[e(t)^2 + \frac{1}{k} \Delta u(t)^2 \right] dt \quad (42)$$

$$\text{mit } \Delta u(t) = u(t) - u(t-1)$$

Der zweite Term soll die vom Regler geforderten Stellgrößenänderungen begrenzen und die Stabilität des Regelkreises gewährleisten. Da der Regler weiterhin die Varianz der Regeldifferenz minimal halten soll (*ISE*), muß die Konstante k so groß wie möglich gewählt werden [18]. Durch die Wahl von k kann hier ein Kompromiß eingegangen werden zwischen schneller Ausregelung der bestehenden Regeldifferenz und akzeptablen Werten der Stellgröße [70].

Für die Ausführung eines Adaptionalgorithmus parallel zum Regelkreis stehen zwei Strategien zur Auswahl [41], die sich in der zeitlichen Reihenfolge der Regel- und Anpassungsaufgaben unterscheiden.

Tab. 1: Adaptive Regelstrategien

Strategie A	Strategie B
1. Erfassung der Regelgröße	1. Erfassung der Regelgröße
2. Prozeßmodell aktualisieren	2. Stellgröße berechnen
3. Reglerparameter anpassen	3. Stellgröße ausgeben
4. Stellgröße berechnen	4. Prozeßmodell aktualisieren
5. Stellgröße ausgeben	5. Reglerparameter anpassen

Bei Strategie A wird der Regler vor Bestimmung der Stellgröße an das aktualisierte Streckenmodell angepaßt und kann so eine bessere Stellgrößenänderung ausgeben. Damit verbunden ist aber der Nachteil, daß eine lange zeitliche Verzögerung zwischen Regelgrößenerfassung und Ausgabe der Stellgröße besteht. Bei Strategie B wird (wie bei einem nicht-adaptiven Regler) die Stellgröße direkt nach der Erfassung der Regelgröße ausgegeben, und erst danach erfolgt die Aktualisierung des Modells und die Reglerparameteranpassung. Auf diese Weise kommt es zu keiner weiteren Totzeit im geschlossenen Regelkreis; allerdings folgt das Regelverhalten dem Modell um einen Regelzyklus versetzt. Dies ist bei sich langsam ändernden Streckenparametern zulässig.

Für den Adaptionalgorithmus wird das Modell als fehlerfrei betrachtet. Ein daran angepaßter Regler ist dann in der Lage, die reale Regelstrecke zu regeln [18]. Für eine Systemidentifikation der Regelstrecke muß der Prozeß hinreichend angeregt sein, so daß im gesamten Arbeitsbereich des Modells verteilte Meßdaten ausreichend vorhanden sind. Da die Identifizierung der Strecke aber im geschlossenen Regelkreis erfolgt, kann die Anregung sehr gering sein. Damit wären auch das Regelstreckenmodell und die daran angepaßten Reglerparameter unzureichend [41]. Hilfreich ist hier ein robustes Verhalten des Reglers, der mit einer geringeren Modellgüte auskommt und Abweichungen zwischen Modell und Regelstrecke zuläßt, dabei aber ein zufriedenstellendes Regelverhalten gewährleistet.

Realisierung mit einem Neuro-Fuzzy-Regler

Die Entwicklung einer adaptiven Regelung für das Reaktionskalorimeter resultierte in einem Neuro-Fuzzy-Regler und einem Adaptionalgorithmus, der sich eines neuronalen Netzes zur Systemidentifikation bedient. Der Neuro-Fuzzy-Regler basiert auf dem in Anhang C beschriebenen Neuro-Fuzzy-System (NFS).

Neuro-Fuzzy-Regler können wie Fuzzy-Logik-Regler entlang linguistischer Pfade entworfen werden. Dabei sind einige Beschränkungen zu berücksichtigen, die ebenfalls in Anhang C angesprochen sind. Neuro-Fuzzy-Regler besitzen die gleiche Durchschaubarkeit wie Fuzzy-Logik-Regler, sind aber nicht nur rein statische Übertragungsglieder, sondern können anhand von Beispieldaten angepaßt werden, sind also lernfähig. Die Anpassung eines Neuro-Fuzzy-Systems basiert auf dem für neuronale Netze bekannten Fehlerrückführungsalgorithmus, der an die Besonderheiten von Neuro-Fuzzy-Systemen angepaßt ist.

Für die folgende Darstellung der Reglerparameteradaption (Adaption der Parameter des Neuro-Fuzzy-Reglers) wird angenommen, daß der Sollwert der Regelgröße $y_{k,soll}$ ein zeitlich konstanter Wert ist. Der Adaptionalgorithmus fußt darauf, daß zum Zeitpunkt t ein Prozeßmodell der Regelstrecke besteht, das das Ein-/Ausgangsverhalten der Strecke ausreichend gut abbildet (Kapitel 3.1.2). Darüber hinaus sind zu diesem Zeitpunkt die Meßdaten für die vorangehenden Zeitschritte $t-1$ und $t-2$ bekannt und können im Adaptionalgorithmus verwendet werden. Nennenswerte Totzeiten werden zunächst noch nicht berücksichtigt. Zum Zeitpunkt t besteht die Regelabweichung:

$$x_w(t) = y_k(t) - y_{k,soll} \quad (43)$$

Diese Regelabweichung (43) kommt zustande, weil die letzte Stellgrößenänderung $\Delta u(t-1)$ nicht ausreichend war, um die Regelgröße y_k auf ihren Sollwert $y_{k,soll}$ zurückzuführen. Die Stellgrößenänderung $\Delta u(t-1)$ ist vom Neuro-Fuzzy-Regler anhand der Regelabweichung (43) und der zeitlichen Regelgrößenänderung (45) des vorhergehenden Zeitschritts $t-1$ gebildet worden.

$$x_w(t-1) = y_k(t-1) - y_{k,soll} \quad (44)$$

$$\frac{\Delta y_k(t-1)}{\Delta t_a} = \frac{y_k(t-1) - y_k(t-2)}{\Delta t_a} \quad (45)$$

Da zum Zeitpunkt t aufgrund der falschen Stellgrößenänderung $\Delta u(t-1)$ immer noch die Regelabweichung $x_w(t)$ besteht, erhebt sich die Frage, um welchen Wert $\Delta \Delta u(t-1)$ die Stellgrößenänderung $\Delta u(t-1)$ verändert werden müßte, um einer Regelabweichung von Null näher zu kommen. In erster Näherung entspricht $\Delta \Delta u(t-1)$ der Regelabweichung, dividiert durch die Ableitung der Regelgröße nach der Stellgröße $y_k(u)$ (46). Diese Ableitung ist nicht analytisch bekannt und muß numerisch anhand des Regelstreckenmodells approximiert werden (41).

$$\Delta \Delta u(t-1) = \frac{x_w(t)}{y'_k(u)} \approx x_w(t) \cdot \frac{\Delta u}{\Delta o_k(u)} \quad (46)$$

Mit diesem Wert wird eine korrigierte Stellgrößenänderung $\Delta u_{korr}(t-1)$ berechnet, welche die Regeldifferenz vom Zeitpunkt $t-1$ zum Zeitpunkt t auf Null gebracht hätte (47).

$$\Delta u_{korr}(t-1) = \Delta u(t-1) + \Delta \Delta u(t-1) \quad (47)$$

Damit verfügt man neben der tatsächlichen Stellgrößenänderung $\Delta u(t-1)$ über die korrigierte Stellgrößenänderung $\Delta u_{korr}(t-1)$, die der Neuro-Fuzzy-Regler aufgrund der Regelabweichung (44) und der zeitlichen Änderung der Regelgröße (45) hätte berechnen sollen. Mit den beiden Größen $\Delta u(t-1)$ und $\Delta u_{korr}(t-1)$ läßt sich nun der Fehler (48) des Neuro-Fuzzy-Systems bilden, der sich zu (49) vereinfacht.

$$E_t = \frac{1}{2} (\Delta u_{korr}(t-1) - \Delta u(t-1))^2 \quad (48)$$

$$E_t = \frac{1}{2} (\Delta \Delta u(t-1))^2 \quad (49)$$

Dieser Fehler E_t kann mit dem Fehlerrückführungsalgorithmus zur Anpassung des Übertragungsverhaltens des Neuro-Fuzzy-Reglers verwendet werden.

In Abbildung 5 ist der beschriebene Adaptionalgorithmus grafisch dargestellt. Der primäre Regelkreis besteht aus Neuro-Fuzzy-Regler und zu regelndem Prozeß. Parallel zum Prozeß wird das neuronale Netz betrieben,

welches eine experimentelle Modellbildung vornimmt. Die Adaption der Reglerparameter des Neuro-Fuzzy-Reglers geht von der Regelabweichung x_w aus, die durch das Prozeßmodell in eine Korrektur der Stellgrößenänderung $\Delta\Delta u$ umgewandelt wird.

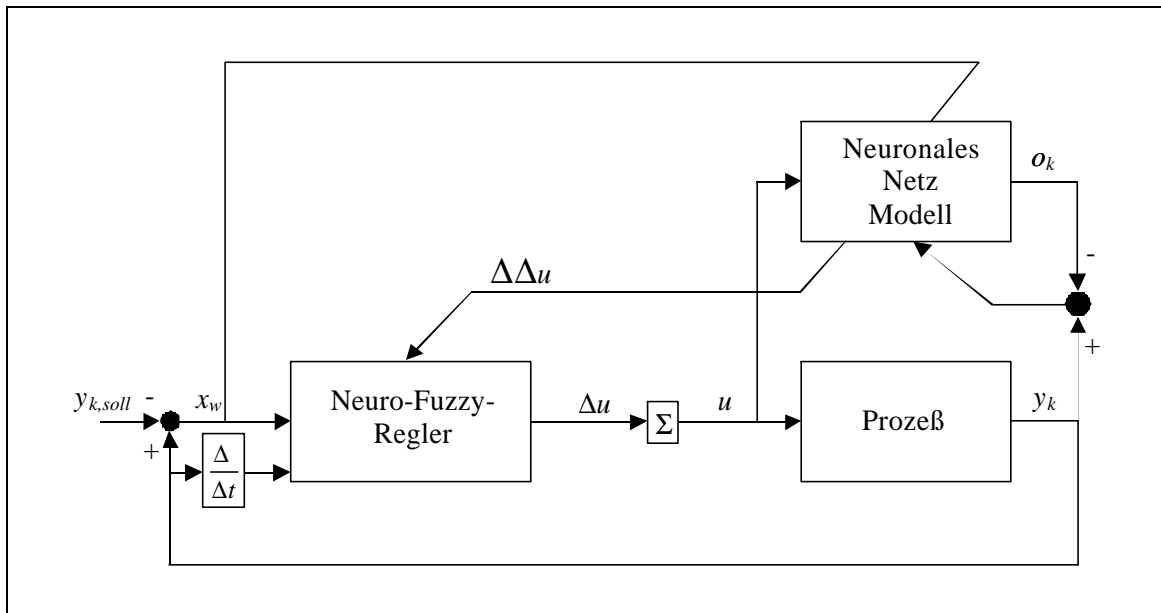


Abb. 5 Regelkreis mit paralleler Prozeßidentifikation und adaptivem Neuro-Fuzzy-Regler

Bei der Implementierung des Adaptionalgorithmus für den Neuro-Fuzzy-Regler wird noch eine Begrenzung eingeführt [18]: Die Reglerparameter dürfen nicht weiter angepaßt werden, wenn die Stellgröße ihren minimalen Wert erreicht hat, und durch (46) eine weitere Verringerung gefordert wird. Gleiches gilt für den Fall, daß die Stellgröße ihr Maximum erreicht hat, und eine weitere Erhöhung der Stellgrößenänderung gefordert wird.

Vorteil des oben beschriebenen Adaptionalgorithmus für einen Neuro-Fuzzy-Regler ist, daß lediglich die Parameter (Eingangs-/Ausgangsmengen oder Regeln) des Reglers angepaßt werden, die bei der gegebenen Regelabweichung $x_w(t-1)$ und Regelgrößenänderung $\Delta y_k(t-1)/\Delta t_a$ für die unzureichende Stellgrößenänderung $\Delta u(t-1)$ verantwortlich sind. Werden die Skalierungsfaktoren der Ein- und Ausgangsgrößen bei der Adaption nicht verändert, so wird das Übertragungsverhalten des Neuro-Fuzzy-Reglers nur in einem lokalen Ausmaß variiert. Abhängig von den gewählten Lernraten für den Fehlerrückführungsalgorithmus kann es bei einer solchen Änderung trotzdem zu einem instabilen Verhalten des ge-

schlossenen Regelkreises kommen. Hier ist es von Vorteil, die Stabilität des Regelkreises und die Regelgüte in einer Simulation zu testen. Daher wird der Adaptionalgorithmus in ein übergeordnetes Bewertungssystem eingebunden. Wie aus Abbildung 6 zu ersehen, wird die Regelgüte des Reglers vor der Parameteranpassung anhand des Gütekriteriums J_{vor} in einer Simulation (*long range prediction control* [18]) bestimmt. Dann wird die Parameterveränderung vorgenommen und anschließend die Regelgüte J_{nach} des Reglers mit den neuen Parametern bestimmt. Nur wenn sich die Regelgüte verbessert hat, wird die Parameteränderung beibehalten.

Die hier vorgestellte adaptive Regelung bedient sich eines Adaptionalgorithmus, der sich bequem in das Konzept der Fuzzy-Logik-Regler einfügt, welche sich trotz eines mathematisch exakten Verfahrens umgangssprachlich erfassen lassen. Genau so einfach verständlich sollte der Adaptionalgorithmus dann auch sein [18].

Der robuste, statische Fuzzy-Logik-Regler wird in ein lernfähiges Neuro-Fuzzy-System übertragen und erhält damit die Fähigkeit, sich über einen Adaptionalgorithmus an wandelnde Streckenparameter zeitvarianter Prozesse anzupassen [12]. Damit vereint er die beiden sich ergänzenden, komplementären Konzepte der robusten und adaptiven Regelung.

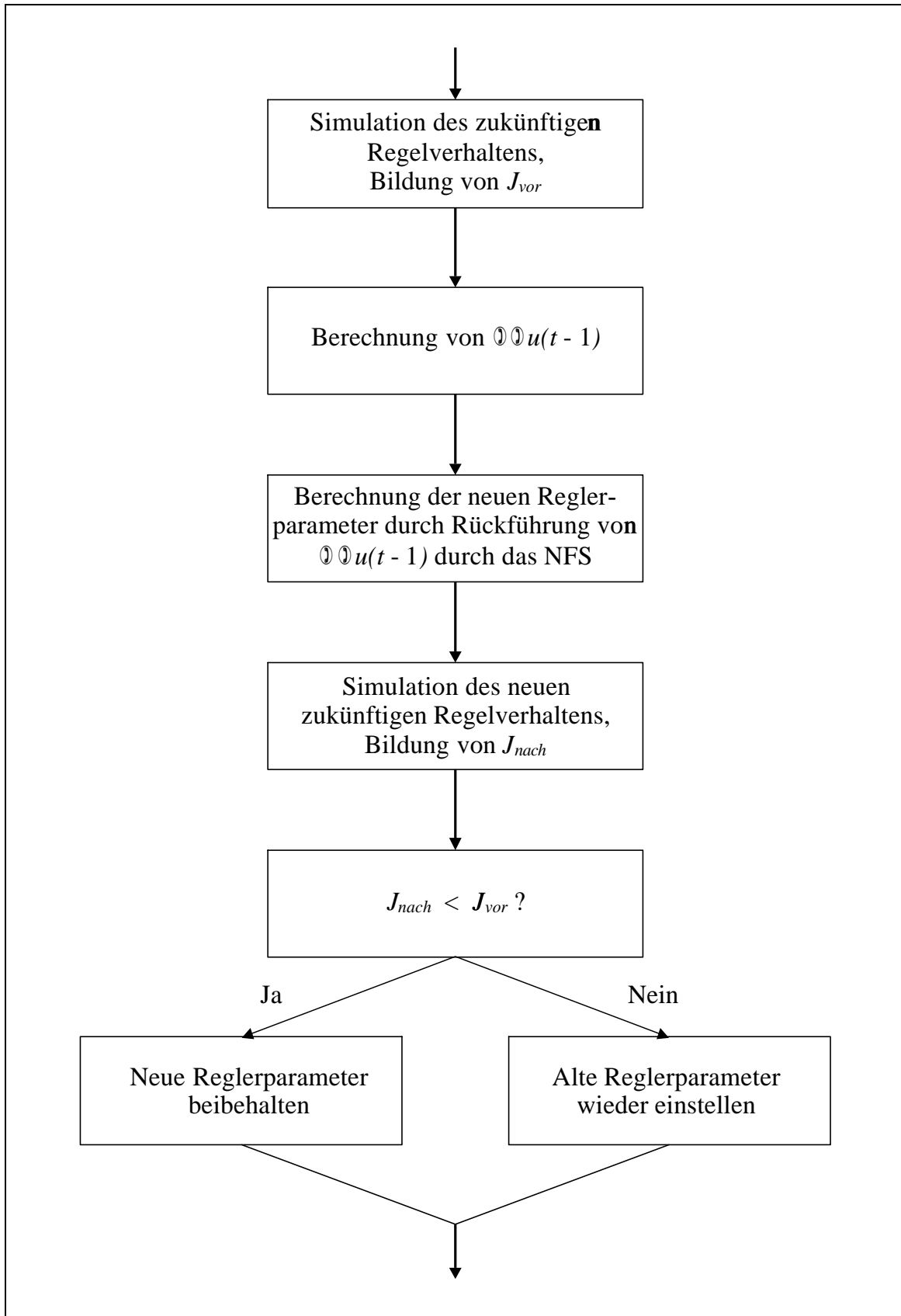


Abb. 6 Erweiterter Adaptionsalgorithmus, der nur Verbesserungen der Reglgüte zulässt

3.2 Implementierung am Reaktionskalorimeter

3.2.1 Grundentwurf eines Fuzzy-Logik-Reglers

Bereits beim Entwurf eines Fuzzy-Logik-Reglers kann ein hohes Maß an Information über den zu regelnden Prozeß berücksichtigt werden. Die turbulent gerührte Flüssigkeit (500 g Wasser) im Reaktionskalorimeter wird mittels einer Kompensationsheizung auf einer Temperatur T_r gehalten, die 10 K über der Temperatur des Kühlmediums liegt. Dazu ist eine Heizleistung von ca. 40 W nötig. Die maximale Heizleistung der Kompensationsheizung liegt bei 80 W. Die Reaktortemperatur soll sich bei konstanter Führungs- und Störgröße in einem Toleranzband von ± 0.010 K um den Sollwert $T_{r,soll}$ bewegen. Das Toleranzband wird so ausgewählt, daß bei einer Meßgenauigkeit von einem Millikelvin noch genügend große Temperaturabweichungen und -änderungen zur Verfügung stehen um einen Regeleinriff berechnen zu können. Bei Einwirken einer Störgröße soll möglichst keine Temperaturabweichung größer als ± 0.100 K auftreten.

Die Ein- und Ausgangsgrößen des Fuzzy-Logik-Reglers sind in Tabelle 2 zusammengestellt. Auf die Arbeitsweise des Reglers ist in Kapitel 3.1 eingegangen worden. Der mathematische Algorithmus wird in Anhang A beschrieben.

Tab. 2: Übersicht der Eingangs- und Ausgangsgrößen

	Symbol	Einheit	Linguistische Variable	Skalierungsfaktor
Eingangsgröße	x_w	[K]	<i>Temperaturabweichung</i>	c_{Tabw}
Eingangsgröße	$\Delta T_r / \Delta t$	[K s ⁻¹]	<i>Temperaturänderung</i>	$c_{Tänd}$
Ausgangsgröße	Δu	[W]	<i>Heizleistungsänderung</i>	$c_{Händ}$

Die Eingangsgrößen x_w und $\Delta T_r / \Delta t$ (Basisvariablen) werden auf die Terme (unscharfe Mengen) der linguistischen Variablen *Temperaturabweichung* und *Temperaturänderung* abgebildet. Auf diesen Termen basiert das Fuzzy-Logik-System des Reglers. Jede linguistische Variable wird mit 7 Termen beschrieben. Als Zugehörigkeitsfunktionen für die Terme (unscharfe Mengen) finden die dreieckförmige und die trapezförmige Funktion Ver-

wendung, wobei die Randterme mit seitlich offener Trapezfunktion beschrieben werden, um den gesamten Wertebereich der Eingangsgrößen abzudecken. Die Zugehörigkeitsfunktionen der einzelnen Terme werden so ausgelegt, daß sie sich überschneiden. Jedem Wert der Eingangsgröße werden aber Zugehörigkeitsgrade zu maximal zwei Termen gleichzeitig zugewiesen. Als UND-Operator für die Aggregation der zwei Teilbedingungen im Prämissenteil einer Regel wird der *MIN*-Operator (A.15) verwendet. Die ODER-Operation bei der Akkumulation aller Entschlossenheitsgrade, die zu ein und derselben Schlußfolgerungen führen, wird mit dem *MAX*-Operator (A.11) vorgenommen. Die Defuzzifizierung erfolgt mittels der Höhenmethode, so daß die Terme der Ausgangsgröße (*Heizleistungsänderung*) als Singletons dargestellt werden. Die Ausgangsmengen werden nur durch ihre Modalwerte $z_{m,ck}$ dargestellt, so daß auf Variationen des Reglerverhaltens durch Wahl von Form und Breite der Zugehörigkeitsfunktionen der Ausgangsterme verzichtet wird.

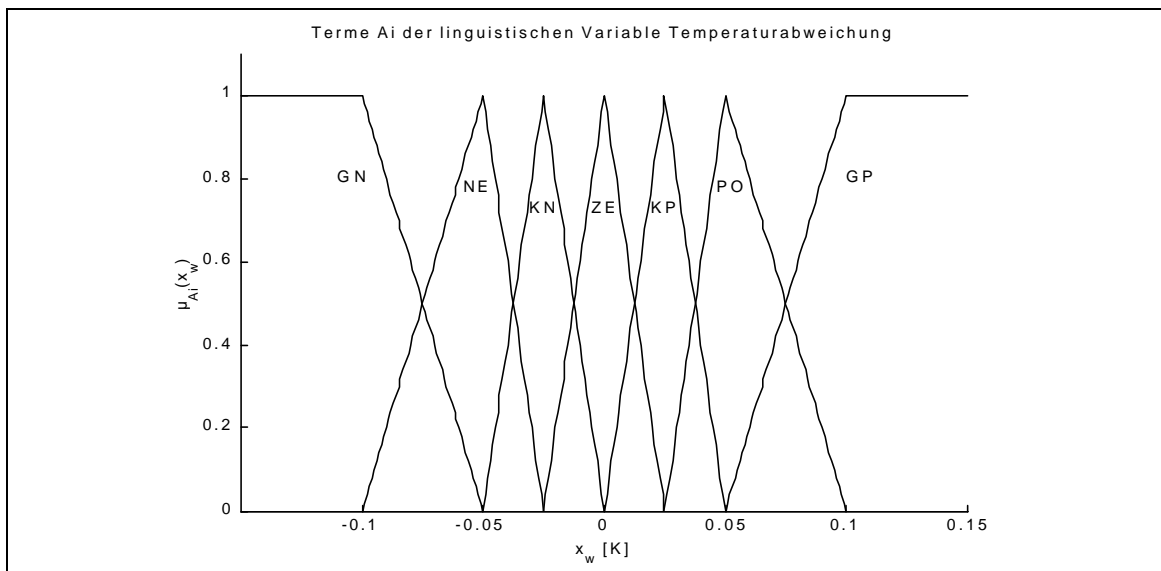
In Tabelle 2 ist jeder linguistischen Variable noch ein Skalierungsfaktor c zugeordnet, mit dem die Modalwerte x_m , bzw. Begrenzungswerte x_r und x_l und die Breiten a und b multipliziert werden. Über die Skalierungsfaktoren läßt sich das Übertragungsverhalten des Fuzzy-Logik-Reglers global verändern. Im hier beschriebenen ersten Entwurf des Reglers werden die Skalierungsfaktoren auf eins gesetzt.

Die Abtastperiode Δt_a des Reglers entspricht der in Kapitel 2.2.2 genannten Abtastzeit von 3 Sekunden. Die Regelperiode Δt_c (Kapitel 3.1.1) wird mit 6 Sekunden auf das Zweifache der Abtastzeit festgelegt.

Eine Anforderung, die an die Regelung der Reaktortemperatur gestellt wird, ist eine maximale Regelabweichung x_w von ± 0.100 K. Die unscharfe Menge für den Term GROSSPOSITIV (GP) wird daher so ausgelegt, daß er bei Regelabweichungen größer $+0.1$ K einen Zugehörigkeitsgrad von 1 aufweist. Eine weitere Unterteilung jenseits dieses Wertes scheint zunächst nicht sinnvoll. Die gleiche Argumentation gilt für den Term GROSSNEGATIV (GN) und Werten kleiner -0.1 K. Die übrigen Terme der linguistischen Variable *Temperaturabweichung* werden zwischen GROSSPOSITIV und GROSSNEGATIV angeordnet (Tabelle 3). Die Terme um den Nullpunkt der Temperaturabweichung sind etwas enger gewählt, um in diesem Bereich ein bessere Differenzierung des Regeleingriffs vornehmen zu können. In Abbildung 7 sind die Terme m_{A_i} , in die die *Temperaturabweichung* unterteilt ist, grafisch dargestellt.

Tab. 3:

Linguistische Variable: <i>Temperaturabweichung</i>						
	Term A_i	Abk.	a_{A_i}	b_{A_i}	x_{m,A_i} (bzw. x_{l,A_i})	x_{r,A_i}
A1	GROßNEGATIV	GN	∞	0,050	$-\infty$	-0,100
A2	NEGATIV	NE	0,050	0,025	-0,050	--
A3	KLEINNEGATIV	KN	0,025	0,025	-0,025	--
A4	NULL	ZE	0,025	0,025	0,000	--
A5	KLEINPOSITIV	KP	0,025	0,025	0,025	--
A6	POSITIV	PO	0,025	0,050	0,050	--
A7	GROßPOSITIV	GP	0,050	∞	0,100	∞

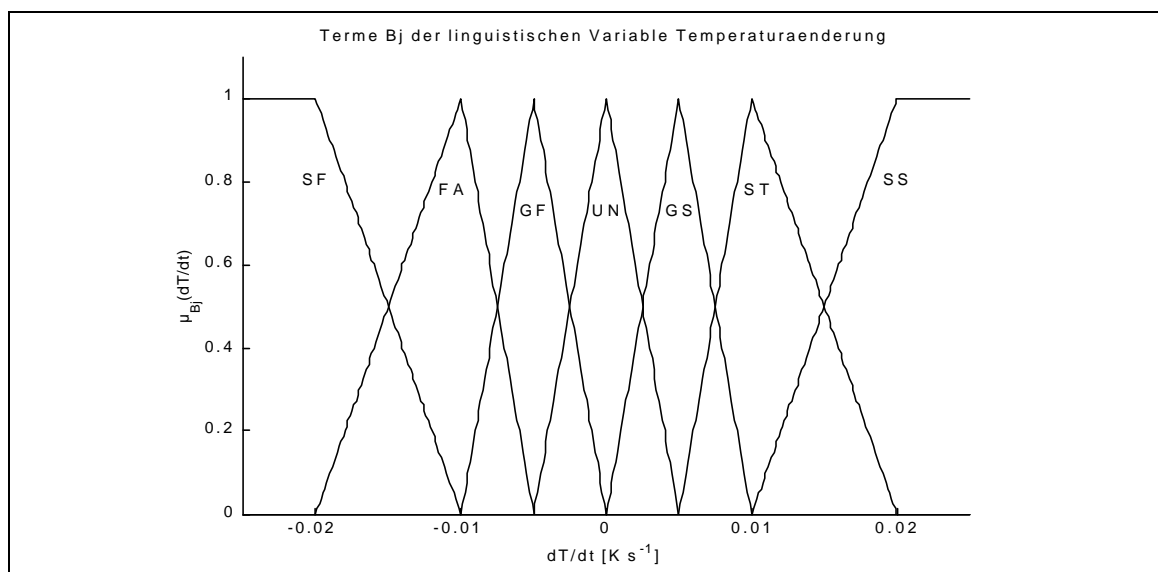
Abb. 7 Auftragung der Terme A_i (linguistische Variable: *Temperaturabweichung*)

Bei einer Reaktorbefüllung mit 500 g Wasser (spezifische Wärmekapazität $4.18 \text{ J g}^{-1} \text{ K}^{-1}$ bei 55 °C) beträgt die Gesamtwärmekapazität $C_p = 2090 \text{ J K}^{-1}$. Da für die Aufrechterhaltung der Temperaturdifferenz zwischen Reaktor und Kühlmantel bereits eine Heizleistung von ca. 40 W benötigt wird, können bis zur Leistungsgrenze von 80 W noch maximal 40 W zugeschaltet werden. Rechnet man die zusätzlichen 40 Watt in eine zeitliche Temperaturänderung um, so ergibt sich ein Wert von 0.019 K s^{-1} (vereinfachte Berechnung unter adiabaten Bedingungen). Der Term STARKSTEIGEND (SS) der linguistischen Variable *Temperaturänderung* wird so gewählt, daß er ab Werten größer 0.020 K s^{-1} Zugehörigkeitsgrade von 1 aufweist. Eine entsprechende Überlegung gilt für die Definition des Terms

STARKFALLEND (SF), der sich aus der zeitlichen Temperaturänderung beim Ausschalten der Heizleistung ergibt. In Tabelle 4 sind die Modalwerte und Breiten der übrigen Terme B_j aufgeführt. Eine grafische Darstellung der Zugehörigkeitsfunktionen m_{B_j} ist in Abbildung 8 zu sehen.

Tab. 4:

Linguistische Variable: <i>Temperaturänderung</i>						
Term B_j		Abk.	a_{B_j}	b_{B_j}	y_{m,B_j} (bzw. y_{l,B_j})	y_{r,B_j}
B_1	STARKFALLEND	SF	∞	0,010	$-\infty$	-0,020
B_2	FALLEND	FA	0.010	0.005	-0.010	--
B_3	GERINGFALLEND	GF	0.005	0.005	-0.005	--
B_4	UNVERÄNDERT	UN	0.005	0.005	0.000	--
B_5	GERINGSTEIGEND	GS	0.005	0.005	0.005	--
B_6	STEIGEND	ST	0.005	0.010	0.010	--
B_7	STARKSTEIGEND	SS	0.010	∞	0.020	∞

Abb. 8 Auftragung der Terme B_j (linguistische Variable: *Temperaturänderung*)

In Tabelle 5 sind die Terme der linguistischen Variable *Heizleistungsänderung* aufgeführt. Der Term ERHÖHESTARK (ES) wird mit 10 W auf ein Viertel der verfügbaren zusätzlichen Heizleistung gesetzt. Die Terme ERHÖHE (ER) und ERHÖHEGERING (EG) sind bei 4 W und 1.6 W lokalisiert. Hinter dieser ungleichmäßigen Verteilung der Singeltons über dem Wertebereich der Ausgangsgröße Δu steht die Idee, daß die großen Schlußfolgerungen ERHÖHESTARK (ES) oder VERRINGERESTARK (VS) in Fällen

eingesetzt werden, wo ein starker Regeleingriff nötig ist. In Bereichen geringerer *Temperaturabweichungen* und geringer *Temperaturänderungen* werden die Terme ERHÖHEGERING (EG) bzw. VERRINGEREGERING (VG) eingesetzt, die einen sehr sachten Regeleingriff zur Folge haben. Da im folgenden Kapitel 3.2.2 eine Anpassung des globalen Übertragungsverhaltens des Fuzzy-Logik-Reglers über die Skalierungsfaktoren beschrieben wird, kommt es bei den Positionen der Singletons weniger auf die absoluten Werte an, als vielmehr auf die relative Lage der Terme zueinander. Zur besseren Übersicht sind die Terme der *Heizleistungsänderung* in Abbildung 9 dargestellt.

Tab. 5:

Linguistische Variable: <i>Heizleistungsänderung</i>			
	Term C_k	Abk.	z_m
$C1$	VERRINGERE STARK	VS	-10.0
$C2$	VERRINGERE	VE	-4.00
$C3$	VERRINGEREGERING	VG	-1.60
$C4$	KEINE VERÄNDERUNG	KV	± 0.00
$C5$	ERHÖHEGERING	EG	+1.60
$C6$	ERHÖHE	ER	+4.00
$C7$	ERHÖHE STARK	ES	+10.0

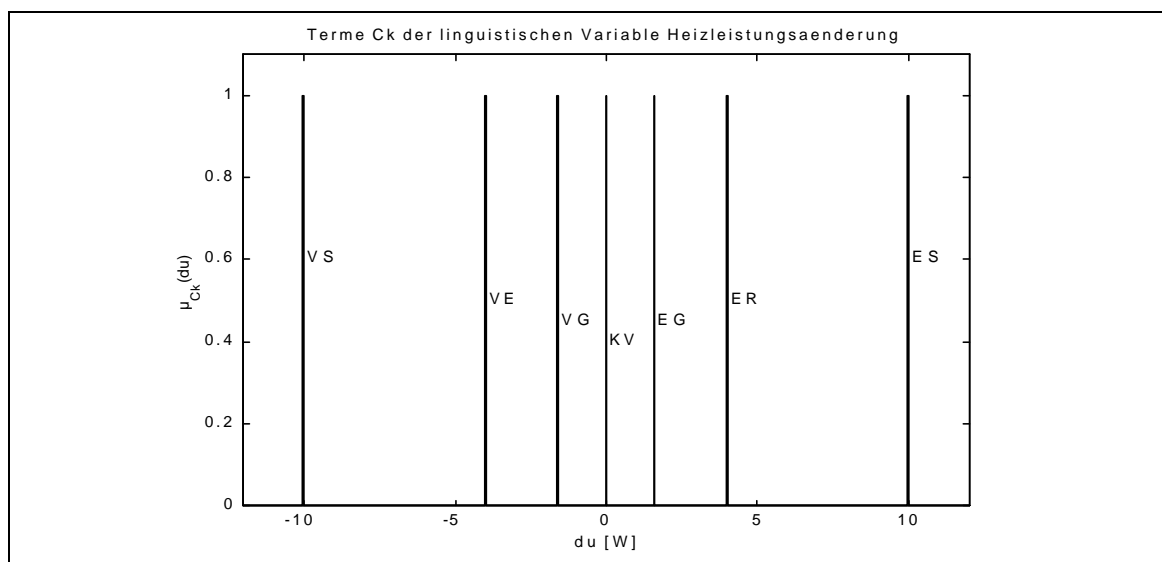


Abb. 9: Auftragung der Terme C_k (linguistische Variable: *Heizleistungsänderung*)

In der Regelbasis werden die Prämissen (WENN-Teil) mit den Konklusionen (DANN-Teil) verknüpft. Da die Prämisse jeder Regel aus zwei Teilbedingungen zusammengesetzt ist, die Konklusion aber nur eine Schlußfolgerung enthält, kann die Regelbasis in Form einer Matrix dargestellt werden. Prinzipiell ist jede Regelbasis zulässig, die eine vernünftige Strategie zur Regelung des Prozesses enthält. Die Regelbasismatrix in Tab. 6 orientiert sich an den Vorschlägen in [18] und [32].

Tab. 6: Matrix der vollständigen und konsistenten Regelbasis

Heizleistungs- änderung		Temperaturabweichung						
		GN	NE	KN	ZE	KP	PO	GP
Temperaturänderung	SF	ES	ES	ES	ES	ES	ER	KV
	FA	ES	ES	ES	ES	ER	KV	VG
	GF	ES	ER	ER	EG	KV	VG	VE
	UN	ES	ER	EG	KV	VG	VE	VS
	GS	ER	EG	KV	VG	VE	VE	VS
	ST	EG	KV	VE	VS	VS	VS	VS
	SS	KV	VE	VS	VS	VS	VS	VS

Die Einträge in die Matrix sind folgendermaßen zu lesen: Wenn die *Temperaturabweichung* NEGATIV (NE, Spalte 2) ist und die *Temperatur* FALLEND (FA, Zeile 2) ist, dann ERHÖHE die *Heizleistung* STARK (ES). Die Regelabweichung x_w ist also negativ und nimmt weiter ab, so daß mit einer großen positiven Heizleistungsänderung darauf reagiert werden muß.

Eine andere Situation ergibt sich beispielsweise, wenn die *Temperaturabweichung* nur KLEINNEGATIV (KN, Spalte 3) ist und die *Temperatur* STEIGT (ST, Zeile 6). Dann sollte eine VERRINGERUNG (VE) der *Heizleistung* vorgenommen werden, damit die Temperatur T_r nach Möglichkeit nicht über den Sollwert $T_{r,soll}$ hinausschießt, sondern sachte an ihn herangeführt wird.

Die Erprobung des entworfenen Fuzzy-Logik-Reglers findet mit einer Reaktorbefüllung von 500 g Wasser bei einem Sollwert der Reaktortemperatur $T_{r,soll} = 55^\circ\text{C}$ statt. Die Temperatur des Kühlmantels wird auf $T_j = 45^\circ\text{C}$ geregelt, die gleiche Temperatur besitzt das umgebende Luftbad. Die Rührerdrehzahl beträgt $N = 250 \text{ min}^{-1}$. Der Regler stellt die gewünschte

Reaktortemperatur ein, danach wird die Meßdatenerfassung gestartet. Nach 1000 Sekunden wird über die Kalibrierheizung eine Heizleistung von 20 W in den Reaktorraum eingebracht, nach weiteren 1000 Sekunden wird die Kalibrierheizleistung wieder ausgeschaltet. Der sich ergebende Reaktortemperaturverlauf ist in Abbildung 10 wiedergegeben.

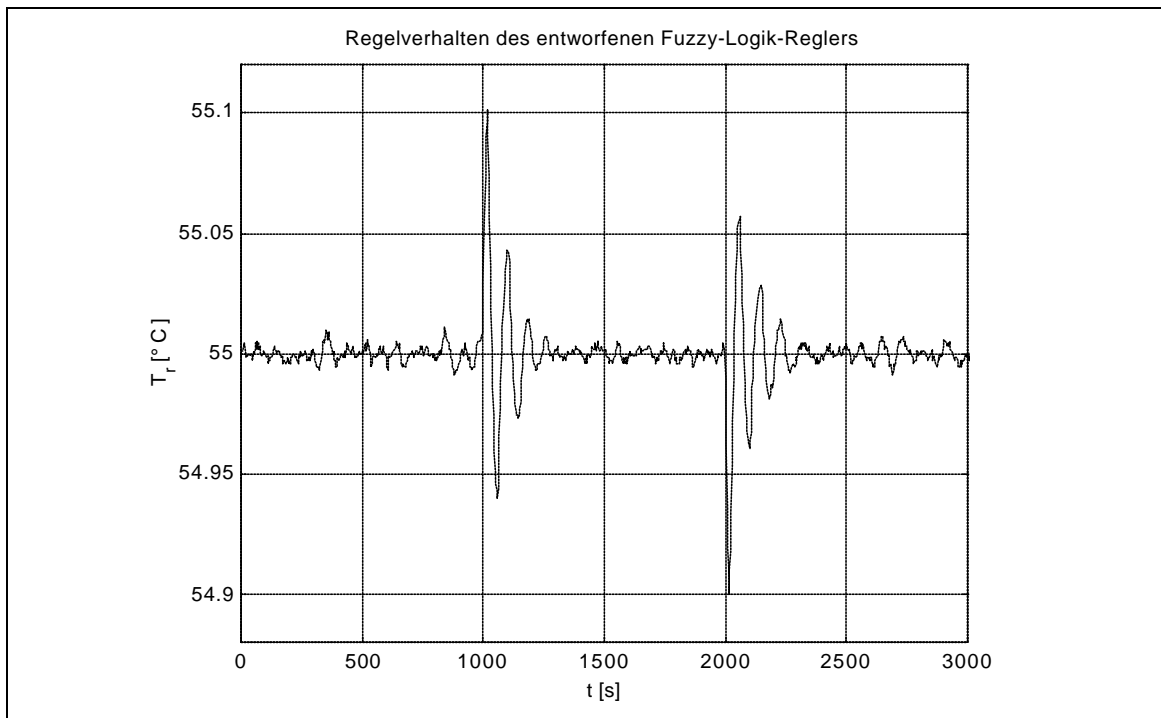


Abb. 10 Temperaturverlauf bei Aufgabe einer rechteckförmigen Kalibrierheizleistung q_{cal} und Regelung durch den entworfenen Fuzzy-Logik-Regler

Es ist deutlich zu erkennen, daß noch kein zufriedenstellendes Regelverhalten erzielt wird. So wird zwar im ungestörten Betrieb ein Toleranzband von ± 0.010 K um den Temperatursollwert $T_{r,soll}$ eingehalten, und es wird die gewünschte maximale Temperaturabweichung von ± 0.100 K beim Ein- und Ausschalten der Kalibrierheizung eingehalten. Aber nach Eintritt der Störungen zeigt der Temperaturverlauf mehrere Überschwinger, wodurch sich eine lange Ausregelzeit bis zum endgültigen Wiedereintritt in den Toleranzbereich ergibt.

Es ist allerdings bezeichnend für das vorgestellte Konzept der Regelung auf Basis der unscharfen Logik, daß allein aufgrund recht grober Abschätzungen und Überschlagsrechnungen ein Regler entworfen worden ist, der zu einem stabilen Verhalten des Regelkreises führt.

3.2.2 Optimierung der Skalierungsfaktoren des Fuzzy-Logik-Reglers mittels eines Genetischen Algorithmus

Eine erste Optimierung des in Kapitel 3.2.1 vorgestellten Fuzzy-Logik-Reglers wird anhand der Skalierungsfaktoren der linguistischen Variablen c_{Tabw} , $c_{Tänd}$ und $c_{Händ}$ vorgenommen (Tab. 2). Die Regelung der Reaktortemperatur T_r wird besonders beansprucht bei Beginn und Ende der Dosierung einer Reaktionskomponente, deren Temperatur T_{dos} nicht genau der Reaktortemperatur entspricht, oder beim Start einer Reaktion, wenn aufgrund der hohen Reaktandenkonzentration im Reaktor eine hohe Reaktionswärme q_r freigesetzt wird. Beide Störungsarten lassen sich als einen sprunghaftigen Verlauf der Wärmestromkurve q_r oder q_{dos} auffassen. Ein solcher zeitlicher Verlauf wird durch eine Rechteckfunktion der Kalibrierheizleistung q_{cal} simuliert (Abb. 11). Ein weiterer Vorteil der Rechteckfunktion ist die frequenzreiche Anregung des Regelkreises, so daß instabiles Regelverhalten bei ungünstigen Skalierungsfaktoren erkennbar wird.

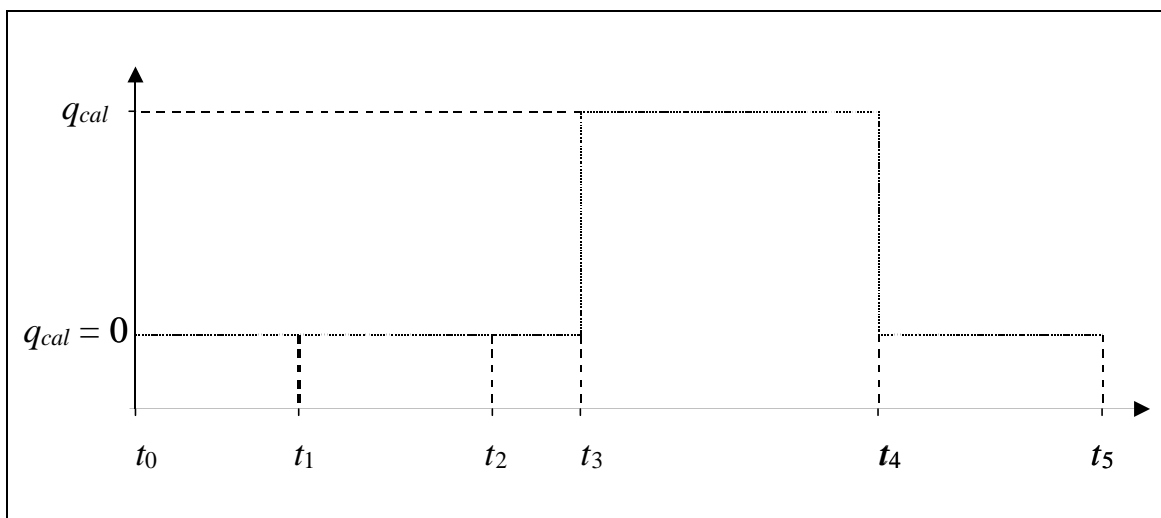


Abb. 11 Zeitlicher Verlauf der Kalibrierheizleistung q_{cal}

Die in den Reaktionsraum eingebrachte Kalibrierheizleistung bleibt zunächst in der Zeit von t_0 bis t_3 ausgeschaltet. Vom Zeitpunkt t_3 bis zum Zeitpunkt t_4 wird sie sprunghaft auf die Heizleistung q_{cal} gesetzt. Anschließend bleibt sie von t_4 bis t_5 wieder ausgeschaltet.

Für die Bewertung der Regelgüte wird nicht der gesamte Zeitraum von t_0 bis t_5 betrachtet, sondern nur das Regelverhalten ab t_2 . Um für die Bewertung unterschiedlicher Reglerparametersätze (bestehend aus c_{Tabw} , $c_{Tänd}$ und $c_{Händ}$) die gleichen Anfangsbedingungen zu schaffen, wird in der Zeit von t_0 bis t_1 die Reaktortemperatur T_r auf ihren Sollwert $T_{r,soll}$ zurückgeführt. Dies ist

gewährleistet, wenn c_{Tabw} recht groß, $c_{Tänd}$ und $c_{Händ}$ recht klein gewählt werden. Ab dem Zeitpunkt t_1 werden die zu bewertenden Parameter beim Fuzzy-Logik-Regler eingestellt. Bei ungünstigen Reglerparameterkombinationen kommt es bereits ohne äußere Anregung zur Ausbildung einer Dauerschwingung in T_r .

Bewertet wird das Regelverhalten nur in dem Zeitraum von t_2 bis t_5 . Eine Beurteilung der Regelgüte erfolgt anhand des Gütekriteriums J (50), das in Kapitel 3.1.3 angesprochen wird.

$$J = \int_{t_2}^{t_5} \left[x_w(t)^2 + \frac{1}{k} \Delta u(t)^2 \right] dt \quad (50)$$

Der Faktor $1/k$ macht einen Kompromiß zwischen schneller Ausregelung der Störgröße und akzeptablen Stellgrößenänderungen möglich. Darüber hinaus berücksichtigt er die unterschiedlichen Größenordnungen, in denen sich die Regelabweichung x_w (10^{-3} bis 10^{-1} K) und die Stellgrößenänderung Δu (10^{-1} bis 10^1 W) bewegen.

In Vorversuchen hatte sich gezeigt, daß die Integralkriterien *IAE* (*integral of absolute error*) und *ISE* (33) nicht verwendet werden können. Bei Verwendung des *IAE* als Gütekriterium für eine Parameteroptimierung werden Parametersätze als gut klassifiziert, deren Temperatur- und Stellgrößenverläufe im stationären Betrieb (ohne Störgrößeneinfluß) Dauerschwingungen mit hoher Frequenz und geringer Amplitude aufweisen. Beim Ein- und Ausschalten der Kalibrierheizleistung q_{cal} ist ein Regler mit diesen Regelparametern dann in der Lage, auf eine Temperaturabweichung schnell und mit großen Stellgrößenänderungen zu reagieren. Auf diese Weise treten keine großen Abweichungen vom Temperatursollwert auf, was insgesamt zu einem kleineren als gut beurteilten *IAE* führt. Noch weniger geeignet ist der *ISE* (33) allein, da bei diesem Gütekriterium große Temperaturabweichungen wegen des Quadrats stärker in den Wert des *ISE* eingehen. Um die Dauerschwingungen um den Sollwert zu sanktionieren, die mit permanenten Stellgrößenänderungen verbunden sind, ist in J (50) neben dem *ISE* der zweite Term $(\Delta u(t)^2)$ enthalten.

Als Optimierverfahren wird ein genetischer Algorithmus verwendet. Dieses Verfahren benötigt lediglich die Parametersätze (c_{Tabw} , $c_{Tänd}$, $c_{Händ}$) und die dazugehörigen Gütekriterien J , um Parametersätze mit besserer Güte ausfindig zu machen. Weitere Annahmen über den funktionalen Zusam-

menhang zwischen J und dem Parametersatz sind nicht nötig, was eine recht einfache Implementierung ermöglicht. Ein genetischer Algorithmus wird in [29] zur Reglerparameteroptimierung bei einem klassischen PID-Regler herangezogen. Seine Funktionsweise ist in Anhang D näher erläutert.

Ein genetischer Algorithmus ist ein Maximierungsverfahren. Bei der Parameteroptimierung wird aber eine Minimierung des Gütekriteriums J angestrebt. Da J nur positive Werte annehmen kann, wird die Minimierungsaufgabe durch Anwendung des Kehrwertes (51) in eine Maximierungsaufgabe überführt.

$$f_i = \frac{1}{J_i} \quad (51)$$

Der Algorithmus arbeitet mit einer Populationsgröße von 25 Parametersätzen. Die anfänglichen Parameter c_{Tabw} , $c_{Tänd}$ und $c_{Händ}$ werden im ganzen Parameterraum zufällig verteilt. Als Genauigkeit für jeden Parameter werden zwei Stellen nach dem Komma als ausreichend erachtet. Auf diese Genauigkeit werden die Parameter der Startpopulation gerundet. Bei Anwendung des Mutationoperators wird darauf geachtet, daß diese Genauigkeit beibehalten bleibt.

Für jeden Parametersatz i wird anhand des oben gezeigten zeitlichen Störgrößenverlaufs das Gütekriterium J_i bestimmt. Um für einen beurteilten Parametersatz nicht ein zweites Mal das Gütekriterium J_i zu bestimmen, wird der Parametersatz mit zugehörigem J_i in einer Tabelle abgelegt. Als Abbruchbedingung wird eine Meßzeit von ca. 24 Stunden gewählt.

Als Reproduktionsoperation wird das Rouletterad-Verfahren angewendet. Dabei wird das Prinzip der Elitisierung berücksichtigt (d.h. der Parametersatz mit der besten Güte wird auf jeden Fall in die Population der nachfolgenden Generation übertragen). Die Rekombinationsoperation wird mit dem 1-Punkt-Kreuzungsoperator ausgeführt, der einmal zwischen den Parametern c_{Tabw} und $c_{Tänd}$ und einmal zwischen den Parametern $c_{Tänd}$ und $c_{Händ}$ wirkt. Jeder Parametersatz nimmt so mit einer Wahrscheinlichkeit von 16.7 Prozent an der Rekombinationsoperation teil. Ein auf diese Weise neu erzeugter Parametersatz wird nicht vor einem zweiten Zugriff des Kreuzungsoperators oder dem Mutationsoperator geschützt.

Die Mutationsoperation bedient sich eines Mutationsoperators, der zu drei zufällig ausgewählten Parametern c_{Tabw} , $c_{Tänd}$ und $c_{Händ}$ bis zu ± 10 Prozent

des aktuellen Parameterwertes addiert. Durch die Beschränkung auf ± 10 Prozent werden hauptsächlich Parametersätze in der Nähe eines aktuellen Parametersatzes erzeugt. Die Wahrscheinlichkeit, daß ein einzelner Parameter durch die Mutationsoperation verändert wird, beträgt 4.2 Prozent.

Der Reaktor wird mit 500 g Wasser befüllt. Verwendet wird der Ankerührer mit aufgesetztem Schrägblattrührer, der mit einer Drehzahl von $N = 200 \text{ min}^{-1}$ betrieben wird. Der Sollwert der Reaktortemperatur wird auf $T_{r,soll} = 55 \text{ °C}$ gesetzt, die Kühlmantel- und die Luftbadtemperatur werden auf $T_j = 45 \text{ °C}$ geregelt. Die von der Kalibrierheizleistung ab dem Zeitpunkt t_3 abgegebene Leistung beträgt $q_{cal} = 20 \text{ W}$. Die Zeiten des Störgrößenprogramms (s. Abb. 11) sind in Tabelle 7 aufgeführt.

Tab. 7: Zeiten des Störgrößenverlaufs von q_{cal} bei Reaktorbefüllung mit Wasser

t_1 [s]	t_2 [s]	t_3 [s]	t_4 [s]	t_5 [s]
120	240	300	540	720

Der anfängliche Wertebereich, in welchem sich die Skalierungsfaktoren bewegen, wird in Tabelle 8 dargestellt. Dabei wird bewußt ein recht großer Wertebereich zugelassen, um die Parametersuche nicht auf einen möglicherweise ungünstigen Suchbereich zu beschränken. Es sei angemerkt, daß Werte von 3.00 für den Skalierungsfaktor der *Temperaturabweichung* c_{Tabw} eigentlich viel zu groß bemessen sind. Berücksichtigt man die Grundeinstellungen des Fuzzy-Logik-Reglers (Kap. 3.2.1), so müßten Abweichungen von über $\pm 0.150 \text{ K}$ auftreten, damit die äußersten Terme GROßPOSITIV oder GROßNEGATIV einen Zugehörigkeitsgrad > 0 aufweisen. Dies widerspricht aber den Anforderungen, die an die Regelung gestellt werden.

Tab. 8: Anfänglicher Wertebereich der Skalierungsfaktoren

Skalierungsfaktor	Wertebereich	
	von	bis
c_{Tabw}	0.30	3.00
$c_{Tänd}$	0.10	1.00
$c_{Händ}$	0.60	3.60

Es wird je ein Versuch für $k=3 \cdot 10^3$ und für $k=1 \cdot 10^4$ durchgeführt. Aus der abschließenden Beurteilung des Regelverhaltens wird ersichtlich, daß eine hohe Dämpfung des Regelgrößenverlaufs ($k=3 \cdot 10^3$) nicht nötig ist. Im folgenden werden daher die Ergebnisse des Versuchs für $k=1 \cdot 10^4$ diskutiert. In Abbildung 12 ist die Verteilung der 25 Parametersätze der Startpopulation dargestellt. Da jeder Parametersatz aus den drei Parametern c_{Tabw} , $c_{Tänd}$ und $c_{Händ}$ besteht, ist die dreidimensionale Darstellung (rechts oben) um die Projektionen auf die Grundebenen ergänzt. Deutlich erkennbar ist die anfängliche Gleichverteilung der Parametertripel im Parameterraum.

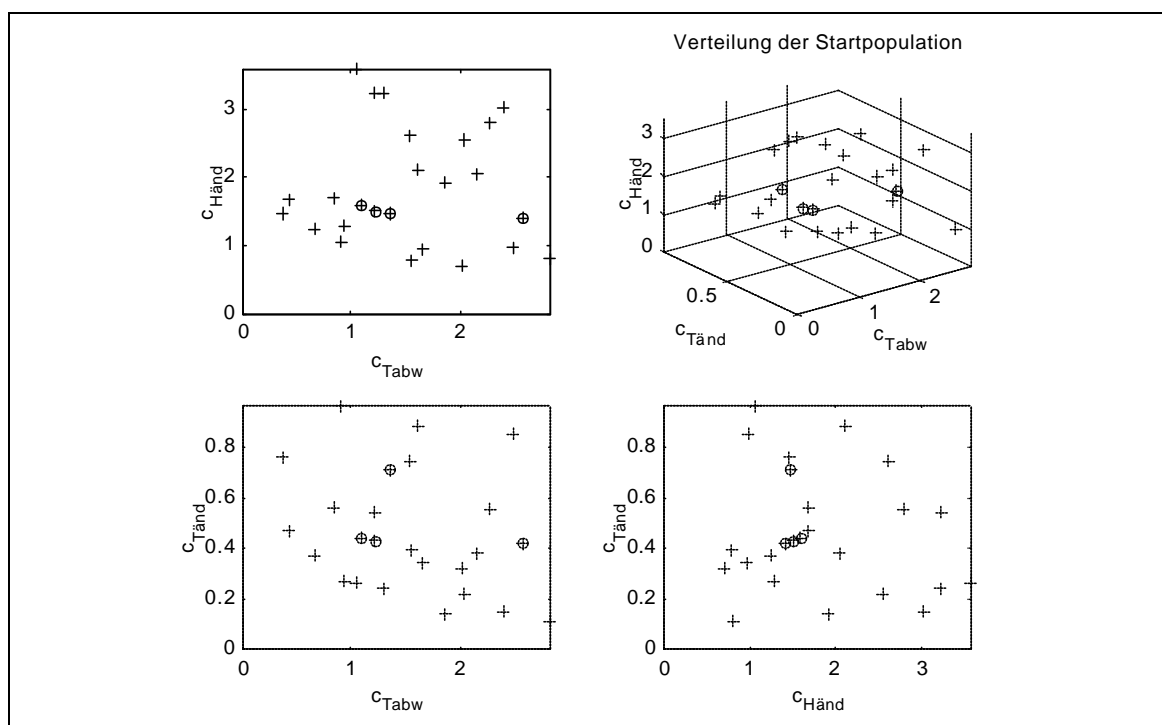


Abb. 12 Parametersätze der Startpopulation bei Wasser

Die vier Individuen mit besonders hoher Güte sind in Abb. 12 durch einen zusätzlichen Kreis kenntlich gemacht. Bereits in der Startpopulation zeigt sich, daß Parametersätze als gut beurteilt werden, wenn c_{Tabw} im Bereich von 1.1 bis 1.5 liegt, wenn $c_{Tänd}$ um 0.4 beträgt und wenn $c_{Händ}$ im Bereich von 1.3 bis 1.7 lokalisiert ist.

Wie aus der Auftragung aller 116 beurteilten Parametersätze (Abb. 13) ersichtlich, wird besonders dieser Bereich vom genetischen Algorithmus nach möglicherweise besseren Parametersätzen abgesucht. Allerdings orientiert

sich der genetische Algorithmus nicht nur an den besten Parametersätzen, sondern erzeugt auch neue in nicht von der Startpopulation besetzten Bereichen des Parameterraumes.

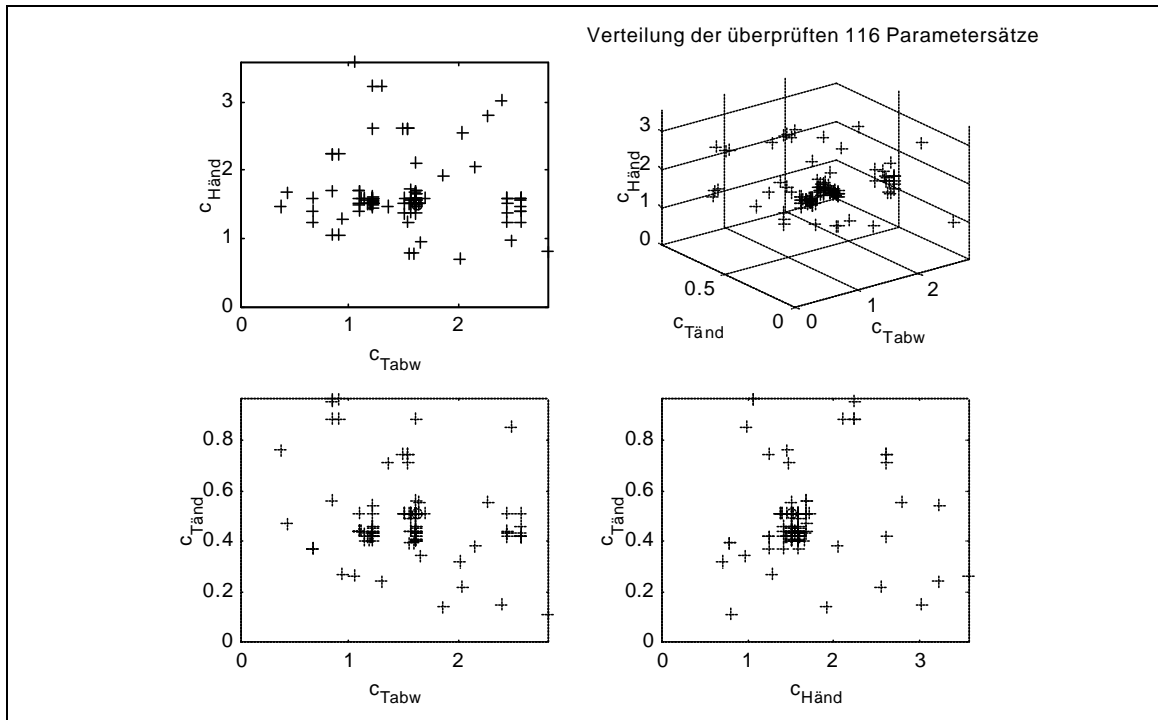


Abb. 13 Alle beurteilten Parametersätze bei Wasser

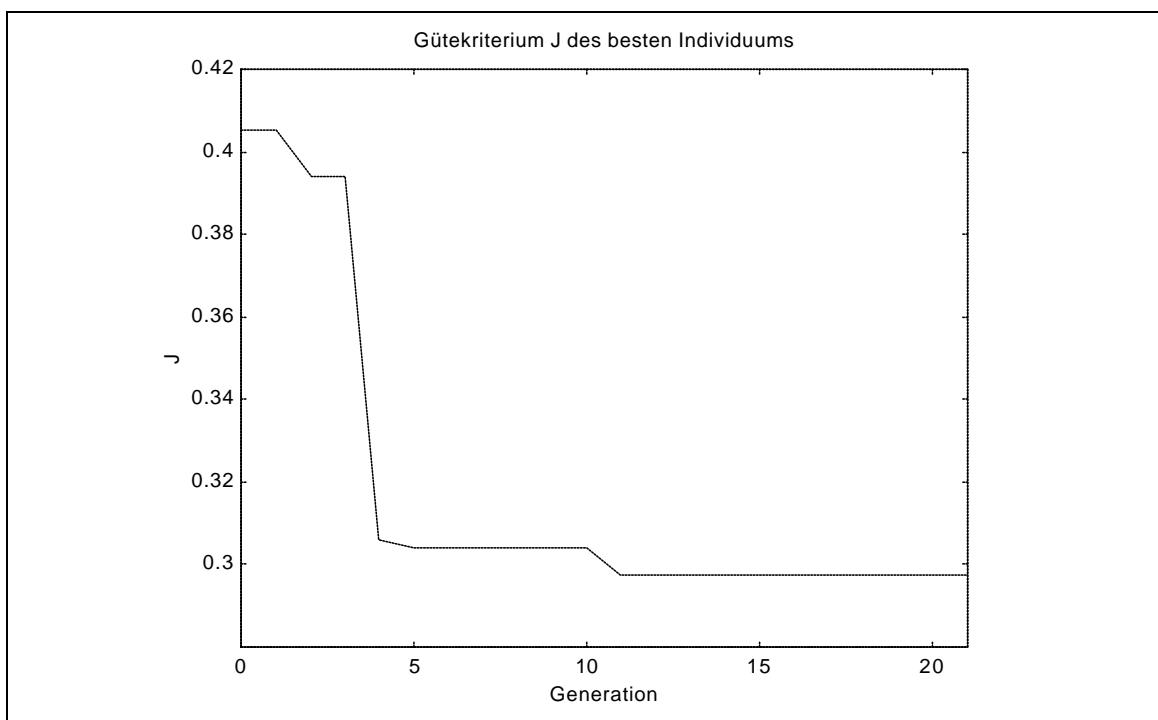


Abb. 14 Gütekriterium J des besten Parametersatzes als Funktion der Generationen

Der beste Parametersatz mit den Werten $c_{Tabw} = 1.61$, $c_{Tänd} = 0.51$ und $c_{Händ} = 1.51$ ist in Abb. 13 durch einen zusätzlichen Kreis gekennzeichnet. Aus der Auftragung der Güte J des jeweils besten Strings einer Generation in Abbildung 14 ist zu erkennen, daß die sich die Güte ab der 11. Generation nicht weiter verbessert hat. Dies ist der Grund für die Häufung der überprüften Parametersätze im Bereich um den optimalen Parametersatz.

Wie Abbildung 14 entnommen werden kann, wird das Gütekriterium J von einem Startwert von 0.405 im Laufe der Generationen auf den endgültigen Wert von 0.297 verbessert. Betrachtet man die optimalen Werte der Skalierungsfaktoren, so wird deutlich, daß der optimierte Regler große *Temperaturabweichungen* eher toleriert ($c_{Tabw} > 1$), jedoch nur weniger große *Temperaturänderungen* zuläßt ($c_{Tänd} < 1$) und insgesamt mit größeren *Heizleistungsänderungen* reagiert ($c_{Händ} > 1$).

Um die Einsetzbarkeit der Regleroptimierung an einem weiteren Beispiel zu überprüfen, werden als Reaktorbefüllung 500 g 6%ige CMC-Lösung verwendet (sonstige Betriebsbedingungen wie bei Wasser als Versuchsflüssigkeit). Aufgrund der höheren Viskosität der CMC-Lösung werden an die Regelung höhere Anforderungen gestellt. Wesentliche Herausforderung ist die längere Totzeit t_D der Regelstrecke aufgrund der veränderten Durchmischungsverhältnisse.

Die in die Flüssigkeit eingebrachte Kalibrierheizleistung q_{cal} beträgt 10 W, die Zeiten des Störgrößenprogramms sind in Tabelle 9 eingetragen. Die Zeitspanne von t_0 bis t_1 , in der die Reaktortemperatur T_r auf ihren Sollwert zurückgeführt wird, ist etwas verlängert worden. Zum einen muß der Regler in dieser „Beruhigungszeit“ durch einen sehr sanften Regeleingriff den Angleich an den Sollwert vornehmen, zum anderen treten u.U. aus dem vorangehenden Meßlauf große Temperaturabweichungen auf, wenn sich der überprüfte Parametersatz als sehr schlecht herausgestellt hat (s. Abb. 11).

Tab. 9: Zeiten des Störgrößenverlaufs von q_{cal} bei Reaktorbefüllung mit 6%iger CMC-Lösung

t_1	t_2	t_3	t_4	t_5
[s]	[s]	[s]	[s]	[s]
300	540	600	840	1020

Der anfängliche Wertebereich, aus dem die Startpopulation der Parametersätze gebildet wird, ist in Tabelle 10 aufgeführt. Wegen der höheren Totzeit der Regelstrecke wird für den optimalen Parametersatz ein kleiner Wert für $c_{Händ}$ erwartet, weshalb der Wertebereich für diesen Parameter kleiner gewählt wird. Eine höhere Reglerverstärkung (großes $c_{Händ}$) würde die Schwingungsneigung des Regelkreises erhöhen.

Tab. 10: Anfänglicher Wertebereich der Skalierungsfaktoren

Skalierungsfaktor	Wertebereich	
	von	bis
c_{Tabw}	0.30	3.00
$c_{Tänd}$	0.10	1.00
$c_{Händ}$	0.20	2.00

Es werden Versuche mit $k=3 \cdot 10^3$ und mit $k=1 \cdot 10^4$ durchgeführt. Die abschließende Beurteilung des Regelverhaltens weist die bessere Parameterkombination für $k=3 \cdot 10^3$ (höhere Dämpfung) aus, weshalb die Ergebnisse zu diesem Versuch im folgenden dargestellt sind.

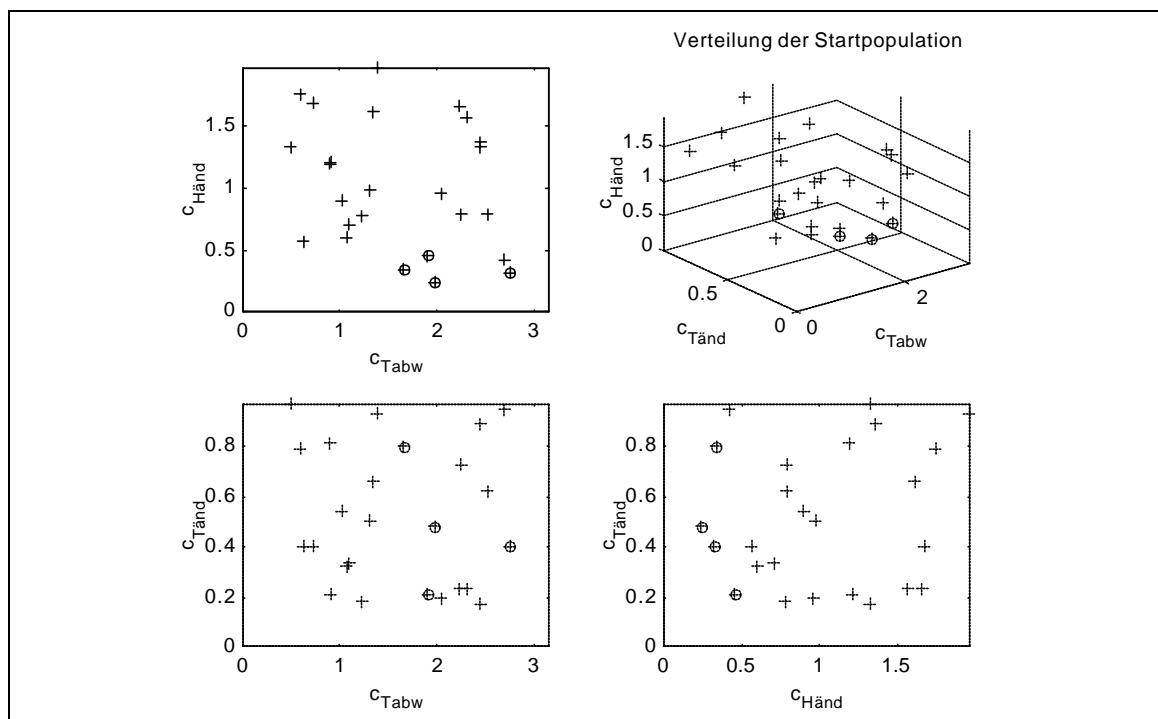


Abb. 15 Parametersätze der Startpopulation bei 6%iger CMC-Lösung

In Abbildung 15 sind die Parametersätze der Startpopulation eingetragen. Die besten vier Parametersätze sind wieder mit einem Kreis gekennzeichnet. Nur der Skalierungsfaktor für die Heisleistungsänderung $c_{Händ}$ der besten Parametersätze bewegt sich in einer engen Bandbreite von 0.20 bis 0.50 (kleine Reglerverstärkung).

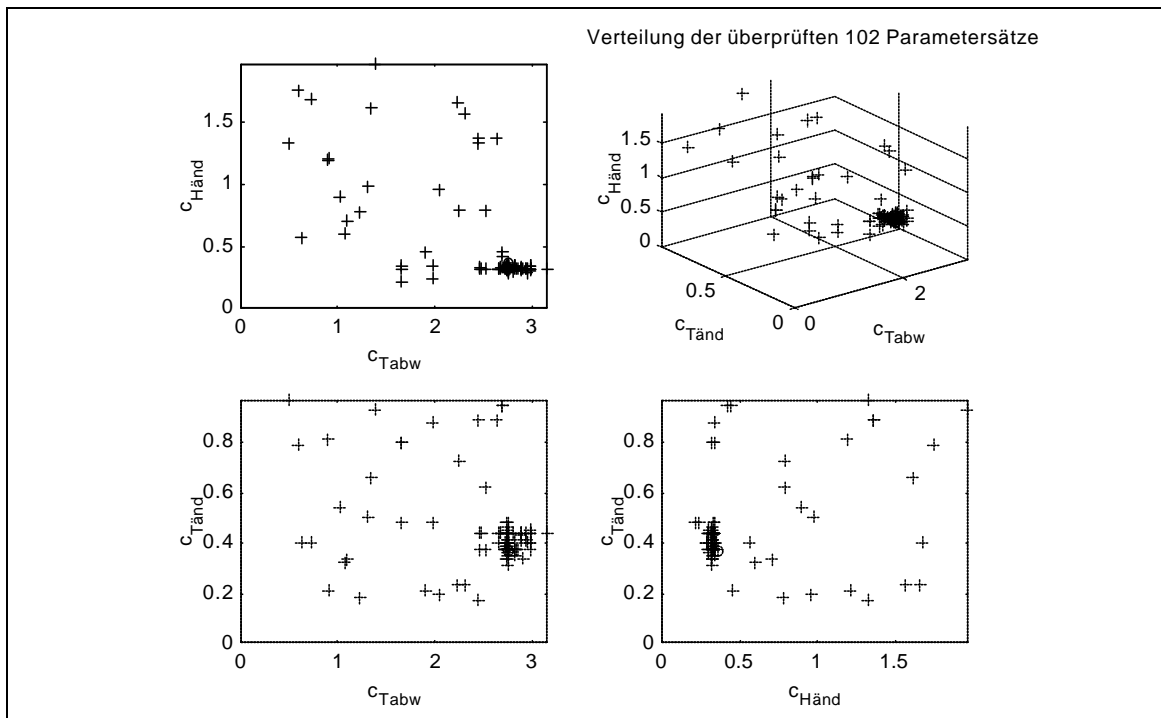


Abb. 16 Alle beurteilten Parametersätze bei 6%iger CMC-Lösung

Bei Betrachtung der Verteilung aller 102 beurteilten Parametersätze in Abbildung 16 wird deutlich, daß der genetische Algorithmus sich rasch auf einen Bereich um den besten Parametersatz der Startpopulation konzentriert. Erklärt werden kann dieses Verhalten, wenn das Gütekriterium J der besten Parametersätze der Startpopulation berücksichtigt wird. Der beste Parametersatz der Startpopulation besitzt ein J von 1.46, der zweitbeste ein mehr als anderthalbfach so großes Gütekriterium $J=2.31$. Der viertbeste hat ein mehr als dreimal so großes $J=4.50$. Das Gütekriterium diskriminiert bereits in der Startpopulation sehr stark zwischen dem besten und zweitbesten Parametersatz. Im Vergleich dazu besitzt bei der Reaktorbefüllung mit Wasser der zweitbeste Parametersatz der Startpopulation nur ein 1.06fach höheres J als der beste ($J=0.405$). Und der viertbeste Parametersatz hat lediglich ein 1.09fach höheres J , so daß die Diskriminierung durch den Algorithmus wesentlich geringer ausfällt.

Das Gütekriterium wird im Verlauf von 25 Generationen von $J=1.46$ auf $J=1.16$ (Abbildung 17) verbessert. Obwohl insgesamt nur 102 Parametersätze überprüft werden, werden 25 Generationen durchlaufen. Dies kann als weiteres Zeichen einer zügigen Konvergenz gewertet werden. Da die beste Parameterkombination ab Generation 5 wegen ihrer überragenden Güte viele gleichartige Nachkommen in den Folgegenerationen erzeugt hat, wird der Einfluß des Kreuzungsoperators schnell ausgeschaltet. So kommen in vielen Generationen wenig neuen Parameterkombinationen vor.

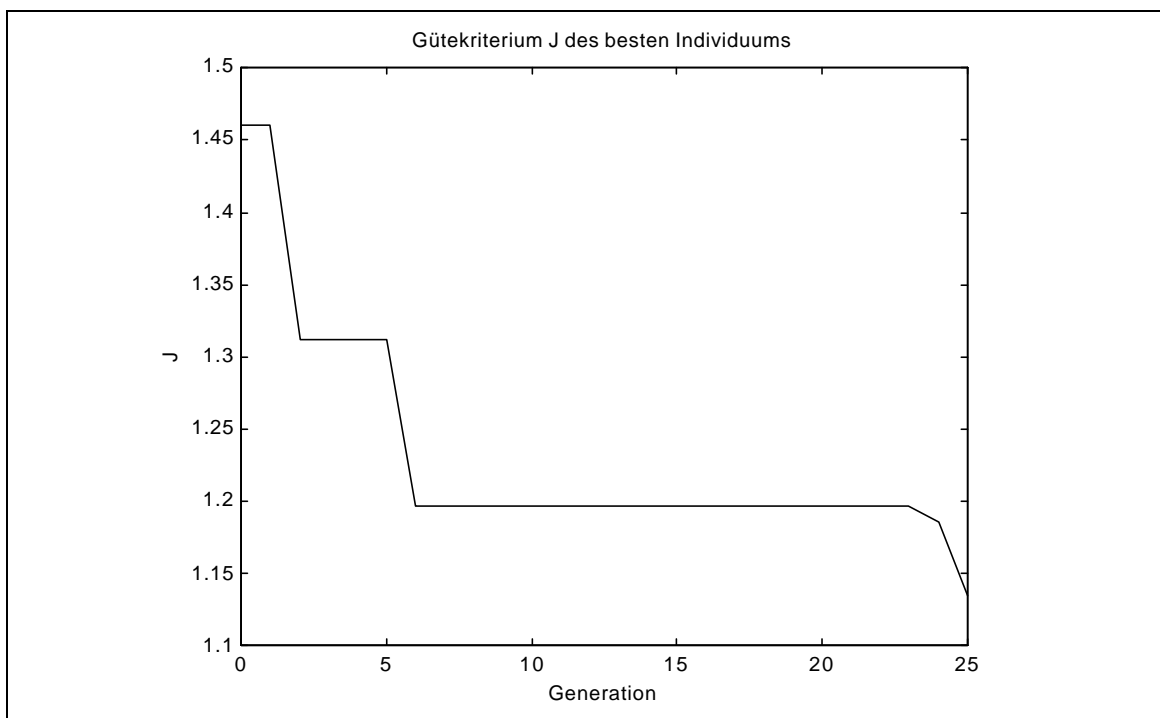


Abb. 17 Gütekriterium J des besten Parametersatzes als Funktion der Generationen

Die Werte des optimalen Parametersatzes liegen mit $c_{Tabw} = 2.76$, $c_{Tänd} = 0.37$ und $c_{Händ} = 0.36$ am Rande des anfänglichen Wertebereichs. Der mit diesen Skalierungsfaktoren betriebene Regler akzeptiert nur geringe *Temperaturänderungen* ($c_{Tänd} < 1$) und reagiert mit kleinen *Heizleistungsänderungen* ($c_{Händ} < 1$). Aus der Interpretation des Skalierungsfaktors der *Temperaturabweichung* von $c_{Tabw} = 2.76$ folgt, daß Regelabweichungen größer ± 0.138 K auftreten müßten, damit die Terme GROßPOSITIV oder GROßNEGATIV aktiviert werden. Eine solche Abweichung entspricht aber nicht den Anforderungen und wird im Regelbetrieb auch nicht beobachtet.

Ein anhand von groben Abschätzungen entworfener Fuzzy-Logik-Regler kann durch die in diesem Kapitel vorgestellte Anpassung seiner Ska-

lierungsfaktoren mittels eines genetischen Algorithmus für die Regelung der Reaktortemperatur bei so unterschiedlichen Reaktionsbefüllungen wie Wasser und viskoserer 6%iger CMC-Lösung angepaßt werden. Über die Skalierungsfaktoren wird das Übertragungsverhalten des Regler global verändert. Eine noch bessere Anpassung des Regelverhaltens sollte erzielt werden, wenn in die Optimierung (dann am Modell) die einzelnen Zugehörigkeitsfunktionen und Regeln mit einbezogen werden.

3.2.3 Identifikation der Regelstrecke

Für die beiden Versuchsflüssigkeiten Wasser und 6%ige CMC Lösung soll eine experimentelle Modellbildung der Regelstrecke vorgenommen werden. Die Regelstrecke reicht vom Stellort Heizung bis zum Meßort Temperatursensor. Um ein aussagekräftiges Modell zu erhalten, muß die Regelstrecke ausreichend angeregt werden. Dazu wird zunächst der Sollwert der Reaktortemperatur $T_{r,soll}$ in einem Bereich von 54.7 bis 55.3 °C zufällig variiert, wobei die einzelnen Sollwerte solange auf einem Wert gehalten werden, bis dieser erreicht ist. Danach wird über die Kalibrierheizung eine definierte Heizleistung als Störung an die Versuchsflüssigkeit abgegeben. Die Rührerdrehzahl N des Ankerrührers mit aufgesetztem Schrägblattrührer beträgt 200 min⁻¹.

Die Regelung der Reaktortemperatur T_r erfolgt mit dem Fuzzy-Logik-Regler (Kap. 3.2.1), der die in Kap. 3.2.2 optimierten Parametern verwendet. Für die Gewinnung von Beispieldaten für die Systemidentifikation der Regelstrecke ist ein optimales Regelverhalten nicht nötig, sie könnte auch am geöffneten Regelkreis erfolgen. Die Verwendung des optimierten Reglers im hier vorgestellten Führungsgrößen- und Störgrößenprogramm zeigt gleichzeitig die Leistungsfähigkeit der Regelung auf Basis der unscharfen Logik.

In Abbildung 18 ist das Führungs- und Störverhalten des Regelkreises für die Reaktorbefüllung mit Wasser dargestellt. Zunächst wird der Sollwert der Reaktortemperatur $T_{r,soll}$ mehrmals sprunghaft verändert und dann für 180 s beibehalten. Obwohl die Skalierungsfaktoren in Kapitel 3.2.2 auf das Störverhalten optimiert worden sind, zeigt der Regler ein gutes Führungsverhalten. Selbst bei großen Sollwertänderungen treten keine merklichen Überschwingweiten in der Reaktortemperatur auf. Im weiteren Versuchsablauf wird die Kalibrierheizleistung im Bereich von 0 bis 25 W sprunghaft variiert, wobei die Leistung für je 120 s konstant beibehalten

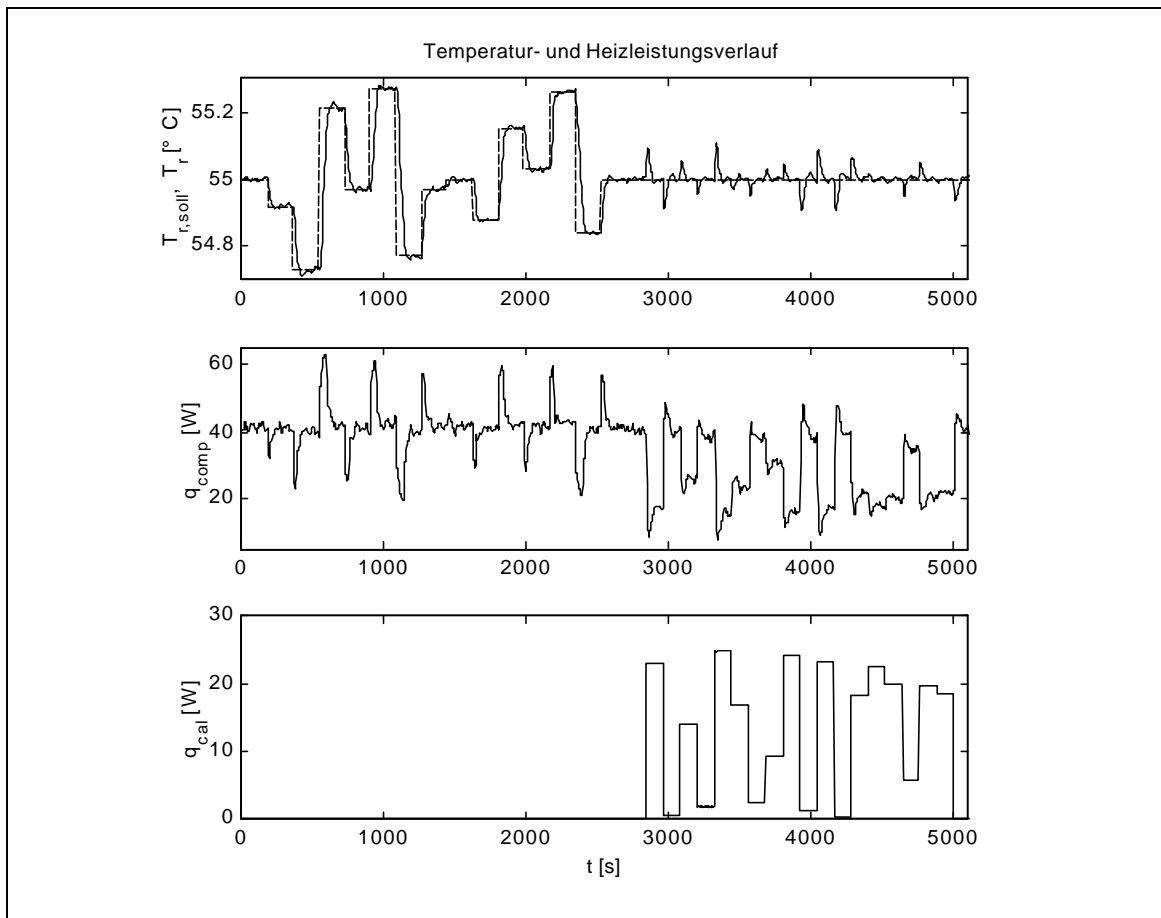


Abb. 18 Führungsgrößen- und Störgrößenprogramm bei einer Reaktorbefüllung mit Wasser

wird. Die geforderte maximale Temperaturabweichung bleibt auch bei Störgrößenänderungen von ca. 20 W im Bereich von ± 0.1 K.

Der Verlauf der Reaktortemperatur bei Befüllung mit 6%iger CMC-Lösung und optimalen Reglerparametern ist in Abbildung 19 dargestellt. Bei der sprunghaften Veränderung der Solltemperatur $T_{r,soll}$ sind die Schwierigkeiten des Reglers zu erkennen, den neuen Sollwert innerhalb von 180 s einzustellen. Der Regler ist bei der Reaktorbefüllung mit 6%iger CMC-Lösung auf ein Regelverhalten mit höherer Dämpfung ($k = 3 \cdot 10^3$) angepaßt worden. Deshalb zeigt er nur recht sachte Regeleinriffe, so daß eine Einstellzeit von 180 s nicht ausreicht. Eine Ausregelzeit von über 180 s zeigt sich auch im anschließenden Störgrößenprogramm, bei dem die Kalibrierheizleistung im Bereich von 0 bis 12 W verändert wird. Charakteristisch ist hier das Auftreten eines zweiten Temperaturpeaks bei größeren Heizleistungsänderungen ($\Delta q_{cal} > 8$ W). Die maximale Temperaturabweichung bleibt im geforderten Rahmen von ± 0.1 K.

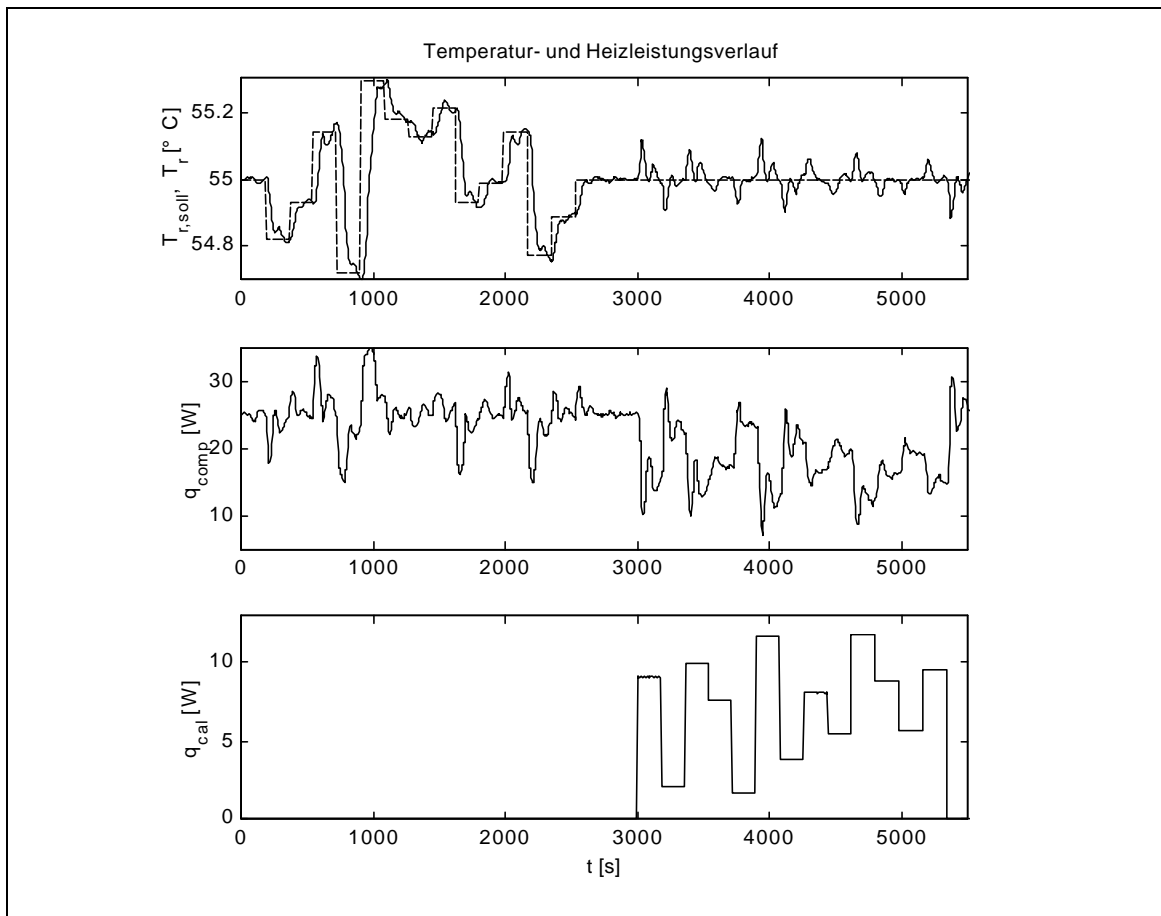


Abb. 19 Führungsgrößen- und Störgrößenprogramm bei einer Reaktorbefüllung mit 6%iger CMC-Lösung

Durch die Aufzeichnung des Temperatur- und Heizleistungsprogramms für eine Reaktorbefüllung mit Wasser oder mit 6%iger CMC-Lösung liegen genügend Meßwerte vor, um eine experimentelle Identifikation der Regelstrecke durchführen zu können.

Bei der Systemidentifikation wird ein funktionaler Zusammenhang zwischen zurückliegenden Ein-/Ausgangswerten des Systems ($(q_{ges}(t-f), T_r(t-f), f=0..(N_V-1))$) und dem zukünftigen Ausgangswert ($T_r(t+1)$) hergestellt. Die N_V Ein-/Ausgangswerte des Systems werden zu einem Vektor $x_h(t)$ ($h=2 \cdot N_V$), dem Eingabevektor des Modells, zusammengefaßt (52). N_V wird im folgenden als Anzahl der Vergangenheitswerte bezeichnet.

$$x_h(t) = (q_{ges}(t), q_{ges}(t-1), \dots, q_{ges}(t-N_V+1), T_r(t), T_r(t-1), \dots, T_r(t-N_V+1)) \quad (52)$$

Die experimentelle Modellbildung wird mit einem in Anhang B beschriebenen Mehrschicht-Perzeptron durchgeführt. Die Ein- und Ausgangswerte der Regelstrecke $q_{ges}(t)$ und $T_r(t)$ werden zunächst separat auf

das Intervall $[-1, 1]$ skaliert, erst danach wird der Vektor $x_h(t)$ (52) zusammengesetzt. Der Sollausgabewert des Modells $y_k(t+1)$ ist die Reaktortemperatur $T_r(t+1)$ (ebenfalls skaliert auf das Intervall $[-1, 1]$):

$$y_k(t+1) = T_r(t+1) \quad (53)$$

Eingabevektor $x_h(t)$ und Ausgabewert des Modells $y_k(t+1)$ werden zum Trainingsdatenpaar l zusammengefaßt:

$$(x_h(t), y_k(t+1))_l \quad (54)$$

Das Mehrschicht-Perzeptron besitzt die Fähigkeit, jeden für eine Modellbildung nötigen funktionalen Zusammenhang darstellen zu können (universelle Funktionsapproximation). Es ist aus drei Neuronenschichten aufgebaut. Die Eingabeschicht besteht aus einem Bias-Neuron und $(n-1)$ nicht-funktionalen Neuronen, die die Elemente des Eingabevektors $x_h(t)$ ($h=2 \cdot N_V$) auf die Neuronen der folgenden verborgenen Schicht verteilen. Die verborgene Neuronenschicht besteht aus $u = (m-1)$ verarbeitenden Neuronen und einem Bias-Neuron. Ist von der Anzahl u der Neuronen die Rede, so sind damit die Anzahl operationaler Neuronen in der verborgenen Schicht gemeint. Die Ausgabeschicht des Mehrschicht-Perzeptron besteht aus einem Ausgabeneuron, das die vom Modell berechnete Reaktortemperatur $T_{r,sim}(t+1)$ (skaliert $o_k(t+1)$) ausgibt. Die u Neuronen der verborgenen Schicht besitzen die logistische Transferfunktion (B.2), das Ausgabeneuron k hat eine lineare Transferfunktion (B.4).

Um keine Einschränkungen bezüglich der Totzeit, der Ordnung oder der Nichtlinearität der Regelstrecke zu machen, sind die Anzahl der Vergangenheitswerte N_V und die Anzahl der Neuronen u als frei wählbare Parameter zu bestimmen. Das Mehrschicht-Perzeptron wird daher anhand aller Eingabe-Ausgabe-Paare trainiert, wobei die Werte für N_V und u ganzzahlig von 1 bis 10 verändert werden. Aus Vorversuchen ist bekannt, daß eine Unterteilung in einen Trainings- und einen Testdatensatz nicht notwendig ist. Für die Anpassung der Netzgewichte wird der stochastische Fehlerrückführalgorithmus (Anhang B) verwendet. Um die Konvergenz des Algorithmus zu gewährleisten, wird die Lernrate auf $h=0.05$ gesetzt und der Momentumterm mit $m=0.001$ nahezu ausgeschaltet. Ferner wird die Anzahl der Epochen auf einen großen Wert von 10^4 gesetzt. Um die Ergebnisse des Fehlerrückführalgorithmus unabhängig von den

anfänglichen, zufällig gewählten Netzgewichten zu machen, wird das Verfahren für jede Kombination aus N_V und u dreimal durchlaufen. Wenn ein durchschnittlicher *rms*-Fehler angesprochen wird, ist damit der Mittelwert über die drei Durchläufe gemeint.

In Abbildung 20 sind die durchschnittlichen *rms*-Fehler für den Trainingsdatensatz mit Wasser als Reaktionsbefüllung dargestellt. Sowohl mit zunehmender Anzahl der Neuronen u als auch mit zunehmender Anzahl der Vergangenheitswerte N_V nimmt der *rms*-Fehler ab. Dies ist verständlich, da in beiden Fällen die Gesamtzahl der Netzgewichte zunimmt und das Mehrschicht-Perzeptron damit eher in der Lage ist, das Regelstreckenverhalten darzustellen. Ein Plateau des *rms*-Fehlers wird ab einer Neuronenanzahl von $u \geq 5$ und ab einer Anzahl Vergangenheitswerte $N_V \geq 5$ mit Werten kleiner $7.1 \cdot 10^{-3}$ erreicht. Der *rms*-Fehler in Abb. 20 bezieht sich auf die skalierten Werte der Reaktortemperatur $y_k(t+1)$, eine Umrechnung in $^{\circ}\text{C}$ kann mit dem Faktor 0.280 erfolgen.

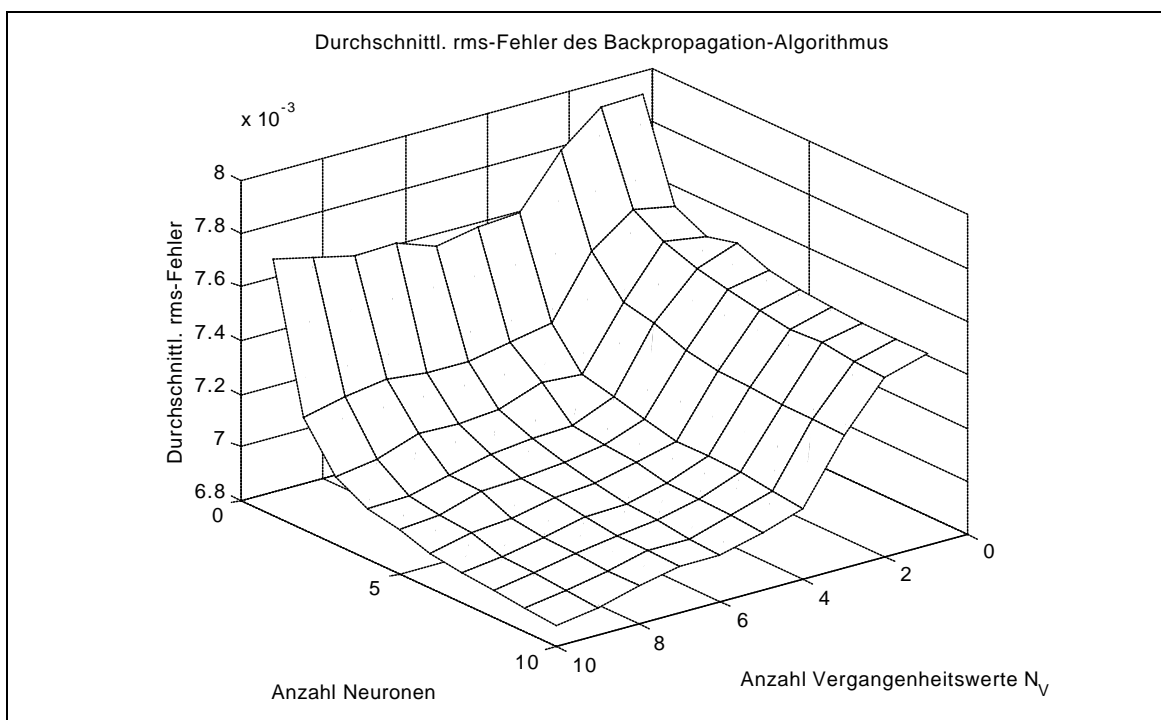


Abb. 20 Durchschnittlicher *rms*-Fehler der Modellbildung bei Wasser als Reaktorbefüllung, Fehlerrückführalgorithmus

Die *rms*-Fehler für dem Trainingsdatensatz mit 6%iger CMC-Lösung als Reaktorbefüllung sind in Abbildung 21 eingetragen. Es fällt auf, daß der *rms*-Fehler für eine Anzahl Vergangenheitswerte $N_V < 4$ sehr große Werte annimmt. In Abbildung 22 wird dieser Bereich daher ausgeblendet.

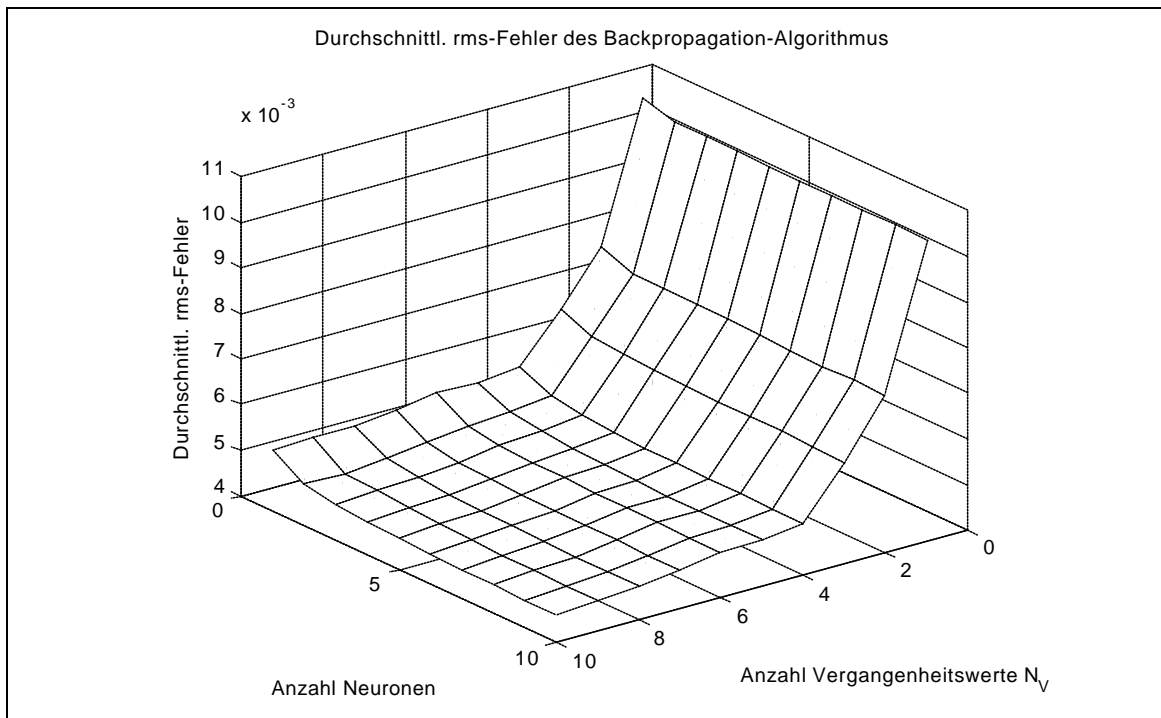


Abb. 21 Durchschnittlicher *rms*-Fehler der Modellbildung bei 6%iger CMC-Lösung als Reaktorbefüllung, Fehlerrückführungsalgorithmus

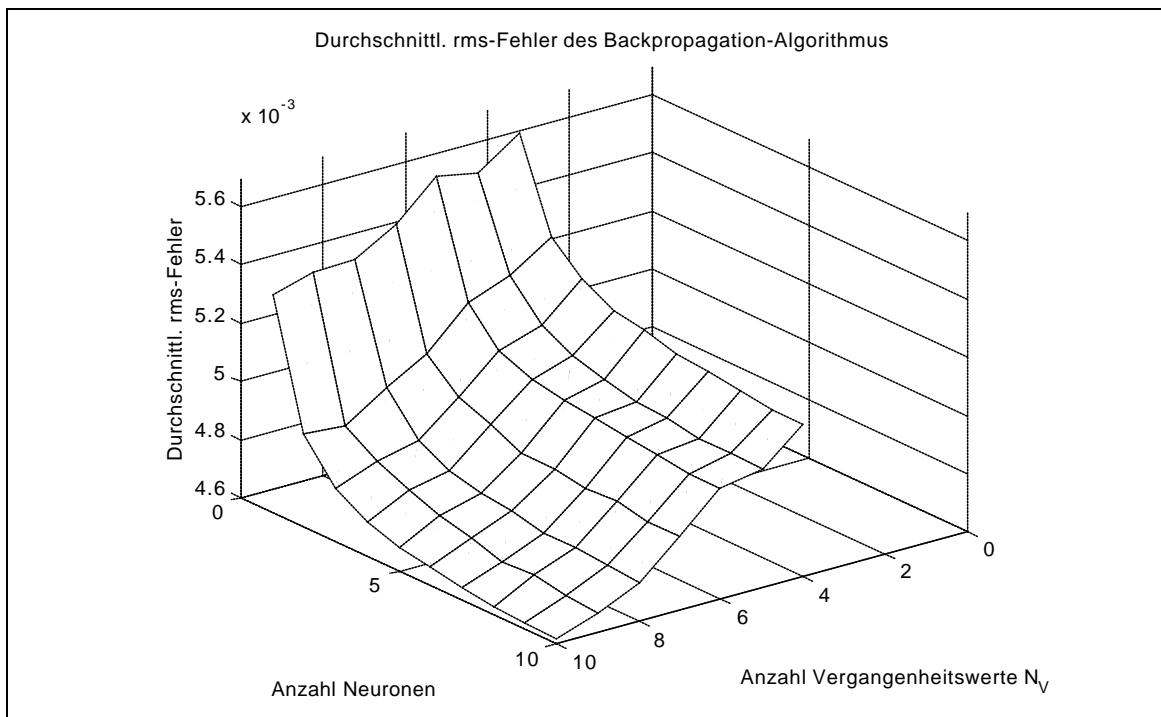


Abb. 22 Durchschnittlicher *rms*-Fehler (Ausschnitt) der Modellbildung bei 6%iger CMC-Lösung als Reaktorbefüllung

Bei weniger als 4 Vergangenheitswerten kann das neuronale Netz nicht den korrekten Temperaturwert $y_k(t+1)$ ausgeben. Da die Regelstrecke bei 6%iger CMC-Lösung als Reaktorbefüllung eine längere Totzeit t_D aufweist, müssen

dem Modell entsprechend ältere Ein-/Ausgangsdaten im Eingabevektor $x_h(t)$ zur Verfügung gestellt werden. Ab einer Neuronenanzahl von $u \geq 5$ (im Einklang mit den Ergebnissen für eine Reaktorbefüllung mit Wasser) und einer Anzahl Vergangenheitswerte $N_V \geq 7$ werden genügend kleine *rms*-Fehler von kleiner $5.0 \cdot 10^{-3}$ erreicht. Der *rms*-Fehler bezieht sich auf skalierte Temperaturwerte $y_k(t+1)$ und kann mit dem Faktor 0.300 in °C umgerechnet werden.

Mit zunehmender Anzahl u und N_V nähert sich der *rms*-Fehler einem Grenzwert von ca. $6.8 \cdot 10^{-3}$ bei der Reaktorbefüllung mit Wasser bzw. von ungefähr $4.5 \cdot 10^{-3}$ bei der Befüllung mit 6%iger CMC-Lösung. Bei der maximalen Neuronenzahl $u = 10$ und der maximalen Anzahl Vergangenheitswerte $N_V = 10$ verfügt das Mehrschicht-Perzeptron über genügend Informationen und Flexibilität, um sich an ein beliebiges Systemverhalten anpassen zu können. Daß der *rms*-Fehler nicht auf Null zurückgeht, muß demnach auf dem Meßrauschen beruhen, das bei der Aufnahme der Temperatur- und Heizleistungswerte ($T_r(t)$, $q_{ges}(t)$) auftritt. Um eine gute Generalisationsfähigkeit des neuronalen Netzes sicherzustellen, wird die Anzahl der Neuronen u so gering wie möglich gewählt.

Es soll jedoch ausgeschlossen werden, daß der stochastische Fehlerrückführungsalgorithmus ein lokales Minimum der Netzfehlerfunktion als beste Anpassung der Netzgewichte an eine Identifikationsaufgabe ausgibt. Zu diesem Zweck wird der Fehlerrückführungsalgorithmus mit der Methode des simulierten Abkühlens (*simulated annealing*) ergänzt. Für jede Kombination aus Neuronenanzahl u und Anzahl Vergangenheitswerte N_V werden drei Optimierungsläufe durchgeführt. Der Fehlerrückführungsalgorithmus bleibt mit einer Lernrate von $h = 0.05$ (ohne Momentumterm) und 10^4 Epochen nahezu unverändert. Allerdings werden nach je 10^2 Epochen zehn Zyklen des simulierten Abkühlens bei einem konstanten Wert der künstlichen Temperatur durchlaufen. Die künstliche Temperatur ändert sich im zeitlichen Ablauf des zusammengesetzten Verfahrens exponentiell von 373 auf 10^{-3} . Die maximal erlaubte Änderung der Netzgewichte fällt exponentiell von $\pm 151\%$ auf $\pm 1\%$ ab. Während eines Zyklus des simulierten Abkühlens wird ein zufällig ausgewählter Anteil von 10% aller Netzgewichte verändert.

In den Abbildungen 23 und 24 sind zur besseren Vergleichbarkeit die Differenzen der durchschnittlichen *rms*-Fehler zwischen dem reinen Fehlerrückführungsalgorithmus und dem zusammengesetzten Verfahren darge-

stellt. Das zusammengesetzte Verfahren liefert nur unwesentlich bessere *rms*-Fehler von durchschnittlich 1.5 bis 3.2 Prozent.

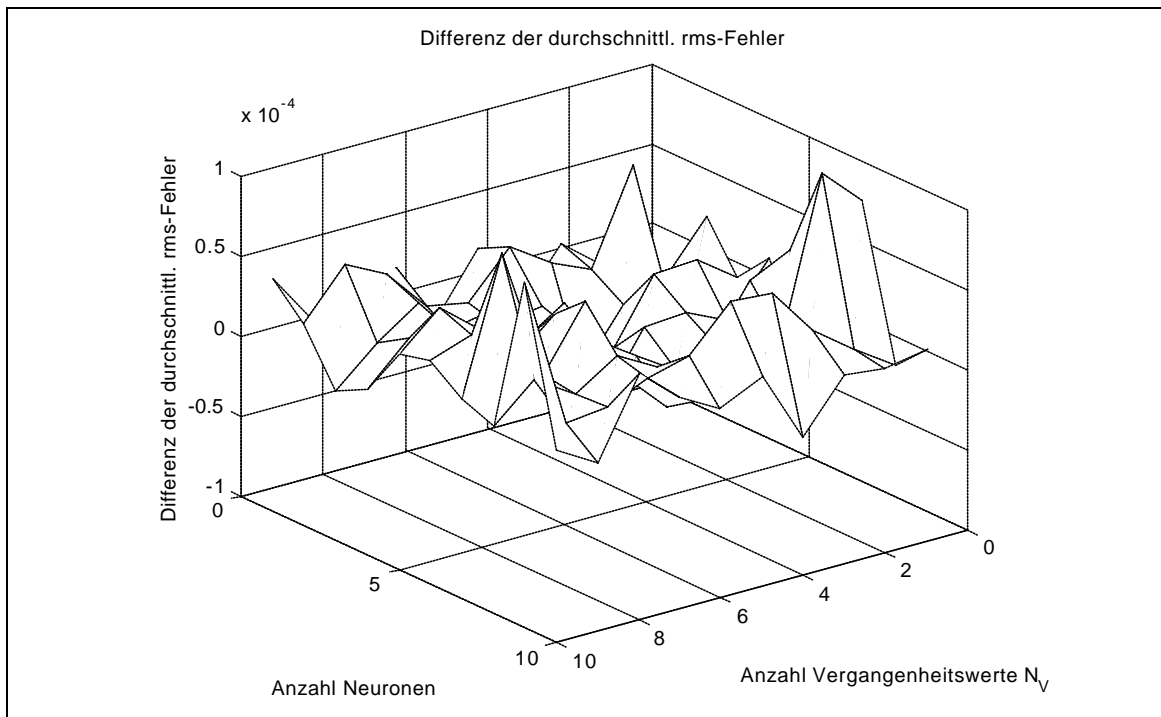


Abb. 23 Differenz der *rms*-Fehler des Fehlerrückführalgorithmus und des mit Simulated Annealing ergänzten Verfahrens (Wasser)

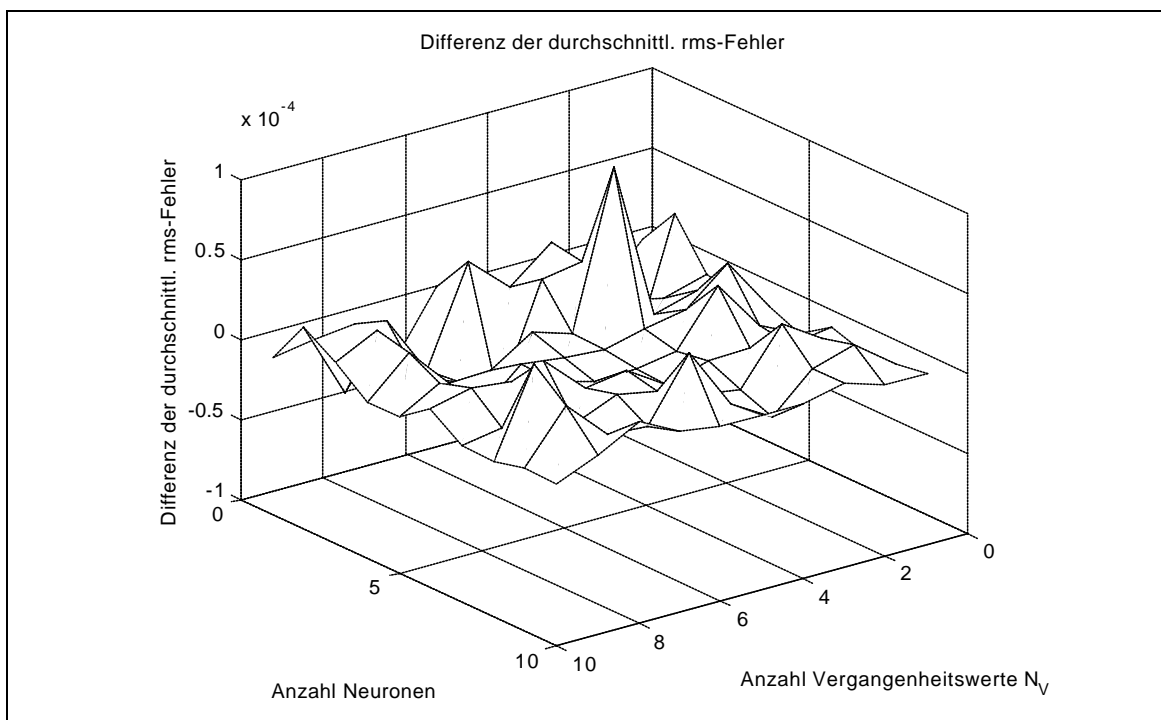


Abb. 24 Differenz der *rms*-Fehler des Fehlerrückführalgorithmus und des mit Simulated Annealing ergänzten Verfahrens (6%ige CMC-Lösung)

Die Verwendung des reinen Fehlerrückführungsalgorithmus ist für die Anpassung des Mehrschicht-Perzeptrons demnach völlig ausreichend. Aus den Ergebnissen dieser Trainingsläufe ergeben sich eine Neuronenanzahl von $u=5$ und eine Anzahl Vergangenheitswerte von $N_V=7$, die für eine Identifikation der Regelstrecke benötigt werden.

Da die Systemidentifikation mit dem Mehrschicht-Perzeptron parallel zur Regelstrecke durchgeführt werden soll, werden die Parameter des Fehlerrückführungsverfahrens geeignet angepaßt. Die Systemidentifikation ist ein zeitkritischer Prozeß, der *rms*-Fehler sollte innerhalb weniger Epochen verringert werden. Daher wird die Anzahl der Epochen auf 500 begrenzt und bei einer Netzkonfiguration von $u=5$ und $N=7$ der *rms*-Fehler in 3 Läufen für die unterschiedlichen Kombinationen der Lernrate h und der Momentumterm-Konstante m bestimmt. Die Lernrate wird zwischen $h=0.05$ und $h=0.90$ und die Momentumterm-Konstante zwischen $m=0.00$ und $m=0.40$ variiert.

Die so ermittelten *rms*-Fehler für den Trainingsdatensatz der Reaktorbefüllung mit Wasser sind in Abbildung 25, die *rms*-Fehler für den Trainingsdatensatz der Reaktorbefüllung mit 6%iger CMC-Lösung sind in Abbildung 26 dargestellt.

Unabhängig vom verwendeten Trainingsdatensatz fällt auf, daß sowohl für sehr kleine Lernraten (<0.3) als auch für sehr große Lernraten (>0.7) größere *rms*-Fehler resultieren. Bei zu kleinen Lernraten ist die geringe Schrittweite für eine schleppende Optimierung verantwortlich (lokale Minima), bei zu großen Lernraten kommt es hingegen zu einem oszillierenden Verhalten des Verfahrens, das keine Konvergenz auf ein Minimum zuläßt. Die Momentumterm-Konstante m hat in dem betrachteten Bereich ($0 \leq m \leq 0.4$) für den Fehlerrückführungsalgorithmus keinen Einfluß auf den *rms*-Fehler.

Trotz der Mittelwertbildung des *rms*-Fehlers über 3 Läufe des Algorithmus mit je 500 Epochen sind die erhaltenen *rms*-Fehler-Funktionen $f(h, m)$ stark zerklüftet. Die Ergebnisse sind bei der geringen Anzahl Epochen abhängig von den zufällig gewählten Startgewichten des Netzes. Der stochastische Fehlerrückführungsalgorithmus besitzt wegen der zufällig gewählten Reihenfolge der Datenpräsentation einen hohen Zufallsanteil, der eine Vorhersagbarkeit des Verfahrens unmöglich macht.

Für das Training des Mehrschicht-Perzeptrons parallel zur Regelstrecke werden eine Lernrate von $h=0.40$ und eine Momentumterm-Konstante von

$m=0.05$ verwendet. Diese Wahl stellt eine zügige Konvergenz des Fehler-rückführungsalgorithmus im *Online*-Betrieb sicher.

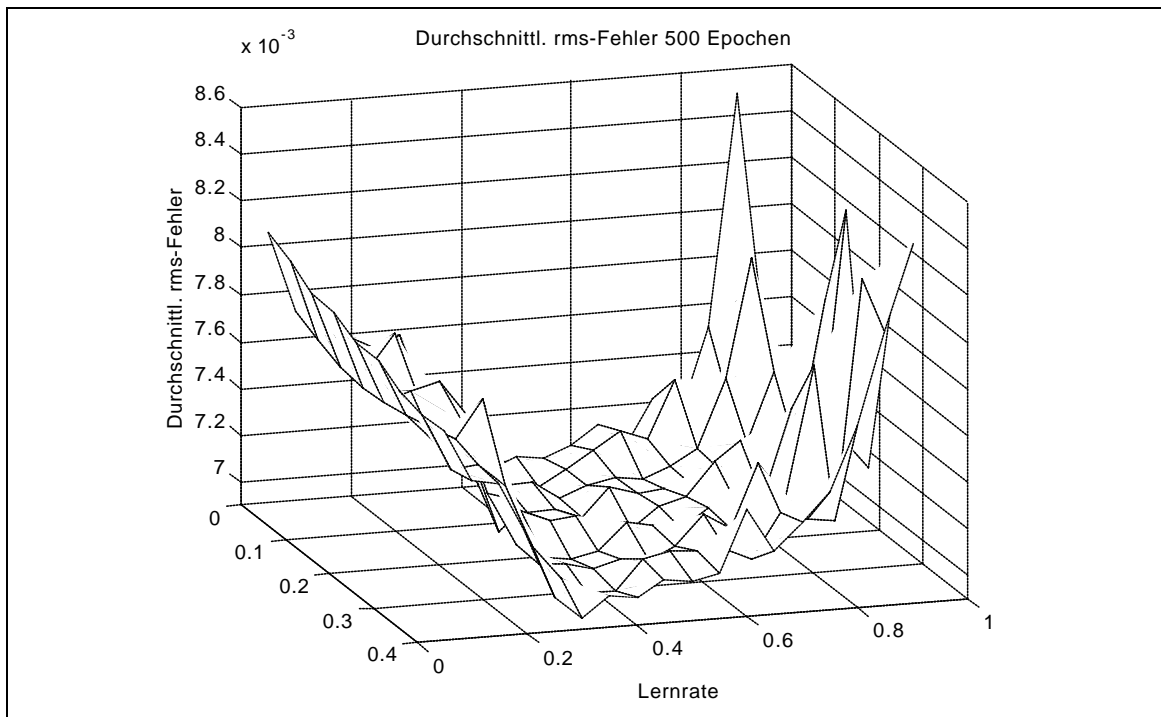


Abb. 25 Durchschnittlicher *rms*-Fehler nach 500 Epochen bei unterschiedlichen Lernraten h und Momentumterm-Konstanten m (Wasser)

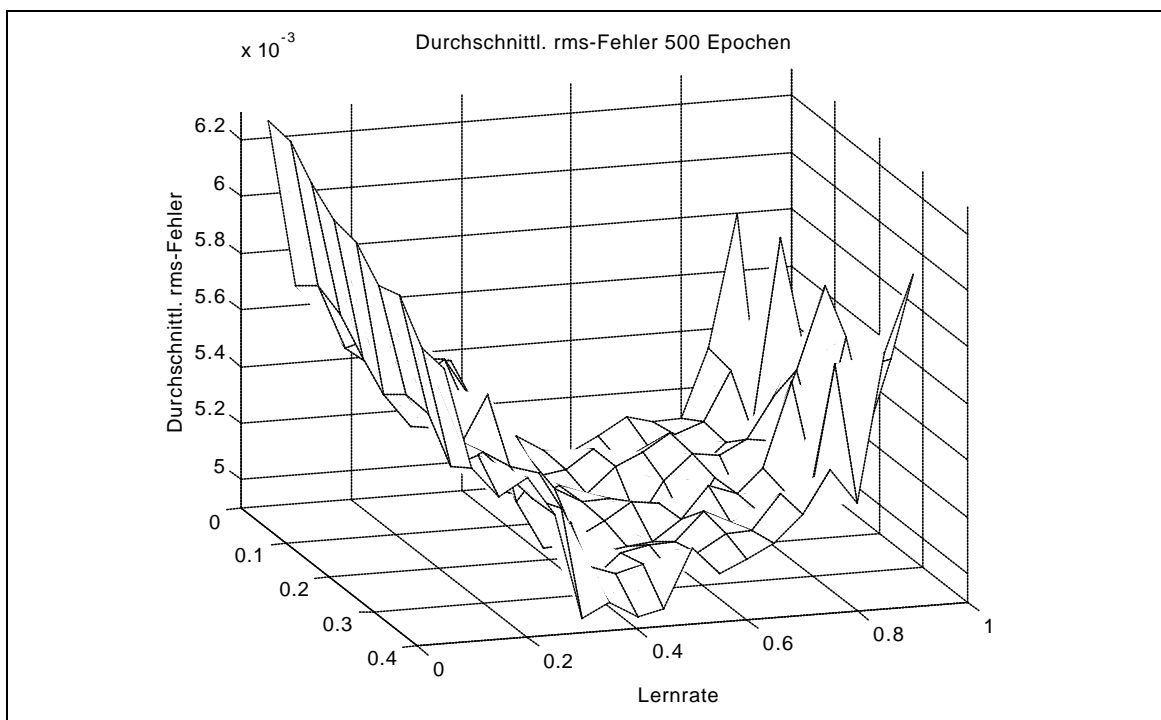


Abb. 26 Durchschnittlicher *rms*-Fehler nach 500 Epochen bei unterschiedlichen Lernraten h und Momentumterm-Konstanten m (6%ige CMC-Lösung)

Die Güte des trainierten Mehrschicht-Perzeptrons für eine Reaktorbefüllung mit Wasser bei der Ein-Schritt-Prädiktion geht aus Abbildung 27 hervor. Neben der Reaktortemperatur $T_r(t+1)$ und der vom Modell berechneten Temperatur $T_{r,sim}(t+1)$ ist die Differenz beider Werte dargestellt. Innerhalb der ersten 2800 s (Führungsgrößenprogramm) bewegt sich die Temperaturdifferenz $T_r - T_{r,sim}$ in einem engen Bereich von ± 0.004 K. In der Zeit des Störgrößenprogramms von 2800 bis 5100 s treten einige größere Abweichungen von ± 0.015 K auf. Diese beruhen auf die sprunghaften Änderungen der Gesamtheizleistung q_{ges} und den resultierenden schnellen Änderungen der Reaktortemperatur T_r . An dieses sprunghafte Verhalten kann sich das neuronale Netz nicht anpassen; auch deshalb, weil ihre Häufigkeit im Trainingsdatensatz gering ist.

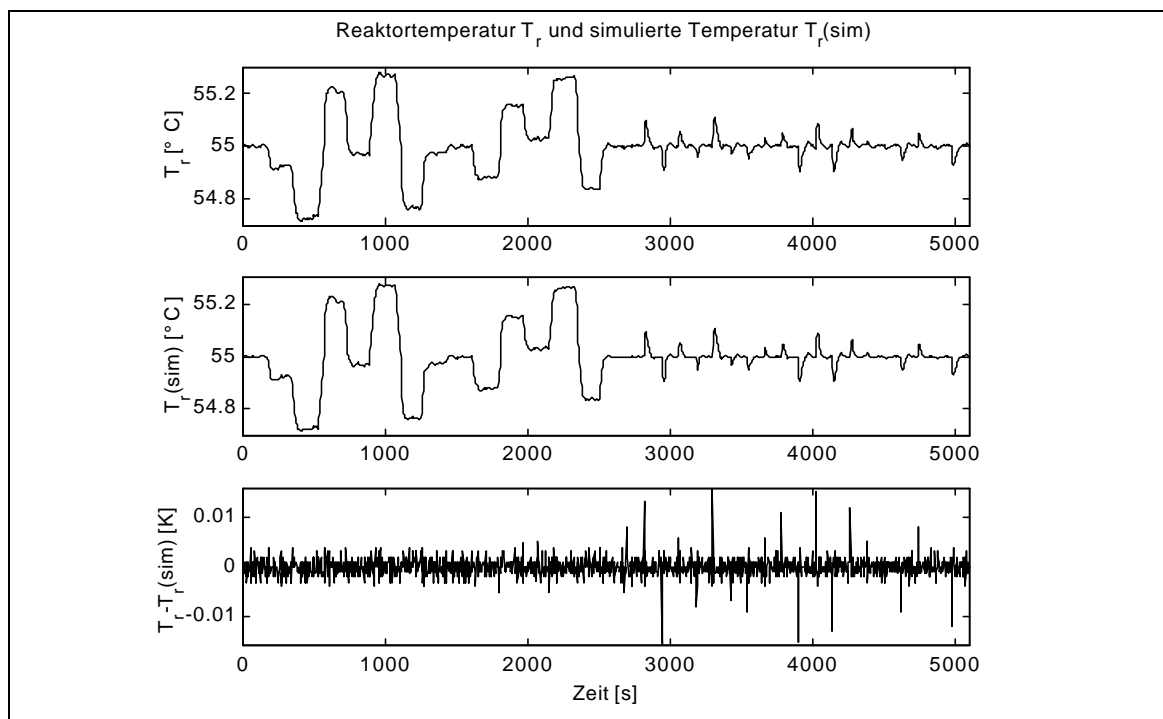


Abb. 27 Gemessene und simulierte Temperaturwerte (Wasser) bei einer Modellierung mit einem [14-5-1] Mehrschicht-Perzeptron ($N_V = 7$, $u = 5$)

Der positive Eindruck von der Modellgenauigkeit wird durch Abbildung 28 bestätigt. Die Darstellung der absoluten Häufigkeiten der Temperaturabweichungen zeigt, daß am häufigsten Differenzen bis ± 0.004 K vorkommen. Größere Differenzen werden sehr selten vom Modell verursacht.

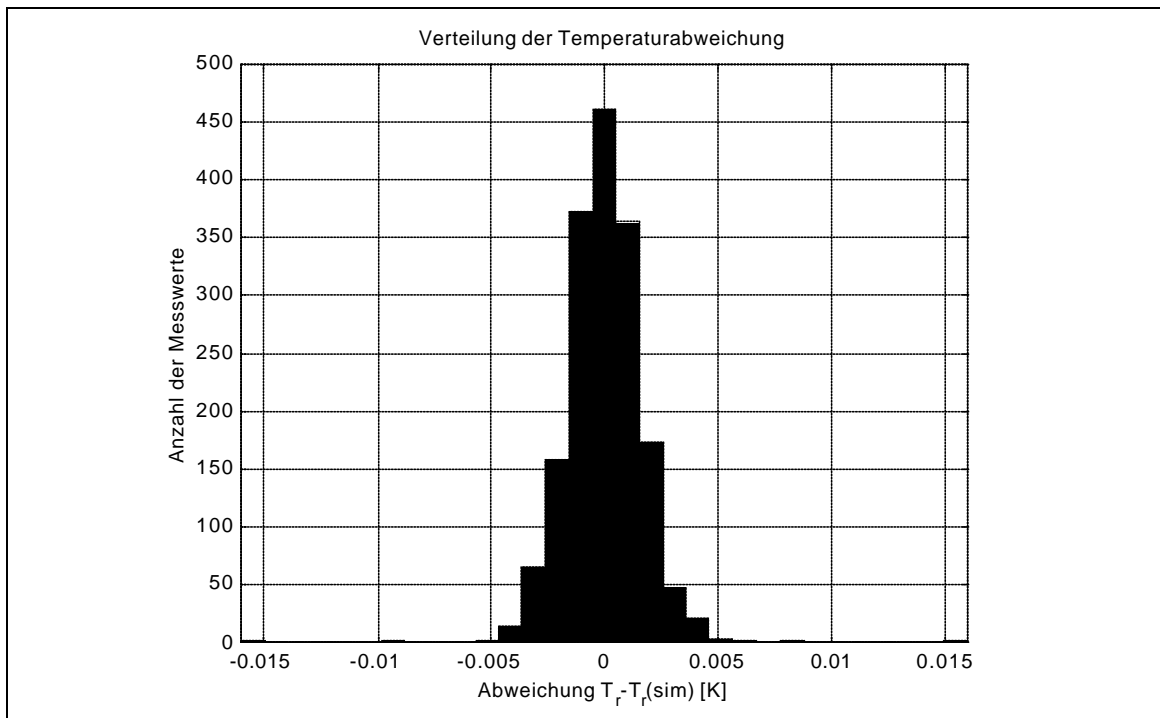


Abb. 28 Absolute Häufigkeit der Abweichung zwischen gemessener und simulierter Temperatur (Wasser)

Noch besser gelingt die Ein-Schritt-Prädiktion dem Mehrschicht-Perzeptron, das anhand des Datensatzes für 6%ige CMC-Lösung als Reaktorbefüllung trainiert worden ist (Abb. 29). Über den gesamten Zeitraum bewegt sich die Temperaturdifferenz $T_r - T_{r,sim}$ im kleinen Bereich von ± 0.003 K, wobei kein Unterschied zwischen Führungsgrößenänderungen und Störgrößenänderungen auszumachen ist. Da die Reaktortemperatur T_r aufgrund der geänderten hydrodynamischen Verhältnisse bei einer Befüllung mit 6%iger CMC-Lösung auf Heizleistungsänderungen wesentlich träger reagiert, sind auch die Anforderungen an die Modellbildung geringer. Die absoluten Häufigkeiten der Temperaturdifferenzen zwischen realen und simulierten Temperaturwerten sind in Abbildung 30 aufgezeigt. Abweichungen > 0.003 K treten nur sehr selten auf, was die hohe Modellgüte bestätigt.

Die trainierten neuronalen Netze geben ein sehr gutes Regelstreckenmodell ab, das für die weitere Optimierung des Fuzzy-Logik-Reglers verwendet werden soll. Wird ein Mehrschicht-Perzeptron als Modell parallel zur Regelstrecke trainiert, sollte es bereits vor dem *Online*-Betrieb anhand von Beispieldaten trainiert worden sein, damit eine zügige Konvergenz garantiert wird.

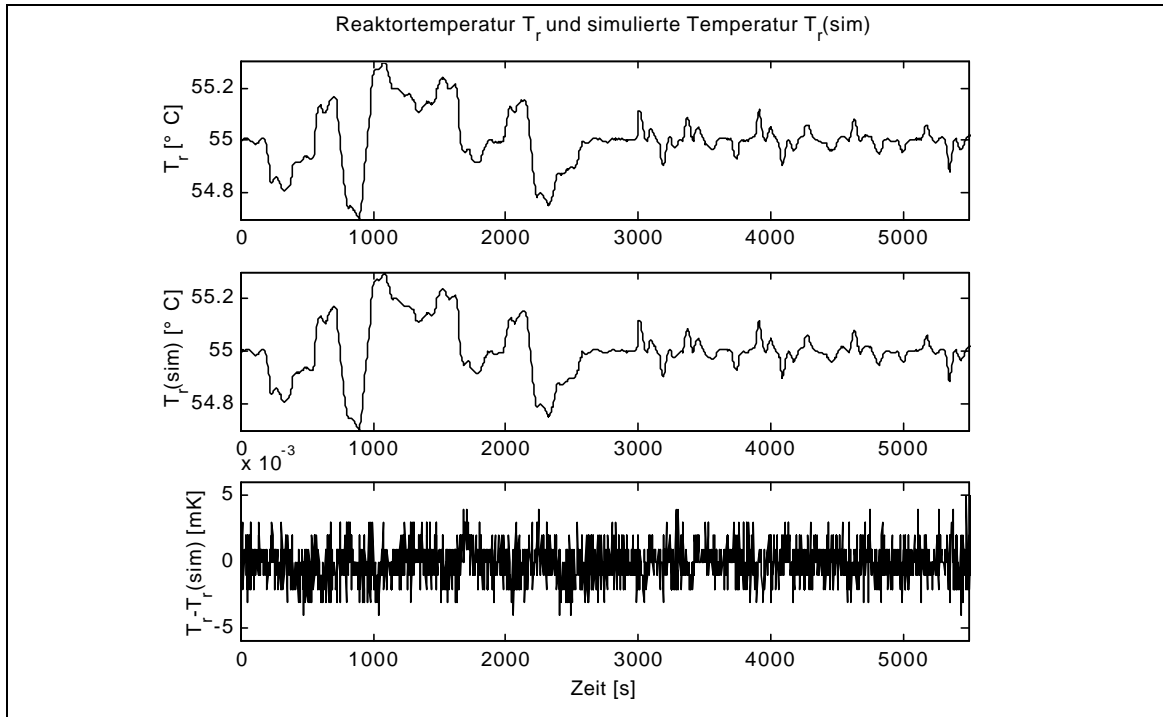


Abb. 29 Gemessene und simulierte Temperaturwerte (6% CMC) bei einer Modellierung mit einem [14-5-1] Mehrschicht-Perzeptron ($N_V = 7$, $u = 5$)

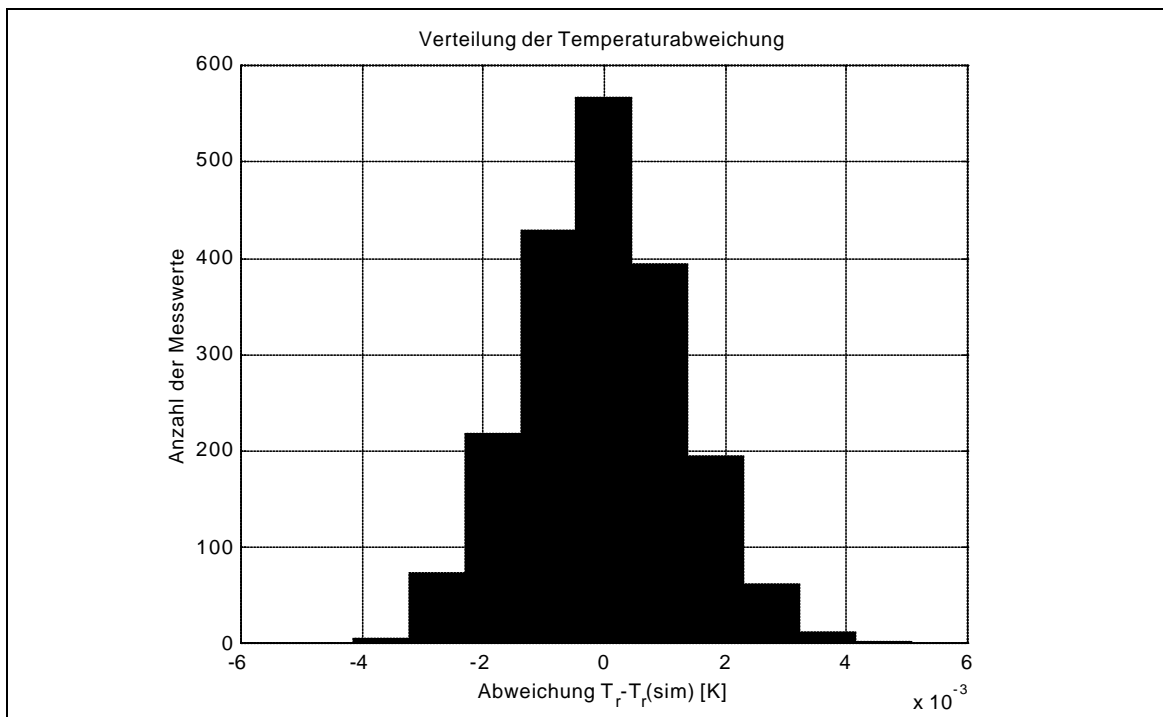


Abb. 30 Absolute Häufigkeit der Abweichung zwischen gemessener und simulierter Temperatur (6% CMC)

3.2.4 Modellgestützte Validierung der Skalierungsfaktoren und Parameteroptimierung des Fuzzy-Logik-Regler

Mit den in Kapitel 3.2.3 beschriebenen Modellen der Regelstrecke ist es möglich, eine Überprüfung der in Kapitel 3.2.2 durch direkte Messungen an der Anlage ermittelten Skalierungsfaktoren c_{Tabw} , $c_{Tänd}$ und $c_{Händ}$ vorzunehmen. Darüber hinaus kann anhand der Steckenmodelle ein optimaler Regler entworfen werden, wobei alle Parameter eines Fuzzy-Logik-Reglers (Parameter der Zugehörigkeitsfunktionen) mittels eines genetischen Algorithmus angepaßt werden. Dieser Reglerentwurf wird im Anschluß an die Validierung der Skalierungsfaktoren beschrieben.

Die Überprüfung der Skalierungsfaktoren erfolgt am gleichen rechteckförmigen Verlauf der Kalibrierheizleistung q_{cal} wie in Abb. 11. Die maximale simulierte Heizleistung beträgt 20 W beim Modell für Wasser und 10 W beim Modell für 6%ige CMC-Lösung als Reaktorbefüllung. Da es sich um einen Simulation mit immer gleichen Startwerten von T_r und q_{comp} handelt, kann auf den Beruhigungszeitraum t_0 bis t_2 verzichtet werden. Charakterisiert wird jeder Parametersatz (bestehend aus c_{Tabw} , $c_{Tänd}$ und $c_{Händ}$) anhand des Gütekriteriums J mit $k = 1 \cdot 10^4$ beim Modell für Wasser bzw. mit $k = 3 \cdot 10^3$ beim Modell für 6%ige CMC-Lösung. Jeder Skalierungsfaktor wird in 0.01-Schritten verändert und das Gütekriterium ermittelt. Die besten gefundenen Skalierungsfaktoren sind in Tabelle 16 zusammen mit den in Kapitel 3.2.2 gefundenen Werten eingetragen.

Tab. 11: Optimierte Skalierungsfaktoren

Reaktorbefüllung (Versuch)	Skalierungsfaktor		
	c_{Tabw}	$c_{Tänd}$	$c_{Händ}$
Wasser (Kapitel 3.2.2)	1.61	0.51	1.51
Wasser (Simulation)	1.55	0.43	1.41
6% CMC (Kapitel 3.2.2)	2.76	0.37	0.36
6% CMC (Simulation)	2.77	0.38	0.43

Tabelle 11 kann entnommen werden, daß die durch Simulation gefundenen Skalierungsfaktoren, den Skalierungsfaktoren recht ähnlich sind, die experimentell bestimmt worden sind. Aufgrund dieses Ergebnisses können zwei Aussagen getroffen werden: Zum einen liefert der in Kapitel 3.2.2 eingesetzte genetische Algorithmus tatsächlich die besten Skalierungs-

faktoren und ist somit als Verfahren zur Regleroptimierung einsetzbar, zum anderen liefern die in Kapitel 3.2.3 auf der Basis künstlicher neuronaler Netze ermittelten Ein-/Ausgangsmodelle brauchbare und einsetzbare Prozeßmodelle von ausreichender Güte.

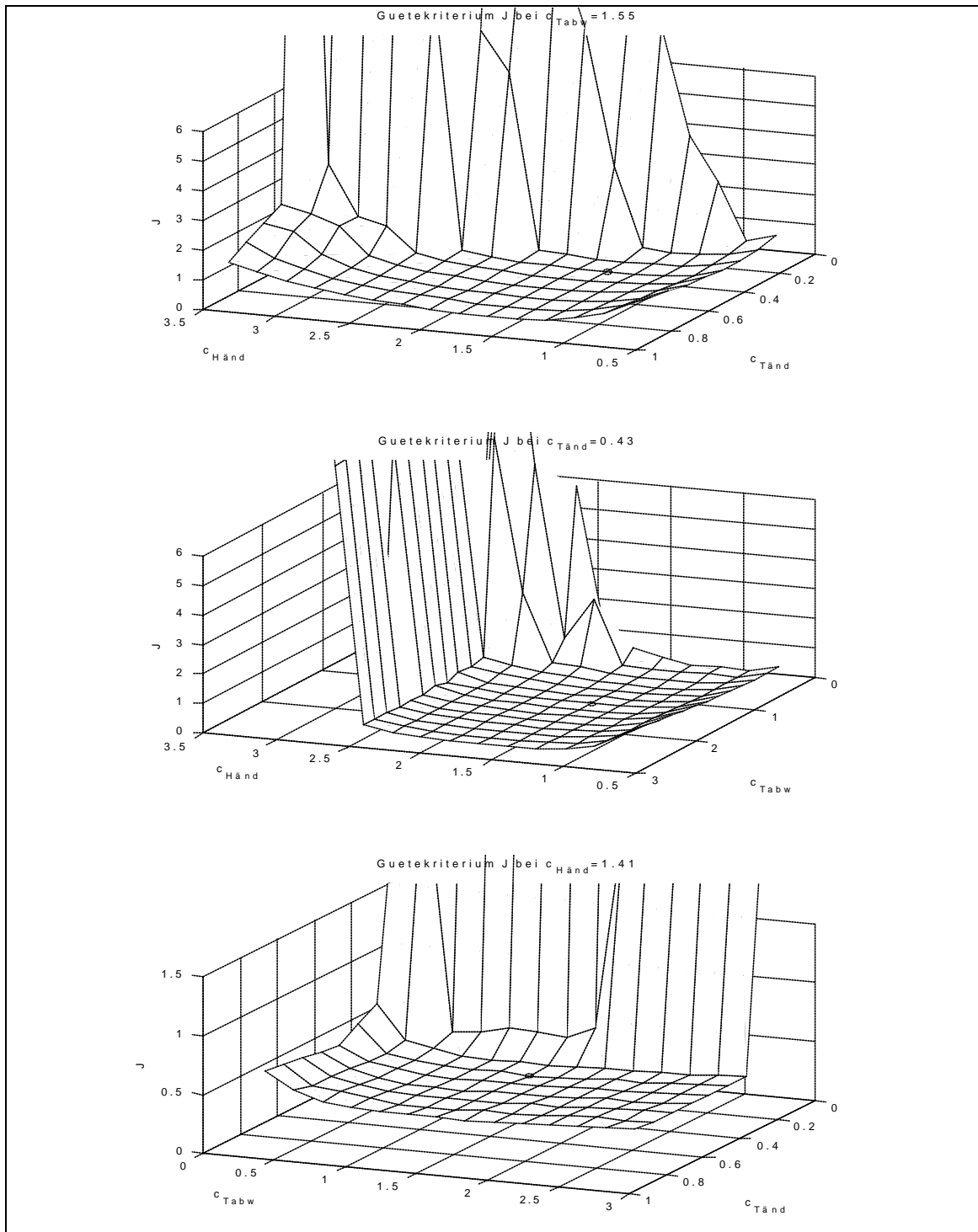


Abb. 31 Gütekriterium J in der Umgebung der optimalen Skalierungsfaktoren, Simulation der Reaktorbefüllung mit Wasser

Um einen Eindruck von den Anforderungen zu gewinnen, die an das Optimierungsverfahren gestellt werden, sind in den Abbildungen 31 und 32 die Gütekriterien J als Funktion der Skalierungsfaktoren dargestellt. Dabei wird je ein Skalierungsfaktor konstant gehalten, und die beiden anderen

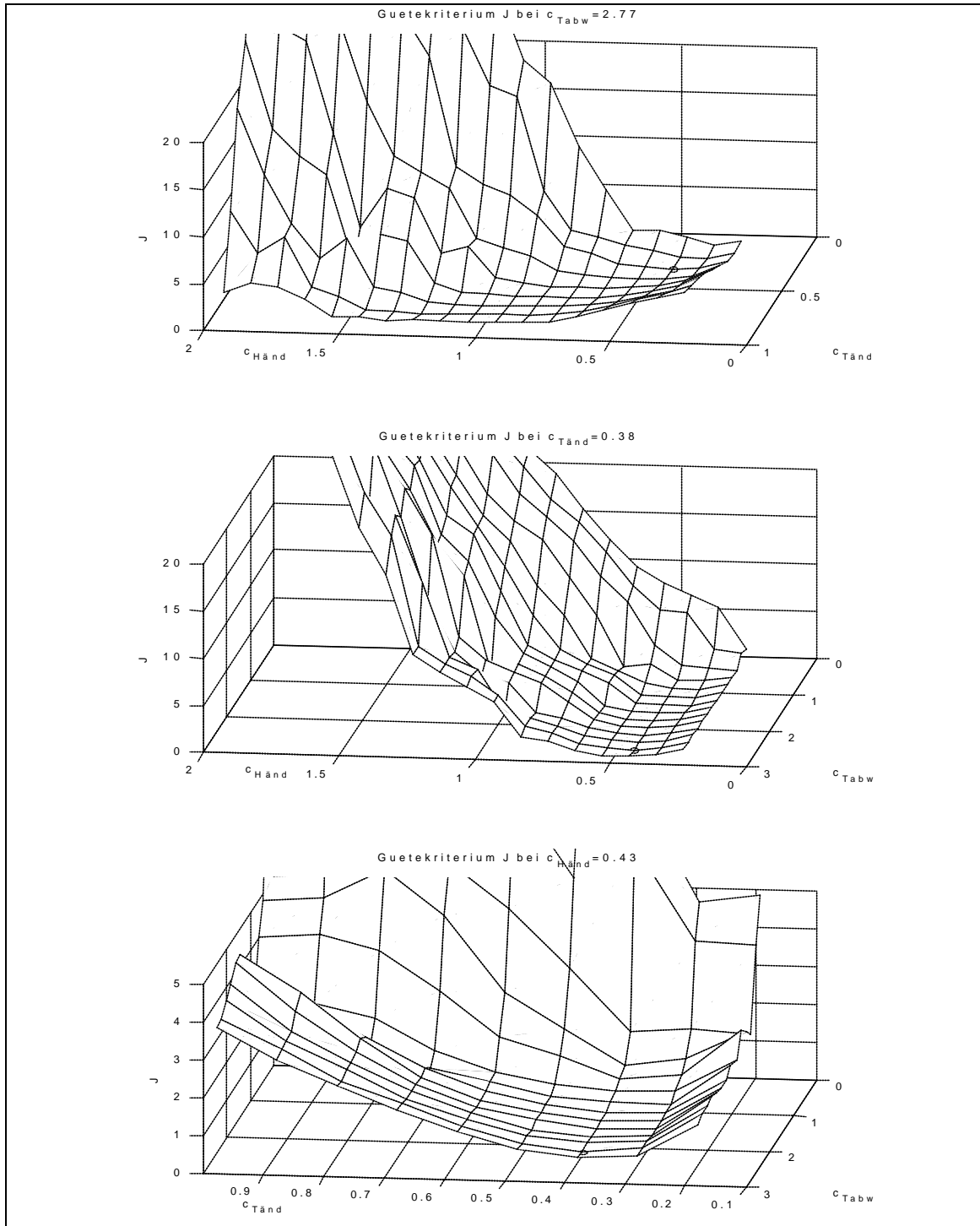


Abb. 32 Gütekriterium J in der Umgebung der optimalen Skalierungsfaktoren, Simulation der Reaktorbefüllung mit 6%iger CMC-Lösung

werden in 0.10-Schritten innerhalb des ursprünglichen Wertebereichs (Tab. 8 und Tab. 10) verändert. Der beste gefundene Parametersatz ist durch einen zusätzlichen Kreis gekennzeichnet. Eine Schrittweite von 0.10 wird gewählt, um eine klarere Darstellung zu erzielen. Diese Form der Darstellung täuscht aber einen glatten Verlauf des Gütekriteriums J vor, der bei der Verwendung kleinerer Schrittweiten (0.01) nicht besteht. Ein gradientenbasiertes Verfahren zur Optimierung der Skalierungsfaktoren c_{Tabw} , $c_{Tänd}$ und $c_{Händ}$ wäre mit erheblichen Problemen konfrontiert. Der beste Parametersatz befindet sich meist am Fuße eines steilen Abfalls des Gütekriteriums J (Abb. 31 oben und unten). Innerhalb der hohen Werte von J bei kleinem c_{Tabw} und $c_{Tänd}$ sowie großem $c_{Händ}$ ist die Fläche von J stark zerklüftet, was ein Gradientenverfahren zusätzlich behindern würde.

Anhand der vorliegenden Modelle kann nun eine Optimierung des Reglers vorgenommen werden, die die Parameter der Zugehörigkeitsfunktionen der Ein- und Ausgangsmengen anpaßt. Das Regelverhalten wird nicht mehr nur global über die Skalierungsfaktoren angepaßt, weshalb auch das Störgrößenprogramm die (simulierte) Regelstrecke reichhaltiger anregen sollte.

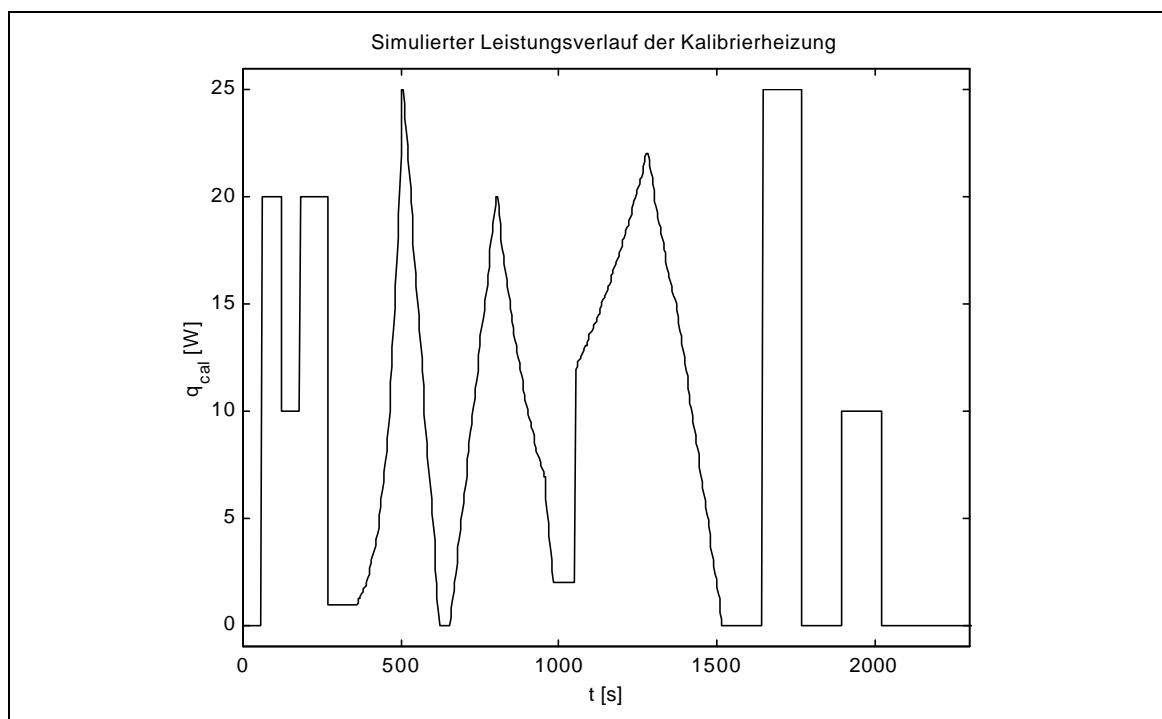


Abb. 33 Erweitertes Störgrößenprogramm für die Parameteroptimierung des Fuzzy-Logik-Reglers

Als simuliertes Störgrößenprogramm wird der in Abbildung 33 dargestellte Verlauf der Kalibrierheizleistung verwendet, der sich aus mehreren sprung- und rampenförmigen Funktionsverläufen zusammensetzt.

Es sind mehrere Versuche mit einem genetischen Algorithmus zur Parameterfindung durchgeführt worden. Dabei zeigt sich, daß man dem Fuzzy-Logik-Regler nicht zuviele Freiheitsgrade einräumen darf, da sonst die Interpretierbarkeit des Reglerverhaltens darunter leidet.

Es wird eine Mehrfachüberschneidung der unscharfen Mengen einer Eingangsgröße vermieden, indem lediglich die Modal- bzw. Begrenzungswerte $x_{m,Ai}$ ($y_{m,Bj}$) als zu optimierende Parameter zugelassen werden, wobei sich die Breiten a und b einer Zugehörigkeitsfunktion aus den Werten der benachbarten Terme ergibt. Die äußersten Terme der linguistischen Eingangsvariablen *Temperaturabweichung* und *Temperaturänderung* sind als seitlich offene Trapezfunktionen konzipiert, und die mittleren Terme sind durch Dreiecksfunktionen beschrieben. Den Termen ZE (*Temperaturabweichung*) und UN (*Temperaturänderung*) wird ein unveränderlicher Modalwert von Null zugeordnet. Eine Optimierung hinsichtlich der besten Zugehörigkeitsfunktion (\wedge -, Π - oder Gauß-Funktion) wird nicht verfolgt.

Ferner zeigt sich, daß eine Optimierung der Regelbasis selbst bei Beschränkung der Relevanzfaktoren auf 0 und 1 (konsistente, vollständige Regelbasis) eine derart hohe kombinatorische Vielfalt liefert, daß der genetische Algorithmus zu lange Laufzeiten benötigt, um eine logisch schlüssige Regelbasis zu finden. Sie wird daher ebenfalls als unveränderlich vorgegeben.

Im folgenden ist die Anpassung des Fuzzy-Logik-Reglers anhand des Modells für Wasser als Reaktorbefüllung beschrieben. Es wird ein Algorithmus für jede Kombination aus den drei Einstellmöglichkeiten durchlaufen:

1. unterschiedliche UND-Verknüpfung (A.15) und (A.16) der zwei Bedingungsteile,
2. unterschiedliche ODER-Verknüpfung (A.11) und (A.12 / C.5) bei der Akkumulation,
3. unterschiedliche Anzahl n , m und o der Terme der Ein- und Ausgangsvariablen (jeweils 3, 5 und 7).

Der genetische Algorithmus verfügt über eine Populationsgröße von 1000 Individuen. Die Werte der Startpopulation für $x_{m,Ai}$, $y_{m,Bj}$ und $z_{m,Ck}$ liegen innerhalb des in Tabelle 12 aufgeführten Wertebereichs.

Tab. 12: Wertebereiche für die Terme der Startpopulation

	Wertebereich	
	von	bis
<i>Temperaturabweichung</i> [K]	-0.120	+0.120
<i>Temperaturänderung</i> [K s ⁻¹]	-0.020	+0.020
<i>Heizleistungsänderung</i> [W]	-16.0	+16.0

Die Fitness einer Parameterkombination wird anhand von Gleichung (51) bestimmt. Zur Bildung der nächsten Generation wird das Rouletterad-Verfahren mit Elitisierung verwendet. Die Mutationswahrscheinlichkeit beträgt 5 Prozent. Wird ein Parameter der Mutationsoperation unterzogen, so wird ihm ein zufälliger, normalverteilter Wert addiert (Mittelwert = 0, Standardabweichung = Wertebereich/Anzahl der Terme). Die Wahrscheinlichkeit, daß ein Individuum der Kreuzungsoperation unterzogen wird, beträgt 10 Prozent. Die Anzahl zu durchlaufender Generationen wird auf 75 festgelegt.

Die besseren Ergebnisse werden erzielt, wenn die UND-Verknüpfung der beiden Bedingungsteile über die *MIN*-Operation (A.15) erfolgt. Es ist demnach von Vorteil, wenn der Entschlossenheitsgrad einer Regel jeweils vom kleinsten Erfüllungsgrad eines seiner Bedingungsteile abhängt, ohne den Erfüllungsgrad des anderen Bedingungsteils zu berücksichtigen.

Bei der Akkumulation aller Entschlossenheitsgrade, die zu ein und derselben Schlußfolgerung führen, werden bessere Ergebnisse bei Verwendung der modifizierten algebraischen Summe (C.5) erreicht. Diese Funktion faßt die unterschiedlichen Entschlossenheitsgrade einer Schlußfolgerung detaillierter zusammen als dies bei der *MAX*-Funktion (A.11) der Fall wäre, die lediglich den maximalen Entschlossenheitsgrad ausgibt.

Wesentliches Ergebnis der unterschiedlichen Durchläufe des genetischen Algorithmus ist aber, daß die beste Regelgüte erreicht wird, wenn jeweils 5 Terme für die drei linguistischen Variablen *Temperaturabweichung*, *Temperaturänderung* und *Heizleistungsänderung* verwendet werden. Bei nur drei Termen für jede linguistische Variable kann das Reglerverhalten nicht fein genug abgestimmt werden, bei sieben Termen hingegen ist die Anzahl zu optimierender Parameter zu groß, als daß der Algorithmus in 75 Generationen eine passende Parametrisierung des Fuzzy-Logik-Reglers findet.

Die für den besten Fuzzy-Logik-Regler gefundenen Zugehörigkeitsfunktionen der Eingangsvariablen *Temperaturabweichung* und *Temperaturänderung* ($n = m = 5$) sind in Abbildung 34 dargestellt. In Tabelle 13 folgen die Positionen der Ausgangssingletons der *Heizleistungsänderung* ($o = 5$). Bis auf die Terme VERRINGERESTARK und ERHÖHESTARK weist der Regler eine sehr symmetrische Struktur auf.

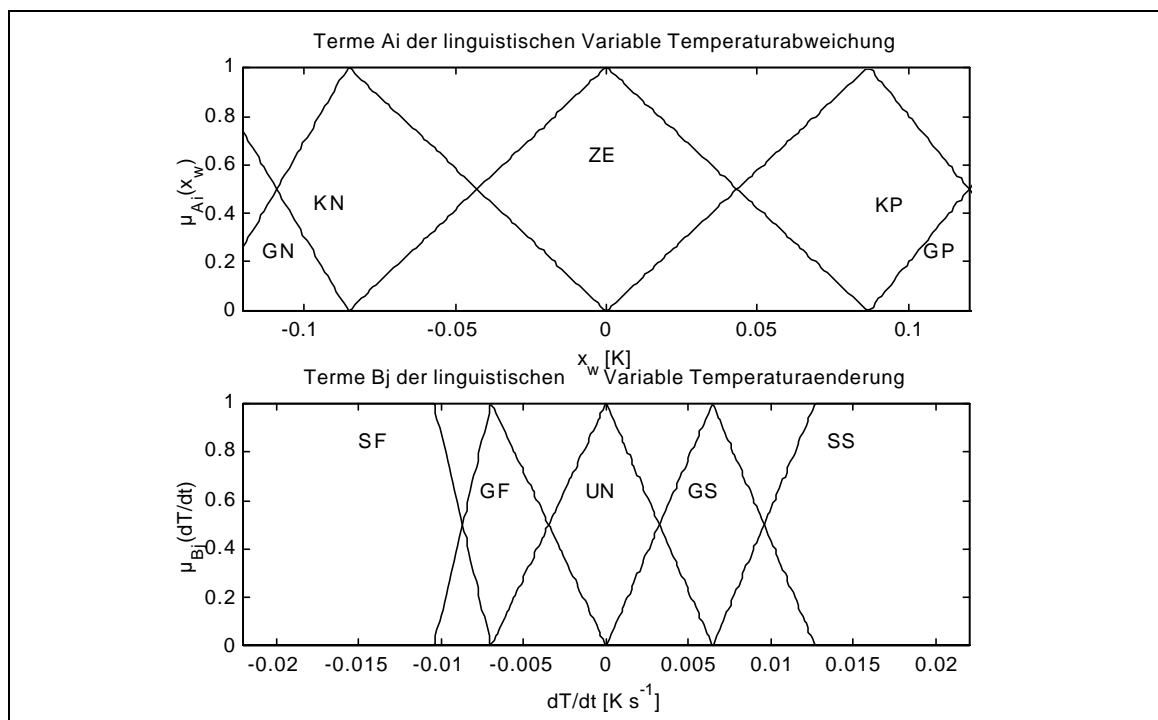


Abb. 34 Auftragung der Terme A_i und B_j der linguistischen Variablen *Temperaturabweichung* und *Temperaturänderung*

Tab. 13:

Linguistische Variable: <i>Heizleistungsänderung</i>			
	Term C_k	Abk.	z_m
$C1$	VERRINGERESTARK	VS	-21.43
$C2$	VERRINGEREGERING	VG	-9.16
$C3$	KEINEVERÄNDERUNG	KV	± 0.00
$C4$	ERHÖHEGERING	EG	+9.86
$C5$	ERHÖHESTARK	ES	+19.35

Da beide Eingangsvariable durch jeweils fünf Terme beschrieben wird, ist eine 5-5-Regelbasismatrix (Tab. 14) vorgegeben worden. Diese Matrix ist nicht symmetrisch zur Diagonale von unten rechts nach oben links, sondern

läßt die Einteilung des Reglerverhaltens in zwei Bereiche unterschiedlicher Reglerverstärkung deutlich werden. In der Mitte befindet sich der Bereich kleiner Temperaturabweichungen und kleiner Temperaturänderungen, die nur kleine Änderungen der Heizleistung benötigen (ERHÖHEGERING, VERRINGEREGERING). Um diesen Zentralbereich herum ist der Außenbereich mit großen Temperaturabweichungen und -änderungen angeordnet, welche entsprechend größere Heizleistungsänderungen nötig machen. Während die Regelung im ungestörten Betrieb sich lediglich im mittleren Bereich bewegt, wird der Außenbereich nur beim Eintritt größerer Störungen beansprucht. Diese Unterteilung in unterschiedliches Regelverhalten bei gestörtem und ungestörtem Betrieb entspricht sehr gut den Anforderungen der isothermen Reaktionskalorimetrie.

Tab. 14: Regelbasis des optimalen Fuzzy-Logik-Reglers

<i>Heizleistungs- änderung</i>		<i>Temperaturabweichung</i>				
		GN	KN	ZE	KP	GP
Temperatur- änderung	SF	ES	ES	ES	EG	KV
	GF	ES	EG	EG	KV	VG
	UN	ES	EG	KV	VG	VS
	GS	EG	KV	VG	VG	VS
	SS	KV	VG	VS	VS	VS

3.2.5 Übersetzung des Fuzzy-Logik-Reglers in ein Neuro-Fuzzy-Regler

Mit dem in Kapitel 3.2.4 optimierten Fuzzy-Logik Regler steht ein bestens an die Regelstrecke angepaßter Regler zur Verfügung, der als robuster Regler sehr gut mit kleineren Parameteränderungen der Regelstrecke zurechtkommt. Damit aber auch größere Änderungen der Streckenparameter vom Regler verarbeitet werden können, soll das robuste Regelkonzept um ein adaptives Konzept ergänzt werden. Dies ist mit einem Fuzzy-Logik-Regler nicht realisierbar, weshalb er in ein Neuro-Fuzzy-Regler übersetzt werden soll.

Ein Neuro-Fuzzy-Regler ist im Gegensatz zu einem Fuzzy-Logik-Regler ein lernfähiger (anpassungsfähiger) Regler. Diese Fähigkeit wird allerdings erkauft mit einigen Restriktionen des Neuro-Fuzzy-Reglers, die eine direkte Übersetzung des Fuzzy-Logik-Reglers verhindern (Anhang C).

Für die Übertragung in ein Neuro-Fuzzy-System wird daher der Weg gewählt, zunächst die Reglerkennfläche Δq_{comp} als Funktion der beiden Eingangsvariablen x_w und $\Delta T_r/\Delta t$ darzustellen und anschließend einen grob parametrisierten Neuro-Fuzzy-Regler an diese Kennfläche anzupassen. Auf diese Weise wird gleichzeitig die Leistungsfähigkeit des in Anhang C beschriebenen Fehlerrückführungsalgorithmus für Neuro-Fuzzy-Systeme unter Beweis gestellt.

Natürlich hätte dem Neuro-Fuzzy-Regler wie einem Fuzzy-Logik-Regler (Kap. 3.2.4) ein gewünschtes Reglerverhalten aufgeprägt werden können. Dann hätte aber nicht der Fehlerrückführungsalgorithmus zum Einsatz kommen können, der das geforderte Ein-/Ausgangsverhalten des Reglers anhand von Beispieldaten für eine Anpassung benötigt.

Die Wertebereiche der *Temperaturabweichung* und der *Temperaturänderung*, für die die Reglerkennfläche berechnet wird, sind in Tabelle 15 aufgeführt. Für diesen Bereich des Reglerkennfelds erfolgt die Anpassung des Neuro-Fuzzy-Reglers.

Tab. 15: Wertebereiche des Reglerkennfelds

	Wertebereich		
	von	bis	Schrittweite
<i>Temperaturabweichung</i> [K]	-0.1400	+0.1600	0.0050
<i>Temperaturänderung</i> [K s ⁻¹]	-0.0110	+0.0135	0.0005

In Abbildung 35 ist das Kennfeld für einen größeren Bereich dreidimensional dargestellt. Recht deutlich wird in dieser Darstellungsart die unterschiedliche Reglerverstärkung durch die Partitionierung der Eingangsvariablen. So bildet die Reglerkennfläche im Bereich des Ursprungs ($x_w = 0$, $\Delta T_r/\Delta t = 0$) eine Fläche mit kleinen Gradienten. Je weiter man sich vom Ursprung entfernt, desto steiler wird die Kennfläche (größere Reglerverstärkungen), bis sie schließlich ihre extremen Plateaus erreicht.

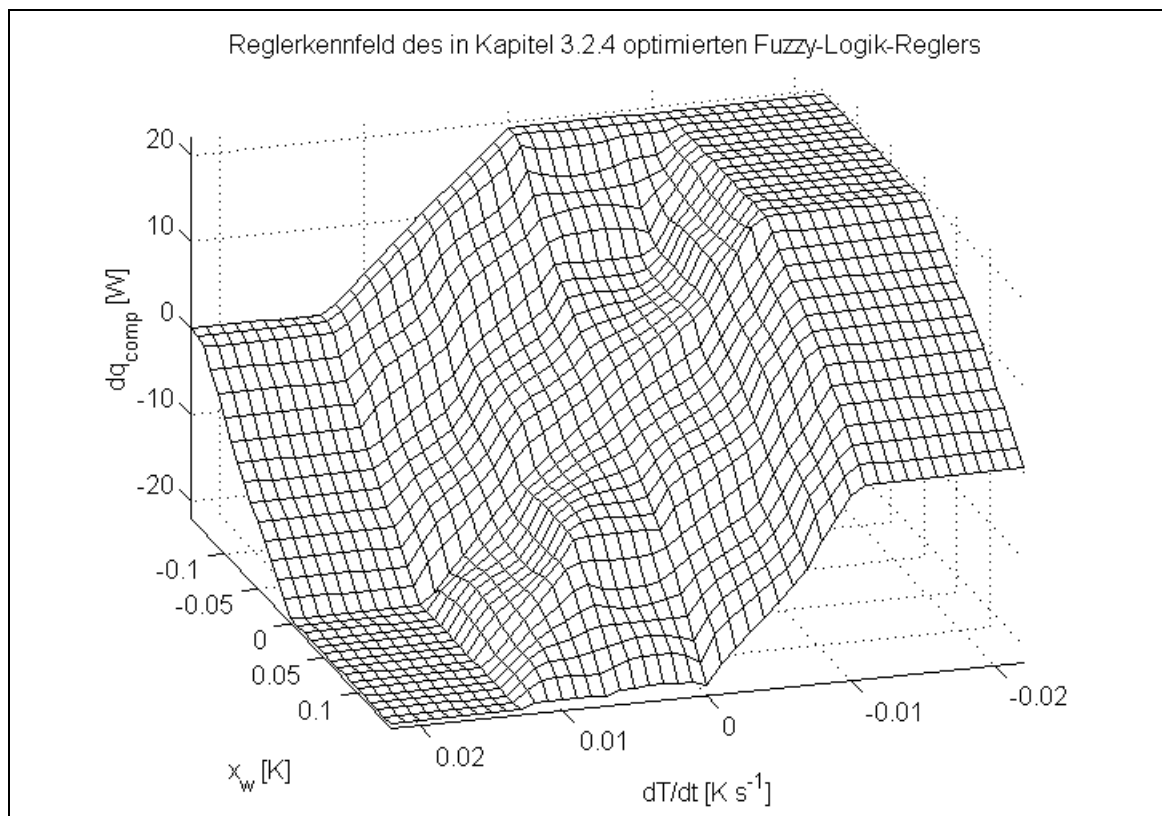


Abb. 35 Kennfeld des in Kapitel 3.2.4 optimierten Fuzzy-Logik-Reglers

Die für den Fehlerrückführungsalgorithmus verwendeten Lernraten sind in Tabelle 16 aufgeführt. Die Lernraten für die Modal- und Begrenzungswerte sowie für die Breiten der Zugehörigkeitsfunktionen sind bewußt klein gewählt, um eine möglichst genaue Anpassung der Reglerkennfläche des Neuro-Fuzzy-Reglers an die Sollkennfläche zu erreichen. Da es sich noch nicht um eine zeitkritische *Online*-Adaption handelt, ist diese Wahl gerechtfertigt. Eine Anpassung der Skalierungsfaktoren ($s_A = c_{Tabw}$, $s_B = c_{Tänd}$ und $s_C = c_{Händ}$) wird nicht vorgenommen: $h_{sA} = h_{sB} = h_{sC} = 0$. Ebenfalls nicht in die Kennfeldanpassung einbezogen sind die Relevanzfaktoren in der Regelbasis ($h_{sC} = 0$). Die Modifizierung der Relevanzfaktoren hätte zwar eine bessere

Übereinstimmung der Reglerkennflächen zur Folge gehabt, allerdings wäre das zu Lasten der einfachen Interpretierbarkeit der Wenn-Dann-Regeln gegangen. Die in Tabelle 14 aufgeführte Regelbasis wird daher unveränderlich beibehalten.

Tab. 16: Lernraten für die Anpassung des Neuro-Fuzzy-Reglers

Lernraten				
h_{x_m, A_i}	h_{a, A_i}	h_{y_m, B_j}	h_{a, B_j}	h_{z_m, C_k}
$1 \cdot 10^{-6}$	$1 \cdot 10^{-6}$	$1 \cdot 10^{-8}$	$1 \cdot 10^{-8}$	$1 \cdot 10^{-3}$

Die Reglerkennfläche des resultierenden Neuro-Fuzzy-Reglers ist in Abbildung 36 dargestellt. Unter den gewählten Bedingungen ist eine durchschnittliche Abweichung von 0.74 W (*rms-Fehler*) erreicht worden, wobei Wert darauf gelegt wurde, daß die Reglerkennfläche durch den Ursprung geht. Das Reglerkennfeld in Abb. 36 zeigt eine viel glattere Struktur als Abb. 35, was auf den geänderten Zugehörigkeitsfunktionen und Operatoren des Neuro-Fuzzy-Systems beruht.

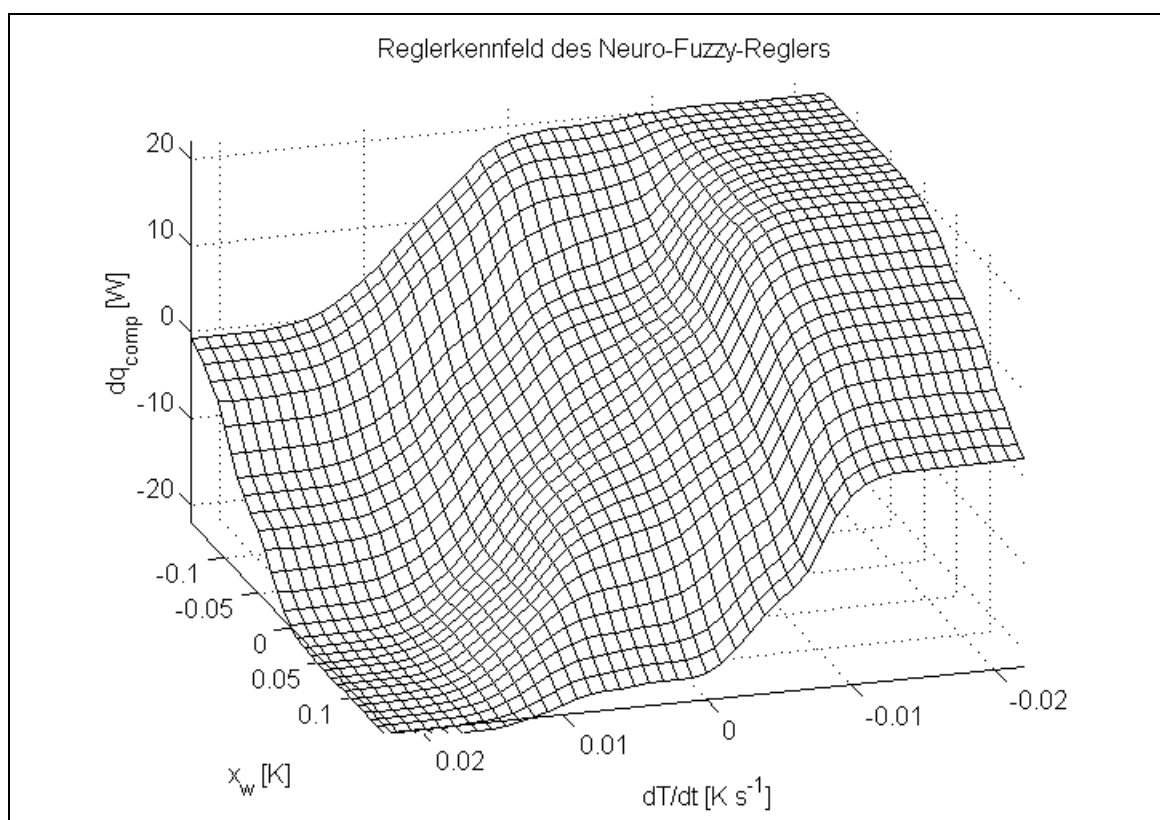


Abb. 36 Kennfeld des anhand des Kennfelds in Abb. 35 trainierten Neuro-Fuzzy-Reglers

Der Verlauf der Zugehörigkeitsfunktionen der einzelnen Terme ist in Abbildung 37 aufgeführt. Die Position der Ausgangssingletons findet sich in Tabelle 17. Die Bedingung, daß für einen Wert x^* (y^*) nur maximal zwei Terme einen Zugehörigkeitsgrad größer Null aufweisen, ist nicht mehr erfüllt, was die sprachliche Interpretierbarkeit des Neuro-Fuzzy-Systems trotz konstanter Regelbasismatrix (Tab. 14) erschwert. Dieselben Probleme bereitet die deutliche Verschiebung der Terme NULL (ZE), UNVERÄNDERT (UN) und KEINEVERÄNDERUNG (KV) in den negativen Bereich der Basisvariable. Diese Verschiebung macht es schwer verständlich, daß der Neuro-Fuzzy-Regler für $x_w = 0$ und $\Delta T_r / \Delta t = 0$ wirklich eine Heizleistungsänderung von $\Delta q_{comp} = 0$ ausgibt.

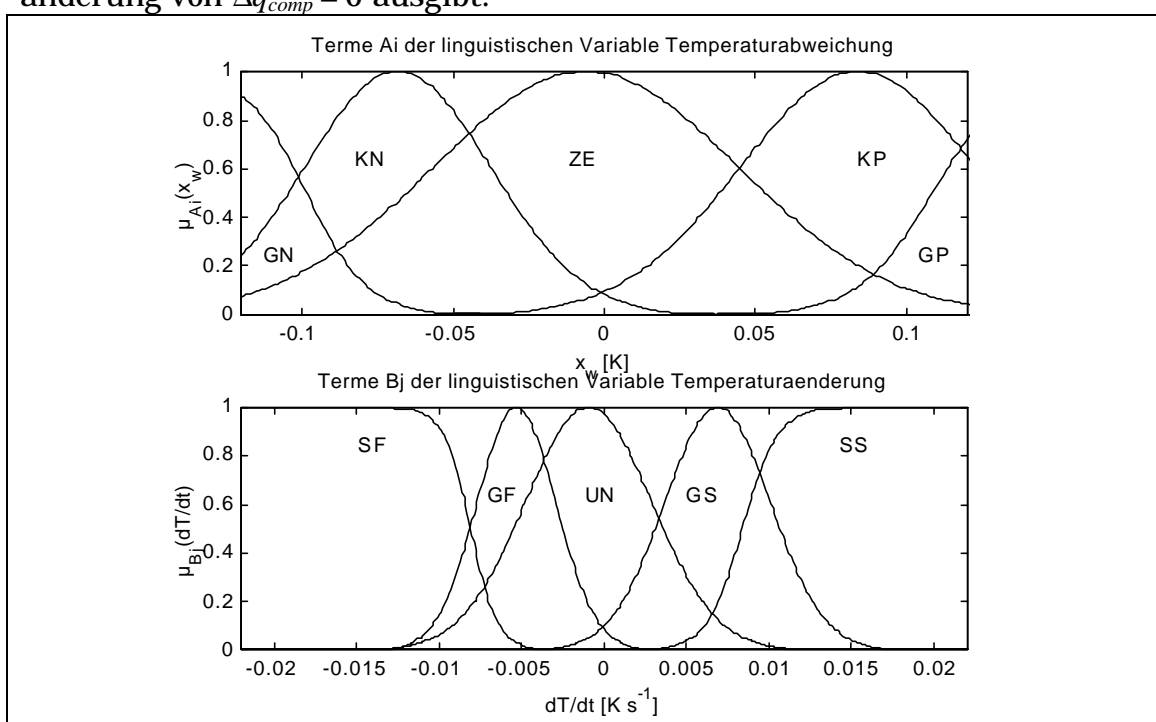


Abb. 37 Auftragung der Terme A_i und B_j der linguistischen Variablen *Temperaturabweichung* und *Temperaturänderung*

Tab. 17:

Linguistische Variable: <i>Heizleistungsänderung</i>			
Term C_k		Abk.	z_m
$C1$	VERRINGERESTARK	VS	-22.91
$C2$	VERRINGEREGERING	VG	-8.34
$C3$	KEINEVERÄNDERUNG	KV	-0.10
$C4$	ERHÖHEGERING	EG	+10.25
$C5$	ERHÖHESTARK	ES	+19.98

3.2.6 Das modelladaptive Regelungskonzept

Ziel des modelladaptiven Regelungskonzepts ist die kontinuierliche Verbesserung des Regelverhaltens im geschlossenen Regelkreis. Es kann nicht für jeden Arbeitspunkt mit den in den vorangegangenen Kapiteln behandelten Methoden ein optimaler Neuro-Fuzzy-Regler bestimmt und eingesetzt werden. Vielmehr kommt es bei der Temperaturregelung eines Reaktors im Satzbetrieb durch Änderungen der Durchmischungs- und Wärmeübergangsverhältnisse sowie des Füllstands zu Streckenparameteränderungen (Totzeit, Verstärkung, Ordnung), die eine dauernde Regleradaption nötig machen.

Die Regelung der Reaktortemperatur wird von einem Neuro-Fuzzy-Regler übernommen, der anhand des vorhandenen Prozeßwissens recht grob entworfen oder wie in Kapitel 3.2.5 beschrieben angepaßt wird. Der Regler errechnet unabhängig von den im folgenden beschriebenen Schritten anhand der Regelgrößenabweichung x_w und der zeitlichen Temperaturänderung $\Delta T_r/\Delta t$ die Stellgrößenänderung Δq_{comp} , die sofort eingestellt wird. Der Fehlerrückführalgorithmus (Anhang C) für das Neuro-Fuzzy-System macht diesen Regler zu einem adaptionfähigen System. Damit aber eine Adaption stattfinden kann, die zu einem besseren Regelverhalten führt, wird eine experimentelle Modellbildung der Regelstrecke mittels eines neuronalen Netzes vorgenommen.

Grundlage einer Systemidentifikation ist die Gewinnung von aussagekräftigen Daten, die bei einer Anregung der Regelstrecke entstehen. Bei der Anregung der Regelstrecke tritt bei der leistungskompensierten Reaktionskalorimetrie die Beschränkung auf, daß eine externe Anregung (Führungsgrößenänderung, zusätzliche Störgröße) ausscheidet, da eine Messung zur Gewinnung kalorimetrischer Daten streng-isotherm zu erfolgen hat. Die erhaltenen Daten bewegen sich für die Reaktortemperatur T_r hauptsächlich in einem Bereich von ± 0.020 K um den Sollwert und für die Kompensationsheizleistung in einer Schwankungsbreite von 4 W.

Bei einer Anzahl von $u = 5$ verarbeitenden Neuronen in der verborgenen Schicht und von $N_V = 7$ Vergangenheitswerten für die Ein- und Ausgangsvariable ($q_{comp}(t)$, $T_r(t)$) wird ein Zeitraum von 1 Sekunde festgelegt, innerhalb dessen die Parameteranpassung des Modells ausgeführt werden kann (Netztraining). Abhängig von der Größe des Trainingsdatensatzes N_T ergibt sich bei der Anpassung des neuronalen Netzes eine unterschiedliche

Anzahl Epochen, die durchlaufen werden. Der Zusammenhang zwischen N_T und der Epochenanzahl ist in Tabelle 18 dargestellt.

Tab. 18: Anzahl Epochen bei unterschiedlichem Umfang des Trainingsdatensatzes

N_T	Anzahl Epochen	N_T	Anzahl Epochen
5	385	20	101
10	198	25	81
15	133	30	68

Wird der Umfang des Trainingsdatensatzes N_T groß gewählt, so wird verhindert, daß das neuronale Netz vorhandenes Meßrauschen mitlernt, da eine Mittelung über mehrere Meßwerte erfolgt. Nachteil eines großen N_T ist das Vorhandensein von alten Meßwerten im Trainingsdatensatz, die u.U. das Regelstreckenverhalten nicht mehr ausreichend repräsentieren, und die geringe Anzahl Epochen, die für eine Anpassung des Netzes an den Datensatz zur Verfügung stehen.

Um das Anpassungsproblem des *Overfitting* (Anhang B) zu vermeiden, wird die Forderung aufgestellt, daß die Größe des Trainingsdatensatzes N_T größer sein sollte als die Anzahl der Netzgewichte des neuronalen Netzes. Bezieht man diese Forderung auf das verwendete Netz ($N_V = 7$, $u = 5$), so müßte der Datensatz mindestens 76 Ein-/Ausgabepaare $(x_h(t), y_k(t))_l$ enthalten. Bei einer Abtastrate Δt_a von 3 Sekunden würde dies bedeuten, daß Meßwerte berücksichtigt werden, die über 4 Minuten alt sind.

Um mit einem Trainingsdatensatz geringeren Umfangs auszukommen, wird das neuronal Netz bereits vorher anhand von Meßdaten aus einem Führungs- und Störgrößenprogramm (Kap. 3.2.3) trainiert, so daß bereits ein gutes Startmodell der Regelstrecke besteht. Bei der Identifikation parallel zur Regelstrecke wird dann ein Trainingsdatensatz von $N_T = 20$ verwendet. Die ältesten Daten sind dann 1.3 Minuten alt, was zu einem stets aktuellen Modell der Regelstrecke führt. Bei einem solch kleinen Datensatz wird auf eine Aufteilung in Trainingsdatensatz und Testdatensatz verzichtet. Für diese Vorgehensweise spricht auch der Umstand, daß der Datensatz aus Ein-/Ausgangspaaren unterschiedlichen Alters besteht, was eine Qualitätsbeurteilung an einem zufällig ausgewählten Trainingsdatensatz fraglich erscheinen läßt.

Kommt es zu einer schnellen Änderung der Regelstreckenparameter z.B. infolge einer schnellen Dosierung, so treten recht hohe Modellfehler auf, weil im Trainingsdatensatz Meßwerte enthalten sind, die das Regelstreckenverhalten vor und nach der Parameteränderung beschreiben. Hier wird ein Modellschutz eingeführt, der das alte Modell solange als gültig annimmt, bis im Trainingsdatensatz wieder in sich schlüssige Daten enthalten sind und das neue Modell einen *rms*-Fehler unter 0.010 K aufweist. Es ist sinnvoller, den Regler an das alte Model adaptiert zu lassen und ein suboptimales Regelverhalten hinzunehmen, als den Regler an ein neues Modell anzupassen, das eine geringere Güte besitzt.

Der Adaptionalgorithmus gliedert sich in die folgenden fünf Teilschritte, die nacheinander abgearbeitet werden:

1. Berechnung der Korrektur der Heizleistungsänderung $\Delta\Delta q_{comp}$
2. Berechnung des Gütekriteriums J_{vor} vor der Adaption
3. Rückführung von $\Delta\Delta q_{comp}$ zur Anpassung des Neuro-Fuzzy-Reglers
4. Berechnung des Gütekriteriums J_{nach} nach erfolgter Adaption
5. Entscheidung, ob die Regleranpassung beibehalten wird ($J_{nach} < J_{vor}$)

Die Berechnung des Korrekturwertes $\Delta\Delta q_{comp}$ (55) erfolgt in Anlehnung an Gleichung (46, Kap. 3.1.3). Allerdings wird die Ableitung der Reaktortemperatur T_r' nach der Heizleistung q_{comp} nicht vereinfachend wie in (41, Kap. 3.1.2) approximiert. Dies hätte zur Folge, daß die Ableitung nur auf der letzten Stellgrößenänderung basiert, wobei eine mögliche Totzeit t_D der Regelstrecke unberücksichtigt bleibt und sich ein falscher Korrekturwert $\Delta\Delta q_{comp}$ ergibt.

$$\Delta\Delta q_{comp} = \frac{T_r(t) - T_{r,soll}}{T_r'(q_{comp})} = (T_r(t) - T_{r,soll}) \cdot \frac{\Delta q_{comp}}{\Delta o_k(q_{comp})} \quad (55)$$

Die Ableitung T_r' wird durch $\Delta o_k / \Delta q_{comp}$ angenähert, indem die Ausgabewerte des Modells $o_{k,-0.5}$ und $o_{k,+0.5}$ für zwei Eingabevektoren $x_{h,-0.5}$ und $x_{h,+0.5}$ (56) gebildet werden. Beim Vektor $x_{h,-0.5}$ werden alle eingehenden Werte von q_{comp} um 0.5 W verringert, entsprechend werden beim Vektor alle q_{comp} -Werte um 0.5 W erhöht. Auf diese Weise wird die Ableitung von T_r unabhängig von einer möglichen Totzeit gebildet. Zusätzlich ergibt sich der rechnerische Vorteil, daß in (57) $\Delta q_{comp} = 1$ wird.

$$x_h = (q_{comp}(t), q_{comp}(t-1), \dots, q_{comp}(t-6), T_r(t), T_r(t-1), \dots, T_r(t-6)) \quad (56)$$

$$T_r'(q_{comp}) = \frac{o_{k,+0.5}(x_{h,+0.5}) - o_{k,-0.5}(x_{h,-0.5})}{\Delta q_{comp}} \quad (57)$$

In Abbildung 38 ist der zeitlicher Verlauf von $\Delta\Delta q_{comp}$ bei sich sprunghaft änderndem Sollwert $T_{r,soll}$ dargestellt (eine Adaption des Reglers findet noch nicht statt). Bei 300 s wird der Sollwert von 80.0 °C auf 80.3 °C erhöht. Dieser Sollwertänderung versucht der Regler zu folgen; allerdings benötigt er ca. 50 s bis der neue Sollwert erreicht ist (Abb. 38 oben). Gemäß Gleichung (55) wird eine Korrektur der Heizleistungsänderung $\Delta\Delta q_{comp}$ berechnet, die eine größere positive Heizleistung zur Folge hätte, wodurch der Regler den neuen Sollwert schneller erreichen könnte. Die hierbei auftretenden Korrekturwerte von über ± 50 W bei der Sollwertänderung scheinen jedoch viel zu groß, als daß sie durch den Neuro-Fuzzy-Regler zurückgeführt werden sollten. Es wird daher ein Begrenzer eingeführt, der einen maximalen Korrekturwert von ± 10 W zuläßt. Es soll so gewährleistet werden, daß sich der Neuro-Fuzzy-Regler iterativ an ein neues Regelverhalten anpaßt.

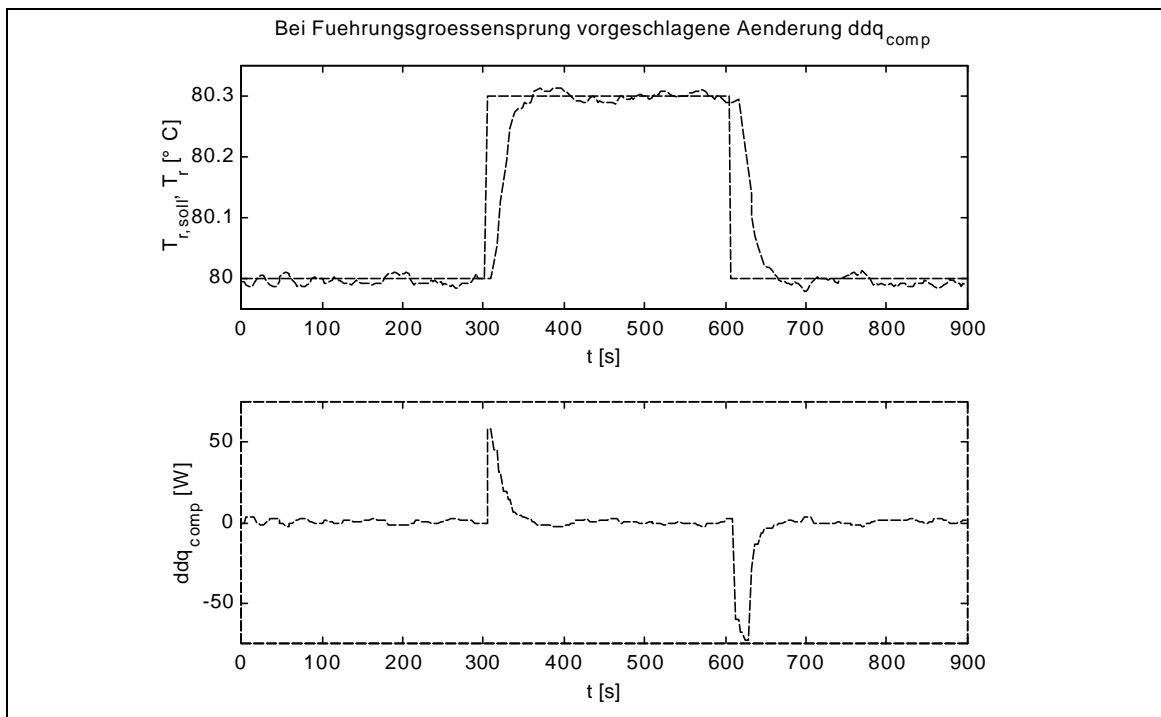


Abb. 38 Berechnete Änderung der Heizleistungsänderung $\Delta\Delta q_{comp}$

Bevor auf die Bildung der Gütekriterien J_{vor} und J_{nach} eingegangen wird, soll zunächst die Rückführung der Korrektur der Heizleistungsänderung durch den Neuro-Fuzzy-Regler behandelt werden. Hierbei taucht ebenfalls das

Problem einer unbekanntem Totzeit t_D der Regelstrecke auf, so daß bei unterschiedlichen Reaktorbefüllungen nicht von vornherein bekannt ist, welcher zeitlich zurückliegende Regeleingriff Δq_{comp} für die aktuelle Regelabweichung x_w verantwortlich ist. Da jedoch für die Aktualisierung des Trainingsdatensatzes die $N_V = 7$ vergangenen Werte von T_r bekannt sind, können die drei letzten Ausgangswerte $\Delta q_{comp}(t-1)$, $\Delta q_{comp}(t-3)$ und $\Delta q_{comp}(t-5)$ des Neuro-Fuzzy-Regler erneut ermittelt werden. Bei der *sample and hold*-Methode nach Shinsky (Kap. 3.1.1) haben in den vergangenen 7 Abtastperioden diese 3 Regeleingriffe stattgefunden. Versehen mit einem zusätzlichen Gewichtungsfaktor wird $\Delta \Delta q_{comp}$ für die drei zurückliegenden Eingangskombinationen von Regelabweichung und Temperaturänderung durch den Neuro-Fuzzy-Regler zurückgeführt. Für eine Reaktorbefüllung haben sich die Gewichtungsfaktoren 0.3 für $\Delta q_{comp}(t-1)$, 0.5 für $\Delta q_{comp}(t-3)$ und 0.2 für $\Delta q_{comp}(t-5)$ als erfolgreich bewiesen. Berücksichtigt man die Totzeit $t_D = 6$ s, die bei einer Reaktorbefüllung mit Toluol auftritt, so ist der erfolgreiche Einsatz dieser Gewichtungsfaktoren nicht verwunderlich. Zum Zeitpunkt t liegt die letzte Stellgrößenänderung $\Delta q_{comp}(t-1)$ noch nicht lange genug zurück, um für die aktuelle Regelabweichung x_w voll zum Tragen zu kommen, daher der Faktor 0.3. Bei dem zeitlich weiter zurückliegenden Regeleingriff $\Delta q_{comp}(t-3)$ kann von einem deutlichen Einfluß auf die Regelabweichung x_w ausgegangen werden, weshalb der größere Gewichtungsfaktor 0.5 verwendet wird. Insgesamt sollte sich dieses Konzept aber auch auf unterschiedliche Reaktionsmischung anwenden lassen. Die Parameteränderung am Neuro-Fuzzy-Regler können lediglich als Vorschläge für eine Verbesserung des Reglers angesehen werden. Ob die Parameteränderungen beibehalten werden, wird durch eine Simulation des geschlossenen Regelkreises und der Ermittlung der Gütekriterien J_{vor} und J_{nach} bestimmt.

Bei der Wahl der Lernraten des Fehlerrückführungsalgorithmus kann man ebenfalls Werte einstellen, die durch Versuch und Irrtum ermittelt worden sind (Tab. 19). Grundsätzlich sollten die Position der Ausgangssingletons stärker bei einer Regleranpassung variiert werden als die Parameter der Eingangsmengen [67]. Auf diese Weise bleibt die Partitionierung der Eingangsvariablen weitgehend erhalten, und das Reglerkennfeld wird dennoch ausreichend angepaßt. Ob die Anpassung erfolgreich war, entscheidet sich allein über die Veränderung der Regelgüte.

Tab. 19: Lernraten für die *Online*-Anpassung des Neuro-Fuzzy-Reglers

Lernraten				
$h_{xm,Ai}$	h_{aAi}	$h_{ym,Bj}$	h_{aBj}	$h_{zm,Ck}$
$1 \cdot 10^{-5}$	$1 \cdot 10^{-5}$	$1 \cdot 10^{-7}$	$1 \cdot 10^{-7}$	$1 \cdot 10^{-2}$

Für die Bestimmung der Regelgüte wird das durch die Systemidentifikation ermittelte Modell als gültig angenommen und mit dem zu bewertenden Neuro-Fuzzy-Regler in einem simulierten Regelkreis zusammengeschlossen. Ausgehend von der aktuellen Temperatur T_r wird das zukünftige Regelverhalten am Modell vorhergesagt und das Gütekriterium J (42) für einen gewissen Prädiktionszeitraum berechnet. Um bei verschiedenen Reaktionsmedien die Stabilität des Regelkreises zu gewährleisten, wird $k = 3 \cdot 10^3$ gewählt. Zu große Stellgrößenänderungen Δq_{comp} werden sanktioniert, und ein eher ruhiger Regler wird bevorzugt (kein reiner Minimum Varianz Regler). Andere Gütekriterien (Überschwingweite, An- und Ausregelzeit) kommen bei den streng-isothermen Bedingungen der leistungskompensierten Reaktionskalorimetrie nicht in Betracht.

Der zeitliche Horizont des Prädiktionszeitraums wird auf $N_F = 30$ Zeitschritte festgelegt (entsprechend 90 s). Diese Länge hatte sich in Versuchen als ausreichend herausgestellt. Wird der Prädiktionszeitraum zu kurz gewählt, tritt möglicherweise eine Verbesserung der Reglerparameter nicht hervor. Wird N_F zu groß gewählt, so ergibt sich möglicherweise ein zeitlicher Rechenaufwand, der länger ist als die Abtastperiode von $\Delta t_a = 3$ s.

Betrachtet man den Prädiktionszeitraum N_F und den Umfang des Trainingsdatensatzes N_T , so drängt sich die Frage auf, ob das Modell Vorhersagen weiter in die Zukunft treffen darf (N_F), als es über Informationen über die Vergangenheit (N_V) verfügt. Hier seien die in den Netzgewichten des neuronalen Netzes gespeicherten Informationen genannt, so daß ein Datenpaar immer noch im Netz gespeichert sein kann, obwohl es schon im Trainingsdatensatz durch aktuellere Meßwerte ersetzt worden ist.

In Abbildung 39 (oben) ist ein Beispiel für eine erfolgreiche Adaption des Neuro-Fuzzy-Reglers dargestellt. Die Simulation ergibt nach erfolgter Adaption (durchgezogene Linie) mit $J_{nach} = 3.930 \cdot 10^{-2}$ ein etwas besseres Regelverhalten als vorher $J_{vor} = 4.098 \cdot 10^{-2}$. Das Gegenbeispiel einer zurückgewiesenen Adaption zeigt Abbildung 39 (unten). Es ist zu erkennen, daß die Regleranpassung dazu geführt hat, daß der Sollwert der Reaktortemperatur nicht erreicht, sondern unterschritten wird (durchgezogene Linie).

In diesem Fall bleiben die alten Reglerparameter eingestellt (gestrichelte Linie), die das bessere Regelverhalten zeigen.

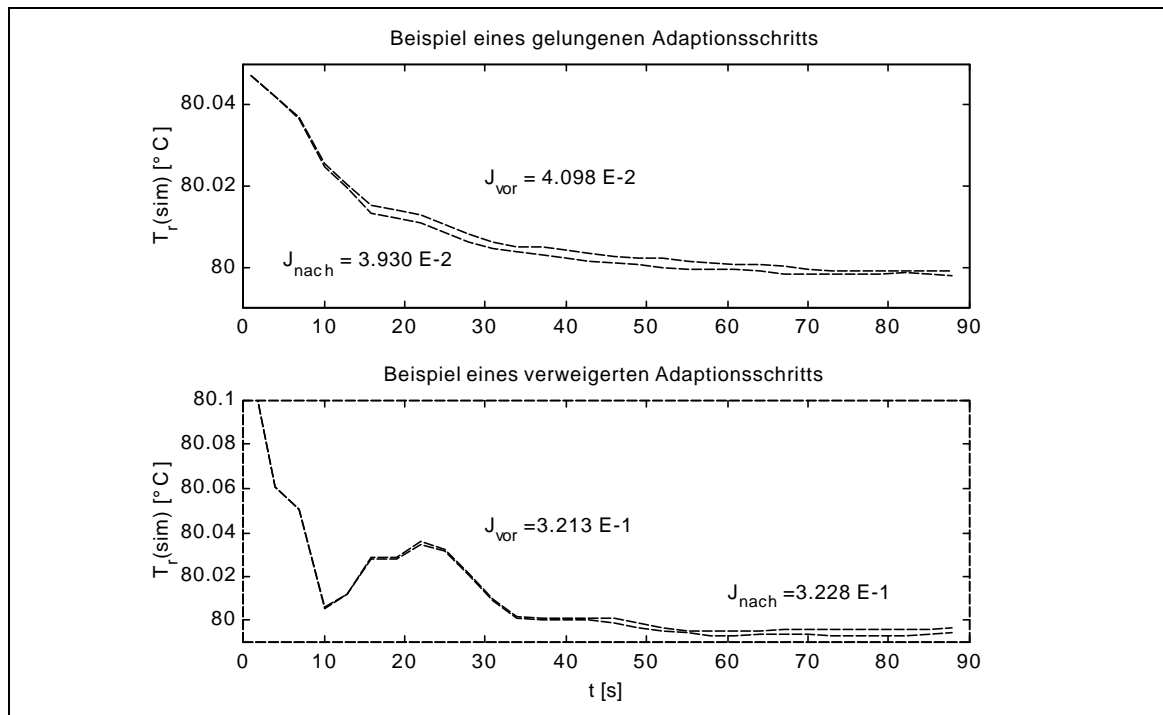


Abb. 39 Beispiel für einen akzeptierten und einen zurückgewiesenen Adaptionsschritt ($N_F = 30$)

Die Adaption des Reglers soll beispielhaft an zwei besonderen Fällen unzureichender Regelgüte beschrieben werden. Ist die Regelung zu langsam, so daß sich die Reaktortemperatur T_r längere Zeit vom Sollwert entfernt aufhält, so kommt es zu mehreren sich akkumulierenden Adaptionsschritten, die den Regler zu einer größeren Stellgrößenänderung Δq_{comp} für diese Situation bringen als ursprünglich. Der Neuro-Fuzzy-Regler wird in der gleichen Situation (*Temperaturabweichung, Temperaturänderung*) ein nächstes Mal mit einer größeren Heizleistungsänderung Δq_{comp} reagieren.

Der gegenläufige Fall zeigt sich, wenn der Regler so große Stellgrößenänderungen vornimmt, daß der Regelkreis instabil wird und die Regelgröße T_r permanent um den Sollwert schwingt. In dieser Situation lernt der Neuro-Fuzzy-Regler hauptsächlich, wenn die Reaktortemperatur T_r die Solltemperatur $T_{r,\text{soll}}$ schneidet. Da der Adaptionalgorithmus den Einfluß zeitlich zurückliegender Regeleingriffe berücksichtigt, wird beim Sollwertdurchgang ein Regeleingriff als zu stark bewertet und das Ausmaß der Stellgrößenänderung verringert. Hier akkumuliert sich die Adaption nach mehreren Schwingungen. Die Reglereingriffe werden sacher.

4 Messungen und Diskussion

4.1 Mischzeitexperimente

Es werden die Mischzeiten q für drei unterschiedliche Rührerformen gemessen. Als Rührer kommen ein Schrägblattrührer ($d = 0.050$ m), ein Ankerrührer ($d = 0.076$ m) und derselbe Ankerrührer mit aufgesetztem Schrägblattrührer zum Einsatz. Als Modellflüssigkeiten werden technisches Glycerin oder eine 85%ige wässrige Lösung des Glycerins als vorgelegte Flüssigkeit verwendet.

Das Reaktionsgefäß wird während der gesamten Versuchsdauer auf 25 °C temperiert. Der Rührer bleibt während der Vorbereitung ausgeschaltet. Es werden 566 g ($450,6$ mL) des technischen Glycerins (bzw. 548 g der 85%igen Glycerinlösung) in den Reaktor gefüllt. Darüber werden mit der Dosierpumpe bei geringem Dosierstrom 100 g Wasser geschichtet, so daß sich eine Phasengrenze zwischen den beiden Flüssigkeiten deutlich sichtbar ausbildet. Mit Einschalten des Rührers, dessen Drehzahl bereits vor Befüllung des Reaktors auf den gewünschten Wert eingestellt worden ist, beginnt der Versuch. Durch den Glasreaktor läßt sich das Mischen der beiden Flüssigkeiten gut beobachten, wobei die Schlieren deutlich hervortreten. Die Mischzeit q entspricht der Zeit, die vom Einschalten des Rührers bis zum Verschwinden der Schlieren vergeht.

Die Drehzahl N des Rührers wird mit einem mechanischen Drehzahlmesser bestimmt. Da es sich bei den verwendeten Flüssigkeiten nicht um aufgereinigte Chemikalien handelt, werden die Dichte r mit einem Pyknometer und die Viskosität h mit einem Rotationsviskosimeter experimentell bestimmt. Sie sind in Tabelle 20 angegeben.

Tab. 20: Dichte und dynamische Viskosität der Hauptkomponente und Mischung

	Dichte r [Kg m ⁻³]	Dynamische Viskosität h [Pa s]
Glycerin (techn.)	1256	$663.0 \cdot 10^{-3}$
85%ige Glycerinlsg.	1216	$66.26 \cdot 10^{-3}$
72%ige Glycerinlsg.	1181	$19.36 \cdot 10^{-3}$

Um die Reynolds-Zahl über einen großen Bereich zu variieren, werden die Rührerdrehzahlen N beim Ankerrührer und beim Ankerrührer mit aufge-

setztem Schrägblattrührer von 170 min^{-1} bis ca. 500 min^{-1} verändert. Beim Schrägblattrührer wird wegen des geringeren Rührerdurchmessers der Drehzahlbereich von 250 min^{-1} bis 850 min^{-1} abgedeckt. Der Maximalwert von 850 min^{-1} ist hierbei die obere Grenze des Rührsystems. Die gemessenen Mischzeiten sind in Tabelle 21 dargestellt. Die angegebenen Reynolds-Zahlen sind mit den physikalischen Eigenschaften der entstehenden Mischung berechnet worden.

Tab. 21: Gemessene Mischzeiten

Schrägblattrührer			Ankerrührer			Ankerrührer mit aufges. Schrägblattrührer		
N [s ⁻¹]	Re	q [s]	N [s ⁻¹]	Re	q [s]	N [s ⁻¹]	Re	q [s]
Hauptkomponente: techn. Glycerin								
6.70	307	2678	3.22	341	277.44	3.18	337	208.88
7.33	337	1796	3.73	396	181.72	3.48	369	67.03
9.17	421	1068	4.10	435	102.69	3.95	419	48.88
10.83	497	718.4	4.58	486	57.28	4.28	454	37.56
12.50	574	358.0	4.98	528	44.44	4.55	482	34.03
14.17	650	211.7	5.93	629	32.50	5.53	587	22.22
			6.73	714	27.47	5.88	624	19.53
			7.37	781	23.40	6.51	691	15.47
						7.05	747	12.16
						8.23	873	9.84
Hauptkomponente: 85%ige wässrige Glycerinlösung								
4.15	633	1762	2.85	1004	182.84	2.85	1004	107.97
5.83	890	795.3	3.17	1158	109.90	3.18	1122	76.44
6.70	1022	495.4	3.70	1304	76.22	3.48	1227	45.25
7.55	1151	295.4	4.21	1486	62.53	3.93	1386	34.72
9.17	1398	84.44	4.50	1586	47.60	4.27	1503	20.44
10.83	1652	36.22	4.88	1721	37.87	4.50	1586	14.32
12.50	1906	20.78	5.97	2102	28.06	5.43	1714	11.28
14.17	2161	16.88	6.83	2408	22.31	5.88	2073	9.09

Die Auftragung der Mischzeit-Kennzahlen $N \times q$ gegen die entsprechenden Reynolds-Zahlen Re wird in den Abbildungen 40 und 41 vorgenommen.

Für Schrägblattrührer wird eine Entfernung vom Behälterboden empfohlen, die einem Drittel der Flüssigkeitshöhe entspricht [54]. Aufgrund der Heizungen kann diese Empfehlung nicht eingehalten werden, so daß der Schrägblattrührer etwas tiefer auf der Rührerwelle angebracht ist. Trotz der von ihm primär verursachten axialen Strömung ist die Durchmischungswirkung an der Phasengrenzfläche sehr gering. Bei Glycerin als vorgelegter Hauptkomponente ist keine Annäherung der Mischzeit-Kennzahl an einen Minimalwert zu erkennen. Aufgrund der großen Mischzeit-Kennzahlen ($N\tau > 3000$), die für Glycerin als Hauptkomponente gemessen werden, wird auf die Eintragung in Abbildung 40 verzichtet. Bei vorgelegter 85%iger wässriger Glycerinlösung wird eine minimale Mischzeit-Kennzahl von $N\tau = 249.4$ erreicht (Abb. 41).

Eine bessere Mischzeitcharakteristik weist der Ankerrührer auf. Aufgrund seiner bis durch die Phasengrenzfläche ragenden Anker verringert sich die minimal erreichbare Mischzeit-Kennzahl auf 159.9, wobei sowohl bei Vorlage von Glycerin als auch bei der Vorlage von 85%iger Glycerinlösung dieser Wert angestrebt wird (Abb. 40 und 41). Der Ankerrührer verursacht in der Flüssigkeit nur eine tangential und radiale Strömung, so daß das Innere der Heizungsspiralen schlecht durchströmt wird. Am oberen Ende der Heizspirale verschwinden die Schlieren erst gegen Ende der Meßzeit, da lediglich die radiale Strömungskomponente, verursacht durch die Anker im unteren Behälterbereich, eine Strömung im Inneren der Heizung verursacht.

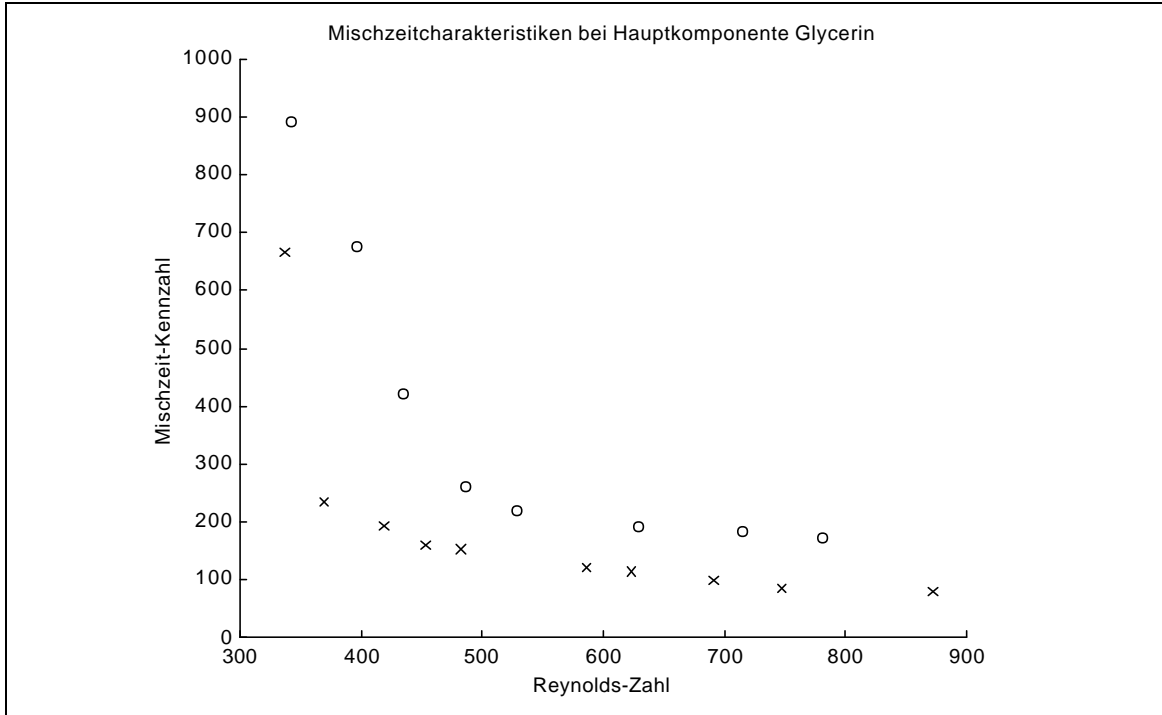


Abb. 40 Mischzeitcharakteristiken bei Hauptkomponente Glycerin (o Ankerrührer, x Ankerrührer mit aufgesetzem Schrägblattrührer)

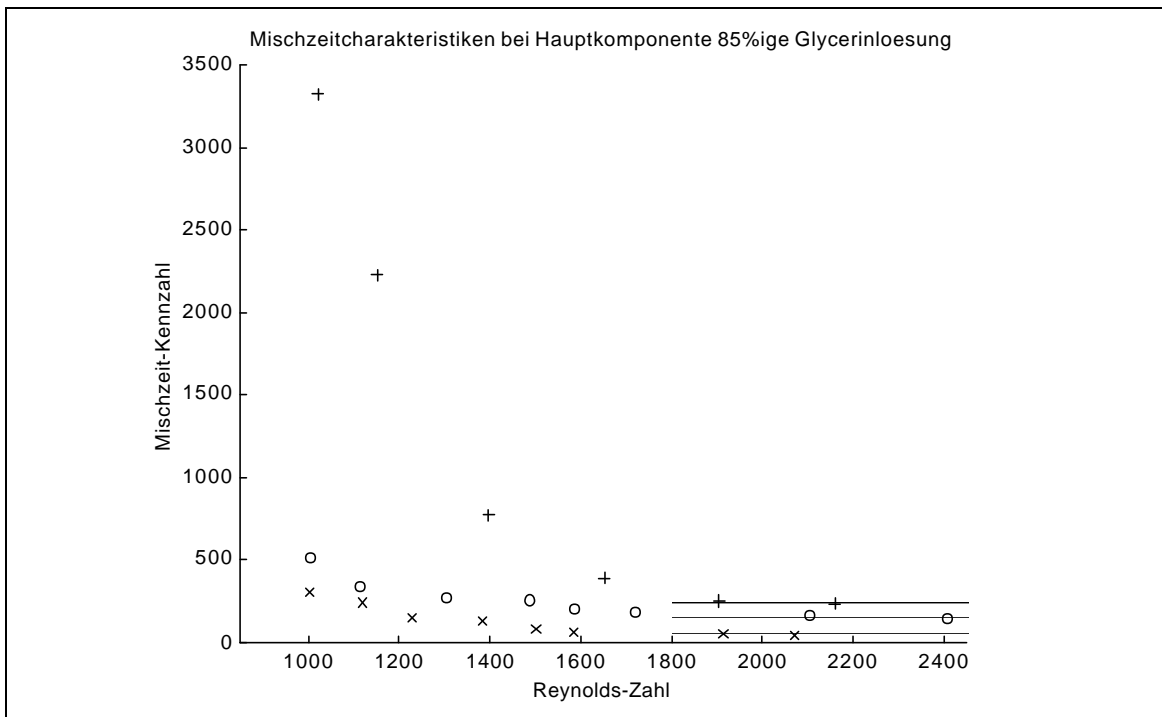


Abb. 41 Mischzeitcharakteristiken bei Hauptkomponente 85%ige Glycerinlösung (+ Schrägblattr., o Ankerr., x Ankerr. m. aufges. Schrägblattr.)

Die beste Mischzeitcharakteristik weist der Ankerrührer mit aufgesetztem Schrägblattrührer auf. Die minimale Mischzeit-Kennzahl liegt hier bei $N \cdot q = 59.7$, was nur rund einem Drittel des Wertes für den einfachen Ankerrührer und nur ca. einem Viertel des Wertes für den Schrägblattrührer entspricht (Abb. 41). Die minimale Mischzeit-Kennzahl wird ab einer Reynolds-Zahl von ungefähr 1600 erreicht. Dies entspricht einer Rührerdrehzahl von $N = 270 \text{ min}^{-1}$. Bei dieser Rührerdrehzahl wird eine Mischzeit von $q \gg 14$ Sekunden zur Homogenisierung des Glycerinlösung-Wassergemischs benötigt.

Bei den Versuchen zur Mischzeitbestimmung werden Flüssigkeiten mit einem deutlichen Dichteunterschied von $\Delta \rho = 259 \text{ Kg m}^{-3}$ bzw. 219 Kg m^{-3} im Vergleich zum darüber geschichteten Wasser verwendet. Dieser Dichteunterschied müßte in einer Kennzahlbeziehung berücksichtigt werden, da er einer Vermischung entgegenwirkt. Für eine qualitative Unterscheidung der drei Rührerformen sind die Meßergebnisse jedoch ausreichend. Hierbei zeigt sich die Überlegenheit des Ankerrührers mit aufgesetztem Schrägblattrührer gegenüber den anderen Rührerformen.

Bei der Dosierung einer Komponente, die im gleichen Lösungsmittel gelöst ist wie eine vorgelegte Komponente, ist der auftretende Dichteunterschied geringer, so daß zufriedenstellende Mischzeiten auch bei niedriger Rührerdrehzahl der Ankerrührer-Schrägblattrührer-Kombination zu erreichen sind.

4.2 Untersuchung des Wärmeübergangs mit der Wilson-Methode

Die Bestimmung der Gerätekonstanten $\Phi(T_r)$ (25) ist die Voraussetzung für die Bestimmung des Wärmeübergangskoeffizienten a_r aus der Messung des Wärmedurchgangskoeffizienten h während einer Reaktion. Da eine Lösungspolymerisation in Toluol als Lösungsmittel für die Erprobung des Reaktionskalorimeters dienen soll, wird zur Untersuchung des Wärmeübergangs nach der Wilson-Methode der Reaktor mit Toluol befüllt.

Unter Verwendung unterschiedlicher Rührerformen wird der Wärmedurchgangskoeffizient h bei Rührerdrehzahlen zwischen 150 min^{-1} und 300 min^{-1} gemessen (beim Schrägblattrührer von 175 min^{-1} bis 335 min^{-1}). Die Beschränkung auf den unteren Drehzahlbereich des Rührsystems wird vorgenommen, weil Teile der Flüssigkeit bei höheren Drehzahlen an der Behälterwand hochgeschleudert werden, so daß eine rechnerische Bestim-

mung der Wärmeaustauschfläche A nicht mehr möglich ist. Es handelt sich hierbei nicht um die Ausbildung einer Trombe; diese wird durch die Heizungen als Strömungsbrecher verhindert, sondern um Flüssigkeitselemente, die sich in unmittelbarer Nähe der Anker des Ankerrührers befinden.

In Abbildung 42 sind die Wilson-Plots für die drei unterschiedlichen Rührerformen Schrägblattrührer, Ankerrührer und Ankerrührer mit aufgesetztem Schrägblattrührer dargestellt. Das Flüssigkeitsvolumen beträgt bei diesen Versuchen $V_r = 450$ mL. Gemessen wird bei einer Reaktortemperatur von $T_r = 80$ °C und einer Manteltemperatur von $T_j = 70$ °C. Die Luftbadtemperatur entspricht der Manteltemperatur.

Deutlich zu erkennen sind die niedrigen Wärmedurchgangswerte für den Schrägblattrührer in diesem Drehzahlbereich. Die höchsten Werte für h liefert der einfache Ankerrührer. Dazwischen liegt der Ankerrührer mit aufgesetztem Schrägblattrührer. Die gleiche Reihenfolge wurde schon in Vorversuchen mit Wasser als Versuchsflüssigkeit gefunden.

Der niedrigere Wärmedurchgangswert für den Ankerrührer mit aufgesetztem Schrägblattrührer im Vergleich zum einfachen Ankerrührer kann durch eine Änderung der Strömungsverhältnisse im Reaktionsgefäß erklärt werden. Im unteren Bereich der Rührerwelle herrscht beim einfachen Ankerrührer eine radiale, auf die Behälterwand gerichtete Strömung vor, welche zu einem intensiven Wärmeaustausch führt. Beim Ankerrührer mit aufgesetztem Schrägblattrührer ist in diesem Bereich die axiale Strömung maßgeblich. Dadurch wird aber der Behälterboden angeströmt, in den der Ablaßstutzen integriert ist. Diese Fläche trägt demnach nicht zum Wärmeaustausch mit dem Mantel bei.

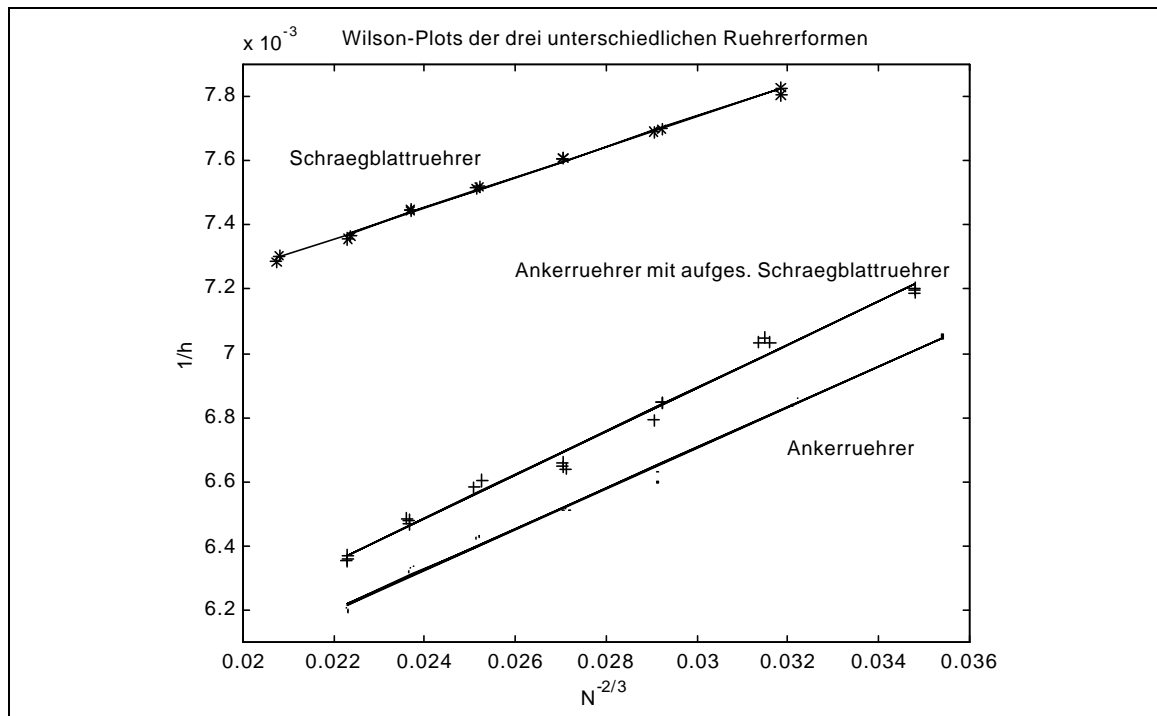


Abb. 42 Wilson-Plots bei unterschiedlichen Rührerformen ($T_r = 80 \text{ }^\circ\text{C}$, $V_r = 450 \text{ mL}$, Toluol)

In den Abbildungen 43 und 44 werden die Wilson-Plots für Reaktorbefüllungen mit $V_r = 450 \text{ mL}$ und $V_r = 550 \text{ mL}$ bei unterschiedlichen Temperaturen dargestellt. Erwartungsgemäß sind die höheren Wärmedurchgangskoeffizienten unabhängig vom Flüssigkeitsvolumen bei der höheren Temperatur $T_r = 90 \text{ }^\circ\text{C}$ zu beobachten. Recht deutlich zeigt sich die in [51] beobachtete und hier bestätigte Abhängigkeit des Wärmedurchgangskoeffizienten h vom Flüssigkeitsvolumen. Durch das zusätzliche Flüssigkeitsvolumen bei 550 mL im Vergleich zu 450 mL wird die Wärmeaustauschfläche vergrößert. Allerdings wächst der beobachtete Wärmestrom in den Kühlmantel nicht proportional dazu, so daß sich der Wärmedurchgangskoeffizient verringert. Grund für den unterproportional zunehmenden Wärmestrom ist die Tatsache, daß die dazugekommene Flüssigkeit eine größere Entfernung vom Rührer besitzt und daher in Wandnähe weniger turbulent durchmischt ist. Grund dafür sind die Anker des Ankerrührers, die lediglich bis auf 9.4 cm an den Reaktordeckel heranreichen.

Der beschriebene Effekt sollte auf einer Veränderung des reaktionsseitigen Wärmeübergangskoeffizienten a_r mit dem Füllstand beruhen. Dies geht allerdings nicht aus den Messungen hervor, wie in Tabelle 4 beispielhaft für eine Rührerdrehzahl von $N = 250 \text{ min}^{-1}$ aufgezeigt ist. Auch liefert die Wilson-Methode für unterschiedliche Flüssigkeitsvolumina unterschiedliche

Gerätekonstanten. Das gleiche Bild zeigt sich bei gleichem Volumen aber unterschiedlichen Rührerformen. Ausgerechnet der Schrägblattrührer mit dem geringsten Wärmedurchgangskoeffizienten h besitzt den größten reaktionsseitigen Wärmeübergang a_r (Tab. 22).

Wie aus den Korrelationskoeffizienten r der Wilson-Plots ersichtlich, liefert diese Methode für ein Flüssigkeitsvolumen und einen Rührertyp in sich schlüssige Ergebnisse. Bei der Vergleichbarkeit der Rührertypen untereinander ist aber Vorsicht geboten, da das Modell hier scheinbar erhebliche Schwächen aufweist. So ist der Meßbereich, in welchem die Rührerdrehzahl und damit $N^{-2/3}$ variiert wird, aus den oben genannten Gründen klein gegenüber der Entfernung von der $1/h$ -Achse. Dies führt dazu, daß jede Gerätekonstante $\Phi(T_r)$ ein großes Konfidenzintervall $\Delta_{68}\Phi(T_r)$ [72] besitzt (Tab. 22), welches die übrigen berechneten Gerätekonstanten beinhaltet. Eine Verringerung der Konfidenzintervalle wäre bei der gegebenen Meßgenauigkeit nur durch eine Erhöhung der Meßpunkte möglich. Messungen bei höheren Rührerdrehzahlen sind allerdings nicht möglich, da es scheinbar schon bei den in den Abbildungen 42 bis 44 aufgezeigten Messungen bei hohen Drehzahlen (kleinen $N^{-2/3}$) zu tendenziell höheren Wärmedurchgangswerten h (kleinere Werte von $1/h$) kommt.

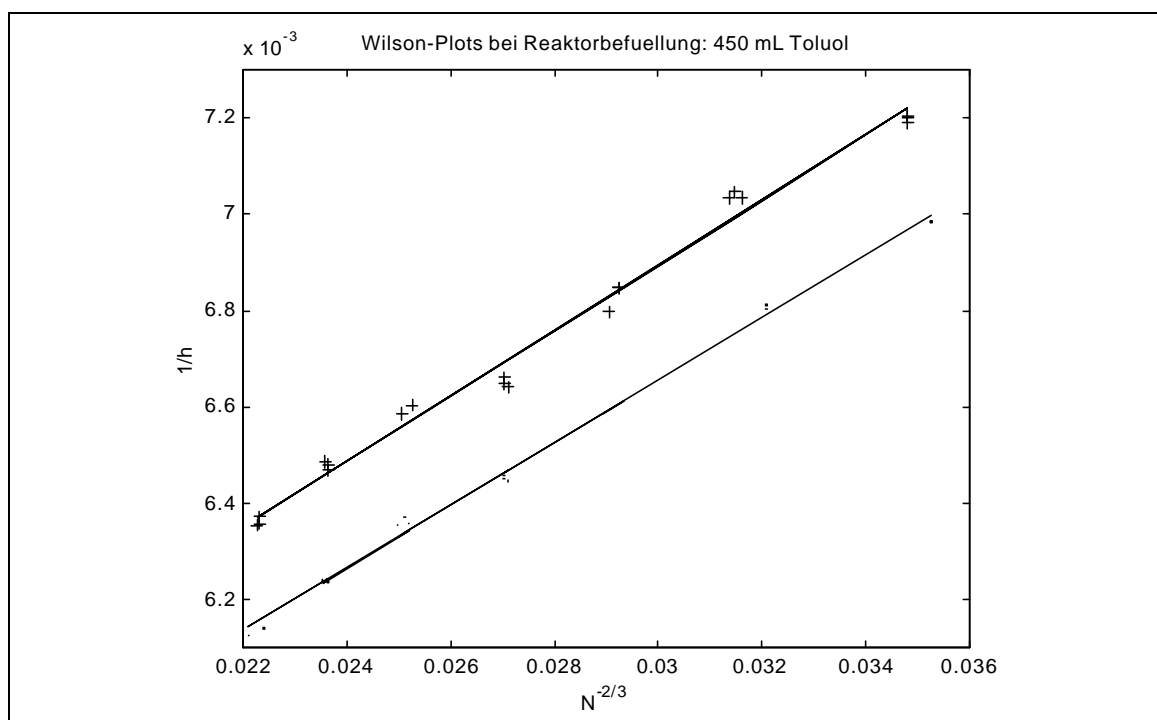


Abb. 43 Wilson-Plot: Ankerrührer mit aufgesetztem Schrägblattrührer
 $T_r = 80^\circ\text{C}$ (+), $T_r = 90^\circ\text{C}$ (·)

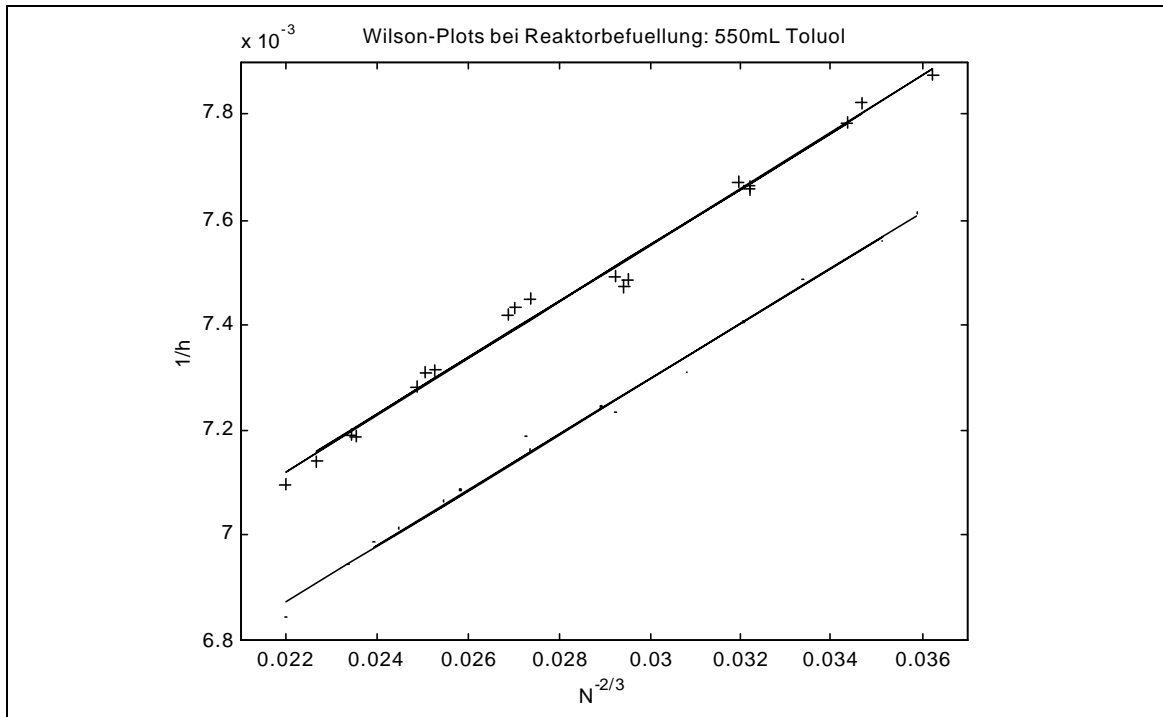


Abb. 44 Wilson-Plot: Ankerrührer mit aufgesetztem Schrägblattrührer
 $T_r = 80\text{ °C}$ (+), $T_r = 90\text{ °C}$ (·)

Tab. 22: Parameter der Wilson-Plots

Rührer	SBR	AR	ARSBR	ARSBR	ARSBR	ARSBR
T_r [°C]	80	80	80	90	80	90
V_r [mL]	450	450	450	450	550	550
$\Phi(T_r)$ [K m ² W ⁻¹]	$6.315 \cdot 10^{-3}$	$4.821 \cdot 10^{-3}$	$4.868 \cdot 10^{-3}$	$4.708 \cdot 10^{-3}$	$5.944 \cdot 10^{-3}$	$5.717 \cdot 10^{-3}$
$\Delta_{68} \Phi(T_r)$ [K m ² W ⁻¹]	$6.56 \cdot 10^{-3}$	$4.06 \cdot 10^{-3}$	$3.86 \cdot 10^{-3}$	$5.66 \cdot 10^{-3}$	$4.50 \cdot 10^{-3}$	$5.79 \cdot 10^{-3}$
b_r [K m ² min ^{2/3} W ⁻¹]	$4.740 \cdot 10^{-2}$	$6.283 \cdot 10^{-2}$	$6.749 \cdot 10^{-2}$	$6.490 \cdot 10^{-2}$	$5.358 \cdot 10^{-2}$	$5.266 \cdot 10^{-2}$
r	0.998	0.997	0.994	0.998	0.994	0.997
$a_r(250\text{ min}^{-1})$ [W K ⁻¹ m ⁻²]	837.3	631.6	588.0	611.4	740.6	753.6
$h(250\text{ min}^{-1})$ [W K ⁻¹ m ⁻²]	133.2	156.1	152.2	157.6	137.1	142.0

In Tabelle 23 sind neben den für unterschiedliche Flüssigkeitsvolumina bei $N = 250 \text{ min}^{-1}$ gemessenen Wärmeübergangskoeffizienten a_r , die Werte aufgeführt, die eine Berechnung nach (24) [58] liefert. Die gemessenen Werte liegen deutlich unter denen mittels der dimensionsanalytischen Berechnung bestimmten. In Abbildung 45 ist diese Differenz für verschiedene Rührerdrehzahlen N aufgetragen.

Grundlage für eine dimensionsanalytische Beschreibung ist die geometrische Ähnlichkeit der Versuchsanordnungen. Sowohl der hier eingesetzte Ankerrührer als auch der Reaktor sind aber den in [54, 55, 58] beschriebenen Apparaten nicht geometrisch ähnlich, so daß die Berechnung nur eine Abschätzung liefern kann. Eine in der Literatur [58] vorgeschlagene Berechnung für wandgängige Rührer, welche die Reynolds-Zahl Re mit den Stoffwerten bei Wandtemperatur bildet, hat nur geringe Auswirkungen auf die Differenz zwischen abgeschätzten und gemessenen Werten.

Da die experimentelle Ermittlung von reaktionsseitigen Wärmeübergangskoeffizienten a_r zu den wichtigen Aufgaben der Reaktionskalorimetrie gehört, sollte hierzu auf die Möglichkeit zurückgegriffen werden, das Reaktionskalorimeter im Wärmeflußmodus zu betreiben. Durch einen geräte-technischen Umbau und geeignete Rührerwahl kann dann die geometrische Ähnlichkeit zu Betriebsreaktoren hergestellt werden, was eine Übertragbarkeit der experimentellen Ergebnisse ermöglicht.

Tab. 23: Berechnete und gemessene Wärmeübergangskoeffizienten für Toluol

N [min ⁻¹]	T [°C]	Re	Pr	Nu	$a_{r,ber}$ [W K ⁻¹ m ⁻²]	a_r [W K ⁻¹ m ⁻²] $V_r=450 \text{ mL}$	a_r [W K ⁻¹ m ⁻²] $V_r=550 \text{ mL}$
250	80	$59.5 \cdot 10^3$	4.75	601.6	962.6	588.0	740.6
250	90	$65.0 \cdot 10^3$	4.43	622.6	984.1	611.4	753.6

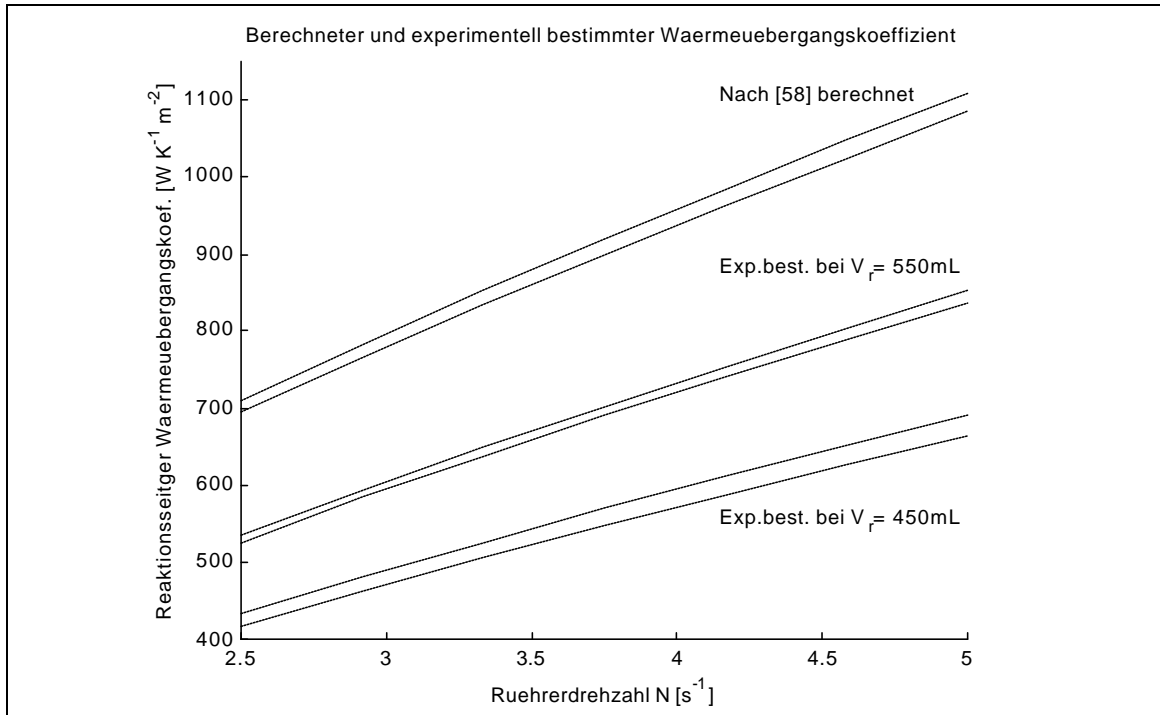


Abb. 45 Vergleich von berechneten und gemessenen Wärmeübergangskoeffizienten α_r bei unterschiedlichen Drehzahlen N

Im Vergleich der unterschiedlichen Rührerformen schneidet der Ankerrührer hinsichtlich des Wärmeübergangs vom Reaktor zum Kühlmantel am besten ab. Nur etwas schlechter stellt sich der Wärmedurchgang beim Ankerrührer mit aufgesetztem Schrägblattrührer dar. Wird die Wahl auf Basis des Wärmedurchgangsverhaltens und der Mischzeitcharakteristik getroffen, so muß der Ankerrührer mit aufgesetztem Schrägblattrührer als bester Rührer angesehen werden.

Unterstützt wird diese Entscheidung, wenn man zusätzlich die Regelbarkeit der Reaktortemperatur T_r bei Verwendung der drei unterschiedlichen Rührerformen berücksichtigt. Eine Regelstrecke mit kurzer Totzeit t_D wird im allgemeinen als besser regelbar angesehen als eine Strecke mit längerer Totzeit. Aus diesem Grund wird zusätzlich zum Wilson-Plot noch die Antwortfunktion in T_r auf einen Stellgrößensprung in q_{comp} aufgezeichnet, um die Totzeit t_D des System bei verschiedenen Rührerformen zu bestimmen. Gemessen wird bei einer Reaktortemperatur von $T_r = 80^\circ\text{C}$, bei einem Flüssigkeitsvolumen von $V_r = 450$ mL und einer Rührerdrehzahl von $N = 250$ min^{-1} . Die erhaltenen Totzeiten sind in Tabelle 24 aufgeführt.

Tab. 24: Totzeiten bei Verwendung unterschiedlicher Rührer

Rührer	Totzeit t_D [s]
Schrägblattrührer	6
Ankerrührer	12
Ankerrührer m.aufg. Schrägblattrührer	6

Bei Betrachtungen der Totzeiten zeigt sich recht deutlich das von den Mischzeiten her bekannte Bild. Der einfache Ankerrührer besitzt zwar einen höheren Wärmedurchgang durch die Behälterwand (was aus den stationären Messungen der Wilson-Methode hervorgeht), zeigt allerdings im Vergleich zum Ankerrührer mit aufgesetztem Schrägblattrührer Schwächen bei der Verteilung der durch die Kompensationsheizung eingebrachten Leistung. Durch den aufgesetzten Schrägblattrührer wird erreicht, daß der Ankerrührer wieder die kürzere Totzeit des Schrägblattrührers aufweist. Eine axiale Strömungskomponente ist bei den gerätetechnischen Vorgaben unerlässlich, weshalb auch bei den folgenden Messungen der Ankerrührer mit aufgesetztem Schrägblattrührer zum Einsatz kommt.

4.3 Kinetische Untersuchung der Lösungspolymerisation von Methylmethacrylat

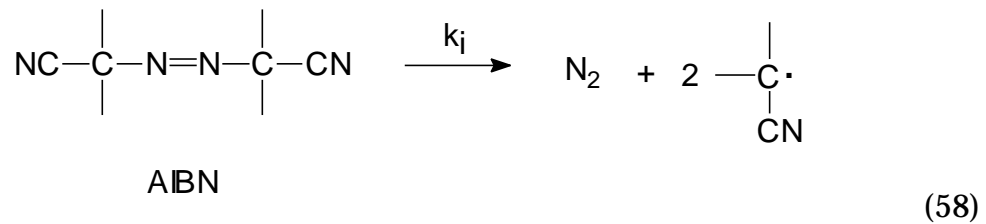
Die radikalische Lösungspolymerisation von Methylmethacrylat (MMA, Methacrylsäuremethylester) in Toluol wird aus mehreren Gründen für die Erprobung des Reaktionskalorimeters in Betracht gezogen [59]. So sind die Stoffdaten und kinetischen Konstanten der Reaktion aus der Literatur bekannt [60]. Als Lösungspolymerisation ist die Gefahr der Belagbildung an der Wand oder auf der Heizung gering, die Reaktion kann bis zu hohen Umsätzen verfolgt werden, und es wird im Gegensatz zu einer Substanzpolymerisation eine gute Wärmeabfuhr gewährleistet. Weitere Vorteile der Reaktion sind die geringe Zunahme der Viskosität im Versuchsablauf und die Vermeidung des Trommsdorff-Norrish-Effekts (Gel-Effekt).

Dieser Effekt, der meistens bei Substanzpolymerisationen und Lösungspolymerisationen in ungünstigen Lösungsmitteln auftritt, äußert sich darin, daß die Bruttoreaktionsgeschwindigkeit der Polymerisation und der Polymerisationsgrad trotz abnehmender Monomerkonzentration wieder zunehmen. Er ist auf die Verhakung der Polymerketten untereinander zurückzuführen. Dadurch wird die freie Diffusion der Polymerradikale stärker gehemmt als die Diffusion der Monomere, so daß der Kettenabbruch durch Disproportionierung oder Kombination der Polymerradikale unterbunden wird. Da die Radikalbildung durch Initiatorzerfall ungehindert weitergeht, nimmt die Radikalkonzentration im Reaktionsmedium beim Trommsdorff-Norrish-Effekt und damit die Bruttoreaktionsgeschwindigkeit zu.

Als Nachteil der Lösungspolymerisation sei die geringe Reaktorkapazität genannt, da der Hauptbestandteil der Reaktionsmischung aus Lösungsmittel besteht. Dieses Lösungsmittel muß bei der Aufarbeitung des Polymethylmethacrylats (PMMA) im technischen Maßstab energie- und kostenaufwendig entfernt werden.

Die radikalische Polymerisation besteht aus den Reaktionsschritten Zerfall des Initiators, Start der Kettenreaktion, Kettenfortpflanzungsreaktion und Abbruch der Kettenreaktion.

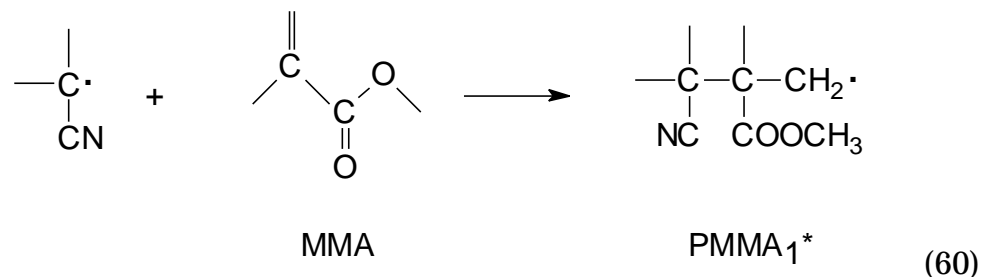
Die Bildung von Radikalen erfolgt durch thermische oder photochemische Spaltung des Initiators Azobisisobutyronitril (AIBN, 2,2'-Azobis(2-methylpropionitril)). Dabei wird ein Molekül Stickstoff aus dem Initiatormolekül abgespalten, so daß zwei Radikale entstehen (58).



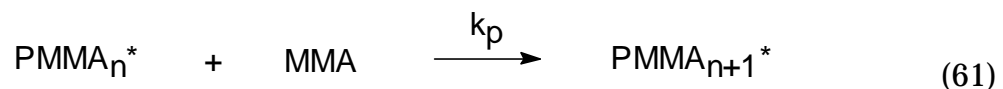
Die Zerfallsreaktion gehorcht einem Geschwindigkeitsgesetz 1. Ordnung (59), die Abnahme der Initiatorkonzentration ist proportional der vorhandenen Konzentration. Geschwindigkeitskonstante der Reaktion ist die Konstante k_i (*initiation*).

$$-\frac{dC_I}{dt} = k_i \cdot C_I \quad (59)$$

Die entstehenden Radikale rekombinieren bzw. kombinieren nur zu einem geringen Teil [61]. Der größere Anteil f_i von ihnen startet eine Kettenreaktion gemäß (60).



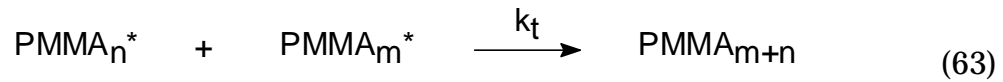
Im Kettenwachstumsschritt (61) wird ein weiteres Monomermolekül MMA an ein Polymerradikal PMMA_n^* angelagert. Dabei wird davon ausgegangen, daß die Wachstumsgeschwindigkeit (ausgedrückt durch k_p) unabhängig von der Kettenlänge des Polymerradikals ist. Daher wird in (61) die allgemeine Formulierung mit $n \geq 1$ gewählt.



Die Wachstumsgeschwindigkeit (Abnahme der Monomerkonzentration C_M mit der Zeit) gehorcht einem Geschwindigkeitsgesetz 2. Ordnung (62) mit der Geschwindigkeitskonstanten k_p (*propagation*) und ist der Polymerradikalkonzentration C_{RM^*} und der Monomerkonzentration C_M proportional.

$$-\frac{dC_M}{dt} = k_p \cdot C_{RM^*} \cdot C_M \quad (62)$$

Ein Abbruch der Kettenreaktion kann durch Rekombination oder Disproportionierung der Polymerradikale erfolgen. Die Rekombinationsreaktion ist in (63) dargestellt. Im Geschwindigkeitsgesetz (64) der Kettenabbruchreaktion können Rekombination und Disproportionierung mit einer Geschwindigkeitskonstanten k_t (*termination*) zusammengefaßt werden, da in beiden Fällen zwei Radikale beteiligt sind. Lediglich auf die Verteilung der Kettenlänge der entstehenden Polymere wirken sich die unterschiedlichen Abbruchreaktionen aus.



$$-\frac{dC_{RM^*}}{dt} = k_t \cdot C_{RM^*}^2 \quad (64)$$

Nicht berücksichtigt in diesem Reaktionsschema sind die Radikalübertragungsreaktionen auf das Lösungsmittel.

Bei dieser Reaktion läßt sich das Bodensteinsche Stationaritätsprinzip anwenden [61]. Es wird davon ausgegangen, daß sich nach kurzer Zeit ein Gleichgewicht zwischen der Kettenstart- und der Kettenabbruchreaktion einstellt. Die Polymerradikalkonzentration ist dabei sehr gering und praktisch keiner zeitlichen Änderung unterworfen. Der so gewonnene Ausdruck (66) für C_{RM^*} läßt sich in das Geschwindigkeitsgesetz der Kettenwachstumsreaktion (62) einsetzen.

$$2f_I \cdot k_i \cdot C_{I,0} = k_t \cdot C_{RM^*}^2 \quad (65)$$

$$C_{RM^*} = \sqrt{\frac{2f_I \cdot k_i}{k_t}} \sqrt{C_{I,0}} \quad (66)$$

Faßt man alle Konstanten der Reaktion k_i , k_p , k_t und f_I in einer Bruttoreaktionsgeschwindigkeitskonstanten k_{br} zusammen, so ergibt sich als Ergebnis das bekannte Wurzelgesetz der radikalischen Polymerisation:

$$-\frac{dC_M}{dt} = k_p \sqrt{\frac{2f_I \cdot k_i}{k_t}} \cdot \sqrt{C_{I,0}} \cdot C_M = k_{br} \cdot \sqrt{C_{I,0}} \cdot C_M \quad (67)$$

Das durch Aufzeichnung aller Meßwerte und Auswertung der stationären Wärmebilanz des Reaktors (16) erhaltene Signal ist der zeitliche Verlauf der Wärmeproduktionsrate der Reaktion $q_r(t)$. Daraus kann die gesamte während der Reaktion freigesetzte Wärmemenge Q_{ges} (68) bestimmt werden. Die obere Grenze der Integration wird hierbei durch das Ende der Reaktion t_{ende} bestimmt.

$$Q_{ges} = \int_{t=0}^{t_{ende}} q_r(t) dt \quad (68)$$

Ist die Unterschußkomponente a im Verlauf der Reaktion vollständig verbraucht worden, so läßt sich die Reaktionsenthalpie (69) unter den gegebenen Reaktionsbedingungen angeben.

$$\Delta_r H = -\frac{Q_{ges}}{n_a} \quad (69)$$

Mit der Gesamtwärmemenge Q_{ges} (68) läßt sich nach (70) der Umsatz X als Funktion der Zeit ermitteln. Damit einhergehend sind die im Reaktionsgefäß vorhandenen Stoffmengen der einzelnen Komponenten bzw. bei bekanntem Reaktionsvolumen deren Konzentration ebenfalls als Funktion der Zeit bekannt. Aus den Reaktionsgeschwindigkeiten $r(t)$ und den Konzentrationen $C_i(t)$ lassen sich die Parameter eines Reaktionsgeschwindigkeitsansatzes bestimmen.

$$X(t) = \frac{\int_{t=0}^t q_r(t) dt}{Q_{ges}} \quad (70)$$

Für eine Lösungsmittelpolymerisation, die dem Wurzelgesetz (67) folgt, läßt sich jedoch eine wesentlich elegantere Bestimmung der Geschwindigkeitskonstanten der Bruttoreaktion k_{br} vornehmen.

Die Reaktionsgeschwindigkeit r als zeitliche Abnahme der Monomerkonzentration C_M läßt sich nach (71) umformen. Darin enthalten ist die exponentiell abnehmende Verlauf der Monomerkonzentration ausgehend von der anfänglichen Konzentration $C_{M,0}$.

$$r(t) = -\frac{dC_M}{dt} = k_{br} \cdot \sqrt{C_{I,0}} \cdot C_{M,0} \cdot e^{-k_{br} \sqrt{C_{I,0}} t} \quad (71)$$

Der gleiche exponentielle Verlauf läßt sich auch in der zeitlichen Entwicklung der Wärmeproduktionsrate q_r erkennen (72).

$$r(t) = \frac{q_r(t)}{V_r \cdot \Delta_r H} = \frac{1}{V_r \cdot \Delta_r H} \cdot q_{r,0} \cdot e^{-k' \cdot t} \quad (72)$$

Ein Koeffizientenvergleich von (71) und (72) liefert als Ergebnis, daß die anfängliche Reaktionsgeschwindigkeit $r(t=0)$ durch das Produkt vor der Exponentialfunktion ausgedrückt wird (73). Darüber hinaus kann aus dem Vergleich der Exponenten die Beziehung (74) gewonnen werden.

$$r(t=0) = k_{br} \cdot \sqrt{C_{I,0}} \cdot C_{M,0} = \frac{q_{r,0}}{V_r \cdot \Delta_r H} \quad (73)$$

$$k' = k_{br} \cdot \sqrt{C_{I,0}} \quad (74)$$

Werden die Werte von $q_r(t)$ logarithmiert gegen die Zeit aufgetragen (75), so folgt aus der Steigung direkt die Konstante k' und damit bei bekannter Initiatorkonzentration über Gleichung (74) auch die Bruttoreaktionsgeschwindigkeitskonstante k_{br} .

$$\ln(q_r(t)) = \ln(q_{r,0}) - k' \cdot t \quad (75)$$

Bevor die Polymerisation von Methylmethacrylat in Toluol kalorimetrisch untersucht werden soll, werden die Leistungsfähigkeit der Anlage und des Reglers anhand von Kalibriermessungen ermittelt. Dazu wird eine rechteck- und rampenförmige Funktion der Kalibrierheizleistung aufgegeben. Dieser Kalibriervorgang ist auch deshalb notwendig, weil im Gegensatz zu dem in [51] als Kühlmedium verwendeten Wasser nunmehr ein Silikonöl geringerer Wärmekapazität eingesetzt wird.

Der Verlauf von Kalibrierheizleistung q_{cal} und gemessener Heizleistung q_{exp} ist für die Rechteckfunktion in Abbildung 46 dargestellt. (Der Reaktor ist mit 550 mL Toluol befüllt, die Reaktortemperatur beträgt $T_r = 80^\circ\text{C}$.) Die gesamte Wärmemenge, die in die Flüssigkeit eingebracht wird, beträgt $Q_{ges} = 19.327$ kJ. Die experimentell bestimmte Wärmemenge ergibt sich zu $Q_{exp} = 19.382$ kJ, so daß sich eine relative Abweichung unter 0.3 Prozent ergibt. Aus Abbildung 46 ist zu entnehmen, daß der Funktionsverlauf der

Kalibrierheizleistung q_{cal} bis auf die Abweichungen beim Ein- und Ausschalten sehr gut aus dem gemessenen Signal rekonstruiert werden kann.

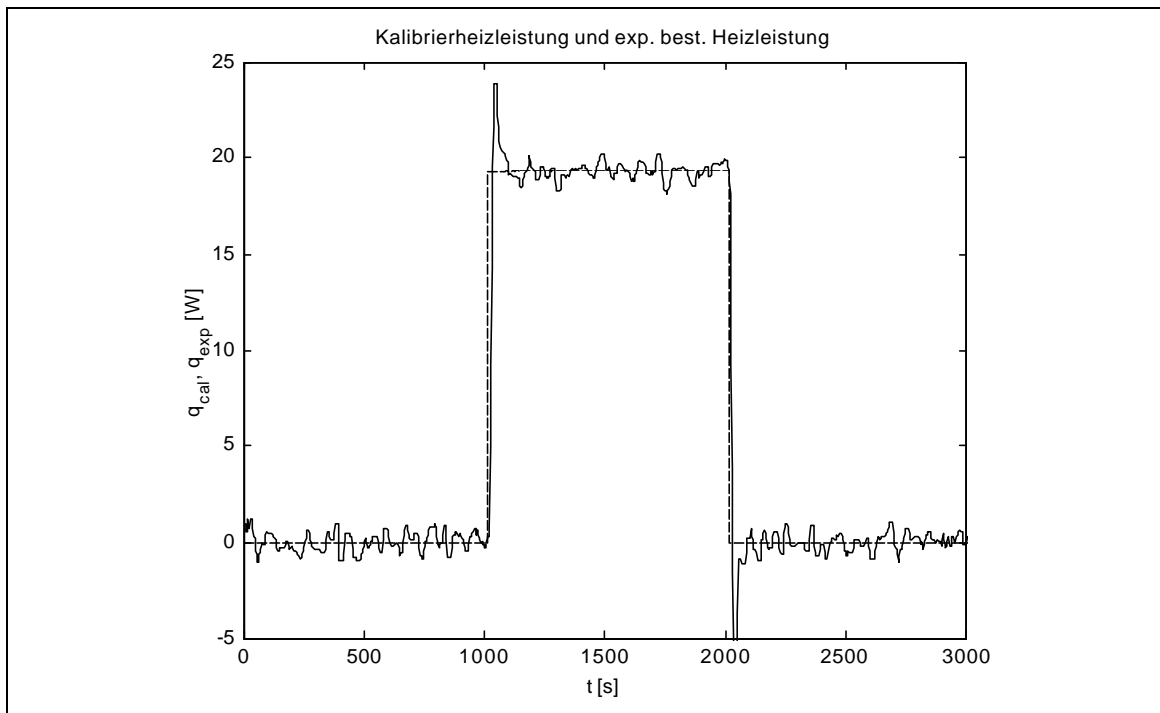


Abb. 46 Kalibrierung des Reaktionskalorimeters anhand einer Rechteckfunktion der Kalibrierheizleistung q_{cal} (---)

Ebenfalls bei $T_r = 80\text{ °C}$ und einer Reaktorbefüllung mit 550 mL Toluol wird die in Abbildung 47 dargestellte Rampenfunktion der Kalibrierheizleistung q_{cal} aufgegeben. Bei dieser Kalibrierung wird eine relative Abweichung von 0.62 Prozent zwischen der abgegebenen Wärmemenge $Q_{ges} = 46.482\text{ kJ}$ und der experimentell ermittelten Wärmemenge $Q_{exp} = 46.195\text{ kJ}$ festgestellt.

Da die Kalibrierheizleistung q_{cal} von ca. 35 W beim Einschalten der Heizung in einer Zeit von 40 Minuten auf ungefähr 5 W abnimmt, wird der Regler wesentlich dynamischer beansprucht als bei der Rechteckfunktion. Wie dem zeitlichen Verlauf der Reaktortemperatur T_r in Abbildung 48 entnommen werden kann, ist der Regler außer beim Ein- und Ausschalten der Heizung sehr gut in der Lage, ein Toleranzband von $\pm 0.020\text{ K}$ einzuhalten.

Beim Einschalten der Kalibrierheizung tritt eine maximale Temperaturabweichung von $+0.400\text{ K}$ auf. Diese recht große Abweichung der Reaktortemperatur von ihrem Sollwert kommt durch die anfängliche hohe Heizleistung von ca. 35 W zustande. Diese Heizleistung kommt den 42.4 W recht nahe, die zur Aufrechterhaltung einer Temperaturdifferenz von 10 K zwischen Reaktorinnenraum und Kühlmantel benötigt werden. Darüber-

hinaus ist der Neuro-Fuzzy-Regler an einem Störgrößenprogramm optimiert, dessen maximale Heizleistung 25 W betragen hat.

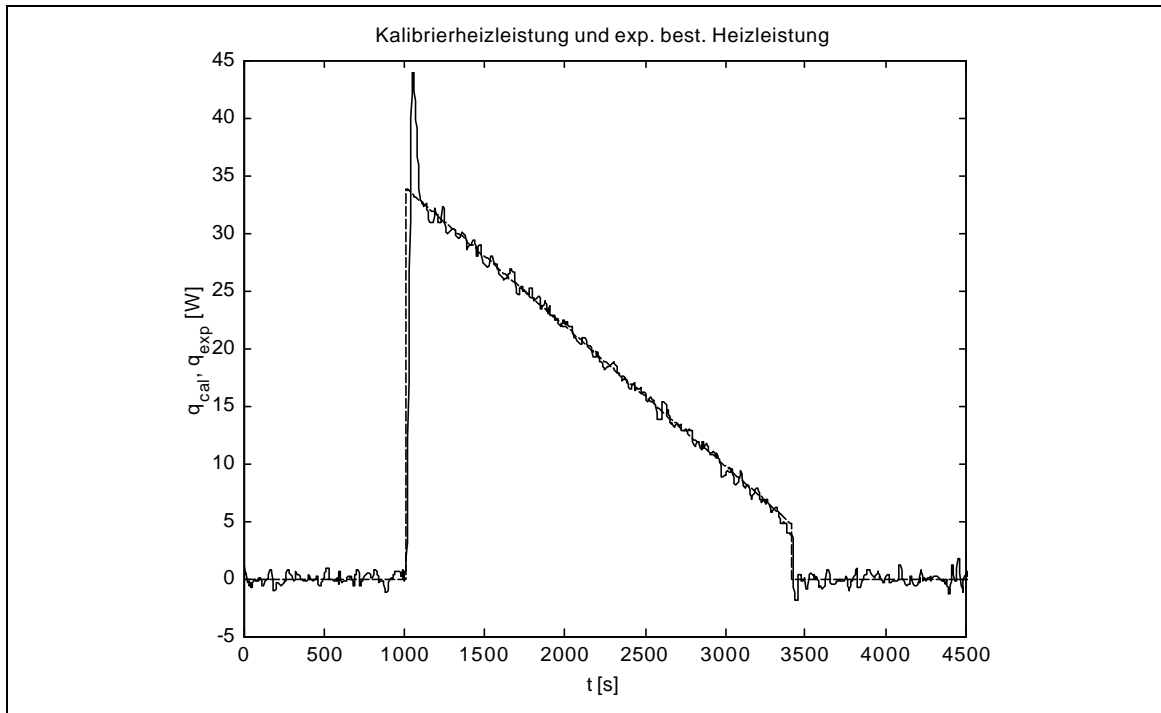


Abb. 47 Kalibrierung des Reaktionskalorimeters und Test des Neuro-Fuzzy-Reglers anhand einer Rampenfunktion der Kalibrierheizleistung q_{cal} (---)

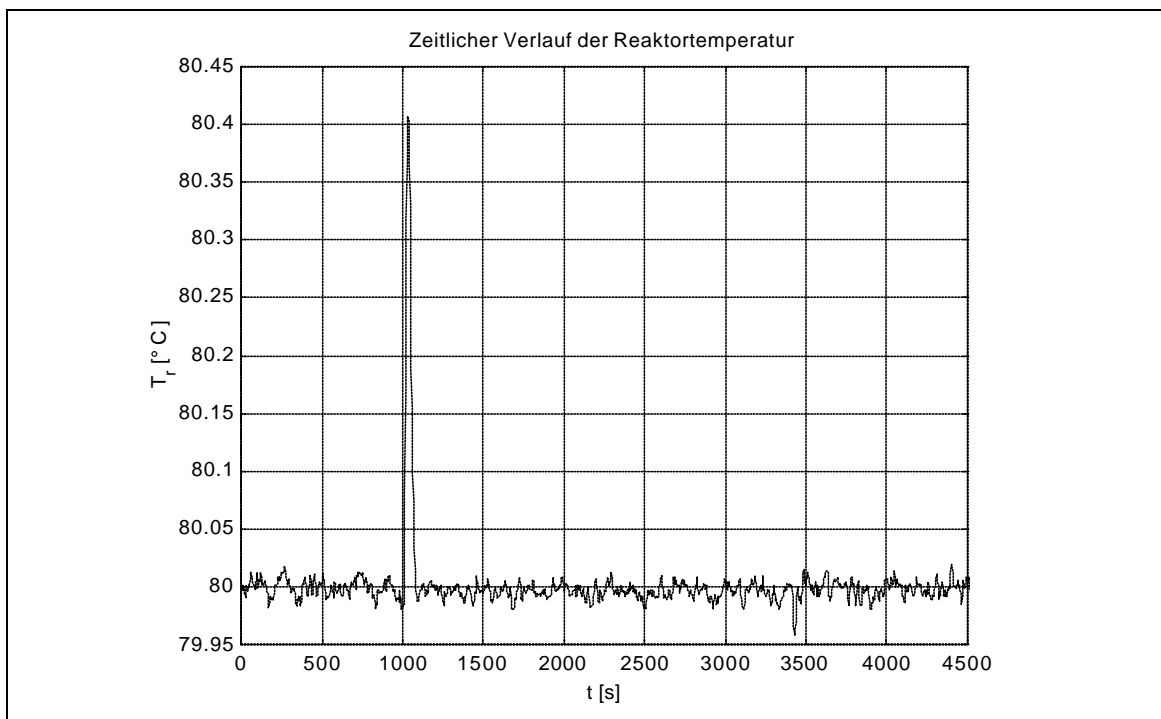


Abb. 48 Verlauf der Reaktortemperatur T_r bei Kalibrierung anhand der Rampenfunktion (Abb. 47)

Trotz dieser Umstände wird die Reaktortemperatur innerhalb von 81 s ohne ein Überschwingen auf den Sollwert zurückgeregelt. Beim Ausschalten der Kalibrierheizung (ein Sprung von 5 W auf 0 W) tritt eine kleine Temperaturabweichung von 0.042 K auf, das ist kaum mehr als das Zweifache des Toleranzbereichs.

Für die Lösungspolymerisation von Methylmethacrylat wird die Temperatur des Kühlmantels auf $T_j = 50\text{ °C}$ eingestellt. Der Reaktor wird durch einen starken Stickstoffstrom getrocknet und mit Inertgas gefüllt. In den so präparierten Reaktor werden 80.2 g Methylmethacrylat in 307.5 g Toluol mittels einer Dosierpumpe aus einem luftdichten Rundkolben gefüllt. Das Methylmethacrylat ist zuvor durch Destillation vom Inhibitor befreit worden. Parallel dazu werden 4.516 g Azobisisobutyronitril in 86.6 g Toluol gelöst.

Nach einem kurzen Spülen des Gasraums über der vorgelegten Toluol-Methylmethacrylat-Mischung mit Stickstoff wird der Rührer eingeschaltet, und das Reaktionsgemisch wird durch Einschalten des Neuro-Fuzzy-Reglers auf die gewünschte Temperatur T_r gebracht. Die Sollwerte der Mantel- und der Luftbadtemperatur werden auf einen Wert 10 K unter der Reaktortemperatur eingestellt.

Nach einer kurzen Zeit, in der die Reaktortemperatur im Toleranzbereich um den Sollwert geblieben ist, wird die AIBN-Lösung innerhalb von 180 s von der Dosierpumpe in den Reaktor befördert. Die Konzentration des Methylmethacrylat beträgt 1.374 mol L^{-1} , die Konzentration des Azobisisobutyronitril 0.047 mol L^{-1} . Es wird je eine Messung bei einer Reaktortemperatur von 80, 85 und 90 °C durchgeführt.

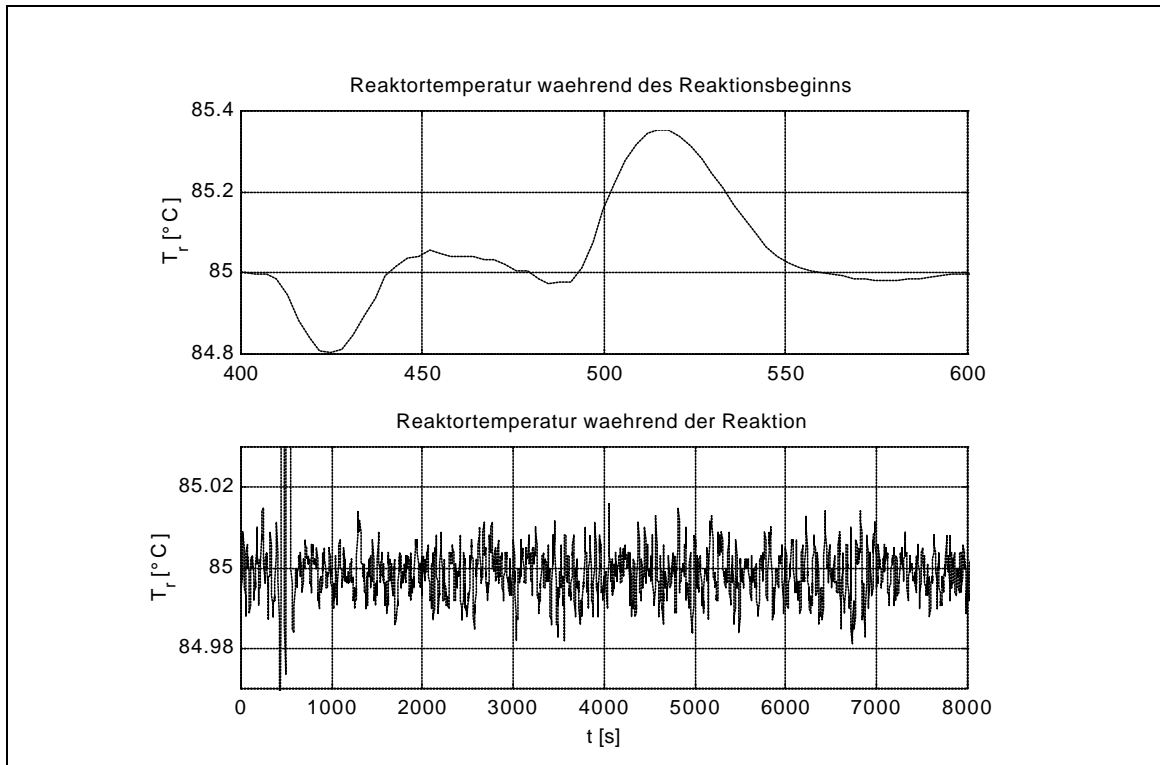


Abb. 49 Verlauf der Reaktortemperatur T_r bei der Lösungspolymerisation von Methylmethacrylat

In Abbildung 49 (oben) ist die Reaktortemperatur T_r während des Beginns der Reaktion dargestellt. Recht deutlich wird die Beanspruchung der Regelung während dieses Zeitraums. Zunächst sinkt die Temperatur infolge der Dosierung der kalten Initiatorlösung 0.2 K unter den Sollwert, worauf der Regler mit einer Erhöhung der Heizleistung reagiert, um die dosierte Flüssigkeit auf Reaktortemperatur zu erwärmen. Kaum jedoch ist der Sollwert $T_{r,soll}$ wieder erreicht, setzt die Reaktion ein, und die Wärmeentwicklung führt zu einem starken Temperaturanstieg auf 0.4 K über den Sollwert, so daß der Regler die Kompensationsheizleistung schnell wieder verringern muß. Sieht man von diesem Anfangsverhalten ab, so ist der adaptive Regler während der gesamten Zeit in der Lage, die Reaktortemperatur T_r in einem Toleranzbereich von 0.020 K um den Sollwert zu halten (Abbildung 49 unten).

Die anfänglichen Wärmeproduktionsraten q_r sind für die verschiedenen Meßtemperaturen in Abbildung 50 aufgeführt. Der Bereich negativer Wärmeproduktionsraten ist nicht mit aufgetragen. Er rührt von der Erwärmung der kalten Initiatorlösung her.

Nicht berücksichtigt ist in der Ermittlung von q_r der Akkumulationsterm q_{accu} (5), was in Betracht der guten Temperaturregelung (Abb. 49) gerechtfertigt ist. Ebenfalls nicht berücksichtigt werden kann der Dosierungsterm q_{dos} (7). Kalibriermessungen bei einer Dosierung von Toluol haben ergeben, daß eine große, zeitlich veränderliche Temperaturdifferenz zwischen gemessenem T_{dos} und der realen Temperatur besteht, bei der die dosierte Flüssigkeit den Reaktor betritt, was die experimentelle Ermittlung von q_{dos} verhindert.

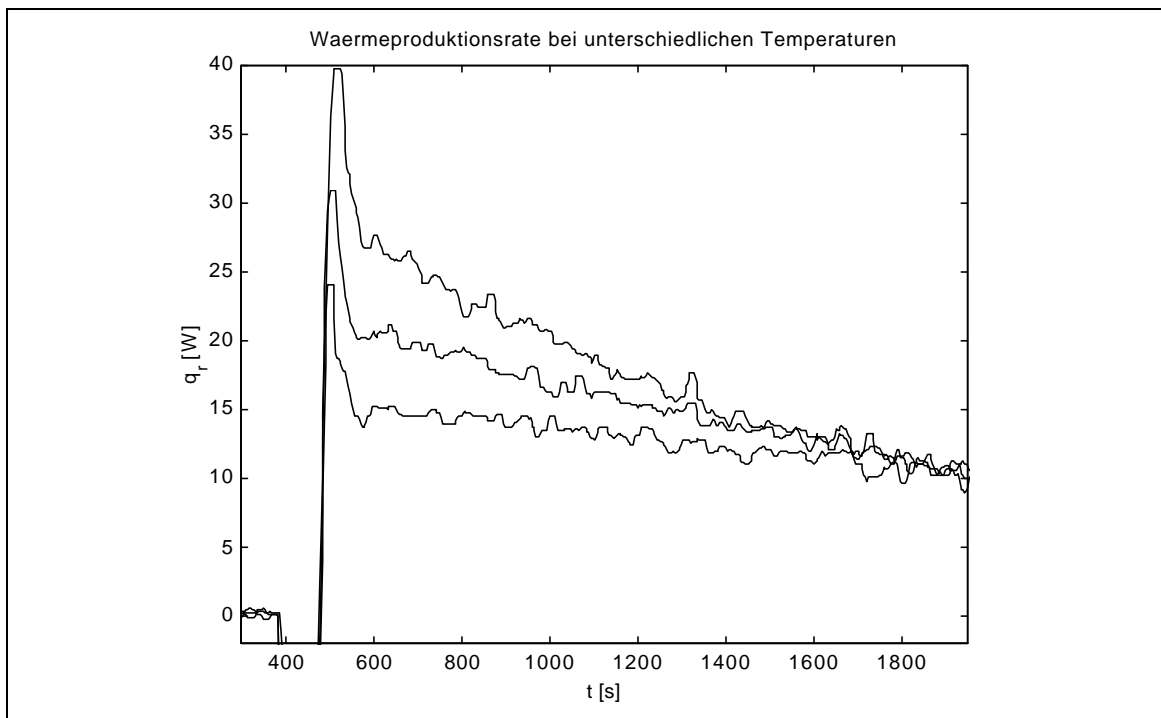


Abb. 50 Anfängliche Wärmeproduktionsraten q_r der MMA-Polymerisationen bei den verschiedenen Reaktortemperaturen ($T_r = 80^\circ\text{C}$, 85°C und 90°C)

Für die Ermittlung der gesamten freigesetzten Wärmemenge Q_{ges} wird daher der Zeitpunkt, an dem die Reaktortemperatur T_r rapide steigt, als Reaktionsbeginn angesehen. Bis zu diesem Zeitpunkt wird die Wärmeproduktionsrate zurückextrapoliert. Das Verfahren ist schon von SCHLEGEL [52] erfolgreich angewendet worden. Anschließend wird über die Wärmeproduktionsrate (68) integriert. Bei vollständigem Umsatz an Methylmethacrylat kann die Reaktionsenthalpie angegeben werden. Eine Überprüfung auf vollständigen Umsatz erfolgt nach Beendigung der Reaktion, als weitere AIBN-Lösung in den Reaktor eingebracht wird und keine Wärmeentwicklung zu beobachten ist.

Die in Tabelle 25 aufgelisteten Werte stimmen sehr gut mit dem in der Literatur [60] aufgeführten Wert überein. Die anfängliche Temperaturerhöhung von 0.4 K über den Sollwert führt lediglich zu einer 3% höheren Reaktionsgeschwindigkeit, was vernachlässigbar ist.

Tab. 25: Reaktionsenthalpie $\Delta_r H$ bei verschiedenen Temperaturen

T_r [°C]	$\Delta_r H$ [kJ mol ⁻¹]
80	55.36
85	55.37
90	55.70

Ein wesentlicher Vorteil der leistungskompensierten Reaktionskalorimetrie ist die Ermittlung des reaktionsseitigen Wärmeübergangskoeffizienten a_r während der Reaktion. Beispielhaft ist a_r für die Polymerisation bei $T_r = 80$ °C in Abbildung 51 aufgetragen. Er berechnet sich aus dem Wärmedurchgangskoeffizienten h (14) unter Verwendung der Gerätekonstante aus Tabelle 22 (für 80 °C, $V_r = 550$ mL). Erkennbar ist der leichte Abfall des Wärmeübergangskoeffizienten a_r im Verlauf der Reaktion. Dies kann aufgrund der zunehmenden Viskosität der Reaktionsmischung erwartet werden. Der gemessene Wert für a_r ist mit 570 bis 480 W K⁻¹ m⁻² immer kleiner als der für Toluol ermittelte (Tab. 22). Bei der Interpretation ist Vorsicht geboten, da die Ermittlung der Gerätekonstanten $\Phi(T_r)$ erheblichen Modellschwächen der Wilson-Methode unterworfen ist (Kap. 4.2).

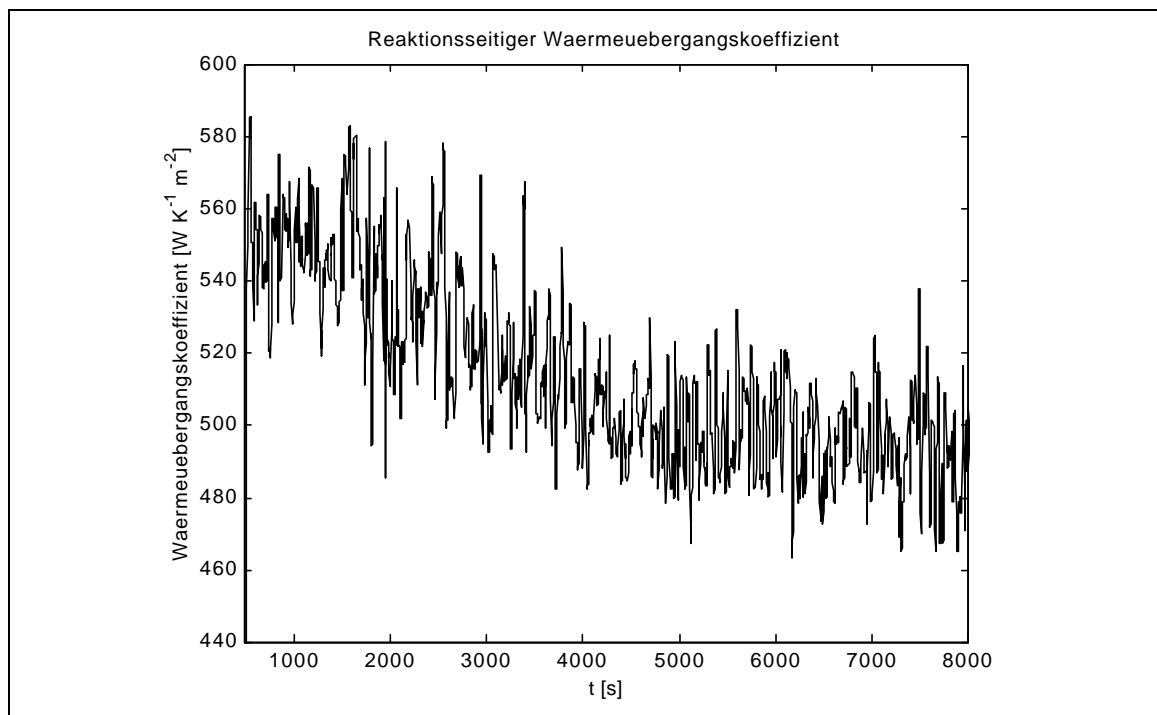


Abb. 51 Zeitlicher Verlauf des reaktionsseitigen Wärmeübergangskoeffizienten a_r während der MMA-Polymerisation bei $T_r = 80^\circ\text{C}$

Für die Bestimmung der Geschwindigkeitskonstanten der Polymerisationsreaktion wird das oben beschriebene Verfahren angewendet. Dazu wird der Logarithmus der Wärmeproduktionsrate $\ln(q_r)$ gebildet. In Abbildung 52 ist dies beispielhaft für eine Reaktortemperatur von $T_r = 85^\circ\text{C}$ vorgenommen worden. Es ist zu erkennen, daß sich für eine Zeit von 40 Minuten eine Gerade ergibt. Aus der Steigung der Gerade läßt sich bei bekannter Initiatorkonzentration die Bruttogeschwindigkeitskonstante k_{br} ermitteln. Die bei verschiedenen Temperaturen bestimmten Werte für k_{br} sind in Tabelle 26 eingetragen. In Abbildung 53 sind diese Werte in ein Arrhenius-Diagramm eingetragen. Der Korrelationskoeffizient von 0.9992 zeigt an, daß die gemessenen Bruttogeschwindigkeitskonstanten in sich schlüssig sind.

Tab. 26: Geschwindigkeitskonstante bei verschiedenen Temperaturen

T_r [K]	k_{br} [$10^{-3} \text{ L}^{1/2} \text{ mol}^{-1/2} \text{ s}^{-1}$]
353.15	1.468
358.15	2.196
363.15	3.090

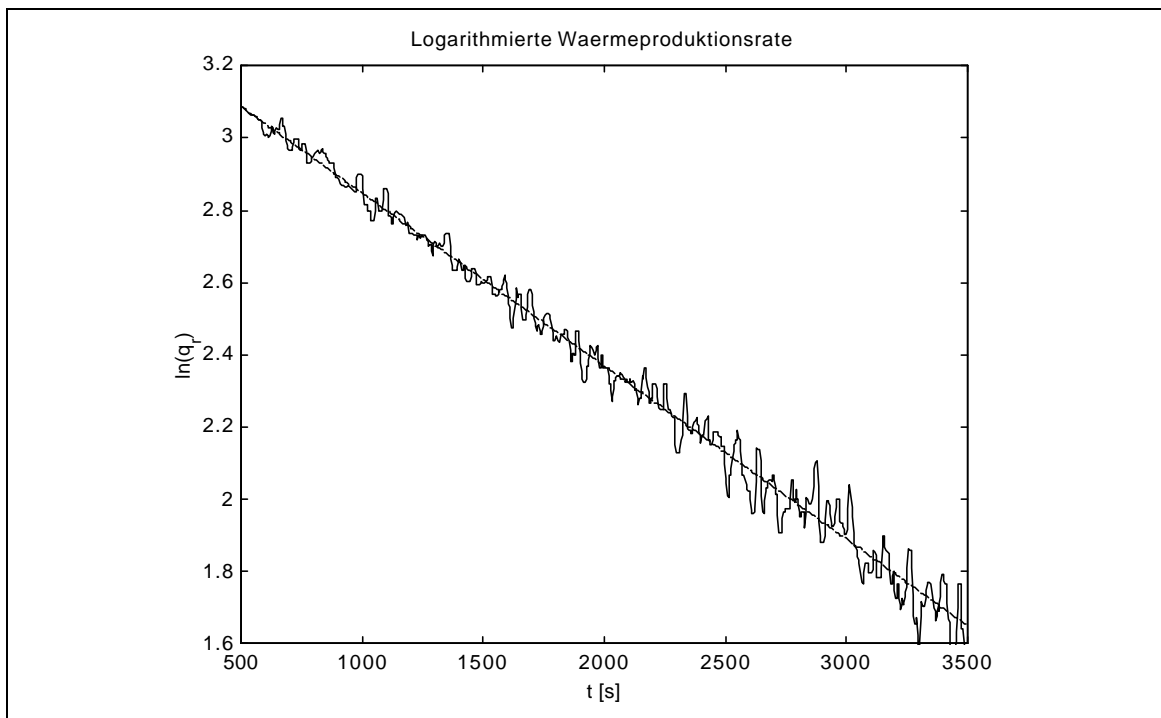


Abb. 52 Logarithmierte Wärmeproduktionsrate $\ln(q_r)$ der MMA-Polymerisation bei $T_r = 85^\circ\text{C}$

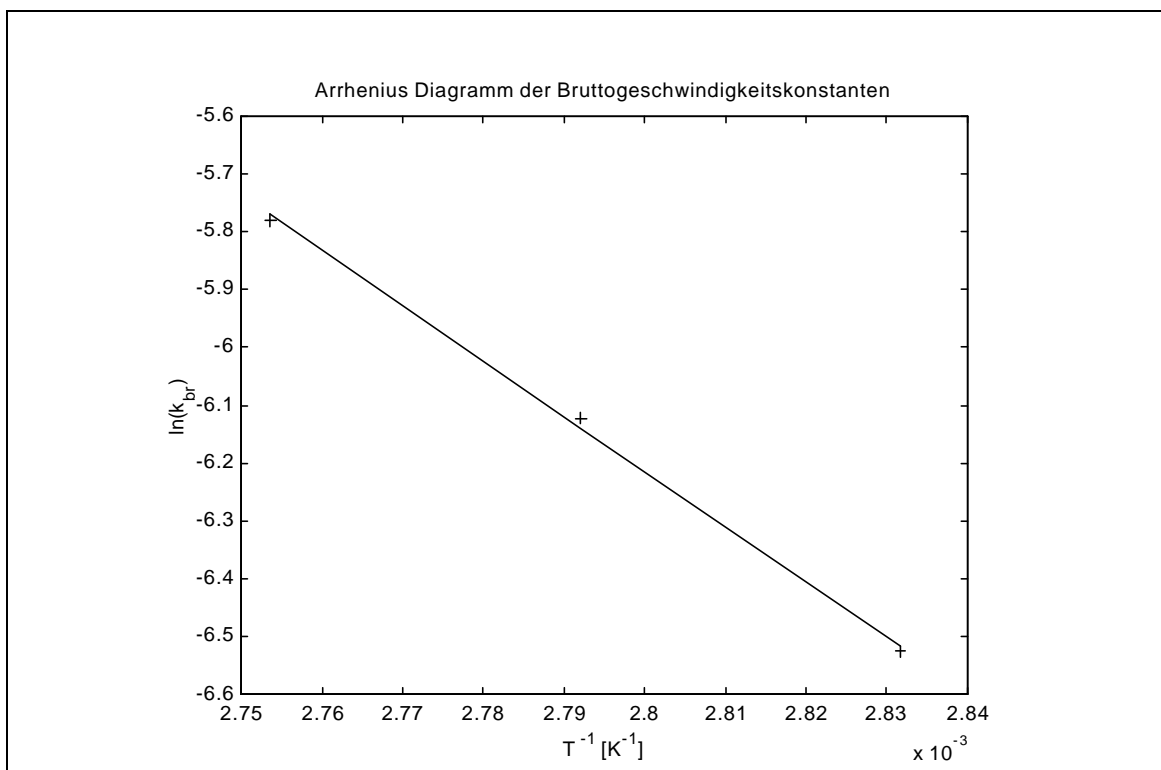


Abb. 53 Arrhenius-Auftragung von $\ln(k_{br})$ gegen T_r^{-1} für die Lösungspolymerisation von Methymethacrylat

5 Ausblick

Die leistungskompensierte Reaktionskalorimetrie stellt bezüglich der Einhaltung eines streng-isothermen Zustands während der Reaktion das beste Meßprinzip dar. Es ist daher nur konsequent, bei der Regelung der Reaktortemperatur T_r nicht beim klassischen PID-Regelkonzept zu bleiben, sondern auf moderne Verfahren der Regelung zuzugreifen. Ein Fuzzy-Logik- bzw. Neuro-Fuzzy-Regler mit seiner Partitionierung des Eingangsraumes paßt sehr gut zu den Anforderungen der isothermen Kalorimetrie, da er über unterschiedliche Reglerverstärkungen für den stationären und gestörten Betrieb verfügt.

Durch die Implementierung eines Adaptionalgorithmus, der die *Online*-Identifikation der Regelstrecke als Grundlage hat, konnte der Schwachpunkt der ursprünglichen Regelung ausgeräumt werden. Bei unterschiedlichen Flüssigkeiten für die Reaktorbefüllung oder während einer Reaktion ändern sich die Durchmischungs- und Wärmeübergangsverhältnisse derart, daß eine kontinuierliche Ermittlung der Reglerparameter nötig ist. Die Reglerparameter wurden bisher experimentell ermittelt [51], worauf nun aufgrund der fortlaufenden Regleradaption verzichtet werden kann. So wird die streng-isotherme Verfahrensweise bei unterschiedlichen Zusammensetzungen der Reaktionsmischung gewährleistet, und der Akkumulationsterm q_{accu} (5) kann ohne Informationsverlust vernachlässigt werden.

Durch die Optimierung des Rührsystems in bezug auf die Homogenisierung des Reaktionsmediums und den Wärmeübergang von der Heizung und auf die Behälterwand wird der Reaktor weiter an das Verhalten eines idealen Rührkessels angepaßt. Auf diese Weise werden Wärmeproduktionsraten q_r ermittelt, die repräsentativ für die gesamte Reaktionsmischung sind.

Ein Schwachpunkt des Reaktionskalorimeters bleibt die Ermittlung der Temperatur T_{dos} einer dosierten Reaktionskomponente. Trotz aller geräte-technischen Ausführungen bleibt hierbei das Grundprinzip jeder Temperaturmessung verletzt, daß nämlich die Wärmekapazität eines Meßobjektes wesentlich größer sein sollte als die Wärmekapazität des Meßfühlers. Die besten Ergebnisse wurden daher bei einer Dosierung bei Raumtemperatur [51] erzielt. Es spricht für die Regelung der Reaktortemperatur, daß gute Ergebnisse auch bei der Dosierung von kalten Reaktionskomponenten erhalten werden.

Mit dem Reaktionskalorimeter und seiner adaptiven Regelung steht jetzt ein Gerät zur Verfügung, das ideal für Flüssigphasereaktionen geeignet ist. Erfolgreich getestet an der Lösungspolymerisation von Methylmethacrylat, sind Versuche zur Substanzpolymerisation [52] mit begrenztem Umsatz oder auch andere Polymerisationen von industriellem Interesse denkbar. Hierbei muß nicht unbedingt die Ermittlung kinetischer Daten im Vordergrund stehen. Angedacht werden kann eine Anpassung von Temperaturprogrammen oder Dosierungsstrategien zur Produkt- und Prozeßoptimierung.

6 Literaturverzeichnis

- [1] SCHULTE, U.: *Einführung in die Fuzzy-Logik*. München: Franzis, 1993
- [2] TILLI, T.: *Fuzzy-Logik*. München: Franzis, 1991
- [3] KAHLERT, J.: *Fuzzy-Control für Ingenieure*. Braunschweig: Vieweg, 1995
- [4] JAMAL, R.; PICHLIK, H.: *LabVIEW Programmiersprache der vierten Generation*. Haar b. München: Prentice Hall, 1997
- [5] ZIMMERMANN, H.-J.: *Fuzzy-Technologien: Prinzipien, Werkzeuge, Potentiale*. Düsseldorf: VDI-Verlag, 1993
- [6] TRAEGER, D. H.: *Einführung in die Fuzzy-Logik*. 2. überarb. u. erw. Aufl., Stuttgart: Teubner, 1994
- [7] VON ALTROCK, C.: *Fuzzy Logik: Technologie*. Bd. 1, 2. Aufl., München: Oldenbourg, 1995
- [8] KAHLERT, J.; FRANK, H.: *Fuzzy-Logik und Fuzzy-Control*. Braunschweig: Vieweg, 1993
- [9] PASSINO, K. M.; YURKOVICH, S.: *Fuzzy Control*. Menlo Park: Addison-Wesley, 1998
- [10] TSOUKALAS, L. H.; UHRIG, R. E.: *Fuzzy and Neural Approaches in Engineering*. New York: John Wiley & Sons, 1997
- [11] DE SILVA, C. W.: *Intelligent Control: Fuzzy Logic Applications*. Boca Raton: CRC Press, 1995
- [12] DRIANKOV, D.; HELLENDORF, H.; REINFRANK, M.: *An Introduction to Fuzzy Control*. 2nd rev. ed., Berlin: Springer Verlag, 1996
- [13] CARTWRIGHT, H. M.: *Applications of Artificial Intelligence in Chemistry*. Oxford: Oxford Univ. Press, 1995
- [14] HOFFMANN, N.: *Kleines Handbuch Neuronale Netze*. Braunschweig: Vieweg, 1993
- [15] ZUPAN, J.; GASTEIGER, J.: *Neural Networks for Chemists*. Weinheim: VCH, 1993
- [16] DEMUTH, H.; BEALE, M.: *Neural Network Toolbox - For use with MatLab*. Version 3.0, Natick: MathWorks Inc., 1998
- [17] PHAM, D. T.; LIU, X.: *Neural Networks for Identification, Prediction and Control*. London: Springer Verlag, 1995
- [18] OMATU, S.; KHALID, M.; YUSOF, R.: *Neuro-Control and its Applications*. London: Springer Verlag, 1996
- [19] ZELL, A.: *Simulation Neuronaler Netze*. 1. Aufl., Bonn: Addison-Wesley, 1994
- [20] FREEMAN, J. A.; SKAPURA, D. M.: *Neural Networks: Algorithms, Applications and Programming Techniques*. Reading: Addison-Wesley, 1991
- [21] FREEMAN, J. A.: *Simulating Neural Networks with Mathematica*. Reading: Addison-Wesley, 1994

-
- [22] WASSERMAN, P. D.: *Neural Computing: Theory and Practice*. New York: Van Nostrand Reinhold, 1989
- [23] HECHT-NIELSEN, R.: *Neurocomputing*. Reading: Addison-Wesley, 1990
- [24] MEDSKER, L. R.: *Hybrid Intelligent Systems*. Boston: Kluwer Acad. Publ., 1995
- [25] DRÖGEMÖLLER, M.: *Optimierungsalgorithmen für das quadratische Zuordnungsproblem*. Braunschweig, Technische Universität, Diplomarbeit, 1998
- [26] BOTHE, H.-H.: *Neuro-Fuzzy-Methoden: Einführung in Theorie und Anwendungen*. Berlin: Springer Verlag, 1998
- [27] MICHALEWICZ, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*. 3rd, rev. and extended ed., Berlin: Springer Verlag, 1996
- [28] CHAMBERS, L. (HRSG.): *Practical Handbook of Genetic Algorithms - Vol. 1: Applications*. Boca Raton: CRC Press, 1995
- [29] CHAMBERS, L. (HRSG.): *Practical Handbook of Genetic Algorithms - Vol. 2: New Frontiers*. Boca Raton: CRC Press, 1995
- [30] GOLDBERG, D. E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading: Addison-Wesley, 1989
- [31] HERRERA, F.; VERDEGAY, J. L. (HRSG.): *Genetic Algorithms and Soft Computing*. Heidelberg: Physica-Verlag, 1996
- [32] KLIR, G. J.; YUAN, B.: *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River: Prentice Hall, 1995
- [33] BULSARI, A. B. (HRSG.) : *Neural Networks for Chemical Engineers*. Amsterdam: Elsevier, 1995
- [34] PATYRA, M. J.; MLYNEK, D. M. (HRSG.): *Fuzzy Logic: Implementation and Applications*. Stuttgart: Teubner, 1996
- [35] ROJAS, R.: *Theorie der neuronalen Netze - eine systematische Einführung*. Berlin: Springer Verlag, 1993
- [36] WIENHOLT, W.: *Entwurf neuronaler Netze*. Frankfurt a. M.: Verlag Harri Deutsch, 1996
- [37] SCHÄFFNER, C.: *Analyse und Synthese neuronaler Regelungsverfahren*. München, Techn. Univ., Diss., 1996
- [38] DRIESCHER, H.-A. B.: *Prädiktive Regelung für Multimodellansätze mit neuronalen Netzwerken*. Magdeburg, Otto-von-Guericke-Universität, Diss., 1998
- [39] HOFFMANN, U.; HOFMANN, H.: *Einführung in die Optimierung*. Weinheim: VCH, 1971
- [40] ISERMANN, R.: *Identifikation dynamischer Systeme*. Bd. I u. II. Berlin: Springer Verlag, 1988
- [41] LANDAU, I. D.; LOZANO, R.; M'SAAD, M.: *Adaptive Control*. London: Springer Verlag, 1998

- [42] BUSCH, P.: *Elementare Regelungstechnik: Allgemeingültige Darstellung ohne höhere Mathematik*. Würzburg: Vogel Verlag, 1999
- [43] KROLL, A.: *Fuzzy-Systeme zur Modellierung und Regelung komplexer technischer Systeme*. Düsseldorf: VDI-Verlag, 1997
- [44] FIRSCHING, P.: *Der Einsatz künstlicher neuronaler Netze in regelungstechnischen Anwendungen*. Erlangen-Nürnberg, Universität, Diss., 1995
- [45] NAUCK, D.; KLAWONN, F.; KRUSE, R.: *Neuronale Netze und Fuzzy-Systeme: Grundlagen des Konnektionismus*. 2., überarb. u. erw. Aufl., Braunschweig: Vieweg, 1996
- [46] SERAPHIN, M.: *Neuronale Netze und Fuzzy-Logik - Verknüpfung der Verfahren, Anwendungen, Vor- und Nachteile*. München: Franzis, 1994
- [47] STRIETZEL, R.: *Fuzzy-Regelung*. München: Oldenbourg, 1996
- [48] PEDRYCZ, W.: *Neurofuzzy Systems*. In: NGUYEN, H.; SUGENO, M.: *Fuzzy Systems - Modeling and Control*. Boston: Kluwer Acad. Publ., 1998
- [49] BUCKLEY, J.J.; FEURING, T.: *Fuzzy and neural - Interactions and Applications*. Heidelberg: Physica-Verlag, 1999
- [50] HEESCHE, K.: *Selbstlernende Fuzzy-Systeme mittels neuronaler und genetischer Algorithmen*. Dortmund, Univ., Diss., 1996
- [51] HORN, K.: *Verbesserung und Charakterisierung eines leistungskompensierten Reaktionskalorimeters mit Differentialkühlung*. Braunschweig, Technische Universität, Dissertation, 1999
- [52] SCHLEGEL, M.: *Entwicklung eines leistungskompensierten Reaktionskalorimeters mit Differentialkühlung*. Braunschweig, Technische Universität, 1997
- [53] SEBORG, D.E.; EDGAR, T.F.; MELLICHAMP, D.A.: *Process dynamics and control*. New York: John Wiley & Sons, 1989
- [54] ZLOKARNIK, M.: *Mixing & Stirring*. Ullmann's Encyclopedia of Industrial Chemistry. Vol. B2. 5.Aufl. Weinheim: VCH Verlag, 1990
- [55] ZLOKARNIK, M.: *Rührtechnik - Theorie und Praxis*. Berlin: Springer Verlag, 1999
- [56] WILKE, H.-P.; ET AL.: *Rührtechnik - Verfahrenstechnische und apparative Grundlagen*. Heidelberg: Hüthig Verlag, 1988
- [57] Verein Deutscher Ingenieure (Hrsg.): *VDI-Wärmeatlas: Berechnungsblätter für den Wärmeübergang*. 6., erw. Aufl., Düsseldorf: VDI-Verlag, 1991
- [58] ZLOKARNIK, M.: Wärmeübergang an der Wand eines Rührbehälters beim Kühlen und Heizen im Bereich $10^0 < Re < 10^5$. In: *Chem.-Ing.-Techn.*, 1969, 41, S. 1195-1242
- [59] REICHERT, K.-H.; MORITZ, H.-U.: *DAKAPO Datenerfassung mittels Kalorimetrie von Polymerisationen*. Ringversuch zur Kalorimeterkalibrierung, 1998
- [60] BRANDRUP, J.; IMMERGUT, E.H.: *Polymer Handbook* 3rd ed., New York: Wiley, 1989

- [61] BRAUN, D.; CHERDRON, H.; RITTER, H.: *Praktikum der makromolekularen Stoffe: Grundlagen, Synthesen, Modifizierungen* Weinheim: Wiley-VCH, 1999
- [62] MEYER, I.: *Verbesserung des dynamischen Verhaltens eines leistungskompensierten Reaktionskalormeters durch Simulationsstudien*. Braunschweig, Technische Universität, Diplomarbeit, 1996
- [63] KASPERS, W.; KÜFNER, H.-J.: *Steuern - Regeln - Automatisieren*. 5., verb. Aufl. Braunschweig: Vieweg, 1996
- [64] REUTER, M.: *Regelungstechnik für Ingenieure*. 9., überarb. und erw. Aufl. Braunschweig: Vieweg, 1994
- [65] KARR, C.L.: *Practical applications of computational intelligence for adaptive control*. Boca Rato: CRC Press, 1999
- [66] SHINSKEY, G.F.: *Process Control Systems*. 4th ed. New York: McGraw-Hill, 1996
- [67] PALM, R.: *Design of Fuzzy Controllers*. In: NGUYEN, H.; SUGENO, M.: *Fuzzy Systems - Modeling and Control*. Boston: Kluwer Acad. Publ., 1998
- [68] ISERMANN, R.: *Mechatronische Systeme: Grundlagen*. Berlin: Springer Verlag, 1999
- [69] UNBEHAUEN, H.: *Regelungstechnik 3: Identifikation, Adaption, Optimierung*. 4., durchges. Aufl. Braunschweig: Vieweg, 1993
- [70] RIEDMILLER, M.: *Selbständig lernende neuronale Steuerung*. Düsseldorf: VDI-Verlag, 1997
- [71] MOSEBACH, M.; REICHERT, K.H.: Bestimmung kinetischer Konstanten der radikalischen Polymerisation mittels adiabatischer Reaktionskalorimetrie. In: *Chem.-Ing.-Techn.*, 1995, 67, S. 1655-1658
- [72] KREYSZIG, E.: *Statistische Methoden und ihre Anwendungen*. 7. Aufl., Göttingen: Vandenhoeck & Ruprecht, 1979

7 Symbol- und Abkürzungsverzeichnis

Indizes:

<i>cal</i>	Kalibrierheizung
<i>comp</i>	Kompensationsheizung
<i>flow</i>	Strom durch die Reaktorwand
<i>dos</i>	Dosierung
<i>I</i>	Intitiator
<i>j</i>	Mantel
<i>M</i>	Monomer
<i>r</i>	Reaktor
<i>RM</i>	Radikal
<i>sec</i>	sekundäre Quellen
<i>soll</i>	Sollwert
<i>w</i>	Wand
0	Start, Zulauf

Symbole:

<i>a</i>	[m ² s ⁻¹]	Temperaturleitzahl
<i>a_f, b_f</i>		Konstanten des ARMA-Modells
<i>A</i>	[m ²]	Wärmeaustauschfläche
AIBN		Azobisisobutyronitril
<i>a</i>	[W K ⁻¹ m ⁻²]	Wärmeübergangskoeffizient
<i>a</i>		linksseitige Breite einer Zugehörigkeitsfunktion
<i>b_r</i>	[K m ² s ^{2/3} W ⁻¹]	Steigung beim Wilson-Plot
<i>b</i>		rechtsseitige Breite einer Zugehörigkeitsfunktion
<i>c</i>	[mol m ⁻³]	Konzentration
<i>c₁, c₂</i>		Konstanten
<i>c_{Händ}</i>		Skalierungsfaktor Heizleistungsänderung
<i>c_p</i>	[J K ⁻¹ kg ⁻¹]	spezifische Wärmekapazität
<i>c_{Tabw}</i>		Skalierungsfaktor Temperaturabweichung
<i>c_{Tänd}</i>		Skalierungsfaktor Temperaturänderung
<i>C</i>	[mol m ⁻³]	Konzentration
<i>d</i>	[m]	Rührerdurchmesser
<i>D</i>	[m]	Reaktordurchmesser
DKD		Deutscher Kalibrierdienst
DMM		Digital Multimeter

d	[m]	Dicke
$e(t)$		Regeldifferenz
$\Delta e(t)$		Änderung der Regeldifferenz
e_l		Prädiktionsfehler des Modells
E		Modellfehler
E_l		Fehler eines Eingabe-Ausgabe-Paares
h	[Pa s]	Dynamische Viskosität
h		Lernrate
h_{index}		Spez. Lernrate für einen Parameter
f		Laufindex ($0..N_V-1$)
f_l		Initiatorausbeute
FLS		Fuzzy-Logik-System
G		Abbildungsvorschrift eines linearen Modells
h	[W K ⁻¹ m ⁻²]	Wärmedurchgangskoeffizient
h		Länge des Eingabevektors x_h ($h = n-1$)
$\Delta_r H$	[J mol ⁻¹]	Reaktionsenthalpie
i		Reaktionskomponente
i, j, k		Laufindezes
I	[A]	elektrischer Strom
IEC		International Electronical Comission
ISE		quadratische Regelfläche (<i>integral of squared error</i>)
ITAE		zeitgewichteter Betrag der Regelfläche (<i>integral of time multiplied absolute error</i>)
ITSE		zeitgewichtete quadratische Regelfläche (<i>integral of time multiplied squared error</i>)
J		Regelgüte (erweiterter ISE)
k_{br}	[L ^{1/2} mol ^{-1/2} s ⁻¹]	Bruttoreaktionsgeschwindigkeitskonstante
k_i	[s ⁻¹]	Geschwindigkeitskonstante der Startreaktion
k_p	[L mol ⁻¹ s ⁻¹]	Geschwindigkeitskonstante der Kettenwachstumsreaktion
k_t	[L mol ⁻¹ s ⁻¹]	Geschwindigkeitskonstante der Abbruchreaktion
k_C	[s ⁻¹]	Reaktionsgeschwindigkeitskonstante
k		Gewichtungsfaktor
KNN		Künstliches neuronales Netz
l		Laufindex über alle r Eingabe-Ausgabe-Paare des Trainingsdatensatzes
l	[W K ⁻¹ m ⁻¹]	Wärmeleitfähigkeit

m	[kg]	Masse
m		Konstante
\dot{m}	[kg s ⁻¹]	Massestrom
MMA		Methylmethacrylat
\mathbf{m}		Momentumterm-Konstante
$\mathbf{m}_i(x)$		Zugehörigkeitsfunktion, -grad
n		Anzahl Neuronen in der Eingabeschicht einschließlich Bias-Neuron
n_D		Brechungsindex
n_i	[mol]	Stoffmenge der Komponente i
\dot{n}_i^0	[mol s ⁻¹]	Stoffmengenstrom
N	[s ⁻¹]	Rührerdrehzahl
N_F		diskreter Prädiktionszeitraum
N_T		Anzahl Datenpaare im Trainingsdatensatz
N_V		Anzahl Vergangenheitswerte der Ein- und Ausgangsgrößen eines Prozesses
NFS		Neuro-Fuzzy-System
\mathbf{n}	[m ² s ⁻¹]	Kinematische Viskosität
\mathbf{n}_{ij}		stöchiometrischer Koeffizient der Komponente i bezüglich der Reaktion j
\mathbf{u}		Anzahl verarbeitender Neuronen
$o_k(t+1)$		Ausgabe ($k = 1$) eines Systemmodells für den Zeitschritt $t+1$
PID		Proportional-Integral-Differential
PMMA		Polymethylmethacrylat
Φ	[K m ² W ⁻¹]	Gerätekonstante beim Wilson-Plot
q_{accu}	[W]	pro Zeiteinheit akkumulierte Wärmemenge
q_{cal}	[W]	Kalibrierheizleistung
q_{comp}	[W]	Kompensationsheizleistung
q_{dos}	[W]	Wärmestrom durch Dosierung
q_{fr}	[W]	Wärmestrom auf die Reaktorwand
q_{ges}	[W]	gesamte Heizleistung (= $q_{comp} + q_{cal}$)
q_r	[W]	Wärmeproduktionsrate
q_{sec}	[W]	sekundäre Wärmeströme
Δq_{comp}	[W]	Kompensationsheizleistungsänderung
$\Delta\Delta q_{comp}$	[W]	Änderung der Kompensationsheizleistungsänderung

Q_{exp}	[J]	experimentell ermittelte Wärmemenge
Q_{ges}	[J]	gesamte Wärmemenge
Θ		Parametervektor eines linearen Modells
r		Anzahl der Eingabe-Ausgabe-Paare im Trainingsdatensatz
r_j	[mol L ⁻¹ s ⁻¹]	Reaktionsgeschwindigkeit der Reaktion j
rms		root mean square
R_i	[mol L ⁻¹ s ⁻¹]	Stoffmengenänderungsgeschwindigkeit
\mathbf{r}	[kg m ⁻³]	Dichte
$\Delta \mathbf{r}$	[kg m ⁻³]	Dichteunterschied
s_A, s_B, s_C		Skalierungsfaktoren
t	[s, min]	Zeit
t		diskreter Zeitschritt
t_r	[s]	Zeitkonstante einer Reaktion
Δt_a	[s]	Abtastperiode
Δt_c	[s]	Regelperiode
T	[°C, K]	Temperatur
T_n	[s]	Nachstellzeit
T_r	[°C]	Reaktortemperatur
$T_{r,sim}$	[°C]	simulierte Reaktortemperatur
$T_r \mathcal{C}$	[K W ⁻¹]	Ableitung der Reaktortemperatur
T_v	[s]	Vorhaltezeit
t_D	[s]	Totzeit
q	[s]	Mischzeit
$u(t)$		(Prozess-)Eingangsgröße u zum Zeitschritt t , Stellgröße
$\Delta u(t)$		Stellgrößenänderung
Δu_{korr}		korrigierte Stellgrößenänderung
$\Delta \Delta u$		Änderung der Stellgrößenänderung
U	[V]	elektrische Spannung
U_h		Stellbereich
v_f, w_f		nichtlineare Funktionen im erweiterten ARMA-Modell
V_r	[m ³]	Reaktionsvolumen
x	[m]	Ortskoordinate
$x_h(t)$		Eingabevektor des Systemmodells zum Zeitschritt t bestehend aus Ein-/Ausgangsgrößen

$x_{l,Ai}$		linksseitiger Begrenzungswert
$x_{m,Ai}$		Mittenswert der Teilmenge Ai
$x_{r,Ai}$		rechtsseitiger Begrenzungswert
$x_w(t)$	[K]	Regelabweichung
X		Umsatz
X_p		Proportionalbereich, Regelbereich
y_k		(Prozeß-)Ausgangsgröße ($k = p = 1$), Regelgröße
$y_{k,soll}$		Sollwert der Regelgröße
$y_k(t)$		(Prozeß-)Ausgangsgröße zum Zeitschritt t , Regelgröße
$y_k'(u)$		Ableitung der Ausgangsgröße y_k
$y_{l,Bj}$		linksseitiger Begrenzungswert
$y_{m,Bj}$		Mittenswert der Teilmenge Bj
$y_{r,Bj}$		rechtsseitiger Begrenzungswert
z^{-1}		Zeitschrittverschiebungsfunktion
z		Störgröße
$z_{m,Ck}$		Position des Singletons Ck

Dimensionslose Kennzahlen:

Ar	Archimedes-Zahl
Nu	Nusselt-Zahl
Pr	Prandtl-Zahl
Re	Reynolds-Zahl

8 Anhänge

Anhang A: Fuzzy-Logik-Systeme

Anhang B: Künstliche neuronale Netze

Anhang C: Neuro-Fuzzy-Systeme

Anhang D: Genetische Algorithmen

Anhang A: Fuzzy-Logik-Systeme [1-12]

Unschärfe Mengen

Die Einführung von unscharfen Mengen (*fuzzy sets*) stellt eine Erweiterung der klassischen Mengen dar. Bei klassischen Mengen ist es nur erlaubt, daß ein Element x der Grundmenge X entweder vollständig der Teilmenge A angehört oder der Teilmenge A vollständig nicht angehört. Bei unscharfen Mengen hingegen ist es erlaubt, daß ein Element x der Grundmenge X einer Teilmenge A auch nur teilweise angehört. Auskunft darüber, inwieweit ein Element x der Menge A angehört, gibt der Zugehörigkeitsgrad $\mathbf{m}_A(x)$ (*degree of membership*) [1]. Um eine unscharfe Menge A darzustellen, muß jedes Element x_i und sein Zugehörigkeitsgrad $\mathbf{m}_A(x_i)$ angegeben werden. Dabei muß für jedes möglichen Element x ein Zugehörigkeitsgrad definiert sein [3]. Der Zugehörigkeitsgrad $\mathbf{m}_A(x)$ kann dabei Werte zwischen 0 (das Element x ist nicht Element der Menge A) und 1 (das Element x ist Element der Menge A) annehmen. Besitzt mindestens ein Element x einen Zugehörigkeitsgrad von $\mathbf{m}_A(x)=1$, so wird die unscharfe Menge als normal bezeichnet [3].

Besteht die Grundmenge X (*universe of discourse*) nur aus diskreten Elementen x_i , so läßt sich eine unscharfe Menge als eine geordnete Menge von Paaren [2, 6] beschreiben.

$$A = \left\{ (x, \mathbf{m}_A(x)) \mid x \in X \right\} \quad (\text{A.1})$$

$$A = \left\{ (x_1, \mathbf{m}_A(x_1)), (x_2, \mathbf{m}_A(x_2)), \dots, (x_n, \mathbf{m}_A(x_n)) \right\} \quad (\text{A.2})$$

Auch die folgende Art der Darstellung ist bei diskreten Grundmengen X gebräuchlich [10, 11]. Der Bruchstrich und das Summenzeichen haben hier nicht die übliche mathematische Bedeutung.

$$A = \frac{\mathbf{m}_A(x_1)}{x_1} + \frac{\mathbf{m}_A(x_2)}{x_2} + \dots + \frac{\mathbf{m}_A(x_n)}{x_n} = \sum_{x \in X} \frac{\mathbf{m}_A(x)}{x} \quad (\text{A.3})$$

Besteht die Grundmenge X nicht aus diskreten Elementen x_i , sondern ist die Grundmenge X kontinuierlich, so wird das Summenzeichen durch das Integralzeichen ersetzt. Das Integralzeichen hat ebenfalls nicht die übliche Bedeutung, was durch das Fehlen des dx des Integraloperators deutlich

wird. Die Elemente x der Grundmenge werden in diesem Fall als Basisvariable bezeichnet.

$$A = \int_{x \in X} \frac{\mathbf{m}_A(x)}{x} \quad (\text{A.4})$$

Eine weitere Möglichkeit der Darstellung einer unscharfen Menge A besteht darin, den Zugehörigkeitsgrad $\mathbf{m}_A(x)$ als Funktion der Basisvariablen x anzugeben. Der funktionale Zusammenhang wird durch die Zugehörigkeitsfunktion $\mathbf{m}_A(x)$ (*membership function*) ausgedrückt.

$$\mathbf{m}_A(x) = f(x; \text{Parameter zur Beschreibung von } A) \quad (\text{A.5})$$

Die unscharfe Menge A wird durch Angabe ihrer Zugehörigkeitsfunktion $\mathbf{m}_A(x)$ vollständig und eindeutig definiert [3]. Die Darstellung über die Zugehörigkeitsfunktion benötigt nur einige Parameter, was die Repräsentation der unscharfen Menge auf einem Computer erleichtert [2].

Als Funktionen $f(x)$ für die Zugehörigkeitsfunktion können unter anderem die dreieckförmige, die trapezförmige und die gaußsche Funktion verwendet werden. Darüber hinaus existieren noch weitere Zugehörigkeitsfunktionen, die lediglich die Aufgabe erfüllen müssen, jedem Wert der Basisvariablen x einen Zugehörigkeitsgrad $\mathbf{m}_A(x)$ zwischen 0 und 1 zuzuweisen.

Die dreieckförmige Zugehörigkeitsfunktion wird durch die Parameter x_m , \mathbf{a} und \mathbf{b} beschrieben. Für Werte der Basisvariablen $x \leq x_m - \mathbf{a}$ ist die Zugehörigkeitsfunktion null. Von $x_m - \mathbf{a}$ bis x_m steigt sie auf 1 an, um dann von x_m bis $x_m + \mathbf{b}$ wieder auf null abzufallen. Für Werte $x \geq x_m + \mathbf{b}$ hat die Zugehörigkeitsfunktion den Wert null. Entsprechend ihrer Form wird die dreieckförmige Funktion auch als Λ -Funktion bezeichnet [7].

$$\mathbf{m}_A(x) = \begin{cases} 0 & \text{für } x \leq x_m - \mathbf{a} \\ 1 - \frac{x_m - x}{\mathbf{a}} & \text{für } x_m - \mathbf{a} < x \leq x_m \\ 1 + \frac{x_m - x}{\mathbf{b}} & \text{für } x_m < x < x_m + \mathbf{b} \\ 0 & \text{für } x \geq x_m + \mathbf{b} \end{cases} \quad (\text{A.6})$$

Bei der dreieckförmigen Zugehörigkeitsfunktion gehört demnach nur das Element x_m vollständig zur unscharfen Menge A . Die übrigen Werte der Basisvariablen x gehören A nur ihrem Zugehörigkeitsgrad entsprechend an. Einen Spezialfall ist die Dreiecksfunktion mit den Parametern $\mathbf{a} = \mathbf{b} = 0$, sie

wird als Singleton bezeichnet. Ein Singleton kann man auch als klassische Menge auffassen, der nur ein Element angehört.

Gehört nicht nur ein Element x_m vollständig der unscharfen Menge A an, sondern alle Werte der Basisvariablen x zwischen x_l und x_r , so wird die trapezförmige Zugehörigkeitsfunktion verwendet. Die Trapezfunktion ist für Werte $x \leq x_l - \mathbf{a}$ null, steigt von $x_l - \mathbf{a}$ bis x_l auf 1 an, und bleibt dann für $x_l \leq x \leq x_r$ auf dem Zugehörigkeitsgrad 1. Im Bereich von x_r bis $x_r + \mathbf{b}$ fällt die Funktion von 1 auf null ab, um für Werte $x \geq x_r + \mathbf{b}$ auf null zu bleiben. Für diese Funktion wird auch die Bezeichnung Π -Funktion benutzt [7].

$$\mathbf{m}_A(x) = \begin{cases} 0 & \text{für } x \leq x_l - \mathbf{a} \\ 1 - \frac{x_l - x}{\mathbf{a}} & \text{für } x_l - \mathbf{a} < x < x_l \\ 1 & \text{für } x_l \leq x \leq x_r \\ 1 + \frac{x_r - x}{\mathbf{b}} & \text{für } x_r < x < x_r + \mathbf{b} \\ 0 & \text{für } x \geq x_r + \mathbf{b} \end{cases} \quad (\text{A.7})$$

Einen Sonderfall stellt die trapezförmige Funktion mit den Parametern $\mathbf{a} = \mathbf{b} = 0$ dar. Die Zugehörigkeitsfunktion beschreibt dann eine klassische, scharfe Menge, der alle Werte der Basisvariablen x von x_r bis x_l angehören.

Dreieckförmige und trapezförmige Zugehörigkeitsfunktionen werden zusammenfassend als Zugehörigkeitsfunktionen mit steilen Flanken bezeichnet. Sie lassen sich leicht programmieren und stellen keine hohen Anforderungen an die Rechnerleistung [2]. Es ist ein Nachteil dieser Funktionen, daß sie nicht stetig differenzierbar sind, was ihren Einsatz in Systemen behindert [3], die mit einem Gradientenverfahren optimiert werden sollen.

Den Vorteil der Differenzierbarkeit weist die glockenförmige gaußsche Zugehörigkeitsfunktion auf. In [3] wird vorgeschlagen die Dreieckfunktion in eine flächengleiche Gauß-Funktion zu überführen.

$$\mathbf{m}_A(x) = \begin{cases} \exp\left[-\frac{\mathbf{p}}{\mathbf{a}^2}(x - x_m)^2\right] & \text{für } x \leq x_m \\ \exp\left[-\frac{\mathbf{p}}{\mathbf{b}^2}(x - x_m)^2\right] & \text{für } x > x_m \end{cases} \quad (\text{A.8})$$

Sind die Parameter a und b gleich, kann bei der gaußschen Funktion die gemachte Fallunterscheidung wegfallen. Es existiert dann nur eine Funktion für alle Werte der Basisvariablen x .

Auf gleiche Weise kann die Trapez-Funktion (Π -Funktion) in eine Gauß-Trapez-Funktion überführt werden. Dabei werden die steilen Flanken der Π -Funktion ersetzt durch Teile der gaußschen Funktion. Wie die trapezförmige Funktion benötigt die Gauß-Trapez-Funktion die vier Parameter x_l , x_r , a und b .

$$\mathbf{m}_A(x) = \begin{cases} \exp\left[-\frac{p}{a^2}(x-x_l)^2\right] & \text{für } x < x_l \\ 1 & \text{für } x_l \leq x \leq x_r \\ \exp\left[-\frac{p}{b^2}(x-x_r)^2\right] & \text{für } x > x_r \end{cases} \quad (\text{A.9})$$

Zu beachten ist bei der Gauß-Funktion und der Gauß-Trapez-Funktion, daß jeder Wert der Basisvariablen x einen Zugehörigkeitsgrad > 0 zur Menge A aufweist, wenn auch die Zugehörigkeitsgrade sehr klein werden können.

Eine weitere Funktion ist das Singleton. Bei ihm ist nur für einen einzigen (scharf definierten) Wert der Basisvariablen $x = x_m$ der Zugehörigkeitsgrad $\mathbf{m}_A(x_m) = 1$. Ein Singleton besitzt keinen Unschärfbereich und ist somit eine klassische Menge mit nur einem Element [1]. Wie bereits angesprochen, geht eine dreieckförmige Zugehörigkeitsfunktion mit den Parametern $a = b = 0$ in ein Singleton über.

$$\mathbf{m}_A(x) = \begin{cases} 1 & \text{für } x = x_m \\ 0 & \text{sonst} \end{cases} \quad (\text{A.10})$$

Die hier aufgeführten Zugehörigkeitsfunktionen $\mathbf{m}_A(x)$ definieren normale und konvexe unscharfe Mengen. Als normale Mengen besitzen sie mindestens ein Element x mit dem Zugehörigkeitsgrad $\mathbf{m}_A(x)=1$. Als konvex werden sie bezeichnet, weil die Zugehörigkeitsfunktionen nur globale Maxima aufweisen ohne weitere lokale Maxima [10].

Mengenoperationen

Mit Mengenoperatoren werden Mengen bearbeitet oder miteinander verknüpft. Die Mengenoperationen umfassen die Bildung der Vereinigungsmenge, die Bildung der Durchschnittsmenge und die Bildung der Komplementmenge, wobei für klassische Mengen die drei Operatoren Vereinigung (\cup), Durchschnitt (\cap) und Komplement zur Anwendung kommen. Für unscharfe Mengen mit Zugehörigkeitsgraden $m_A(x)$ zwischen 0 und 1 lassen sich jedoch unterschiedliche Vereinigungsmengen (oder Durchschnittsmengen) erzeugen, da eine Vielzahl von Operatoren zur Bildung der Vereinigungsmenge (oder Durchschnittsmenge) existieren. Neben den hier aufgeführten Operatoren gibt es noch weitere [2], die aber für Fuzzy-Logik-Systeme weniger nützlich sind. Darüberhinaus gibt es für unscharfe Mengen zusätzlich „unscharfe“ Operatoren, die sich über Parameter „fließend“ zwischen Vereinigung und Durchschnitt verstellen lassen [2].

Während bei Mengenoperationen alle Elemente x der Grundmenge X betrachtet werden, und das Ergebnis einer Mengenoperation wieder eine Menge ist, wird bei Logikoperationen nur die Eigenschaft eines Elements x_i betrachtet, und das Ergebnis einer Logikoperation ist wieder eine Eigenschaft eines Elements [6]. Jeder Mengenoperator kann daher in einen korrespondierenden Logikoperator überführt werden, wenn lediglich ein x_i und nicht alle $x \in X$ betrachtet werden [11]. Die klassische Logik arbeitet mit den Eigenschaften wahr (1, *true*) und falsch (0, *false*), sowie den Logikoperatoren UND (*AND*), ODER (*OR*) und NICHT (*NOT*). In der unscharfen Logik (*fuzzy logic*) ist eine Eigenschaft beliebig zwischen wahr (1) und falsch (0) wählbar, und es existieren mehrere Logikoperatoren, welche sich aus den korrespondierenden Mengenoperatoren ableiten. Alle Logikoperatoren beinhalten den Fall der klassischen zweiwertigen Logik als Spezialfall [2].

Die im folgenden dargestellten Operatoren (Co-T-Normen [2]) eignen sich zur Bildung der Vereinigungsmenge [14]. Diese Operatoren korrespondieren mit der ODER-Verknüpfung (*OR*) in der unscharfen Logik [11].

Maximum

$$m_{A \cup B}(x) = \text{MAX}(m_A(x), m_B(x)) \quad \forall x \in X \quad (\text{A.11})$$

Algebraische Summe (*algebraic sum*)

$$m_{A \cup B}(x) = m_A(x) + m_B(x) - m_A(x)m_B(x) \quad \forall x \in X \quad (\text{A.12})$$

Begrenzte Summe (*bounded sum*, Lukasiewicz-Oder)

$$\mathbf{m}_{A \cup B}(x) = \text{MIN}(1, \mathbf{m}_A(x) + \mathbf{m}_B(x)) \quad \forall x \in X \quad (\text{A.13})$$

Hamacher Summe

$$\mathbf{m}_{A \cup B}(x) = \frac{\mathbf{m}_A(x) + \mathbf{m}_B(x) - 2\mathbf{m}_A(x)\mathbf{m}_B(x)}{1 - \mathbf{m}_A(x)\mathbf{m}_B(x)} \quad \forall x \in X \quad (\text{A.14})$$

Folgende Operatoren (T-Normen [2]) eignen sich zur Bildung der Durchschnittsmenge [14]. Diese Operatoren korrespondieren mit der UND-Verknüpfung (*AND*) in der unscharfen Logik [11].

Minimum

$$\mathbf{m}_{A \cap B}(x) = \text{MIN}(\mathbf{m}_A(x), \mathbf{m}_B(x)) \quad \forall x \in X \quad (\text{A.15})$$

Algebraisches Produkt (*algebraic product*)

$$\mathbf{m}_{A \cap B}(x) = \mathbf{m}_A(x)\mathbf{m}_B(x) \quad \forall x \in X \quad (\text{A.16})$$

Begrenzte Differenz (*bounded difference*, Lukasiewicz-UND)

$$\mathbf{m}_{A \cap B}(x) = \text{MAX}(0, \mathbf{m}_A(x) + \mathbf{m}_B(x) - 1) \quad \forall x \in X \quad (\text{A.17})$$

Hamacher Produkt

$$\mathbf{m}_{A \cap B}(x) = \frac{\mathbf{m}_A(x)\mathbf{m}_B(x)}{\mathbf{m}_A(x) + \mathbf{m}_B(x) - \mathbf{m}_A(x)\mathbf{m}_B(x)} \quad \forall x \in X \quad (\text{A.18})$$

In den oben aufgeführten Fällen wurden zwei Mengen A und B miteinander verknüpft, die auf derselben Grundmenge X definiert sind. Die Mengenoperatoren lassen sich jedoch auch anwenden, wenn eine Menge A auf der Grundmenge X definiert ist ($\mathbf{m}_A(x)$) und die andere Menge B auf der Grundmenge Y definiert ist ($\mathbf{m}_B(y)$). Es muß dann berücksichtigt werden, daß die Vereinigungs- oder Durchschnittsmenge C auf einer zweidimensionalen Grundmenge $X \times Y$ (Cartesisches Produkt) definiert ist. Durch geeignete Operationen (z.B. Projektion) kann die Dimensionalität der Menge C anschließend wieder verringert werden [10].

$$\mathbf{m}_C(x, y) = \text{Operator}(\mathbf{m}_A(x), \mathbf{m}_B(y)) \quad \forall x \in X, y \in Y \quad (\text{A.19})$$

Die Komplementmenge wird gebildet, indem der Komplement-Operator auf die unscharfe Menge angewendet wird. Bei Bildung der Komplementmenge ist es wichtig, daß die unscharfe Menge eine normale Menge ist, in der keine Zugehörigkeitsgrade größer 1 auftreten [6]. Ferner muß die Grundmenge X genau definiert sein. Die Bildung der Komplementmenge entspricht der NICHT-Operation (*NOT*) in der unscharfen Logik.

Komplement

$$\mathbf{m}_{\bar{A}}(x) = 1 - \mathbf{m}_A(x) \quad \forall x \in X \quad (\text{A.20})$$

Multiplikation einer unscharfen Zahl

Ist eine unscharfe Menge normal und konvex (z.B. mit einer dreieckförmigen Zugehörigkeitsfunktion, beschrieben durch x_m , \mathbf{a} und \mathbf{b}), so kann diese Menge auch als unscharfe Zahl (*fuzzy number*) aufgefaßt werden. Sie wird durch den umgangssprachlichen (linguistischen) Ausdruck „ungefähr x_m “ ausgedrückt. In [10] wird gezeigt, wie mit solchen unscharfen Zahlen mathematische Operationen ausgeführt werden können. Für die Multiplikation „ \cdot “ einer klassischen Zahl c (*crisp number*) mit einer unscharfen Zahl A (beschrieben durch x_{m1} , \mathbf{a}_1 , \mathbf{b}_1) ist das Ergebnis:

$$\mathbf{m}_{c \cdot A}(x) = f(x; x_{m2}, \mathbf{a}_2, \mathbf{b}_2) \quad (\text{A.21})$$

$$x_{m2} = c \cdot x_{m1} ; \quad \mathbf{a}_2 = c \cdot \mathbf{a}_1 ; \quad \mathbf{b}_2 = c \cdot \mathbf{b}_1 \quad (\text{A.22})$$

Die unscharfe Zahl „ungefähr x_{m1} “ multipliziert mit der klassischen, scharfen Zahl c führt zu dem unscharfen Ergebnis „ungefähr $(c \cdot x_{m1})$ “, wobei sich der Unschärfebereich ebenfalls um den Faktor c ändert.

Diese Operation kann für die Skalierung der unscharfen Eingangs- und Ausgangsmengen eines Fuzzy-Logik-Systems wichtig sein. Sie ist nicht mit der Mengenoperation Algebraisches Produkt zu verwechseln, bei dem die Zugehörigkeitsgrade $\mathbf{m}_A(x)$ für jedes $x \in X$ mit $\mathbf{m}_B(x)$ multipliziert werden.

Algorithmus eines Fuzzy-Logik-Systems

In einem Fuzzy-Logik-System werden die theoretischen Grundlagen der unscharfen Mengen zu einem Algorithmus verknüpft. Eingangsgrößen des Fuzzy-Logik-Systems sind meist scharfe Meßwerte, und als Ausgangsgröße wird meist ebenfalls ein scharfer Wert verlangt. Fuzzy-Logik-Systeme weisen den Vorteil auf, daß sie entlang linguistischer (umgangs-

sprachlicher) Pfade entwickelt werden, die auch umgangssprachlich verstanden werden können. Hinter dieser Darstellung steht aber die präzise Mathematik der unscharfen Mengen und der unscharfen Logik. Der Algorithmus eines Fuzzy-Logik-Systems wird in drei Schritte unterteilt:

- Fuzzifizierung der Eingangsgrößen (*fuzzification*),
- Bildung der unscharfen Ausgangsmenge (*inference*) anhand der Regelbasis (auch Wissensbasis, *knowledge base*, *fuzzy associative maps* (FAM)) unter Verwendung der Zugehörigkeitsgrade aus der Fuzzifizierung,
- Defuzzifizierung der unscharfen Ausgangsmenge (*defuzzification*) zur Berechnung einer scharfen Ausgangsgröße.

Der Algorithmus wird hier für ein System mit zwei Eingangsgrößen (x^* , y^*) und einer Ausgangsgröße (z^*) beispielhaft beschrieben. In der folgenden Abbildung 55 ist der Algorithmus eines Fuzzy-Logik-Systems mit den drei Schritten Fuzzifizierung, regelbasiertes unscharfes Schließen und Defuzzifizierung grafisch dargestellt.

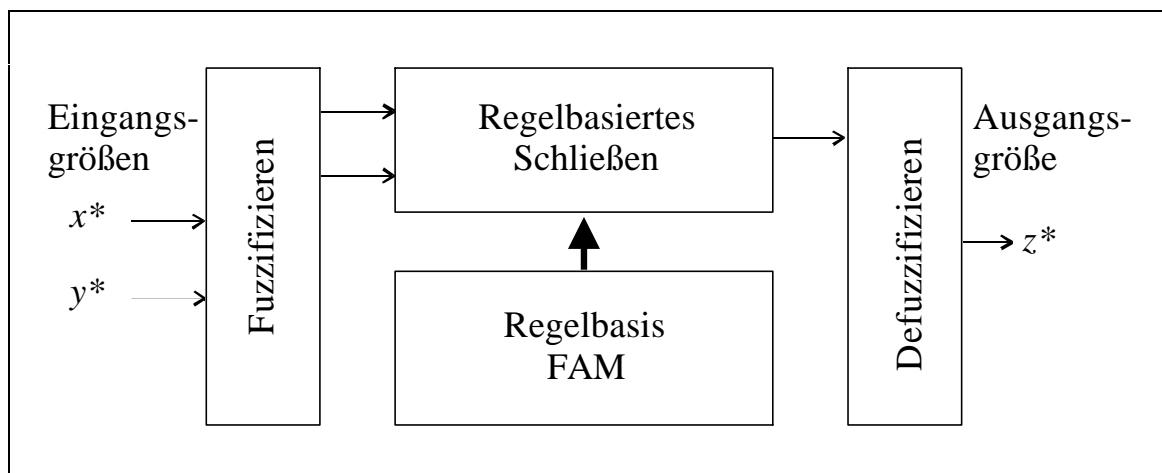


Abb. 54 Funktionsweise eines Fuzzy-Logik-Systems

Fuzzifizierung

Der scharfe Eingangswerte x^* wird vom Fuzzy-Logik-System nicht direkt verarbeitet. Vielmehr wird der gesamte Bereich der Basisvariablen x in n sich überlappende unscharfe Mengen A_i , Terme genannt, unterteilt [1, 3, 5]. Diese Terme A_i sind Werte der linguistischen Variablen A , und mit ihnen arbeitet das Fuzzy-Logik-System. Der Vorgang der Fuzzifizierung ist die Abbildung eines scharfen Wertes x^* der Basisvariablen x auf den Bereich der linguistischen Variable A mit ihren Werten (Termen) A_i [3]. Dazu werden die Zugehörigkeitsgrade $m_{A_i}(x^*)$ des scharfen Wertes x^* zu den einzelnen Termen

A_i der linguistischen Variable A bestimmt. Ergebnis dieses Vorgangs ist ein Vektor $\mathbf{m}_A(i)$, der die Zugehörigkeitsgrade zu den einzelnen Termen A_i enthält. Jeder Term A_i wird durch die Parameter der Zugehörigkeitsfunktion beschrieben (\mathbf{a}_i , \mathbf{b}_i und $(x_m)_i$ bzw. $(x_l)_i$ und $(x_r)_i$).

Der scharfe Wert y^* wird auf gleiche Weise in den Vektor $\mathbf{m}_B(j)$ überführt. Die linguistische Variable B ist unterteilt in m Terme B_j basierend auf der Basisvariablen y . Die so erzeugten Vektoren werden im nächsten Schritt verarbeitet.

$$\mathbf{m}_A(i) = \begin{pmatrix} \mathbf{m}_{A_1}(x^*) \\ \mathbf{m}_{A_2}(x^*) \\ \vdots \\ \mathbf{m}_{A_i}(x^*) \\ \vdots \\ \mathbf{m}_{A_m}(x^*) \end{pmatrix}, \quad \mathbf{m}_B(j) = \begin{pmatrix} \mathbf{m}_{B_1}(y^*) \\ \mathbf{m}_{B_2}(y^*) \\ \vdots \\ \mathbf{m}_{B_j}(y^*) \\ \vdots \\ \mathbf{m}_{B_m}(y^*) \end{pmatrix} \quad (\text{A.23})$$

Regelbasiertes Schließen

Dieser Teil des Algorithmus eines Fuzzy-Logik-Systems wird als regelbasiertes unscharfes Schließen [4] bezeichnet, da in diesem Teil auf Basis der unscharfen Logik gearbeitet wird und auch nur ein unscharfes Resultat erzeugt wird. In der Regelbasis wird das Verhalten eines Fuzzy-Logik-Systems beim Vorliegen der Terme A_i und B_i beschrieben. Eine Regel hat die allgemeine Form.

$$\text{WENN } A = A_i \text{ UND } B = B_j, \text{ DANN } C = C_k \quad (\text{A.24})$$

Der *WENN*-Teil der Regel wird als Prämisse, und der *DANN*-Teil wird als Konklusion (Schlußfolgerung) bezeichnet. Im Fall von zwei Eingangsgrößen gliedert sich die Prämisse in zwei Teilbedingungen, welche durch einen UND-Operator miteinander verknüpft sind. Die Schlußfolgerungen C_k (mit $k=1..o$) sind Terme der linguistischen Variable C . Die unscharfen Mengen C_k sind auf der Basisvariablen z (den Werten der Ausgangsgröße) definiert.

Nur wenn beide Teilbedingungen zumindest teilweise erfüllt sind, d.h. weder $\mathbf{m}_{A_i}(x^*)=0$ noch $\mathbf{m}_{B_j}(y^*)=0$, trägt die Regel mit ihrer Schlußfolgerung C_k zur Ausgangsmenge bei. Für den Grad, mit dem eine Schlußfolgerung zutrifft (Grad der Entschlossenheit), legt man zweckmäßigerweise fest, daß er dem Grad entspricht, mit dem die Prämisse erfüllt ist (Erfüllungsgrad) [1].

Damit gelangt man zu dem Ergebnis, daß eine teilweise erfüllte Prämisse zu einer teilweise gültigen Schlußfolgerung führt.

Entsprechend der Anzahl der Terme n der linguistischen Variable A und der Anzahl der Terme m der linguistischen Variable B würde man zu einer Regelbasis aus $n \times m$ Regeln gelangen, wenn jeder Termkombination der Eingangsgrößen eine von o Schlußfolgerungen zugeordnet würde. Eine solche Regelbasis wird als vollständig bezeichnet [3, 9]. Führt darüber hinaus jede Termkombination nur zu genau einer Schlußfolgerungen (keine sich widersprechenden Regeln), so wird die Regelbasis als konsistent bezeichnet.

Von der Vorstellung, daß jede Regel zu 100 Prozent gültig ist, entfernt sich das Konzept der unscharfen assoziativen Karten (*fuzzy associative maps*, FAMs). Es beinhaltet die Möglichkeit, daß auch eine Regel zu weniger als 100 Prozent gültig sein kann, was näher am Grundverständnis der unscharfen Logik liegt. Jeder Regel ist ein Relevanzfaktor zwischen 0 und 1 zugeordnet, der angibt inwieweit die Regel gültig ist. So können auch Regeln mit gleichen Prämissen und unterschiedlichen Konklusionen gleichzeitig aktiv sein [5]. Die allgemeine Form einer Regel wird erweitert auf.

$$\text{WENN } A = A_i \text{ UND } B = B_j, \text{ DANN } C = C_k \text{ MIT } \mathbf{m}_{FAM}(i, j, k) \quad (\text{A.25})$$

Im Gegensatz zur vollständigen, konsistenten Regelbasis mit $m \times n$ Regeln, müssen die unscharfen assoziativen Karten $n \times m \times o$ Einträge enthalten: Jede Termkombination aus n Termen A_i und m Termen B_j kann zu jeder der o Schlußfolgerungen C_k führen, versehen mit dem Relevanzfaktor $\mathbf{m}_{FAM}(i, j, k)$.

Im folgenden soll das erweiterte Konzept der unscharfen assoziativen Karten verwendet werden.

Zur Berechnung des WENN-Teils der Regeln wird ausgehend von den beiden Vektoren $\mathbf{m}_A(i)$ und $\mathbf{m}_B(j)$ über eine UND-Operation eine Aggregationsmatrix $\mathbf{m}_{Aggregation}(i, j)$ erzeugt. In ihr sind die Erfüllungsgrade jeder denkbaren Prämisse (d.h. jeder möglichen Termkombination von A_i und B_j) vermerkt. Als UND-Operator kann jeder oben genannte UND-Operator der unscharfen Logik verwendet werden. Hier wird einfacherweise der MIN-Operator verwendet.

$$\mathbf{m}_{Aggregation}(i, j) = \text{MIN}(\mathbf{m}_A(i), \mathbf{m}_B(j)) \quad \begin{array}{l} \text{für } i = 1..n \\ \text{und } j = 1..m \end{array} \quad (\text{A.26})$$

Die Aggregationsmatrix $\mathbf{m}_{\text{Aggregation}}(i,j)$, die angibt, welche Termkombinationen erfüllt sind, wird elementweise mit den Relevanzfaktoren $\mathbf{m}_{\text{FAM}}(i,j,k)$ der Regeln multipliziert.

$$\mathbf{m}_{\text{Produkt}}(i,j,k) = \mathbf{m}_{\text{Aggregation}}(i,j)\mathbf{m}_{\text{FAM}}(i,j,k) \quad \text{für } i = 1..n \quad \text{und } j = 1..m \quad \text{und } k = 1..o \quad (\text{A.27})$$

In diesem dreidimensionalen Feld der Relevanzfaktoren $\mathbf{m}_{\text{FAM}}(i,j,k)$ werden alle *WENN-DANN*-Verknüpfungen erfaßt. Lautet eine zu 100 Prozent gültige Regel:

$$\text{WENN } A = A2 \text{ UND } B = B4, \text{ DANN } C = C1, \quad (\text{A.28})$$

so wird dies durch $\mathbf{m}_{\text{FAM}}(2,4,1) = 1$ ausgedrückt. Hieran wird deutlich, daß bei der Verwendung von unscharfen assoziativen Karten eine Regelbasis dann als konsistent und vollständig bezeichnet werden kann, wenn zu jeder Kombination aus $i=1..n$ und $j=1..m$ für genau ein k (aus $k=1..o$) $\mathbf{m}_{\text{FAM}}(i,j,k)=1$ gilt.

Eine Schlußfolgerung Ck , als ein Term der linguistischen Variable C , kann von mehreren Regeln angesprochen werden. Bei mehreren vorliegenden Entschlossenheitsgraden für dieselbe Schlußfolgerung ist es zweckmäßig, nur den höchsten zu berücksichtigen [3] (*first-accumulate-then-defuzzify* [26]). Für diesen Vorgang der Akkumulation werden für jede einzelne Schlußfolgerung Ck mit $k=1..o$ die Elemente des Feldes $\mathbf{m}_{\text{Produkt}}(i,j,k)$ über alle $i=1..n$ und $j=1..m$ über eine ODER-Operation zusammengefaßt. Als Operator kann einer der oben genannten ODER-Operatoren der unscharfen Logik verwendet werden. Hier steht stellvertretend der MAX-Operator.

$$\mathbf{m}_{\text{Akkumulation}}(k) = \underset{i,j}{\text{MAX}}(\mathbf{m}_{\text{Produkt}}(i,j,k)) \quad \text{für } k = 1..o \quad (\text{A.29})$$

Ergebnis dieses Vorgangs ist ein Vektor $\mathbf{m}_{\text{Akkumulation}}(k)$, der angibt inwieweit jede unscharfe Menge Ck (Schlußfolgerung) zur resultierenden unscharfen Ausgangsmenge $\mathbf{m}_{\text{C,res}}(z)$ beiträgt.

Defuzzifizierung

Die unscharfen Mengen Ck (definiert als $\mathbf{m}_{Ck}(z)$ auf der Basisvariablen z) sind benutzerdefiniert festgelegt, und die Entschlossenheitsgrade $\mathbf{m}_{\text{Akkumulation}}(k)$ sind aus dem vorhergehenden Schritt bekannt. Zur Vorgehensweise, wie

eine Menge C_k zu modifizieren ist, um dem Entschlossenheitsgrad $\mathbf{m}_{\text{Akkumulation}}(k)$ Rechnung zu tragen, existieren zwei Verfahren.

Bei dem als *Clipping* [6] bezeichneten Verfahren wird die modifizierte Menge $C_{k,\text{mod}}$ aus der unscharfen Menge C_k gebildet, indem die Menge C_k über den Minimumoperator in Höhe des Entschlossenheitsgrades $\mathbf{m}_{\text{Akkumulation}}(k)$ abgeschnitten wird.

$$\mathbf{m}_{C_{k,\text{mod}}}(z) = \text{MIN}(\mathbf{m}_{C_k}(z), \mathbf{m}_{\text{Akkumulation}}(k)) \quad \begin{array}{l} \text{für } k = 1..o \\ \text{und } \forall z \in Z \end{array} \quad (\text{A.30})$$

Zu einer anderen Form der unscharfen Menge $C_{k,\text{mod}}$ führt das als *Scaling* [6] bezeichnete Verfahren. $C_{k,\text{mod}}$ wird gebildet, indem die Zugehörigkeitsfunktion der unscharfen Menge C_k mit dem Entschlossenheitsgrad $\mathbf{m}_{\text{Akkumulation}}(k)$ algebraisch multipliziert wird.

$$\mathbf{m}_{C_{k,\text{mod}}}(z) = \mathbf{m}_{C_k}(z) \mathbf{m}_{\text{Akkumulation}}(k) \quad \begin{array}{l} \text{für } k = 1..o \\ \text{und } \forall z \in Z \end{array} \quad (\text{A.31})$$

Zur Bildung der resultierenden Ausgangsmenge $\mathbf{m}_{C,\text{res}}(z)$ werden die o modifizierten Ausgangsmengen $C_{k,\text{mod}}$ überlagert. Der Maximumoperator führt dazu, daß für einen Wert der Basisvariablen z jeweils die maximale Zugehörigkeitsfunktion $\mathbf{m}_{C_{k,\text{mod}}}(z)$ aus $k=1..o$ bestimmend wird.

$$\mathbf{m}_{C,\text{res}}(z) = \text{MAX}_k(\mathbf{m}_{C_{k,\text{mod}}}(z)) \quad \forall z \in Z \quad (\text{A.32})$$

Aus der resultierenden Ausgangsmenge $\mathbf{m}_{C,\text{res}}(z)$ muß nun ein sinnvoller scharfer Ausgangswert z^* generiert werden [3]. Bei der Schwerpunkt-methode (*center of area*, COA) wird z^* bestimmt als der Wert der Basisvariablen bei der die resultierende Ausgangsmenge ihren Schwerpunkt hat.

$$z^* = \frac{\int_Z z \mathbf{m}_{C,\text{res}}(z) dz}{\int_Z \mathbf{m}_{C,\text{res}}(z) dz} \quad (\text{A.33})$$

Bei der modifizierte Schwerpunkt-methode, der *Center of Sum*-Methode (COS), wird bereits bei der Bildung der resultierenden Ausgangsmenge $\mathbf{m}_{C,\text{res}}(z)$ eine Änderung vorgenommen. In (A.32) wird der MAX-Operator ersetzt durch die Summe Σ über alle Schlußfolgerungen k .

$$\mathbf{m}_{C,\text{res}}(z) = \sum_k (\mathbf{m}_{C_{k,\text{mod}}}(z)) \quad \forall z \in Z \quad (\text{A.34})$$

Zur Ermittlung des scharfen Ausgangswertes z^* findet anschließend wieder (A.33) Verwendung. Der Unterschied zur COA-Methode besteht bei der COS-Methode darin, daß Werte von z , bei denen sich die Zugehörigkeitsfunktionen $\mathbf{m}_{Ck,mod}(z)$ der o unscharfen Ausgangsmengen Ck,mod überschneiden entsprechend höher gewichtet werden. Diese Überschneidung wird bei der COA-Methode aufgrund des MAX-Operators nicht berücksichtigt.

Die Schwerpunktberechnung bei den Schwerpunktmethoden (COA und COS) kann recht aufwendig werden, weil die Integrale der resultierenden unscharfen Ausgangsmenge $\mathbf{m}_{C,res}(z)$ in (A.33) durch die MAX-Operation bzw. die Summenbildung bei der Überlagerung der einzelnen Mengen $\mathbf{m}_{Ck,mod}(z)$ eine komplexe Struktur annehmen können [3].

Eine einfachere Art der Defuzzifizierung kann angewendet werden, wenn die Terme Ck der linguistischen Variable C Singletons sind. Für die als Höhenmethode (*height method*) bezeichnete Methode wird direkt der Vektor $\mathbf{m}_{Akkumulation}(k)$ verwendet, der vom regelbasierten Schließen her stammt. Der Wert $(z_m)_k$ wird aus der Singleton-Zugehörigkeitsfunktion des Terms Ck entnommen.

$$z^* = \frac{\sum_k (z_m)_k \mathbf{m}_{Akkumulation}(k)}{\sum_k \mathbf{m}_{Akkumulation}(k)} \quad (\text{A.35})$$

Neben den hier aufgeführten Methoden zur Bildung eines scharfen Ausgangswertes (Schwerpunktmethoden, Höhenmethode) existieren noch [3, 10]: die Methode des Mittelwertes der Maxima (*mean of maxima*, MOM), die Methode zur Wahl des linken Randpunktes (*first of maxima*) und die Methode zur Wahl des rechten Randpunktes (*last of maxima*). Ihnen allen ist gemeinsam, daß im wesentlichen die Schlußfolgerung mit dem höchsten Entschlossenheitsgrades zum Ausgangswert z^* beiträgt.

Bei der Programmierung des Algorithmus bestehen viele Wahlmöglichkeiten bezüglich der verwendeten unscharfen Mengen und Operatoren, was es erlaubt, das Übertragungsverhalten eines Fuzzy-Logik-Systems in weiten Grenzen zu variieren. Beim Entwurf eines Fuzzy-Logik-Systems müssen Entscheidungen zu folgenden Einflußfaktoren getroffen werden:

1. die Anzahl der Terme der Eingangsvariablen und Ausgangsvariable,
2. die Form der Zugehörigkeitsfunktionen der Terme,
3. die Lage der Maxima der Zugehörigkeitsfunktionen,

4. die Breite der Zugehörigkeitsfunktionen,
5. der Operator zur UND-Verknüpfung der Teilbedingungen der Prämissen,
6. die Einträge der Regelbasis, bzw. der unscharfen assoziativen Karten,
7. der Operator zur Akkumulation, wenn dieselbe Schlußfolgerung von mehreren Regeln angesprochen wird,
8. der Operator, der den Einfluß der Entschlossenheitsgrade auf die Terme der Ausgangsgröße ausdrückt, und
9. die Defuzzifizierungsmethode.

Mit einem Fuzzy-Logik-System läßt sich z.B. ein Fuzzy-Logik-Regler (*fuzzy logic controller*, FLC) realisieren. Als Eingangsgrößen können die Regelabweichung x_w sowie die zeitliche Änderung der Regelabweichung $\Delta x_w / \Delta t_a$ verwendet werden. Ausgangsgröße kann dann die Änderung der Stellgröße Δu sein (PI-Fuzzy-Logik-Regler). Ein Fuzzy-Logik-Regler gehört zur Gruppe der Kennfeldregler [4]. Er besitzt eine statische Struktur, d.h. dieselben Eingangswerte führen immer zu demselben Ausgangswert. Aufgrund der oben gezeigten Variationsmöglichkeiten eines Fuzzy-Logik-Systems läßt sich jedes gewünschte Übertragungsverhalten einstellen.

Verwendete Symbole Anhang A

a	Linksseitige Breite der Zugehörigkeitsfunktion
A	Teilmenge der Grundmenge X
b	Rechtsseitige Breite der Zugehörigkeitsfunktion
B	Teilmenge der Grundmenge Y
c	Klassische, scharfe Zahl
C	Teilmenge der Grundmenge Z
i, j, k	Laufindizes
m	Anzahl der Teilmengen (Terme) B
$\mathbf{m}_A(x)$	Zugehörigkeitsgrad des Elements x zur Teilmenge A
$\mathbf{m}_{\text{Aggregation}}(i,j)$	Aggregationsmatrix, entstanden aus den Vektoren $\mathbf{m}_A(i)$, $\mathbf{m}_B(j)$
$\mathbf{m}_{\text{Akkumulation}}(k)$	Akkumulationsvektor, enthält den Grad der Entschlossenheit bezüglich der Schlußfolgerung k
$\mathbf{m}_{Ck,mod}(z)$	Modifizierte Ausgangsmenge Ck über der Ausgangsvariable z
$\mathbf{m}_{C,res}(z)$	Resultierende Ausgangsmenge C , entstanden aus der Zusammenfassung der k modifizierten Ausgangsmengen Ck,mod
$\mathbf{m}_{FAM}(i,j,k)$	Unscharfe assoziative Karten, dreidimensionales Feld der Relevanzfaktoren aller ij -Regeln

$\mathbf{m}_{Produkt}(i,j,k)$	dreidimensionales Produktfeld, entstanden aus der elementweisen Multiplikation von $\mathbf{m}_{Aggregation}(i,j)$ und $\mathbf{m}_{FAM}(i,j,k)$
n	Anzahl der Teilmengen (Terme) A
o	Anzahl der Teilmengen (Terme) C
Δt_a	Abtastperiode
u	Stellgröße
x	Element der Grundmenge X
x^*	Eingangsgröße des Fuzzy-Logik-Systems (Element von X)
x_l	Linksseitiger Begrenzungswert der Π -Zugehörigkeitsfunktion
x_m	Modalwert der Λ -Zugehörigkeitsfunktion
x_r	Rechtsseitiger Begrenzungswert der Π -Zugehörigkeitsfunktion
x_w	Regelabweichung
X	Grundmenge
y	Element der Grundmenge Y
y^*	Eingangsgröße des Fuzzy-Logik-Systems (Element von Y)
z	Element der Grundmenge Z
z^*	Ausgangsgröße des Fuzzy-Logik-Systems (Element von Z)

Anhang B: Künstliche Neuronale Netze [13-23]

Obwohl es ihr Name „künstliche neuronale Netze“ vermuten läßt, haben diese Programme nur wenig mit der Biologie oder der Neurophysiologie zu tun. Vielmehr werden neuronale Netze in den Ingenieurwissenschaften als Funktionsapproximatoren eingesetzt, die jeden beliebigen funktionalen Zusammenhang zwischen Ein- und Ausgabedaten darstellen können [18], wenn sie diesen Zusammenhang aus repräsentativen Ein- und Ausgabedaten erlernt haben [17]. Man spricht auch davon, daß das künstliche neuronale Netz anhand vorliegender Eingabe-Ausgabe-Paare trainiert wird.

Diese Fähigkeit macht neuronale Netze wertvoll für eine Modellbildung von realen Systemen, von denen genügend Meßwerte an Eingangs- und Ausgangsgrößen vorliegen. Ein Nachteil neuronaler Netze ist eine fehlende Interpretierbarkeit der in ihnen enthaltenen Parametern [15, 24]. Dies unterscheidet sie von physikalischen Modellen, in denen die enthaltenen Parameter eine physikalische Bedeutung besitzen.

Wegen ihres Ursprungs als rechnerische Modelle des Gehirns [13, 17, 24] existiert eine Vielzahl unterschiedlicher Arten an künstlichen neuronalen Netzen [15, 17]. Der Netztyp des Mehrschicht-Perzeptrons (*multilayer perceptron*, *feedforward neural network* oder *backpropagation neural network*) mit dem dazugehörigen Fehlerrückführungslernverfahren (*error-backpropagation*) kann für eine Systemidentifikation anhand von Ein- und Ausgangsdaten als ausreichend betrachtet werden [18, 33]. Die folgende Darstellung beschränkt sich daher auf diesen Netztyp.

Neuronen und künstliche neuronale Netze

In der folgenden Abbildung 56 ist ein dreischichtiges Perzeptron dargestellt. Es besteht aus einer Anzahl einfacher Verarbeitungseinheiten, den Neuronen. Diese Neuronen sind in drei Schichten (*layers*) angeordnet, daher die Bezeichnung als dreischichtiges Netz. Die Wahl von drei Schichten ist nicht zwingend notwendig. Vielmehr ist die Anzahl der Schichten ein frei wählbarer Parameter des neuronalen Netzes.

Eingegeben wird in das neuronale Netz der Eingabevektor x_h ($h = 1..n-1$), vom Netz ausgegeben wird der Ausgabevektor o_k ($k = 1..p$).

Der Eingabevektor x_h wird an die Neuronen der Eingabeschicht (*input layer*) geleitet. Die Neuronen i der Eingabeschicht führen keine Berechnungen

durch, sondern ergänzen den Eingabevektor x_h um eine weitere Dimension und geben den so erzeugten Vektor o_i ($i = 1..n$) an alle Neuronen der verborgenen Schicht (*hidden layer*) [17] weiter. Der Dimension n wird ein Wert von 1 zugewiesen. Dadurch wird erreicht, daß die Neuronen der verborgenen Schicht einen zusätzlichen Eingabewert besitzen, den Biaswert [14]. Durch den Biaswert wird die Anzahl der Freiheitsgrade eines Neurons erhöht. Formal entspricht seine Einführung der Verschiebung des effektiven Eingangs e_j , wodurch ein von Null verschiedener Schwellenwert ermöglicht wird.

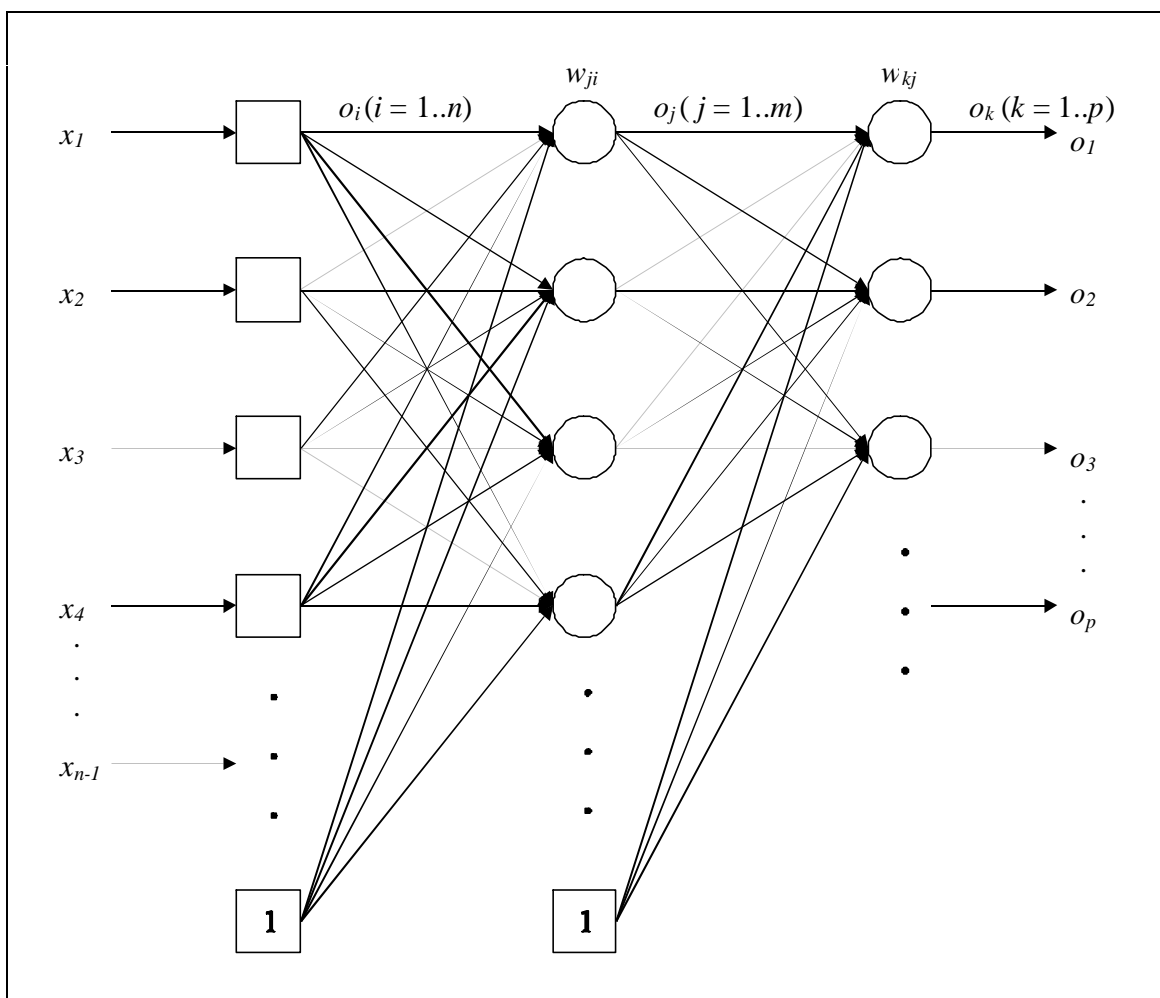


Abb. 55 Dreischichtiges neuronales Netz (Mehrschicht-Perzeptron)

Die Neuronen der Eingabeschicht sind reine Verteilungseinheiten. Daher werden sie meistens als Eingabeeinheiten und nicht als Eingabeneuronen bezeichnet. In der Abbildung kommt dies durch die eckige Darstellung zum Ausdruck. (Aus diesem Grund ist in der Literatur auch die Zählweise der Neuronenschichten unterschiedlich: Einige Autoren zählen die Schicht der

Eingabeneuronen nicht als separate Schicht. In ihrer Nomenklatur wäre das neuronale Netz aus Abbildung 56 zweischichtig.)

Die Neuronen 1 bis $m-1$ der verborgenen Schicht erhalten die Werte des Vektors o_i . In mehreren Berechnungsschritten bildet jedes einzelne Neuron j ($j = 1..m-1$) einen Ausgabewert. Die Ausgabewerte werden in dem Vektor o_j zusammengefaßt. Wobei o_j um die m .te Dimension mit dem Wert 1 erweitert wird, um auch den Neuronen der Ausgabeschicht (*output layer*) einen Biaswert zu ermöglichen.

Der Vektor o_j der verborgenen Schicht wird an alle p Neuronen der Ausgabeschicht weitergeleitet. Jedes Neuron k berechnet daraus in mehreren Schritten einen Wert des Ausgabevektors des Netzes o_k ($k = 1..p$).

Bei dem beschriebenen neuronalen Netz handelt es sich um ein vollständig verbundenes, vorwärtsgekoppeltes Mehrschicht-Perzeptron [13, 14, 17]. In einem vorwärtsgekoppelten Netz gibt jedes Neuron seine Ausgabewert nur an Neuronen der nachfolgenden Schicht weiter, nicht an Neuronen innerhalb seiner eigenen Schicht, oder Neuronen der vorgelagerten Schicht (Rückkopplung). Von einem vollständig verbundenen Netz wird gesprochen, wenn jedem Neuron der nachfolgenden Schicht die Ausgangswerte aller Neuronen der vorgelagerten Schicht für eine Verarbeitung zur Verfügung stehen. Um die Anzahl der Neuronen jeder Schicht in die Bezeichnung des neuronalen Netzes mit aufzunehmen, ist es üblich von einem $[(n-1)-(m-1)-k]$ neuronalen Netz zu sprechen. (Auch hier ist die Nomenklatur nicht eindeutig: Einige Autoren berücksichtigen die Bias-Neuronen und gelangen zu der Bezeichnung eines $[n-m-k]$ neuronalen Netzes.)

Anhand eines Neurons j der verborgenen Schicht werden die Berechnungsschritte innerhalb des Neurons beschrieben. Die Neuronen der Ausgabeschicht sind gleich aufgebaut, so daß die Beschreibung auch für sie gültig ist. Die Verarbeitungsschritte in einem künstlichen Neuron j sind angelehnt an die biologischen Vorgänge in natürlichen Neuronen [16, 24]. Ihre Aufgabe ist es den Eingabevektor zu verarbeiten und daraus einen Ausgabewert zu produzieren.

Zunächst berechnet die Eingangsfunktion einen effektiven Eingang e_j [14]. In Mehrschicht-Perzeptrons ist die Eingangsfunktion die gewichtete Summe der Werte des Eingabevektors des Neurons o_i . In (B.1) sind die Gewichte w_{ji} des Neurons j bezüglich des Wertes o_i enthalten. Diese Gewichte spielen die wichtigste Rolle für die Anpassungsfähigkeit des gesamten neuronalen Netzes an eine gestellte Lernaufgabe.

$$\mathbf{e}_j = \sum_{i=1}^n w_{ji} o_i \quad (\text{B.1})$$

Geometrisch gedeutet bestimmt die Eingangsfunktion den effektiven Eingang \mathbf{e}_j als Skalarprodukt des Eingabevektors o_i und des Gewichtsvektors w_{ji} des Neurons j . Verlaufen die Vektoren parallel zueinander ist der effektive Eingang \mathbf{e}_j am größten. Stehen die Vektoren senkrecht aufeinander wird der effektive Eingang null. Der effektive Eingang ist demnach umso größer, je ähnlicher sich die beiden Vektoren o_i und w_{ji} sind.

Bei der Bestimmung des effektiven Eingangs \mathbf{e}_j wird deutlich, daß die Erweiterung des Vektors o_i um die Dimension n dem Neuron ein zusätzliches Gewicht verschafft [16]. Künstliche neuronale Netze, die aus Neuronen mit einem Bias-Wert aufgebaut sind, können den funktionalen Zusammenhang zwischen dem Eingabe- und Ausgabevektor problemlos darstellen [16].

Der effektive Eingang \mathbf{e}_j des Neurons j wird an seine Transferfunktion t_j weitergeleitet. Die Transferfunktion t_j kann nochmals unterteilt werden in die Aktivierungsfunktion und die Ausgangsfunktion [14]. Dies wird allerdings nur gemacht, wenn sich der aktuelle Ausgabewert o_j des Neurons auch aus vorangegangenen Aktivierungen berechnet.

An die Transferfunktion t_j wird die Bedingung gestellt, daß sie stetig und differenzierbar ist. Dies ermöglicht es, den unten aufgezeigten Fehlerrückführungsalgorithmus auf das neuronale Netz anzuwenden. Die gebräuchlichsten Transferfunktionen sind die logistische [19, 22, 34] oder auch Fermi-Funktion [14] (B.2), die Hyperbeltangens-Funktion (B.3) und die Identität (B.4).

$$t_j(\mathbf{e}_j) = \frac{1}{1 + \exp(-\mathbf{e}_j)} \quad (\text{B.2})$$

$$t_j(\mathbf{e}_j) = \tanh(\mathbf{e}_j) = \frac{\exp(\mathbf{e}_j) - \exp(-\mathbf{e}_j)}{\exp(\mathbf{e}_j) + \exp(-\mathbf{e}_j)} = \frac{\exp(2\mathbf{e}_j) - 1}{\exp(2\mathbf{e}_j) + 1} \quad (\text{B.3})$$

$$t_j(\mathbf{e}_j) = \mathbf{e}_j \quad (\text{B.4})$$

Die Funktionen (B.2) und (B.3) sind s-förmig und werden auch als *squashing*-Funktionen bezeichnet, da sie den in seinem numerischen Wert unbegrenzten effektiven Eingang \mathbf{e}_j auf ein definiertes Intervall abbilden. Bei sehr großen oder sehr kleinen effektiven Eingängen geben diese

Funktionen die entsprechenden Sättigungswerte aus. Ein Nachteil der logistischen Funktion ist, daß ihr Sättigungswert bei effektiven Eingängen $e_j < 0$ null beträgt, dies verlangsamt möglicherweise die Konvergenz des unten beschriebenen Fehlerrückführungsalgorithmus [19, 22].

Die Transferfunktionen ließen sich noch dahingehend modifizieren, daß der effektive Eingang e_j ersetzt wird durch $k_2(e_j - k_1)$. Dies entspräche einer Verschiebung in e_j -Richtung und dem Hinzufügen einer zusätzlichen Steigung. Durch die Verwendung eines Biaseingangs bei jedem Neuron und eines Lernverfahrens, welches alle Gewichte w_{ji} des Neurons verändert, werden diese beiden Variationen allerdings schon in die Transferfunktion t_j integriert [15].

Der Vorteil der oben aufgeführten Transferfunktionen ist die einfache Darstellung ihrer Ableitungen nach dem effektiven Eingang e_j . Diese Ableitungen können bei (B.2) und (B.3) in Abhängigkeit von der Transferfunktion t_j selbst ausgedrückt werden (B.5) und (B.6). Die Ableitung der Identität ist 1 (B.7), was keine weitere Rechnung notwendig macht.

$$\frac{d t_j(e_j)}{d e_j} = \frac{d \left(\frac{1}{1 + \exp(e_j)} \right)}{d e_j} = t_j(e_j)(1 - t_j(e_j)) \quad (\text{B.5})$$

$$\frac{d t_j(e_j)}{d e_j} = \frac{d \tanh(e_j)}{d e_j} = 1 - (t_j(e_j))^2 \quad (\text{B.6})$$

$$\frac{d t_j(e_j)}{d e_j} = \frac{d e_j}{d e_j} = 1 \quad (\text{B.7})$$

Es ist festzuhalten, daß die Ableitungen der Transferfunktionen am größten sind, wenn der Funktionswert um 0.5 bei der logistischen Funktion und um 0.0 bei der Hyperbeltangens-Funktion liegt [18]. Das ist der Fall, wenn der effektive Eingang e_j des Neurons j den Wert null besitzt. Für die Initialisierung der Gewichte w_{ji} wird dieser Umstand wichtig sein.

Die logistische Funktion (B.2) und die Hyperbeltangens-Funktion (B.3) werden meist für Neuronen der verborgenen Schicht eingesetzt. Sie lassen ein neuronales Netz zu einem universellen Approximator werden [17]. Die s-förmigen Funktionen in den verborgenen Neuronen ermöglichen dem

neuronalen Netz die Darstellung nichtlinearer Funktionen zwischen den Ein- und Ausgabedaten [14, 18].

Die Identität als Transferfunktion t_j wird hingegen für Neuronen der Ausgabeschicht verwendet. Diese lineare Transferfunktion ist nicht auf den Wertebereich von 0 bis 1 wie bei der logistischen Funktion bzw. von -1 bis 1 bei der Hyperbeltangens-Funktion beschränkt. (In [15] und [18] wird vorgeschlagen bei Verwendung einer logistischen Funktion in den Ausgabeneuronen die Werte des Soll-Ausgabevektors y_k auf das Intervall von 0.1 bis 0.9 zu skalieren, da die Werte 0 und 1 nur asymptotisch erreicht werden. Durch diese Vorgehensweise wird vornehmlich der lineare Bereich der logistischen Transferfunktion genutzt.)

Es ist deutlich geworden, daß diverse Parameter festgelegt werden müssen, um ein neuronales Netz zu charakterisieren [15]. Diese sind im folgenden:

Die Anzahl $(n-1)$ der Elemente des Eingabevektors x_h ,

Die Anzahl p der Elemente des Ausgabevektors o_k ,

Die Anzahl der verborgenen Schichten,

Die Anzahl der Neuronen (n, m, p, \dots) in den einzelnen Schichten,

Die Wahl der Konnektivität (vollständig oder unvollständig verbunden: d.h. einige Gewichte w_{ji}, w_{kj}, \dots werden unveränderlich auf null gesetzt),

Die Wahl der Transferfunktion t_j, t_k, \dots der Neuronen.

Während sich die Dimensionalitäten des Eingabevektors und des Ausgabevektors des Netzes aus der Lernaufgabe ergeben, sind die übrigen Parameter vom Anwender vorzugeben. Meist kann man sich bei der Anzahl der verborgenen Schichten auf eine Schicht beschränken: Ein dreischichtiges vorwärtsgekoppeltes neuronales Netz ist in der Lage, jede beliebige Funktion zwischen Eingabe- und Ausgabedaten beliebig genau zu approximieren. Es wird allerdings keine Aussage darüber gemacht, wieviele Neuronen in der verborgenen Schicht dazu notwendig sind [20, 32]. Die Festlegung dieser Anzahl sowie der übrigen Parameter ist meistens ein Prozeß von Versuch und Irrtum.

Da für die Systemidentifikation ein neuronales Netz mit nur einem Ausgabewert o_k ($k = p = 1$) (meist die Regelgröße) benötigt wird, wird im folgenden lediglich auf den für einen Ausgabewert vereinfachten Fehler-rückführungsalgorithmus eingegangen. Er kann jedoch auf ein neuronales Netz mit einem mehrdimensionalen Ausgabevektor o_k ($k = 1..p, p > 1$) erweitert werden [15, 17-23, 35].

Die folgende Abbildung 57 stellt das vereinfachte neuronale Netz mit nur einem Ausgabeneuron grafisch dar. Es ergibt sich aus dem in Abbildung 56 abgebildeten Mehrschicht-Perzeptron.

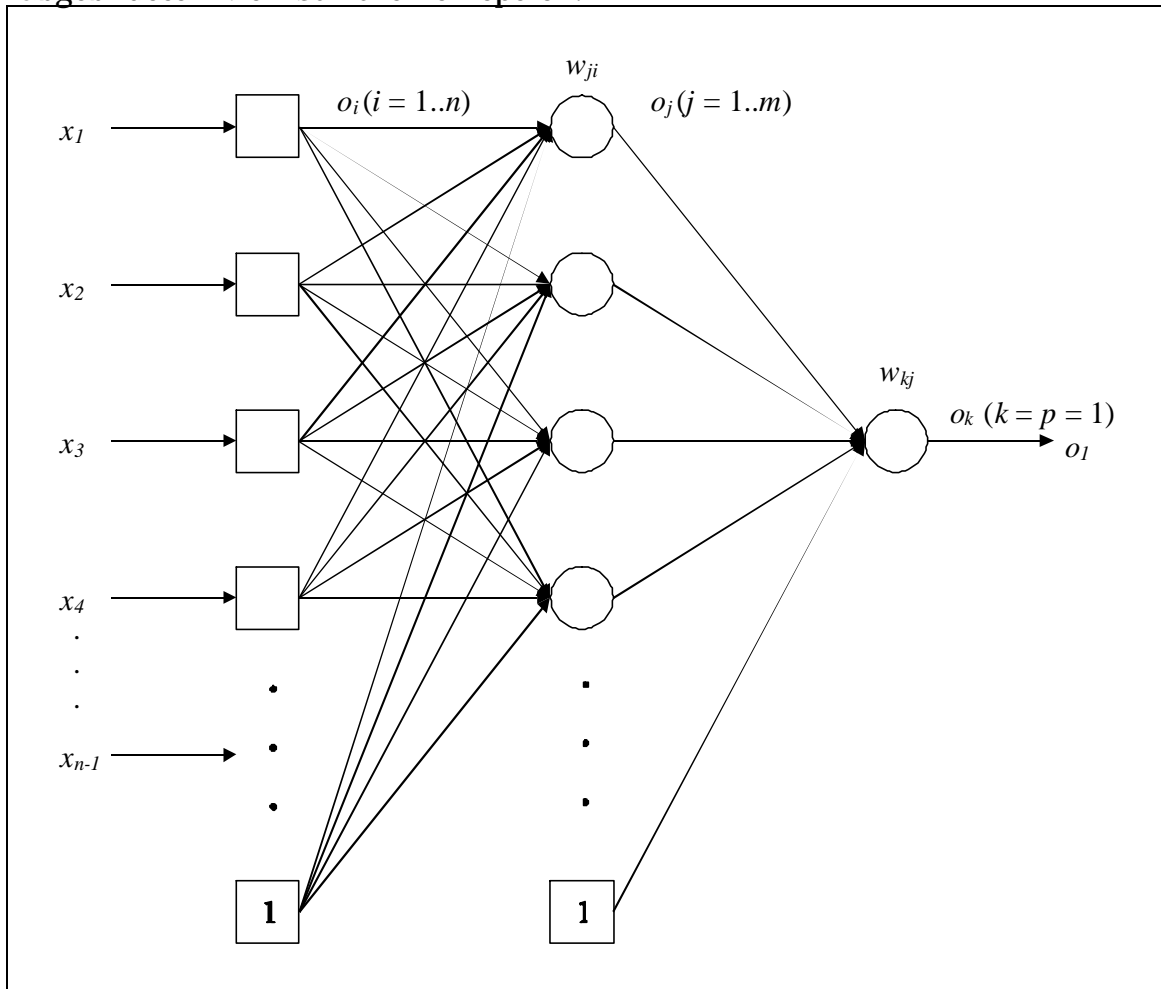


Abb. 56 Dreischichtiges neuronales Netz mit nur einem Ausgabeneuron

Fehlerrückführungs- und weitere Algorithmen [15, 18, 19]

Der funktionale Zusammenhang zwischen dem Eingabevektor und Ausgabewert ist bei einem künstlichen neuronalen Netz nicht fest programmiert, sondern das neuronale Netz wird durch einen Lern- bzw. Trainingsverfahren an diese Daten angepaßt [17]. Zu diesem Zweck verändert das Lernverfahren die Gewichte der Neuronen w_{ji}, w_{kj} [16, 17]. Die Funktionalität des neuronalen Netzes wird bestimmt durch diese Gewichte, die Einfluß auf den effektiven Eingang der Neuronen nehmen [16].

Der hier beschriebene Fehlerrückführungsalgorithmus entstand ursprünglich aus der auf einschichtige Perzeptrons angewendeten Widrow-Hoff-Lernregel oder Delta-Lernregel (*delta rule*), welche auf mehrschichtige

Perzeptrons mit differenzierbaren Transferfunktionen erweitert wurde [13, 14, 16, 17, 19, 23]. Da zur Anwendung des Fehlerrückführungsalgorithmus r Paare an Eingabevektor und Ausgabewerten $(x_h, y_k)_l$ (B.8) vorliegen müssen, gehört dieser Algorithmus zu den Verfahren des überwachten Lernens [24]. Ganz allgemein kann das Lernen in künstlichen neuronalen Netzen (das Training von neuronalen Netzen) als ein Optimierungsproblem verstanden werden [36], welches darin besteht für einen Trainingsdatensatz aus Eingabe-Ausgabe-Paaren (oder Eingabe-Ausgabe-Mustern) die besten Gewichte der Neuronen w_{ji}, w_{kj} zu finden.

$$\begin{aligned} & \text{mit } h = 1..n-1 \\ (x_h, y_k)_l & \text{ und } k = p = 1 \\ & \text{und } l = 1..r \end{aligned} \tag{B.8}$$

Obwohl das neuronale Netz im Verlauf des Fehlerrückführungsverfahrens an die Eingabe-Ausgabe-Paare des Trainingsdatensatzes angepaßt wird und die absoluten Werte des Eingabevektors x_h bzw des Ausgabewertes y_k (zumindest bei Verwendung der Identität als Transferfunktion t_k des Ausgabeneurons) keine Rolle spielen sollten, ist es üblich, diese Werte in einer Vorverarbeitung auf ein begrenztes Intervall zu skalieren. In [18] wird vorgeschlagen, die Werte des Eingabevektors auf Werte zwischen 0 und 1 zu skalieren. Eine bessere Wahl ist das Intervall $[-1, 1]$, da der Fehlerrückführungsalgorithmus keine Änderung eines Gewichtes w_{ji} vornimmt, dessen Eingabewert $o_i=0$ ist. (Aus demselben Grund kann auch die Verwendung der logistischen Funktion als Transferfunktion t_j problematisch werden, da ihr Sättigungswert bei groß-negativen effektiven Eingängen e_j null beträgt.) Setzt sich der Eingabevektor x_h aus unterschiedlichen Meßwerten zusammen, die sich in verschiedenen Größenordnungen bewegen, so sollten diese zunächst separat skaliert werden, und erst anschließend der Eingabevektor konstruiert werden.

Bevor mit dem Fehlerrückführungsverfahren eine Anpassung des neuronalen Netzes vorgenommen werden kann, müssen die Gewichte w_{ji}, w_{kj} zu Beginn mit zufälligen Werten belegt werden. (Dieser Vorgang entspricht der zufälligen Wahl eines Punktes in dem $(n \cdot (m-1) + m)$ -dimensionalen Parameterraum der Gewichte.) Der Fehlerrückführungsalgorithmus verlangt, daß die anfänglichen Gewichte von null und auch untereinander verschieden sind (*symmetry breaking* [19]) [14]. Die Startgewichte w_{ji}, w_{kj} werden daher mit kleinen zufälligen Werten initialisiert.

In [15] wird als Faustregel vorgeschlagen, die Gewichte w_{ji} für ein Neuron j mit kleinen zufälligen Werten so zu wählen, daß die Summe der Beträge eins ergibt (B.9). An gleicher Stelle [15] wird daneben das Intervall $[-1/s, 1/s]$ vorgeschlagen, aus dem die zufälligen Werte für die anfänglichen Gewichte zu wählen sind. Wobei s die Anzahl aller Gewichte innerhalb einer Schicht ist (für die verborgenen Schicht $s = n \cdot (m-1)$, für die Ausgabeschicht $s = m$).

$$\sum_{i=1}^n |w_{ji}| = 1 \quad (\text{B.9})$$

In [19] hingegen wird das Intervall $[-1, 1]$ genannt, in welchem die zufälligen Anfangswerte liegen sollten. Daneben wird das Intervall $[-1/n, 1/n]$ genannt [19], mit n als Anzahl der Eingänge eines Neurons. Auch das Intervall $[-0.5, 0.5]$ wird an anderer Stelle [20] vorgeschlagen.

Die anfängliche Wahl der Gewichte hat Auswirkungen auf die Leistungsfähigkeit des Fehlerrückführungsalgorithmus [16]. Da die Ableitung der Transferfunktion t_j, t_k nach dem effektiven Eingang in die vorzunehmende Gewichtsänderung eingeht, ist die Größe dieser Ableitung entscheidend. Bereits bei der Darstellung der s-förmigen Transferfunktionen (B.2) und (B.3) wurde darauf hingewiesen, daß die Ableitung dieser Funktionen nach dem effektiven Eingangswert e_j am größten ist, wenn e_j gleich null ist [19, 22]. Unabhängig von den Werten des Eingangsvektors wird dies am ehesten gewährleistet, wenn die Gewichte anfänglich auf kleine zufällige Werte gesetzt werden [18, 19]. Ein Neuron j mit einem effektiven Eingang e_j zwischen -4 und 4 wird am besten durch den Fehlerrückführungsalgorithmus angepaßt [15].

Bei dem Fehlerrückführungsalgorithmus durchläuft das neuronale Netz zwei unterschiedliche Phasen: Die Reproduktionsphase und die Lern- oder Trainingsphase [14]. In der Reproduktionsphase wird ein Eingabevektor x_h des Trainingsdatensatzes $(x_h, y_k)_l$ ($l=1..r$) in das Netz eingegeben und der daraus resultierende Ausgabewert o_k berechnet. Bei einem mit zufälligen Gewichten w_{ji}, w_{kj} initialisierten Netz wird der Ausgabewert o_k vom Sollausgabewert y_k verschieden sein. Daher ändert man die Gewichte derart ab, daß die Abweichung zwischen Ausgangswert o_k und Sollausgabewert y_k kleiner wird. Die Größe der Gewichtsänderung wird durch den Algorithmus bestimmt [14].

Bei nur einem Ausgabeneuron k berechnet sich die Abweichung oder Fehler eines Eingabe-Ausgabe-Paares [18] gemäß (B.10). Der Faktor $\frac{1}{2}$ wird zur Vereinfachung der Rechnung eingeführt. Für den Fehlerrückführungsalgorithmus spielt es keine Rolle, ob der quadratische Fehler oder der halbe quadratische Fehler minimiert wird [19].

$$E_l = \frac{1}{2} (y_k - o_k)_l^2 \quad (\text{B.10})$$

Der gesamte Netzfehler E ergibt sich aus der Summation über alle r Eingabe-Ausgabe-Paare des Trainingsdatensatzes (B.11). Der Netzfehler E als Funktion der Gewichte w_{ji} , w_{kj} wird auch als Kostenfunktion bezeichnet, die es im Verlauf des Optimierungsverfahrens zu minimieren gilt.

$$E = \sum_{l=1}^r E_l \quad (\text{B.11})$$

Ein näher an der Ausgleichsrechnung orientierter Netzfehler eines Trainingsdatenpaares ist die Wurzel des mittleren Fehlerquadrates (*root-mean-square-error*) (B.12) [15]. Er kann auf einfache Weise aus dem Netzfehler E abgeleitet werden. Es ist ein Vorteil des *rms*-Fehlers, daß er direkt Auskunft über die zu erwartende Abweichung zwischen dem Ausgabewert des Netzes o_k und dem Sollausgabewert y_k eines jeden Paares l gibt.

$$\text{rms-Fehler} = \sqrt{\frac{\sum_{l=1}^r (y_k - o_k)_l^2}{r}} = \sqrt{\frac{2E}{r}} \quad (\text{B.12})$$

Angewendet in der Lern- oder Trainingsphase wird aber der Fehler E_l (B.10), den das Netz für ein Eingabe-Ausgabe-Paar $(x_h, y_k)_l$ liefert. Zunächst werden die Änderungen der Gewichte Δw_{kj} des Ausgabeneurons berechnet. Anschließend wird der Fehler E_l zu den Neuronen j der verborgenen Schicht zurückgeführt [13] (daher rührt auch der Name: Fehlerrückführungslernverfahren). Die Änderung eines Gewichte Δw_{kj} entspricht einen Bruchteil des negativen Gradienten des Netzfehlers E_l [19]. Damit ist der Fehlerrückführungsalgorithmus ein einfaches Gradientenabstiegsverfahren erster Ordnung. Der in (B.13) enthaltene Parameter h wird als Lernrate bezeichnet. Die Lernrate h dient zur Steuerung der Schrittweite. Auf die Wahl ihres (positiven) Werts wird weiter unten eingegangen.

$$\Delta w_{kj} = -\mathbf{h} \frac{\partial E_l}{\partial w_{kj}} = -\mathbf{h} \frac{\partial E_l}{\partial o_k} \frac{\partial o_k}{\partial \mathbf{e}_k} \frac{\partial \mathbf{e}_k}{\partial w_{kj}} \quad (\text{B.13})$$

Um den Gradienten des Netzfehlers zu bestimmen, wird mehrfach die Kettenregel angewendet, wodurch die drei Terme auf der rechten Seite in (B.13) entstehen. Der erste Term, die Ableitung des Netzfehlers E_l nach dem Ausgabewert des Netzes o_k , wird in (B.14) angegeben. Hieran wird auch deutlich, warum die Berechnung der Gewichtänderungen Δw_{kj} beim Ausgabeneuron k beginnen muß: Die Ableitung des Netzfehlers E_l nach dem Ausgabewert o_j eines verborgenen Neurons j ist nicht so einfach möglich.

$$\frac{\partial E_l}{\partial o_k} = \frac{\partial (\frac{1}{2} (y_k - o_k)^2)}{\partial o_k} = -(y_k - o_k) \quad (\text{B.14})$$

Der zweite Term in (B.13) ist die Ableitung der Transferfunktion t_k nach dem effektiven Eingang \mathbf{e}_k (B.15). Für einige mögliche Transferfunktionen wurde diese bereits in den Gleichungen (B.5)-(B.7) vorgestellt.

$$\frac{\partial o_k}{\partial \mathbf{e}_k} = t'_k(\mathbf{e}_k) \quad (\text{B.15})$$

Zur Vereinfachung der Darstellung werden aus (B.13) das Minuszeichen, die Ableitung des Netzfehlers nach dem Ausgabewert und die Ableitung des Ausgabewertes nach dem effektiven Eingang in einem Korrekturwert \mathbf{d}_k zusammengefaßt (B.16).

$$\mathbf{d}_k = -\frac{\partial E_l}{\partial o_k} \frac{\partial o_k}{\partial \mathbf{e}_k} \quad (\text{B.16})$$

$$\mathbf{d}_k = (y_k - o_k) t'_k(\mathbf{e}_k) \quad (\text{B.17})$$

Der dritte Term aus Gleichung (B.13) beschreibt die Ableitung des effektiven Eingangs \mathbf{e}_k des Neurons nach dem Gewicht, dessen Änderung bestimmt werden soll. Diese Ableitung entspricht dem Element des Vektors \mathbf{o}_j , das zu dem Gewicht w_{kj} gehört (B.18).

$$\frac{\partial \mathbf{e}_k}{\partial w_{kj}} = \frac{\partial \left(\sum_{j=1}^m w_{kj} o_j \right)}{\partial w_{kj}} = o_j \quad (\text{B.18})$$

In (B.19) werden die Betrachtungen zur Berechnung der Gewichtsänderung Δw_{kj} zusammengefaßt. Obwohl eine Abweichung zwischen dem Ausgabewert des Netzes o_k und dem Sollausgabewert y_k errechnet wird, kann die berechnete Änderung des Gewichts Δw_{kj} null betragen, wenn die Ableitung der Transferfunktion t_k oder wenn der Eingabewert o_j null ist. Dies kann die Konvergenz des Lernverfahrens behindern.

$$\Delta w_{kj} = \mathbf{h} \mathbf{d}_k o_j \quad (\text{B.19})$$

Anschließend werden die Änderungen aller n Gewichte von jedem verborgenen Neuronen j berechnet. Jedem Neuron j der verborgenen Schicht wird gewissermaßen ein Anteil am Fehler E_l eines Eingabe-Ausgabe-Musters zugewiesen. Proportional zu diesem Anteil wird die Änderung seiner Gewichte Δw_{ji} vorgenommen [13].

In (B.20) wird die Änderung eines Gewichts wie schon in (B.13) als proportional der negativen Ableitung des Netzfehlers E_l nach dem Gewicht w_{ji} berechnet. Auch hier findet die Kettenregel Anwendung.

$$\Delta w_{ji} = -\mathbf{h} \frac{\partial E_l}{\partial w_{ji}} = -\mathbf{h} \frac{\partial E_l}{\partial o_j} \frac{\partial o_j}{\partial \mathbf{e}_j} \frac{\partial \mathbf{e}_j}{\partial w_{ji}} \quad (\text{B.20})$$

Der dritte Term auf der rechten Seite ist die Ableitung des effektiven Eingangs \mathbf{e}_j nach dem zu ändernden Gewicht w_{ji} . Sie entspricht dem Element des Vektors \mathbf{o}_i bezüglich des Gewichts (B.21).

$$\frac{\partial \mathbf{e}_j}{\partial w_{ji}} = \frac{\partial \left(\sum_{i=1}^n w_{ji} o_i \right)}{\partial w_{ji}} = o_i \quad (\text{B.21})$$

Das Minuszeichen, der erste und der zweite Term werden zur größeren Übersichtlichkeit wieder in einem Korrekturwert \mathbf{d}_j zusammengefaßt (B.22).

$$\mathbf{d}_j = -\frac{\partial E_l}{\partial o_j} \frac{\partial o_j}{\partial \mathbf{e}_j} \quad (\text{B.22})$$

Darin befindet sich die Ableitung der Transferfunktion t_j nach dem effektiven Eingang \mathbf{e}_j (B.23) und die Ableitung des Netzfehlers E_l nach dem Ausgabewert des Neurons o_j . Deren Berechnung gestaltet sich etwas aufwendiger, da sie nur wieder über die Kettenregel ermittelt werden kann (B.24).

$$\frac{\partial o_j}{\partial \mathbf{e}_j} = t'_j(\mathbf{e}_j) \quad (\text{B.23})$$

$$-\frac{\partial E_l}{\partial o_j} = -\frac{\partial E_l}{\partial o_k} \frac{\partial o_k}{\partial \mathbf{e}_k} \frac{\partial \mathbf{e}_k}{\partial o_j} \quad (\text{B.24})$$

In (B.24) entsprechen auf der rechten Seite das Minuszeichen, der erste und zweite Term dem in (B.16) für das Ausgabeneuron definierten Korrekturwert \mathbf{d}_k . Es ist also durchaus sinnvoll von einem Fehlerrückführungslernverfahren zu sprechen, da der Netzfehler durch die einzelnen Schichten des neuronalen Netzes zurückgeführt wird [14]. Der letzte Term in (B.24) (Ableitung des effektiven Eingangs \mathbf{e}_k des Ausgabeneurons k nach der Ausgabe des betrachteten Neurons j der verborgenen Schicht) ist das Gewicht des Ausgabeneurons w_{kj} (B.25).

$$\frac{\partial \mathbf{e}_k}{\partial o_j} = \frac{\partial \left(\sum_{j=1}^m w_{kj} o_j \right)}{\partial o_j} = w_{kj} \quad (\text{B.25})$$

Damit ist die Ableitung des Netzfehlers E_l nach dem Ausgabewert o_j des verborgenen Neurons j bekannt (B.26) und der Korrekturwert \mathbf{d}_j kann angegeben werden (B.27).

$$-\frac{\partial E_l}{\partial o_j} = \mathbf{d}_k w_{kj} \quad (\text{B.26})$$

$$\mathbf{d}_j = \mathbf{d}_k w_{kj} t'_j(\mathbf{e}_j) \quad (\text{B.27})$$

Die Änderung der Gewichte Δw_{ji} der verborgenen Neuronen j wird analog zu Gleichung (B.19) dargestellt (B.28). Die dort gemachten Feststellungen gelten auch hier.

$$\Delta w_{ji} = \mathbf{h} \mathbf{d}_j o_i \quad (\text{B.28})$$

Werden die so berechneten Gewichtsänderungen Δw_{kj} und Δw_{ji} vorgenommen, so ist gewährleistet, daß der Netzfehler E_l für das betrachtete Eingabe-Ausgabe-Paar $(x_h, y_k)_l$ verringert wird. Beim Fehlerrückführungsalgorithmus hat man nun die Wahl zwischen zwei Vorgehensweisen.

Im Standardalgorithmus (*standard backpropagation* [13], *deferred correction* [15], *batch mode* [16], *batch training* [17], kumuliertes [19] oder epochales Lernen [36]) werden die Gewichte erst dann verändert, wenn alle r Eingabe-Ausgabe-Paare $(x_h, y_k)_l$ das Netz durchlaufen haben, wobei die aus den einzelnen Paaren resultierenden Gewichtsänderungen zunächst aufaddiert werden [16], um anschließend nur einmalig eine Änderung vorzunehmen. Sind alle Eingabe-Ausgabe-Daten des Trainingsdatensatzes dem neuronalen Netz einmal präsentiert worden, so bezeichnet man dies als eine Epoche des Algorithmus [13].

Beim stochastischen Fehlerrückführungsalgorithmus (*stochastic backpropagation* [13, 19], *immediate correction* [15], *incremental mode* [16], *pattern based training* [17], *jump every time variant* [23], musterweises Lernen [36] oder *Online-Verfahren* [19]) hingegen wird eine Änderung der Gewichte nach jedem Durchlauf eines Eingabe-Ausgabe-Paares $(x_h, y_k)_l$ vorgenommen.

Während die Gefahr besteht, daß der Standardalgorithmus in einem lokalen Minimum des Netzfehlers E stehen bleibt, wird dies bei dem stochastischen Algorithmus mit höherer Wahrscheinlichkeit vermieden, wenn die Eingabe-Ausgabe-Paare in einer zufällig gewählten Reihenfolge gelernt werden [18, 36]. Voraussetzung dafür ist zusätzlich die Wahl einer genügend kleinen Lernrate h [19]. Durch die zufällige Reihenfolge wird gegenüber einer starren Reihenfolge auch vermieden, daß das Netz anfänglich Erlerntes im Laufe des Algorithmus wieder vergisst [13, 15, 20]. Der stochastische Fehlerrückführungsalgorithmus stellt daher meist die vorteilhaftere Variante dar [14].

Mit Anwendung der in (B.19) und (B.28) berechneten Gewichtsänderungen wäre das Fehlerrückführungsverfahren ein reines Gradientenabstiegsverfahren. Eine einfache Weiterentwicklung ist die Erweiterung um einen Momentum- oder Impulsterm (B.29) und (B.30). Diese Erweiterung soll das Fehlerrückführungslernverfahren zur Minimierung der Netzfehlerfunktion E beschleunigen [17, 18, 35], ohne daß dazu eine Erhöhung der Lernrate h notwendig wird. Die Gewichtsänderung in einem Lernschritts wird bei Einbeziehung des Momentumterms aus dem aktuellen Gradienten der

Netzfehlerfunktion und einem Anteil der Gewichtänderung des vorangegangenen Lernschritts gebildet.

Neben der Lernrate h ist dann noch ein weiterer Parameter des Fehler-rückführungsverfahrens zu wählen: die Momentumterm-Konstante oder auch der Momentumterm-Koeffizient m

$$\Delta w_{kj} = h d_k o_j + m \Delta w_{kj}^{prev} \quad (\text{B.29})$$

$$\Delta w_{ji} = h d_j o_i + m \Delta w_{ji}^{prev} \quad (\text{B.30})$$

Mit dem zusätzlichen Momentumterm wird die Änderung der Gewichte nicht nur auf dem aktuellen lokalen Gradienten basierend berechnet, sondern es wird ein längerfristiger Trend innerhalb der Netzfehlerfunktion E berücksichtigt [16]. Zeigen mehrere aufeinanderfolgende Gewichtsänderungen in dieselbe Richtung wird die aktuelle Änderung beträchtlich erhöht. Als Tiefpassfilter filtert der Momentumterm hochfrequente Änderungen der Netzfehlerfunktion E aus [16, 18] und erlaubt es so dem Algorithmus kleinere Änderungen im Gradienten der Netzfehlerfunktion zu ignorieren. In gleicher Weise dämpft der Momentumterm durch die Form der Netzfehlerfunktion E verursachte Oszillationen ab [15]. Ohne die Einbeziehung eines Momentumterms besteht eine größere Gefahr, daß der Algorithmus in einem flachen, lokalen Minimum stehen bleibt [15, 16].

Das in [19-22] beschriebene Verfahren des exponentiellen Glättens (*exponential smoothing*) kann als der Einbeziehung eines Momentum- oder Impulsterms in die Berechnung der Gewichtsänderung äquivalent betrachtet werden.

Der Fehlerrückführalgorithmus ist ein Gradientenabstiegsverfahren, welches für kontinuierliche Funktionen und für infinitesimal kleine Schrittweiten gerechtfertigt ist [15]. Daher wird für die meisten Anwendungen eine sehr kleine Lernrate h gewählt, um die Konvergenz auf ein Minimum zu gewährleisten [21]. Wenn die Lernrate h zu groß gewählt wird, besteht die Gefahr, daß der Algorithmus instabil wird, und es kann zu oszillierenden Lösungen kommen [16, 19]. Wird die Lernrate h hingegen zu klein gewählt, benötigt der Algorithmus viele Epochen um auf ein Minimum der Netzfehlerfunktion E zu konvergieren [16, 19]. In [17] werden daher Werte von 0.1 bis 0.5 für die Lernrate genannt. In [18] wird eine Lernrate von ca. 0.9 vorgeschlagen. Von demselben Wert $h = 0.9$ wird in [19] ausgegangen, allerdings

wird hier vorgeschlagen, die Lernrate mit zunehmender Anzahl der Epochen zu verringern, um die Konvergenz zu gewährleisten. Mit einer Lernrate zwischen 0.05 und 0.25 werden in [20] die kleinsten Werte genannt.

Eine ähnliche Vielfalt herrscht bei den Vorschlägen für den Wert der Momentumkonstante m . Abgesehen davon, daß einige Autoren sich gar nicht zum Wert von m äußern, bleibt bei den übrigen Autoren meist offen, ob sie den Standardalgorithmus oder den stochastischen Algorithmus der Fehler-rückführung meinen. In [14] wird ein Wert von 0.9 für den Momentumkoeffizienten genannt. In [17] hingegen ein Wert von $m=0.1$. Während in [19] Werte zwischen 0.20 und 0.99 zur Wahl gestellt werden.

Zusammenfassend kann festgehalten werden, daß bei der Wahl der Lernrate h und der Momentumkonstante m ein Kompromiß gefunden werden muß, zwischen zügiger Konvergenz auf ein Minimum der Netzfehlerfunktion E und der Wahrscheinlichkeit des Findens einer Lösung nahe an diesem Minimum [15]. Angesichts der in der Literatur genannten Vielfalt ist man bei der aufgabenspezifischen Festlegung dieser Parameter auf die Vorgehensweise von Versuch und Irrtum angewiesen [22].

Bezüglich des Abbruchkriteriums bleibt bei dem Fehlerrückführungsverfahren die Wahl zwischen drei Varianten [16]. Erfolgt der Abbruch des Algorithmus nach einer festgelegten Anzahl von Epochen oder nach einer festgelegten Rechenzeit, kann der Rechen- oder Zeitaufwand von vornherein begrenzt werden, was in zeitkritischen Anwendungen von Vorteil ist. Wird das Unterschreiten einer bestimmten Schwelle beim Netzfehler E als Bedingung für einen Abbruch gewählt, so ist der Zeitaufwand unbestimmt, dafür wird die Güte einer Lösung garantiert. Einen Nachteil besitzt dieses Kriterium, wenn der minimal erreichbare Netzfehler E im Vorfeld unbekannt ist und möglicherweise eine Schwelle gewählt wird, die nicht erreicht werden kann.

Anpassungsprobleme

Trainierte Mehrschicht-Perzeptrons geben plausibel erscheinende Ausgabe-werte o_k auch auf Eingabevektoren x_h , die nicht im Trainingsdatensatz enthalten sind [16]. Diese Generalisierungsfähigkeit erlaubt es, das neuronale Netz mit einer repräsentativen Auswahl an Eingabe-Ausgabe-Paaren zu trainieren, und nicht mit allen möglichen oder allen vorhandenen Eingabe-Ausgabe-Paaren [16].

Ein Problem, das während der Trainingsphase eines neuronalen Netzes auftreten kann ist das *Overfitting*. Es beschreibt den Vorgang, daß der

Netzfehler E für den Trainingsdatensatz einen sehr kleinen Wert annimmt, das neuronale Netz aber eine große Abweichung E_l produziert, wenn es den Ausgabewert o_k für einen Eingangsvektor x_h eines Eingabe-Ausgabe-Paares berechnet, das nicht im Trainingsdatensatz enthalten war [16].

Die als *early stopping with validation* bezeichnete Variante des Algorithmus versucht dieses Problems Herr zu werden, indem während des Trainings, das anhand des Trainingsdatensatzes erfolgt, der Netzfehler für einen Testdatensatz bestimmt wird. Der Testdatensatz wird vom Trainingsdatensatz getrennt behandelt, das neuronale Netz wird also nicht anhand des Testdatensatzes trainiert. Das Training des neuronalen Netzes wird abgebrochen, wenn der Netzfehler für den Testdatensatz im Laufe des Optimierungsalgorithmus als Zeichen des *Overfitting* wieder zunimmt.

Das Problem des *Overfitting* hängt auch zusammen mit der Anzahl an verarbeitenden Neuronen $(m-1)$ in der verborgenen Schicht. So nimmt durch eine höhere Anzahl $(m-1)$ der erreichbare minimale Netzfehler E bezüglich des Trainingsdatensatzes ab, gleichzeitig aber auch die Generalisierungsfähigkeit des neuronalen Netzes [18]. *Overfitting* tritt immer auf, wenn die Anzahl der in einem neuronalen Netz anzupassenden Gewichte $(n \cdot (m-1) + m)$ größer ist als die Anzahl unterschiedlicher Eingabe-Ausgabe-Paare r im Trainingsdatensatz [15].

Allgemein kann gesagt werden, daß die kleinstmöglich Anzahl an Neuronen $(m-1)$ gewählt werden sollte, um eine glatte, wenig strukturierte Approximationsfunktion zu erhalten, und eine gute Generalisierung zu gewährleisten [37]. Mit einer geringen Anzahl $(m-1)$ wird dem neuronalen Netz weniger Möglichkeit gegeben, ein in der Realität immer vorhandenes Meßrauschen in den Ein- und Ausgabedaten $(x_h, y_k)_l$ möglicherweise mitzulernen. Darüberhinaus wird bei der Implementierung von neuronalen Netzen auf seriellen Rechnern die Rechenzeit sowohl in der Lernphase als auch in der Reproduktionsphase begrenzt, wenn eine geringere Anzahl an Neuronen verwendet wird [20].

Ein weiteres Problem bei der Anpassung eines neuronalen Netzes an die Eingabe-Ausgabe-Paare des Trainingsdatensatzes ist das Verfangen des Fehlerrückführungsalgorithmus in einem lokalen Minimum der Netzfehlerfunktion E [13, 19]. Diese Problematik wurde bereits bei der Entscheidung für die Standard- oder die stochastische Form des Algorithmus angesprochen. Grundsätzlich findet eine Gradientenabstiegsmethode nur lokale Minima [14]. Durch die zufällige Reihenfolge in der die Trainingspaare

$(x_h, y_k)_l$ beim stochastischen Fehlerrückführungsverfahren präsentiert werden, ist jedoch das Verlassen eines lokalen Minimums wahrscheinlicher als beim Standardverfahren. Im Zusammenhang mit den Lernverfahren für neuronale Netze scheint ohnehin zu gelten: Wenn der Algorithmus eine annehmbare Lösung findet, dann besteht keine Sicherheit dafür, daß er das globale Minimum gefunden hat. Wenn die Lösung aber einen ausreichend kleinen Netzfehler E aufweist, dann ist es auch unerheblich, ob das Minimum globaler oder lokaler Natur ist [20].

Mit dem Problem der lokalen Minima beschäftigen sich auch einige Ergänzungen zum Fehlerrückführungsalgorithmus. Ein in [20] vorgeschlagenes Verfahren addiert in jeder Epoche zu jedem Eingabevektor x_h aus dem Trainingsdatensatz eine kleine normalverteilte Abweichung, um die Gefahr des Stehenbleibens in einem lokalen Minimum zu verhindern. Dieses Verfahren soll auch erfolgreich sein, wenn im Trainingsdatensatz eigentlich keine Abweichungen vorhanden sind bzw. erwartet werden, wie dies bei experimentell gewonnenen Daten der Fall ist. Prinzipiell denselben Effekt hat das Schocken des neuronalen Netzes [10]. Dabei wird nach einer gewissen Anzahl an Epochen zu jedem Gewicht des Netzes eine zufällige Abweichung im Intervall von ± 10 Prozent des aktuellen Wertes addiert.

Probleme bei der Anpassung der Gewichte können dem Fehlerrückführungsalgorithmus auch flache Plateaus in der Netzfehlerfunktion E bereiten. In diesen Bereichen liegen nur kleine Gradienten von E bezüglich den Gewichten w_{ji} und w_{kj} vor, so daß deren Änderungen entsprechend klein ausfallen. Dieses Problem tritt vornehmlich bei falscher, zu großer Wahl der anfänglichen Gewichte auf, da dann die s-förmigen Transferfunktionen (B.2) und (B.3) ihre Sättigungswerte ausgeben. Die Folge der flachen Plateaus ist eine langsame Konvergenz [13, 19] des Verfahrens auf ein Minimum. Zur Lösung trägt der Momentumterm bei, da er in solchen Fällen die Gewichtsänderungen Δw_{ji} , Δw_{kj} fortlaufend erhöht und damit zur Beschleunigung des Verfahrens beiträgt.

Eine andere Problematik stellen steile Schluchten in der Netzfehlerfunktion E dar. Bei solch einer Form von E liefert der Fehlerrückführungsalgorithmus oszillierende Lösungen [13, 19], welche sich auf den Flanken der Schluchten gegenüberliegen. Eine Lösung bietet auch hier die Einbeziehung eines Momentumterms, indem er hochfrequente Änderungen herausfiltert und so das Erreichen des Tals von E ermöglicht.

Bei allen positiven Eigenschaften, die eine Einbeziehung des Momentumterms besitzt, soll der Nachteil nicht ungenannt bleiben, daß auch gute lokale Minima durch den beschleunigenden Effekt des Momentumterms wieder verlassen werden können [19]. Unter Umständen ist es angebracht, daß der Fehlerrückführungsalgorithmus die jeweils besten gefundenen Gewichte w_{ji} und w_{kj} zwischenspeichert, um sie bei Beendigung des Verfahrens als Lösung auszugeben.

Weitere Algorithmen

Während die Einbeziehung eines Momentum- oder Impulsterms in den Fehlerrückführungsalgorithmus eine einfache Weiterentwicklung darstellt, wurden noch weitere Algorithmen entwickelt oder bereits aus anderen Bereichen bekannte Optimierungsalgorithmen auf das Training von neuronalen Netzen angewendet. Beispielhaft genannt seien nur der Quickprop-Algorithmus, der Rprop-Algorithmus [19, 35] und der Levenberg-Marquardt Algorithmus [16, 38]. Diese Algorithmen beziehen sich auf die Variante des kumulierten Lernens.

Quickprop ist ein iteratives Verfahren zweiter Ordnung, in dem die Netzfehlerfunktion als näherungsweise lokal quadratisch angenommen wird. Der Levenberg-Marquardt Algorithmus ist ebenfalls eine Gradientenmethode zweiter Ordnung und wird auf die Minimierung von Quadratsummen angewendet [33, 39]. Er kann um den Faktor 10 bis 100 schneller sein als der Fehlerrückführungsalgorithmus. In [16] wird daher vorgeschlagen den Levenberg-Marquardt Algorithmus immer zuerst bei dem Training eines Netzes anzuwenden. Beide Algorithmen verwenden das Fehlerrückführungsverfahren um die partiellen Gradienten des Netzfehlers bezüglich der einzelnen Gewichte zu ermitteln [16].

Nachteil aller Gradientenverfahren ist die Verwendung lokaler Informationen über die Netzfehlerfunktion E . Daher besteht immer die Möglichkeit, daß der Algorithmus in einem lokalen Minimum der Netzfehlerfunktion E stehenbleibt und die Netzgewichte zu diesem lokalen Minimum ausgibt. Mit diesem Nachteil ist auch der Quickprop-, der Rprop-Algorithmus und der Levenberg-Marquardt-Algorithmus behaftet. Diese Algorithmen, die zur Optimierung die zweiten Ableitungen der Netzfehlerfunktion nach den Gewichten benötigen, verlangen darüberhinaus einen höheren Berechnungsaufwand und Speicherbedarf [19, 22, 38]. Der stochastische Fehlerrückführungsalgorithmus scheint daher einen sinnvollen Kompromiß zwischen

der Rechenzeit und der Sicherheit zum Auffinden einer Lösung nahe einem Minimum zu bieten.

Die Gefahr ein lokales Minimum der Netzfehlerfunktion E als beste Lösung für die Adaption eines neuronalen Netzes an die Eingabe-Ausgabe-Paare des Trainingsdatensatzes anzunehmen, wird vermindert durch die Anwendung stochastischer Optimierungsalgorithmen [14, 22]. Dazu zählen der in Anhang D vorgestellte genetische Algorithmus und das Verfahren des simulierten Abkühlens (*simulated annealing*). Bei diesen Verfahren hängt das erreichbare Minimum nicht so sehr von den anfänglichen, zufällig gewählten Gewichten ab, wie dies bei den Gradientenabstiegsverfahren der Fall ist, welche lediglich lokale Informationen im Parameterraum um die aktuellen Netzgewichte für die Optimierung verwenden [10].

Der genetische Algorithmus faßt den in einem neuronalen Netz vorhandenen Satz an Gewichten als ein Individuum auf [10, 17, 19, 24]. Er orientiert sich aber nicht nur an einem zufällig erstellten Satz an Gewichten (an einem Individuum), sondern arbeitet mit mehreren solcher Individuen, die anhand von Zufallszahlen erstellt werden. Durch die gleichzeitige Handhabung mehrerer Individuen und die Anwendung von genetischen Operatoren soll gewährleistet werden, daß der genetische Algorithmus den gesamten Parameterraum absucht und das globale Optimum der Netzfehlerfunktion E findet. Diese globale Suchstrategie mit abschnittsweise ungerichteter Suche verhindert den Einsatz eines genetischen Algorithmus als Lernverfahren für ein größeres neuronales Netz mit vielen Gewichten, da dann das Verfahren zu zeitaufwendig wird [19]. Ferner fällt der genetische Algorithmus in seiner Leistungsfähigkeit hinter den Fehlerrückführungsalgorithmus zurück, wenn er schon in die Nähe des globalen Optimums gelangt ist. Dann ist der anhand des Gradienten der Netzfehlerfunktion E optimierende Fehlerrückführungsalgorithmus schneller [10]. Die Vorteile beider Lernverfahren zu verbinden, versucht das Lamarksche Lernverfahren, in dem sich Zyklen genetischer Optimierung mit Epochen des Fehlerrückführungsalgorithmus abwechseln [10].

Die Methode des simulierten Abkühlens arbeitet mit nur einem Satz an Gewichten für das neuronale Netz. Im Gegensatz zu einem deterministischen Gradientenabstiegsverfahren läßt dieses Verfahren jedoch auch Änderungen der Gewichte zu, die zu einer zeitweisen Erhöhung des Netzfehlers E führen [22]. Dadurch wird eine gewisse Unabhängigkeit von den anfänglich gewählten Gewichten erreicht [27] und die Wahrscheinlichkeit

für das Auffinden des globalen Minimums erhöht [23]. Mit welcher Wahrscheinlichkeit p_{accept} eine Änderung, die eine Erhöhung von E bewirkt, akzeptiert wird, wird durch eine der Boltzmann-Verteilungsfunktion ähnliche Wahrscheinlichkeitsfunktion (B.31) ausgedrückt. Darin ist E_{new} der Netzfehler des neuronalen Netzes mit den neu erzeugten Gewichten und E_{prev} der Netzfehler des Netzes mit den vorherigen Gewichten. a ist eine der Boltzmann-Konstanten k nachempfundene Konstante, deren Wert sich an der Größenordnung des Netzfehlers orientiert [27], und T ist die künstliche Temperatur. Die künstliche Temperatur T stellt den eigentlichen Kontrollparameter des simulierten Abkühlens dar [27]. Sie wird zunächst sehr groß gewählt, so daß auch Gewichtsänderungen, die zu einer großen Erhöhung des Netzfehlers E führen, mit einer hohen Wahrscheinlichkeit akzeptiert werden. Im Verlauf des Algorithmus wird T verringert, daher der Name des Verfahrens. Erreicht die Temperatur gegen Ende des Verfahrens einen Wert nahe $T=0$, werden vom Algorithmus praktisch nur noch Gewichtsänderungen akzeptiert, die zu einer Verringerung des Netzfehlers E führen.

$$p_{accept} = \exp\left[-\frac{(E_{New} - E_{prev})}{a T}\right] \quad (\text{B.31})$$

In der folgenden Abbildung 58 ist das Verfahren des simulierten Abkühlens grafisch dargestellt. Zunächst wird die künstliche Temperatur T auf einen großen Wert gesetzt. Anschließend werden den Gewichten des neuronalen Netzes zufällige Werte zugewiesen und der gesamte Netzfehler E_{prev} aus allen Eingabe-Ausgabe-Daten des Trainingsdatensatzes bestimmt. Wegen der zufälligen Initialisierung wird dieser anfängliche Netzfehler einen hohen Wert besitzen.

Die alten Gewichte werden durch Addition einer Zufallszahl, deren Verteilung und Standardabweichung festzulegen ist, verändert. Im $(n \cdot (m-1) + m)$ -dimensionalen Parameterraum der Gewichte des neuronalen Netzes bedeutet dies das Aufsuchen eines Nachbarpunktes ausgehend vom aktuellen Punkt. Für das neuronale Netz mit den geänderten Gewichten wird wiederum der Netzfehler E_{new} ermittelt. Ist der neue Netzfehler kleiner-gleich dem vorhergehenden $E_{new} \leq E_{prev}$ wird die Änderung der Gewichte beibehalten und E_{prev} nimmt den Wert von E_{new} an. Ist der neue Netzfehler größer als der vorherige $E_{new} > E_{prev}$ wird anhand von (B.31) bestimmt, mit welcher Wahrscheinlichkeit die Gewichtsänderung beibe-

halten werden soll. Dann wird eine im Intervall $[0, 1]$ gleichverteilte Zufallszahl erzeugt und entschieden, ob die Gewichtsänderung und damit die Erhöhung des Netzfehlers akzeptiert wird [22, 23, 27]. Die Wahrscheinlichkeit, daß eine Gewichtsänderung angenommen wird, die zu einem größeren Netzfehler E_{new} führt, ist umso größer je größer die künstliche Temperatur T ist ($p_{accept} \rightarrow 1$ für $T \rightarrow \infty$) und je kleiner die Erhöhung des Netzfehlers ausfällt ($p_{accept} \rightarrow 1$ für $(E_{new} - E_{prev}) \rightarrow 0$).

Der Prozeß Erzeugung neuer Gewichte, Bestimmung des Netzfehlers E_{new} und Entscheidung über die Akzeptanz der neuen Gewichte wird meist bei einem Temperaturwert T mehrfach wiederholt, was durch die innere Schleifenkonstruktion ausgedrückt wird. Man kann sich aber auch bei der Erzeugung neuer Gewichte dafür entscheiden, nicht alle Gewichte auf einmal zu ändern, sondern nur ein einziges Gewicht [23]. Dann wird die innere Schleife für jedes Gewicht einmal durchlaufen.

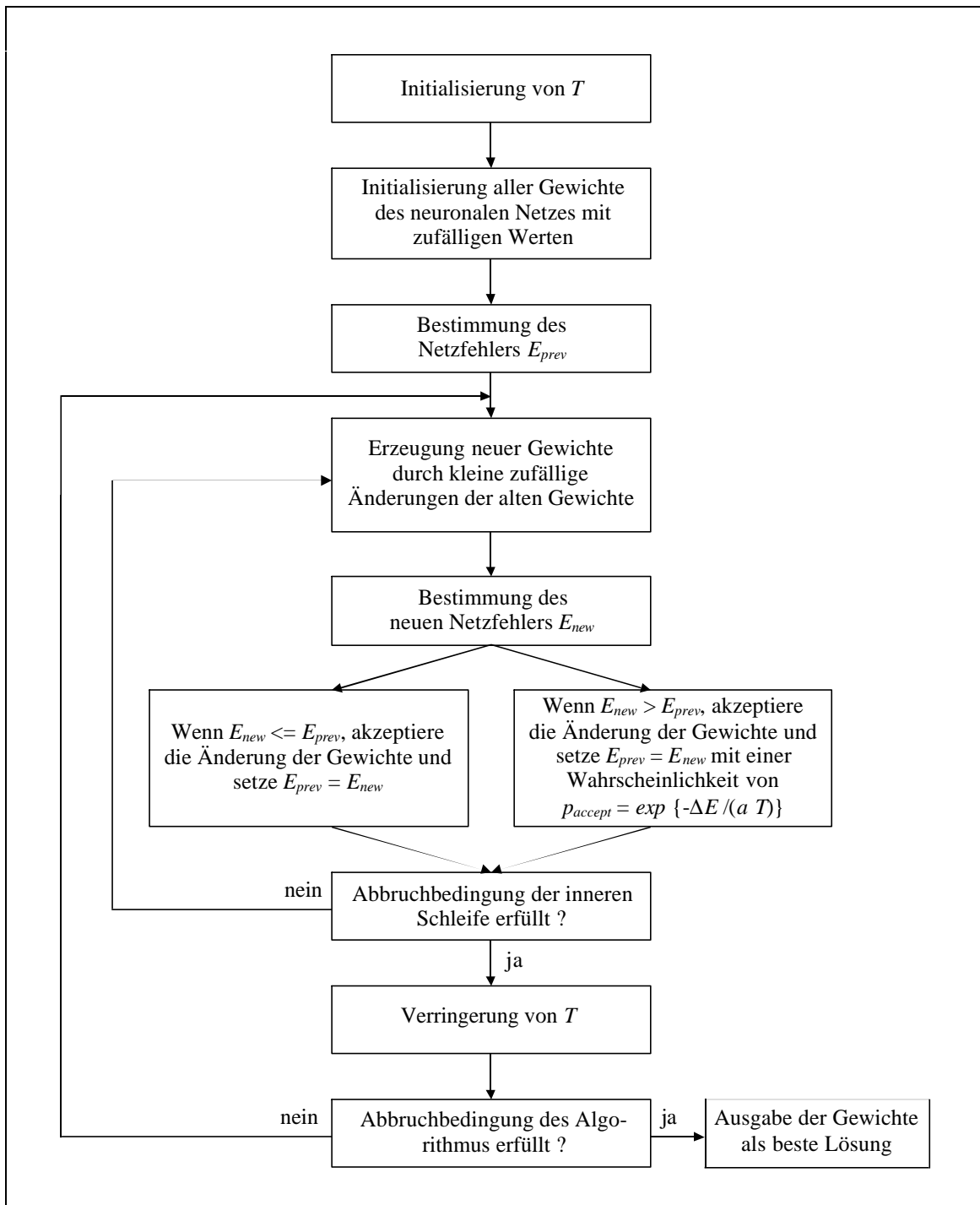


Abb. 57 Simuliertes Abkühlen (*simulated annealing*)

Die Abbruchbedingung für die äußere Schleife ist meist eine im vornherein festgelegte Anzahl an Durchläufen, da die Anzahl mit der Schrittweite der Temperaturverringernung abgestimmt sein muß, um zu verhindern, daß Temperaturen von $T \leq 0$ auftreten, die die Bestimmung von p_{accept} unsinnig werden lassen würden. Der zeitliche Temperaturverlauf kann linear, exponentiell abklingend oder proportional zum reziproken Logarithmus der

Anzahl der äußeren Schleifendurchläufe sein [22, 23]. Die letztere Variante führt zur langsamsten Abnahme der künstlichen Temperatur T .

Da das Verfahren des simulierten Abkühlens wie der Fehlerrückführungsalgorithmus mit nur einem Satz an Gewichten w_{ji} , w_{kj} für das neuronale Netz arbeitet, ist es sinnvoll, diese beiden Verfahren zu verknüpfen [22]. Wenn sich Zyklen des simulierten Abkühlens mit Epochen des Fehlerrückführungsalgorithmus abwechseln, werden zunächst bei hoher künstlicher Temperatur T nahezu alle Gewichtsänderungen akzeptiert und damit alle Minima wieder verlassen, mit abnehmender Temperatur werden immer mehr nur Verringerungen des Netzfehlers E akzeptiert, wodurch ein möglicherweise zerstörerischer Effekt der simulierten Abkühlungszyklen zurückgeht und hauptsächlich der Fehlerrückführungsalgorithmus die Suche nach der besten Gewichtskombination übernimmt.

Der Fehlerrückführungsalgorithmus wurde im Bereich der Modellierung und Systemidentifikation bereits weitläufig und erfolgreich eingesetzt [18]. Auch die fortgeschrittenen Algorithmen gewinnen die notwendigen Informationen über die Fehlerfunktion E aus dem Fehlerrückführungsalgorithmus. In bezug auf Einfachheit der Programmierung, Anspruch an die Rechnerkapazität und Zuverlässigkeit im *Online*-Einsatz scheint eine Konzentration auf den stochastischen Fehlerrückführungsalgorithmus daher zunächst ausreichend.

Verwendete Symbole Anhang B

a	Boltzmann-Konstante im Simulated Annealing Algorithmus
d_j	Korrekturwert des verborgenen Neurons j
d_k	Korrekturwert des Ausgabeneurons k
E	Summe der halben quadratischen Fehler (Netzfehler)
E_l	Halber quadratischer Fehler eines Eingabe-Ausgabe-Paares l
E_{new}	Neuer Netzfehler im Simulated Annealing Algorithmus
E_{prev}	Vorheriger Netzfehler im Simulated Annealing Algorithmus
e_j	Effektiver Eingang des verborgenen Neurons j
e_k	Effektiver Eingang des Ausgabeneurons k
h	Lernrate
h, i, j, k	Laufindizes
k_1, k_2	Konstanten
l	Laufindex über alle r Eingabe-Ausgabe-Paare des Trainingsdatensatzes

m	Anzahl Neuronen in der verborgenen Schicht einschließlich Bias-Neuron
\mathbf{m}	Momentumterm-Konstante
n	Anzahl Neuronen in der Eingabeschicht einschließlich Bias-Neuron
o_i	Vektor der Eingabeschicht an jedes verborgene Neuron j
o_j	Vektor der verborgenen Schicht an jedes Ausgabeneuron k
o_k	Ausgabevektor des neuronalen Netzes
p	Anzahl Neuronen in der Ausgabeschicht, Dimensionalität des Ausgabevektors o_i
p_{accept}	Wahrscheinlichkeit, mit der eine Netzgewichteänderung akzeptiert wird, die zu einer Erhöhung des Netzfehlers E führt
r	Anzahl der Eingabe-Ausgabe-Paare im Trainingsdatensatz
$rms\text{-Fehler}$	Durchschnittliche Abweichung der Netzausgabe o_k vom Sollausgabewert y_k
t_j, t_k	Transferfunktion des Neurons j, k
t'_j, t'_k	Ableitung der Transferfunktion des Neurons j, k nach dem Effektiven Eingang e_j, e_k
T	Künstliche Temperatur im Simulated Annealing Algorithmus
w_{ji}	Netzgewicht des verborgenen Neurons j bezüglich des Elements i des Vektors o_i
w_{kj}	Netzgewicht des Ausgabeneurons k bezüglich des Elements j des Vektors o_j
Dw_{ji}	Gewichtsänderung des Gewichts w_{ji}
Dw_{kj}	Gewichtsänderung des Gewichts w_{kj}
Dw_{ji}^{prev}	Gewichtsänderung von w_{ji} im vorhergehenden Lernschritt
Dw_{kj}^{prev}	Gewichtsänderung von w_{kj} im vorhergehenden Lernschritt
x_h	Eingabevektor des neuronalen Netzes ($h=1..n-1$)
y_k	Sollausgabevektor des neuronalen Netzes beim Trainingsdatenpaar $(x_h, y_k)_l$ ($k = 1..p$), (Prozeß-)Ausgangsgröße ($k = p = 1$)

Anhang C:

Neuro-Fuzzy-Systeme [10, 24, 26, 45, 46]

Verknüpfung Neuronaler Netze und Fuzzy-Logik-Systeme

Aus der Beschreibung von Fuzzy-Logik-Systemen in Anhang A sollte deutlich geworden sein, daß diese Systeme die Möglichkeit bieten Expertenwissen direkt in die Konstruktion eines solchen Systems einzubringen [3]. So werden die Zugehörigkeitsfunktionen und die Regeln der Regelbasis aus dem Prozeßwissen abgeleitet [17]. Darüberhinaus weisen sie zumindest bei einer konsistenten Regelbasis einen hohen Grad der (sprachlichen) Interpretierbarkeit [3, 26] ihres statischen Eingabe-Ausgabe-Verhaltens auf. So kann bei einer Fehlfunktion des Systems durch Rückverfolgung des Ausgabewertes leicht die zugehörige Regel in der Regelbasis und die durch die Eingabewerte angesprochenen, unscharfen Eingangsmengen identifiziert werden. Dies ist durch die strukturierte Ablage des Expertenwissens in unscharfen Eingangs- und Ausgangsmengen, sowie den Regeln der Regelbasis möglich. Allerdings weisen Fuzzy-Logik-Systeme nicht die Fähigkeit auf, sich an gewünschtes Übertragungsverhalten, repräsentiert durch einen Satz Eingabe-Ausgabe-Daten, zu adaptieren [3, 24]. Fuzzy-Logik-Systeme benötigen Wissen in qualitativer, nicht in quantitativer Form [3]. Aus diesem Grund wird für die Anpassung eines Fuzzy-Logik-Systems an vorliegende Beispieldaten auch oft der in Anhang D dargestellte genetische Algorithmus verwendet [10, 17, 26].

Bei der Beschreibung künstlicher neuronaler Netze in Anhang B hingegen wurden als Vorteile dieser Netze die Anpassung des Eingabe-Ausgabe-Verhaltens an vorhandene oder generierte Eingabe-Ausgabe-Daten genannt [7]. Der Vorteil dieser universellen Approximationseigenschaft wird aber erkaufte mit dem Nachteil, daß ein künstliches neuronales Netz als „*black box*“ fungiert, es also schwierig ist für einen Ausgabewert ein bestimmtes Neuron oder ein bestimmtes Netzgewicht verantwortlich zu machen. Das aus den Eingabe-Ausgabe-Daten extrahierte „Wissen“ ist in neuronalen Netzen unstrukturiert, in allen Netzgewichten distributiert gespeichert [3, 7]. Auch besitzen neuronale Netze keine Extrapolationsfähigkeit, so daß für den gesamten Eingabebereich, in dem das Netz später arbeiten soll, Beispieldaten vorhanden sein müssen [10].

Aus dieser verkürzten Darstellung wird deutlich, daß die Schwächen des einen Ansatzes genau dort liegen, wo sich die Stärken des anderen Ansatzes

befinden. Diese Situation hat zur Entwicklung von Neuro-Fuzzy-Systemen geführt. Neuro-Fuzzy-Systeme sollen die Lernfähigkeit eines neuronalen Netzes mit der Transparenz eines Fuzzy-Logik-Systems verknüpfen [3]. In der Konstruktionsphase ist das Neuro-Fuzzy-System in der Lage, vorhandenes qualitatives Prozeßwissen aufzunehmen. Anschließend ist es möglich, daß System datengestützt nachzuoptimieren [3, 47]. Diese Feinabstimmung geschieht anhand von quantitativen Beispieldaten, also nicht mehr nur qualitativ. Nach der Anpassung des Neuro-Fuzzy-Systems ist seine Wirkungsweise immer noch durchschaubar, und sowohl unscharfe Eingangs- und Ausgangsmengen als auch die Regelbasis ist umgangssprachlich interpretierbar [3, 26, 45, 46]. Ein Neuro-Fuzzy-System ist leistungstärker als ein neuronales Netz oder ein Fuzzy-Logik-System allein, da es die Vorteile beider Ansätze vereint [48].

Bei der im folgenden gewählten Darstellung geht es im Prinzip nur darum ein Fuzzy-Logik-System, wie es in Anhang A beschrieben ist, in ein künstliches Neuronales Netz zu übertragen [46]. Das daraus entstehende Neuro-Fuzzy-System ist rechnerisch identisch mit dem ursprünglichen Fuzzy-Logik-System [48, 50]. Macht man bei der Anwendung des Fehlerrückführungsalgorithmus auf das Neuro-Fuzzy-System einige Einschränkungen bezüglich der Parameteränderungen, so bleibt auch die Rückübersetzung eines Neuro-Fuzzy-Systems in ein Fuzzy-Logik-System nach erfolgter Anpassung gewährleistet. Unabhängig von den Einschränkungen des Fehlerrückführungsalgorithmus, soll bereits an dieser Stelle darauf hingewiesen sein, daß Neuro-Fuzzy-Systeme wesentlich weniger Freiheitsgrade als künstliche neuronale Netze haben [7], .

Restriktionen bei der Abbildung eines Fuzzy-Logik-Systems als Künstliches Neuronales Netz

Soll das Neuro-Fuzzy-System mit dem Fehlerrückführungsverfahren an Eingabe-Ausgabe-Paare angepaßt werden, müssen bei der Wahl der Zugehörigkeitsfunktionen, der logischen Operatoren und der Defuzzifizierungsmethode Einschränkungen gemacht werden. Die wichtigste Einschränkung ist die Forderung nach der stetigen Differenzierbarkeit der Funktionen und Operatoren [50].

Der genaue Verlauf der Zugehörigkeitsfunktion $m_A(x)$ über alle Werte der Basisvariablen x kann aus qualitativem Vorwissen oft nicht erschlossen werden [24]. Es ist daher nötig den Mittenwert oder die Breite einer un-

scharfen Menge anzupassen. Die oft verwendete dreieckförmige Zugehörigkeitsfunktion (A.6) scheidet als Funktion aus, da sie an drei Stellen nicht stetig differenzierbar ist [26, 45]. Anwendung findet daher nur die zu einer Dreieckfunktion flächengleiche Gauß-Funktion (A.8), in der folgenden Abbildung 59 ist deren Zusammenhang nochmal verdeutlicht. Um die Anwendung des Adaptionalgorithmus nicht zu aufwendig werden zu lassen, wird die Unterteilung in eine rechts- und linksseitige Breite der Zugehörigkeitsfunktion aufgehoben und nur noch eine für beide Seiten der Funktion gültige Breite a_{Ai} zugelassen (C.1).

Nachteil dieser Umstellung ist, daß die Gauß-Funktion im Bereich des Modalwertes $x_{m,Ai}$ eine geringere Sensitivität gegenüber Schwankungen des Eingabewerts x zeigt als die Λ -Funktion. Dafür wird aber die Zugehörigkeitsfunktion auch in großen Entfernungen vom Mittenwert nie ganz null, so daß das System immer einen Ausgabewert liefert, da immer eine Regel aktiv ist (abgesehen von dem unlogischen Fall, daß die Regelbasis leer ist, $\mathbf{m}_{FAM}(i,j,k) = 0, \forall i, j, k$). In (C.1) wird zusätzlich noch ein Skalierungsfaktor s_A eingeführt. Er gilt für alle Terme Ai einer linguistischen Variable A und verschiebt die Mittenwerte der Terme $x_{m,Ai}$ und verändert die Funktionsbreiten a_{Ai} . Bei einer Veränderung von s_A wird das Übertragungsverhalten des Neuro-Fuzzy-Systems global beeinflusst. Formal kann die Einführung des Skalierungsfaktors als die Multiplikation einer durch $x_{m,Ai}$ und a_{Ai} beschriebenen unscharfen Zahl mit der scharfen Zahl s_A aufgefaßt werden (siehe Anhang A).

$$\mathbf{m}_{Ai}(x) = \exp\left[-\frac{p}{s_A^2 a_{Ai}^2} (x - s_A x_{m,Ai})^2\right] \quad (\text{C.1})$$

Die Gauß-Funktion eignet sich für die mittleren Terme Ai einer linguistischen Variablen A [33, 50]. Für seitliche Fuzzy-Mengen [33] finden die s-förmigen Funktionen [17] Verwendung. Sie ersetzen Gauß-Trapez-Funktionen (A.9), wobei entweder der linke Begrenzungswert x_l bei $-\infty$ liegt (C.2) (linksseitige Randfunktion), oder sich der der rechte Begrenzungswert x_r bei $+\infty$ befindet (C.3) (rechtsseitige Randfunktion).

Die Einführung der beiden Konstanten k_1 und k_2 ist optional. Sie sollen dazu beitragen den Begrenzungswert $x_{m,Ai}$ und die Breite a_{Ai} besser interpretierbar zu machen, wie dies bei einer Gauß-Trapezfunktion der Fall ist. Wie aus der folgenden Abbildung 60 ersichtlich, ergeben sich für

$x_{m,Ai} = 10$ und $a_{Ai} = 2.5$ die Werte der Konstanten zu $k_1 = 0.875$ und $k_2 = 6.521$. Bedauerlicherweise sind die Zahlenwerte der Konstanten auch von $x_{m,Ai}$ und a_{Ai} abhängig, wenn man über den gesamten Bereich der Basisvariable x die Ähnlichkeit zur Gauß-Trapez-Funktion aufrecht erhalten wollte.

$$m_{Ai}(x) = \frac{1}{1 + \exp\left[\frac{k_2(x - k_1 s_A x_{m,Ai})}{s_A a_{Ai}}\right]} \quad (C.2)$$

$$m_{Ai}(x) = \frac{1}{1 + \exp\left[-\frac{k_2(x - k_1 s_A x_{m,Ai})}{s_A a_{Ai}}\right]} \quad (C.3)$$

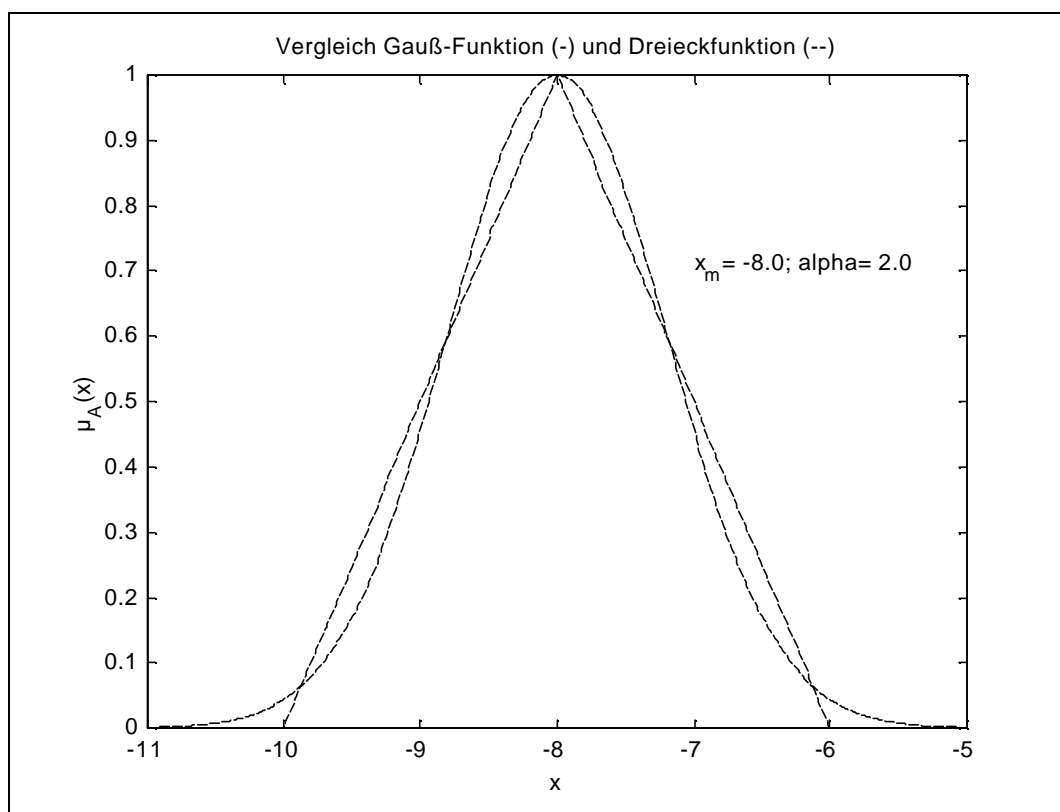


Abb. 58 Vergleich von dreieckförmiger Funktion und Gauß-Funktion mit gleichem Mittenwert x_m und gleicher Breite a

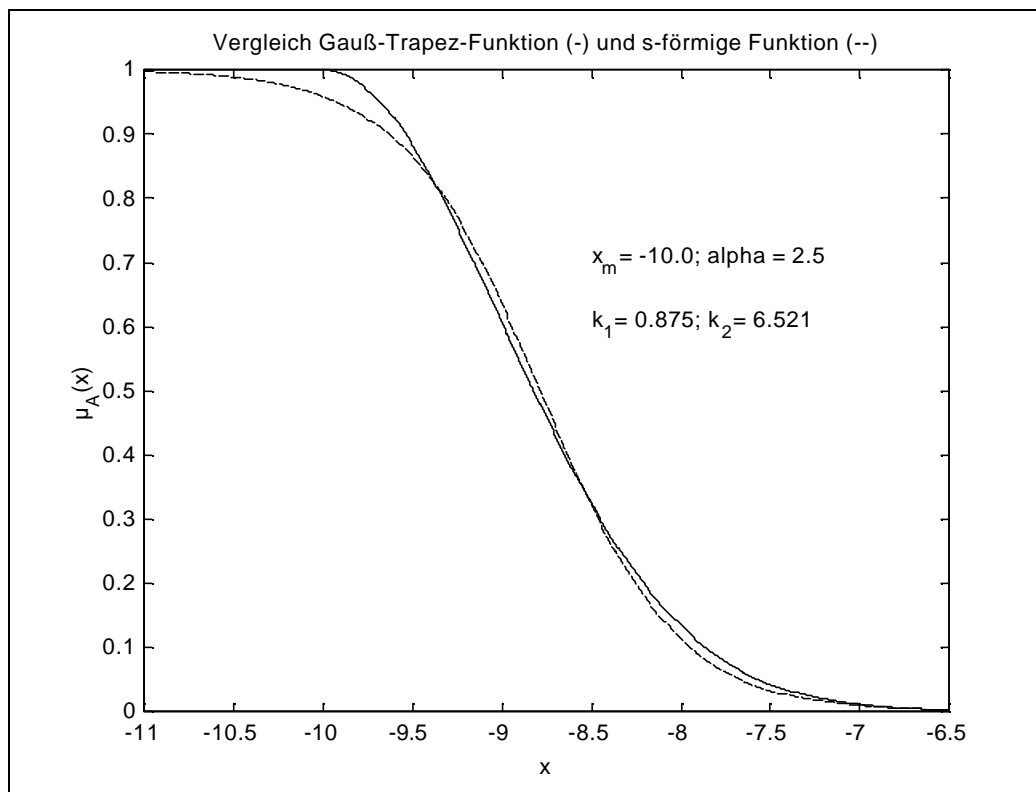


Abb. 59 Vergleich von Gauß-Trapez-Funktion und s-förmiger Funktion mit gleichen Parametern

Auch bei der Wahl der Logik-Operatoren für das Neuro-Fuzzy-System muß auf die Differenzierbarkeit geachtet werden. So sind alle Operatoren, die die *MIN*- oder *MAX*-Funktion enthalten nicht stetig differenzierbar [45]. Zwar ist der *MIN*-Operator gemäß (C.4) differenzierbar, allerdings liegt keine Definition für den Fall $m_A(x) = k$ vor [48].

$$\frac{\partial \text{MIN}(m_A(x), k)}{\partial m_A(x)} = \begin{cases} 1 & \text{für } m_A(x) < k \\ 0 & \text{für } m_A(x) > k \end{cases} \quad (\text{C.4})$$

Als Aggregationsoperator wird daher das algebraische Produkt (A.16) verwendet [50].

Als ODER-Operator bei der Akkumulation aller Regeln, die zu ein und derselben Schlußfolgerung führen, kann aus demselben Grund nicht die *MAX*-Funktion (A.29) benutzt werden. Sie wird ersetzt durch eine modifizierte algebraische Summe (C.5)

$$m_{\text{Akkumulation}}(k) = 1 - \prod_{i,j} (1 - m_{\text{Produkt}}(i, j, k)) \quad (\text{C.5})$$

Weniger aus Gründen der Differenzierbarkeit als vielmehr aufgrund des Rechenaufwand beschränkt sich das hier beschriebene Neuro-Fuzzy-System auf die Höhenmethode (A.35) [50]. Auch hier wird ein Skalierungsfaktor s_C eingeführt (C.6), um die Position der Ausgangssingletons $z_{m,Ck}$ und damit das Übertragungsverhalten global beeinflussen zu können.

$$z^* = \frac{\sum_k s_C z_{m,Ck} \mathbf{m}_{\text{Akkumulation}}(k)}{\sum_k \mathbf{m}_{\text{Akkumulation}}(k)} \quad (\text{C.6})$$

Möglich wäre auch die Ableitung der Schwerpunktmethod und der modifizierten Schwerpunktmethod (A.33). Hierbei müßten die unscharfen Ausgangsmengen Ck ebenfalls als differenzierbare Gauß-Funktionen repräsentiert werden, wobei sowohl die Modalwerte $z_{m,Ck}$ als auch die Breiten a_{Ck} durch einen Fehlerrückführungsalgorithmus angepaßt werden könnten. Wegen der bei diesen Methoden auftretenden Integrale gestaltet sich dieser Weg jedoch erheblich schwieriger als bei der Höhenmethode (A.35).

Da alle oben skizzierten Funktionen in einem Neuro-Fuzzy-System vorkommen, ist verständlich, warum ein solches System nicht aus uniformen Neuronen bestehen kann, wie dies bei einem künstlichen neuronalen Netz der Fall ist. Bei einem Neuro-Fuzzy-System werden die einzelnen auszuführenden Aufgaben Fuzzifizierung, regelbasiertes Schließen und Defuzzifizierung durch ein oder mehrere Schichten repräsentiert. In jeder Schicht befinden sich spezielle Neuronen, die an die rechnerische Aufgabe angepaßt sind [7, 33, 47].

Als Beispiel ist hier das Fuzzifizierungsneuron zur Darstellung der Gauß-Funktion als Zugehörigkeitsfunktion in Abbildung 61 aufgeführt. Eingabewert des Neurons ist der Wert x , Ausgabewert der Fuzzifizierungsneurons ist der Zugehörigkeitsgrad $\mathbf{m}_A(i)$ zur unscharfen Eingangsmenge A_i . Zu diesem Neuron gehören die Netzgewichte x_{m,A_i} , a_{A_i} und s_A , die in (C.1) vorkommen und die Zugehörigkeitsfunktion beschreiben.

Solche speziellen Neuronen existieren für jede Aufgabe in einem Neuro-Fuzzy-System. Einige besitzen Netzgewichte wie das Fuzzifizierungsneuron (x_{m,A_i} , a_{A_i} , s_A), andere wie beispielsweise das UND-Neuron für die Aggregation kommen ohne Netzgewichte aus [33].

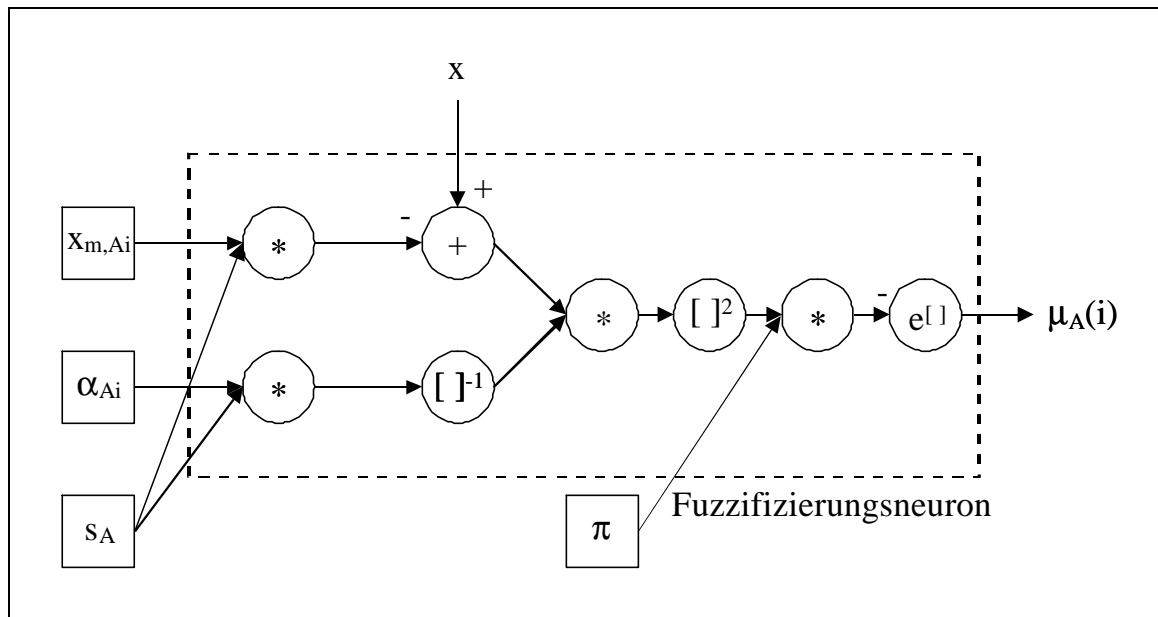


Abb. 60 Verarbeitungsschritte in einem Fuzzifizierungsneuron

In der Abbildung 62 ist ein Neuro-Fuzzy-System mit zwei Eingängen und einem Ausgang abgebildet. Es besteht aus in Schichten angeordneten, aufgabenspezifischen Neuronen und ist rechnerisch äquivalent zu dem im Anhang A beschriebenen Fuzzy-Logik-System.

Die Eingabewerte x^* und y^* gehen an die Neuronen der ersten Schicht. Die Neuronen dieser Eingangsschicht führen keine Rechnung aus, sondern leiten die Werte lediglich an die zugehörigen Fuzzifizierungsneuronen der zweiten Schicht weiter.

Jedem Term A_i (B_j) der linguistischen Variable A (B) ist ein Fuzzifizierungsneuron in der Fuzzifizierungsschicht zugeordnet. Die Fuzzifizierungsneuronen berechnen die Zugehörigkeitsgrade $m_A(i)$ ($m_B(j)$) von x (y) zu den einzelnen Termen A_i (B_j). Mittlere Terme werden beschrieben durch die Gauß-Funktion (C.1); Randterme werden durch die s-förmigen Funktionen (C.2) oder (C.3) ausgedrückt.

In der darauffolgenden Aggregationsschicht wird in UND-Neuronen das algebraische Produkt (A.16) der Zugehörigkeitsgrade $m_A(i)$ und $m_B(j)$ gebildet. Diese Schicht enthält $n \times m$ Neuronen, um alle erdenklichen Kombinationen von $m_A(i)$ ($i = 1..n$) und $m_B(j)$ ($j = 1..m$) behandeln zu können. Ergebnis der Aggregationsschicht ist die Matrix $m_{Aggregation(i,j)}$, die an alle ODER-Neuronen der Akkumulationsschicht geleitet wird.

In der Akkumulationsschicht ist jeder Schlußfolgerung k ($k = 1..o$) ein ODER-Neuron zugeordnet. Im ODER-Neuron k wird die Aggregationsmatrix $m_{Aggregation(i,j)}$ für alle i, j elementweise mit der k -ten Schicht des

dreidimensionalen Felds $m_{FAM}(i,j,k)$ multipliziert. Die resultierende k -te Schicht des Feldes $m_{Produkt}(i,j,k)$ wird über die modifizierte algebraische Summe (C.5) verknüpft. Jedes ODER-Neuron liefert einen Grad der Entschlossenheit $m_{Akkumulation}(k)$ für die Schlußfolgerung k .

Aggregations- und Akkumulationsschicht zusammen können als ein zweischichtiges, heteroassoziatives Netzwerk (Inferenznetz [46]) aufgefaßt werden, das die unscharfen Eingänge in unscharfe Ausgänge abbildet [24].

Die letzte Schicht des Neuro-Fuzzy-Systems ist die Defuzzifizierungsschicht, die aus nur einem Neuron besteht. Dieses Defuzzifizierungsneuron enthält als Netzgewichte den Skalierungsfaktor s_C und die Lage der Ausgangssingletons $z_{m,Ck}$. Es führt die Defuzzifizierung nach der Höhenmethode (C.6) aus.

Es ist zu erkennen, daß die Umsetzung eines Fuzzy-Logik-Systems in ein künstlichen neuronalen Netzes zu einem strukturierten neuronalen Netz mit lokalen Netzwerken führt [3].

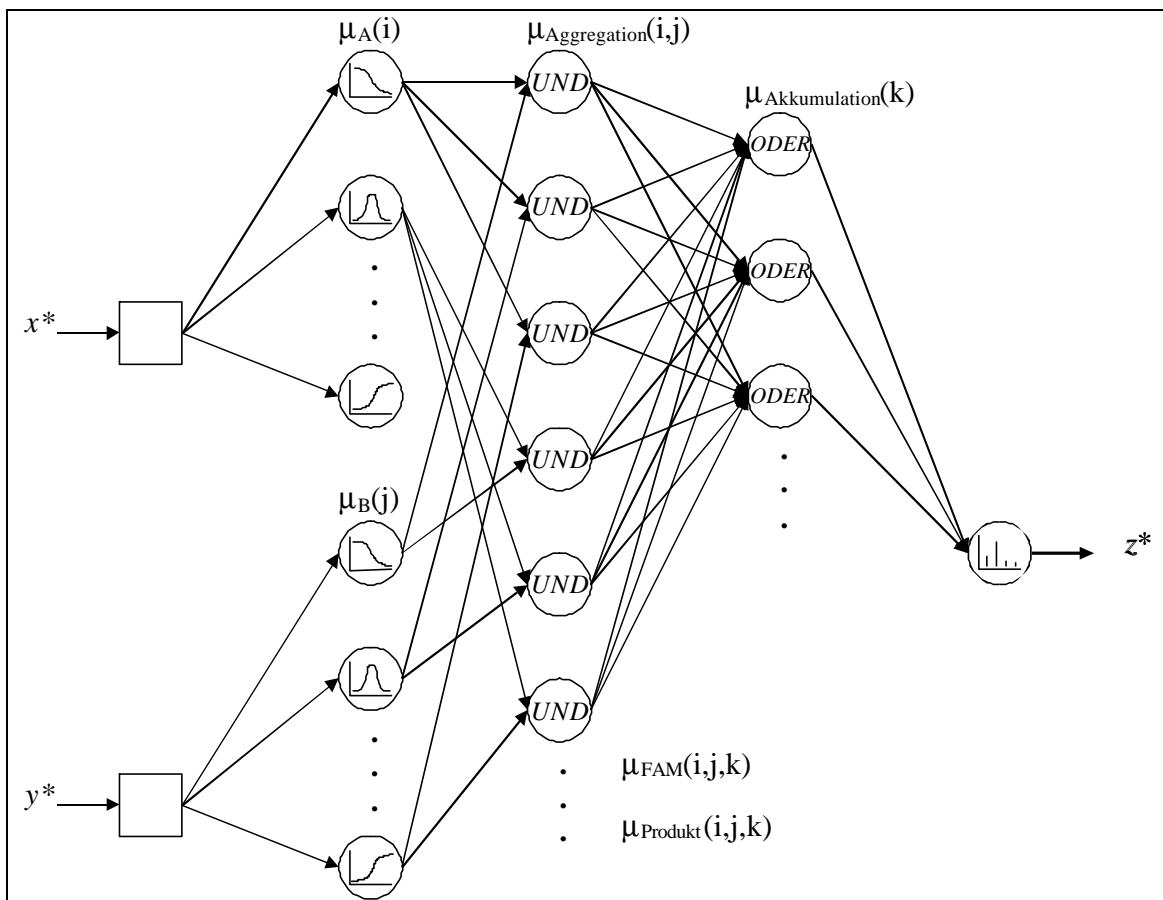


Abb. 61 Grafische Darstellung eines Neuro-Fuzzy-Systems mit zwei Eingängen und einem Ausgang

Fehlerrückführungsalgorithmus [10, 26, 33, 48]

Durch die Anwendung des Fehlerrückführungsalgorithmus auf das Neuro-Fuzzy-System soll das Übertragungsverhalten an die Beispieldatentripel $(x^*, y^*, z^{*,ref})_l$ ($l = 1..r$) angepaßt werden. Die Parameter des Neuro-Fuzzy-Systems werden dabei als Netzgewichte interpretiert [33]. Da es kein Standardlernverfahren für Neuro-Fuzzy-Systeme gibt, wird das Lernverfahren auf das vorgegebene System zugeschnitten [46, 48].

Gegenüber einem künstlichen neuronalen Netz ist der Lernprozeß beschleunigt, da Vorabwissen sowohl in die Struktur als auch in die einzelnen Werte des Neuro-Fuzzy-Systems eingebracht werden kann [46, 47]. Dadurch, daß man mit einem vorstrukturierten System beginnt, sind die Freiheitsgrade des Lernens aber von vornherein eingeschränkt. Das Resultat des Lernens bleibt nachvollziehbar und manuell nachoptimierbar [7, 46]. Die Struktur des zugrunde liegenden Fuzzy-Logik-Systems als lokal strukturierte Netze verhindert die verschlüsselte Ablage von Wissen im Neuro-Fuzzy-System [46], was nach Ende der Anpassung eine Rückübersetzung möglich macht.

Zunächst sollen in (C.7) bis (C.36) die Ableitungen der Funktionen nach den einzelnen Parametern angegeben werden, bevor dann (C.37) bis (C.55) das Gradientenabstiegsverfahren beschrieben wird, mit dem die Parameter an die r Eingabe-Ausgabe-Tripel $(x^*, y^*, z^{*,ref})_l$ angepaßt werden.

Die Ableitungen sind meistens einfache Ausdrücke, die für eine schnellere Abarbeitung auf einem Rechner noch weiter vereinfacht werden können. Dazu ist es nötig, Funktionswerte und Teilergebnisse bereits beim Durchlaufen der Eingabewerte x^*, y^* durch das Neuro-Fuzzy-System zwischenspeichern, um bei der Berechnung der Ableitungen auf diese zurückgreifen zu können. Auf diese Weise ist es gerade bei zeitsensiblen Anwendungen möglich Rechenzeit durch Speicherkapazität zu ersetzen.

Die Ableitung des scharfen Ausgabewertes z^* nach dem Skalierungsfaktor s_C wird in (C.7) vorgenommen. Sie entspricht dem Ausgabewert z^* dividiert durch den Skalierungsfaktor s_C .

$$\frac{\partial z^*}{\partial s_C} = \frac{\sum_k z_{m,Ck} \mathbf{m}_{\text{Akkumulation}}(k)}{\sum_k \mathbf{m}_{\text{Akkumulation}}(k)} = \frac{z^*}{s_C} \quad (\text{C.7})$$

Ein Fehler könnte für $s_C = 0$ gemeldet werden. Diese Situation ist jedoch unsinnig, da dann die Positionen aller Ausgangssingletons bei Null

lokalisiert wären und das Neuro-Fuzzy-System unabhängig von den Eingabewerten $z^* = 0$ ausgeben würde.

In (C.8) wird die Ableitung des Ausgabewertes z^* nach der Position eines Ausgangssingletons $z_{m,Ck}$ berechnet. Wird bereits bei der Anwendung des Neuro-Fuzzy-Systems der Nenner der Defuzzifizierungsfunktion (C.9) zwischengespeichert, läßt sich der Ausdruck weiter zu (C.10) vereinfachen.

$$\frac{\partial z^*}{\partial z_{m,Ck}} = \frac{s_C \mathbf{m}_{\text{Akkumulation}}(k)}{\sum_k \mathbf{m}_{\text{Akkumulation}}(k)} \quad (\text{C.8})$$

$$\text{Nenner} = \sum_k \mathbf{m}_{\text{Akkumulation}}(k) \quad (\text{C.9})$$

$$\frac{\partial z^*}{\partial z_{m,Ck}} = \frac{s_C \mathbf{m}_{\text{Akkumulation}}(k)}{\text{Nenner}} \quad (\text{C.10})$$

Die Ableitung von z^* nach den Elementen des Akkumulationsvektors $\mathbf{m}_{\text{Akkumulation}}(k)$ ist in (C.11) gegeben. Wird neben dem Nenner auch der Zähler der Defuzzifizierungsfunktion gespeichert (C.12), kann der rechnerische Aufwand erheblich reduziert werden (C.13).

$$\frac{\partial z^*}{\partial \mathbf{m}_{\text{Akkumulation}}(k)} = \frac{s_C z_{m,Ck} \sum_k \mathbf{m}_{\text{Akkumulation}}(k) - \sum_k s_C z_{m,Ck} \mathbf{m}_{\text{Akkumulation}}(k)}{\left[\sum_k \mathbf{m}_{\text{Akkumulation}}(k) \right]^2} \quad (\text{C.11})$$

$$\text{Zähler} = \sum_k s_C z_{m,Ck} \mathbf{m}_{\text{Akkumulation}}(k) \quad (\text{C.12})$$

$$\frac{\partial z^*}{\partial \mathbf{m}_{\text{Akkumulation}}(k)} = \frac{s_C z_{m,Ck} \text{Nenner} - \text{Zähler}}{\text{Nenner}^2} \quad (\text{C.13})$$

Sowohl in (C.10), als auch in (C.13) könnte für den Fall $\text{Nenner} = 0$ ein Fehler auftreten. Da aber bereits bei Defuzzifizierung durch den Nenner dividiert wird, taucht der Fehler schon bei der Berechnung von z^* auf, so daß kein Ausgabewert des Neuro-Fuzzy-Systems angegeben werden kann. Ein Nenner von Null bedeutet aber das keine der k Schlußfolgerungen ange-

sprochen wird. Diese Situation sollte schon bei der Erstellung des Systems berücksichtigt und vermieden werden.

Bei der Berechnung der Ableitung von $\mathbf{m}_{\text{Akkumulation}}(k)$ nach $\mathbf{m}_{\text{Produkt}}(i,j,k)$ in (C.14) wurde bereits eine Vereinfachung vorgenommen, die dazu führt, daß die Funktion im Fall von $\mathbf{m}_{\text{Produkt}}(i,j,k) = 1$ (d.h. der Entschlossenheitsgrad einer Schlußfolgerung ist 1) nicht definiert ist. Diese Definitionslücke würde bei Verzicht auf die Vereinfachung nicht auftreten. Bei der Implementierung muß dieser Sonderfall überprüft werden und die Ableitung in diesem Fall gleich 1 gesetzt werden.

$$\frac{\partial \mathbf{m}_{\text{Akkumulation}}(k)}{\partial \mathbf{m}_{\text{Produkt}}(i,j,k)} = \frac{\prod_{i,j} (1 - \mathbf{m}_{\text{Produkt}}(i,j,k))}{1 - \mathbf{m}_{\text{Produkt}}(i,j,k)} = \frac{1 - \mathbf{m}_{\text{Akkumulation}}(k)}{1 - \mathbf{m}_{\text{Produkt}}(i,j,k)} \quad (\text{C.14})$$

Sehr einfach gestaltet sich die Ableitung von $\mathbf{m}_{\text{Produkt}}(i,j,k)$ nach den Einträgen der unscharfen assoziativen Karten $\mathbf{m}_{\text{FAM}}(i,j,k)$ (C.15) und nach der Aggregationsmatrix $\mathbf{m}_{\text{Aggregation}}(i,j)$ (C.16). Ein Fehler würde also nur nach $\mathbf{m}_{\text{Aggregation}}(i,j)$ zurückpropagiert werden, wenn der Eintrag in der Regelbasis $\mathbf{m}_{\text{FAM}}(i,j,k)$ von Null verschieden ist, d.h. der Regeleintrag auch zur Schlußfolgerung k führt.

Andererseits würde der Fehler nur zu einer Änderung der unscharfen assoziativen Karten $\mathbf{m}_{\text{FAM}}(i,j,k)$ führen, wenn die Prämisse $\mathbf{m}_{\text{Aggregation}}(i,j) \neq 0$, zumindest teilweise erfüllt ist.

$$\frac{\partial \mathbf{m}_{\text{Produkt}}(i,j,k)}{\partial \mathbf{m}_{\text{Aggregation}}(i,j)} = \mathbf{m}_{\text{FAM}}(i,j,k) \quad (\text{C.15})$$

$$\frac{\partial \mathbf{m}_{\text{Produkt}}(i,j,k)}{\partial \mathbf{m}_{\text{FAM}}(i,j,k)} = \mathbf{m}_{\text{Aggregation}}(i,j) \quad (\text{C.16})$$

Die Ableitung der Aggregationsmatrix $\mathbf{m}_{\text{Aggregation}}(i,j)$ nach den Vektoren $\mathbf{m}_A(i)$ und $\mathbf{m}_B(j)$ ist in (C.17) und (C.18) gegeben. Ein Fehler kann also nur zum Term A_i einer linguistischen Eingangsvariable A zurückgeführt werden, wenn ein oder mehrere Terme B_j der anderen Eingangsvariable B angesprochen werden und umgekehrt.

$$\frac{\partial \mathbf{m}_{\text{Aggregation}}(i,j)}{\partial \mathbf{m}_A(i)} = \mathbf{m}_B(j) \quad (\text{C.17})$$

$$\frac{\partial \mathbf{m}_{Aggregation}(i, j)}{\partial \mathbf{m}_B(j)} = \mathbf{m}_A(i) \quad (\text{C.18})$$

Ein recht komplizierter Ausdruck entsteht bei der Ableitung der Gauß-Funktion (C.1) nach dem Skalierungsfaktor s_A , da sich dieser auf den Mittenwert $x_{m,Ai}$ und auf die Breite \mathbf{a}_{Ai} der Eingangsmenge Ai auswirkt. Eine Vereinfachung kann erzielt werden, wenn der Exponent der Gauß-Funktion gespeichert wird (C.20). Dann kann $-2Exponent$ ausgeklammert werden und es entsteht (C.21). Bei Berechnung der vereinfachten Ableitung kann für den Fall $x = s_A \mathbf{x}_{m,Ai}$ ein Fehler auftreten, dann kann der gesamte Ausdruck gleich Null gesetzt werden. Für $x = s_A \mathbf{x}_{m,Ai}$ fällt der Eingabewert x^* genau auf den Modalwert der Zugehörigkeitsfunktion, sodaß die Ableitung nach dem Skalierungsfaktor Null wird.

$$\frac{\partial \mathbf{m}_A(i)}{\partial s_A} = \left[\frac{2\mathbf{p}}{s_A^3 \mathbf{a}_{Ai}^2} (x - s_A x_{m,Ai})^2 + \frac{2\mathbf{p}x_{m,Ai}}{s_A^2 \mathbf{a}_{Ai}^2} (x - s_A x_{m,Ai}) \right] \cdot \exp \left[-\frac{\mathbf{p}}{s_A^2 \mathbf{a}_{Ai}^2} (x - s_A x_{m,Ai})^2 \right] \quad (\text{C.19})$$

$$Exponent = -\frac{\mathbf{p}}{s_A^2 \mathbf{a}_{Ai}^2} (x - s_A x_{m,Ai})^2 \quad (\text{C.20})$$

$$\frac{\partial \mathbf{m}_A(i)}{\partial s_A} = -2Exponent \cdot \left[\frac{1}{s_A} + \frac{x_{m,Ai}}{x - s_A x_{m,Ai}} \right] \cdot \mathbf{m}_A(i) \quad (\text{C.21})$$

Die Ableitung der Gauß-Funktion nach dem Mittenwert $x_{m,Ai}$ ist in (C.22) angegeben. Auch hier weist die vereinfachende Darstellung (C.23) eine Definitionslücke für den Fall $x = s_A \mathbf{x}_{m,Ai}$ auf. An dieser Stelle weist die Funktion ihr Maximum auf, an dem die Ableitung Null ist.

$$\frac{\partial \mathbf{m}_A(i)}{\partial x_{m,Ai}} = \frac{2\mathbf{p}}{s_A \mathbf{a}_{Ai}^2} (x - s_A x_{m,Ai}) \cdot \exp \left[-\frac{\mathbf{p}}{s_A^2 \mathbf{a}_{Ai}^2} (x - s_A x_{m,Ai})^2 \right] \quad (\text{C.22})$$

$$\frac{\partial \mathbf{m}_A(i)}{\partial x_{m,Ai}} = -\frac{2s_A Exponent}{(x - s_A x_{m,Ai})} \mathbf{m}_A(i) \quad (\text{C.23})$$

Die Ableitung der Gauß-Funktion nach der Breite \mathbf{a}_{Ai} folgt in (C.24), die Vereinfachung (C.25) kann ohne zusätzlich Überprüfung programmiert werden.

$$\frac{\partial \mathbf{m}_A(i)}{\partial \mathbf{a}_{Ai}} = \frac{2\mathbf{p}}{s_A^2 \mathbf{a}_{Ai}^3} (x - s_A x_{m,Ai})^2 \exp \left[-\frac{\mathbf{p}}{s_A^2 \mathbf{a}_{Ai}^2} (x - s_A x_{m,Ai})^2 \right] \quad (\text{C.24})$$

$$\frac{\partial \mathbf{m}_A(i)}{\partial \mathbf{a}_{Ai}} = \frac{-2 \text{Exponent}}{\mathbf{a}_{Ai}} \mathbf{m}_A(i) \quad (\text{C.25})$$

Bei der Berechnung des Exponenten der Gauß-Funktion (C.20) kann natürlich ein Fehler für den Fall $s_A = 0$ oder $\mathbf{a}_{Ai} = 0$ auftreten. Dieser Fehler tritt dann aber bereits während der Ausführung des Neuro-Fuzzy-Systems auf. Der Fall $s_A = 0$ beschreibt die unsinnige Situation, daß alle unscharfen Eingangsmengen Singletons mit einem Modalwert von Null sind. Der Fall $\mathbf{a}_{Ai} = 0$ hingegen beschreibt den Term Ai ebenfalls als Singleton mit beliebigen Mittenwert. Singletons werden jedoch nicht für Eingangsmengen verwendet, womit dieser Fall ebenfalls unsinnig ist.

Auch die Ableitung der linksseitigen Randfunktion (C.2) nach dem Skalierungsfaktor s_A liefert zunächst einen recht umfangreichen Ausdruck (C.26). Wird bereits bei der Berechnung der Zugehörigkeitsfunktion (C.2) der Wert der Exponentialfunktion $Efkt$ (C.27) zwischengespeichert, gestaltet sich die Berechnung der Ableitung erheblich einfacher (C.28). Definitionslücken bestehen auch hier bei den unlogischen Fällen $\mathbf{a}_{Ai} = 0$ (die Zugehörigkeitsfunktion beschreibt eine klassische Menge ohne Unschärfbereich) oder $s_A = 0$ (die Zugehörigkeitsfunktion ist die klassische Menge aller negativen Zahlen ohne Unschärfbereich).

$$\frac{\partial \mathbf{m}_A(i)}{\partial s_A} = \frac{k_2 x}{s_A^2 \mathbf{a}_{Ai}} \cdot \exp \left[\frac{k_2 (x - k_1 s_A x_{m,Ai})}{s_A \mathbf{a}_{Ai}} \right] \cdot \frac{1}{\left[1 + \exp \left[\frac{k_2 (x - k_1 s_A x_{m,Ai})}{s_A \mathbf{a}_{Ai}} \right] \right]^2} \quad (\text{C.26})$$

$$Efkt = \exp \left[\frac{k_2 (x - k_1 s_A x_{m,Ai})}{s_A \mathbf{a}_{Ai}} \right] \quad (\text{C.27})$$

$$\frac{\partial \mathbf{m}_A(i)}{\partial s_A} = \frac{k_2 x}{s_A^2 \mathbf{a}_{Ai}} Efkt \mathbf{m}_A(i)^2 \quad (\text{C.28})$$

Die Ableitung der linksseitigen Randfunktion nach dem Wert $x_{m,Ai}$ ist in (C.29) dargestellt. Aufgrund des s-förmigen Verlaufs wird die Ableitung

auch für x -Werte in großen Entfernungen vom Wert $x_{m,Ai}$ nie ganz Null. Die vereinfachende Berechnung (C.30) ist für alle Stellen definiert.

$$\frac{\partial \mathbf{m}_A(i)}{\partial x_{m,Ai}} = \frac{k_1 k_2}{\mathbf{a}} \cdot \exp\left[\frac{k_2(x - k_1 s_A x_{m,Ai})}{s_A \mathbf{a}_{Ai}}\right] \cdot \frac{1}{\left[1 + \exp\left[\frac{k_2(x - k_1 s_A x_{m,Ai})}{s_A \mathbf{a}_{Ai}}\right]\right]^2} \quad (\text{C.29})$$

$$\frac{\partial \mathbf{m}_A(i)}{\partial x_{m,Ai}} = \frac{k_1 k_2}{\mathbf{a}} \text{Efkt } \mathbf{m}_A(i)^2 \quad (\text{C.30})$$

In (C.31) wird die Ableitung der linksseitigen Randfunktion nach der Breite \mathbf{a}_{Ai} vorgenommen. Sie kann im Gegensatz zu den beiden obigen Ableitungen für den Fall $x = s_A x_{m,Ai}$ Null werden. An dieser Stelle besitzt die Zugehörigkeitsfunktion einen Wendepunkt mit dem Funktionswert von $\mathbf{m}_{Ai}(x) = 0.5$. Eine über die Breite \mathbf{a}_{Ai} vorgenommene Streckung oder Stauchung in x -Richtung hätte also keine Auswirkungen auf den Zugehörigkeitsgrad $\mathbf{m}_{Ai}(x)$. Die Ableitung ist entsprechend Null.

$$\frac{\partial \mathbf{m}_A(i)}{\partial \mathbf{a}_{Ai}} = \frac{k_2(x - k_1 s_A x_{m,Ai})}{s_A \mathbf{a}_{Ai}^2} \cdot \exp\left[\frac{k_2(x - k_1 s_A x_{m,Ai})}{s_A \mathbf{a}_{Ai}}\right] \cdot \frac{1}{\left[1 + \exp\left[\frac{k_2(x - k_1 s_A x_{m,Ai})}{s_A \mathbf{a}_{Ai}}\right]\right]^2} \quad (\text{C.31})$$

$$\frac{\partial \mathbf{m}_A(i)}{\partial \mathbf{a}_{Ai}} = \frac{k_2(x - k_1 s_A x_{m,Ai})}{s_A \mathbf{a}_{Ai}^2} \text{Efkt } \mathbf{m}_A(i)^2 \quad (\text{C.32})$$

Die Ableitungen der s-förmigen, rechtseitigen Funktion nach dem Skalierungsfaktor s_A , nach dem Wert $x_{m,Ai}$ und der Breite \mathbf{a}_{Ai} der Zugehörigkeitsfunktion sind in den Formeln (C.34) bis (C.36) dargestellt. Für die vereinfachende Berechnung wird der Wert der Exponentialfunktion *Efkt* (C.33) zwischengespeichert. Bezüglich des Auftretens von Definitionslücken gilt das für die linksseitige Randfunktion Ausgeführte.

$$\text{Efkt} = \exp\left[-\frac{k_2(x - k_1 s_A x_{m,Ai})}{s_A \mathbf{a}_{Ai}}\right] \quad (\text{C.33})$$

$$\frac{\partial \mathbf{m}_A(i)}{\partial s_A} = -\frac{k_2 x}{s_A \mathbf{a}_{Ai}} \text{Efkt } \mathbf{m}_A(i)^2 \quad (\text{C.34})$$

$$\frac{\partial \mathbf{m}_A(i)}{\partial x_{m,Ai}} = -\frac{k_1 k_2}{\mathbf{a}} \text{Efkt } \mathbf{m}_A(i)^2 \quad (\text{C.35})$$

$$\frac{\partial \mathbf{m}_A(i)}{\partial \mathbf{a}_{Ai}} = -\frac{k_2 (x - k_1 s_A x_{m,Ai})}{s_A \mathbf{a}_{Ai}^2} \text{Efkt } \mathbf{m}_A(i)^2 \quad (\text{C.36})$$

Nachdem alle benötigten Ableitungen formuliert sind, folgt nun die Darstellung des Fehlerrückführalgorithmus für das Neuro-Fuzzy-System. Der Algorithmus ist ein einfaches Gradientenabstiegsverfahren, bei dem ein Parameter um einen Bruchteil des negativen Gradienten der Fehlerfunktion verändert wird. Er nimmt eine Anpassung der Netzgewichte des Neuro-Fuzzy-Systems nach jedem Trainingsmuster indiziert mit l ($l=1..r$) vor. Bezüglich der möglichen Erweiterungen und Ausführungsformen des Fehlerrückführalgorithmus gilt das in Anhang B zum Fehlerrückführalgorithmus für künstliche neuronale Netze Gesagte. Wie schon bei den künstlichen neuronalen Netzen handelt es sich bei der Anpassung eines Neuro-Fuzzy-Systems um die Minimierung einer Quadratsumme, so daß auch dieselben mathematischen Verfahren zur Lösung dieses Optimierungsproblems herangezogen werden können [33]. Die folgende Darstellung beschränkt sich jedoch auf das einfache Gradientenabstiegsverfahren, ohne weiter auf Verfahren höherer Ordnung einzugehen. Dem Neuro-Fuzzy-System werden die scharfen Eingabewerte eines Beispielsatzes x^* und y^* präsentiert und der Ausgabewert z^* wird berechnet. Ist das Neuro-Fuzzy-System noch nicht vollständig an alle Eingabe-Ausgabe-Tripel adaptiert, wird der Ausgabewert z^* von dem Sollausgabewert $z^{*,ref}$ verschieden sein, der eigentlich zu den Eingabewerten x^* und y^* gehört. Aus $z^{*,ref}$ und z^* wird gemäß (C.37) der Ausgabefehler des Neuro-Fuzzy-Systems bestimmt. Er entspricht der halben quadratischen Differenz von $z^{*,ref}$ und z^* , und ist identisch mit dem Netzfehler (B.10) eines künstlichen neuronalen Netzes.

$$E_l = \frac{1}{2} (z^{*,ref} - z^*)_l^2 \quad (\text{C.37})$$

Die Anpassung des Skalierungsfaktors s_C und der Positionen $z_{m,Ck}$ der Ausgangsmengen Ck gestaltet sich noch recht einfach, da das Defuzzifizierungsneuron direkt am Systemausgang des Neuro-Fuzzy-Systems liegt.

So wird zur Anpassung des Skalierungsfaktors einmal die Kettenregel (C.38) angewendet und die Änderung Δs_C kann berechnet (C.39) und vorgenommen (C.40) werden.

$$\Delta s_C = -\mathbf{h}_{s_C} \frac{\partial E_l}{\partial s_C} = -\mathbf{h}_{s_C} \frac{\partial E_l}{\partial z^*} \frac{\partial z^*}{\partial s_C} \quad (\text{C.38})$$

$$\Delta s_C = \mathbf{h}_{s_C} (z^{*,ref} - z^*) \cdot \left(\frac{z^*}{s_C} \right) \quad (\text{C.39})$$

$$s_C^{neu} = s_C^{alt} + \Delta s_C \quad (\text{C.40})$$

Auch die Anpassungen der Positionen der Ausgangssingletons $z_{m,Ck}$ zur Minimierung des Fehlers E_l läßt sich durch einmalige Anwendung der Kettenregel ermitteln (C.41). Hierbei ist $\Delta z_{m,Ck}$ ein k -dimensionaler Vektor, da für jedes Ausgangssingleton eine Änderung berechnet wird. Trägt eine Schlußfolgerung Ck nicht zum Defuzzifizierungsergebnis bei ($\mathbf{m}_{Akkumulation}(k) = 0$), so wird die Lage des Ausgangssingletons auch nicht verändert $\Delta z_{m,Ck} = 0$.

$$\Delta z_{m,Ck} = -\mathbf{h}_{z_{m,Ck}} \frac{\partial E_l}{\partial z_{m,Ckdf}} = -\mathbf{h}_{z_{m,Ck}} \frac{\partial E_l}{\partial z^*} \frac{\partial z^*}{\partial z_{m,Ck}} \quad (\text{C.41})$$

$$\Delta z_{m,Ck} = \mathbf{h}_{z_{m,Ck}} (z^{*,ref} - z^*) \frac{s_C \mathbf{m}_{Akkumulation}(k)}{Nenner} \quad (\text{C.42})$$

$$z_{m,Ck}^{neu} = z_{m,Ck}^{alt} + \Delta z_{m,Ck} \quad (\text{C.43})$$

Die Adaption der Einträge in die unscharfen assoziativen Karten $\mathbf{m}_{FAM}(i,j,k)$ macht die mehrmalige Anwendung der Kettenregel nötig (C.44). Die Ableitungen sind schon besprochen worden und werden in (C.45) eingesetzt. Es ist zu berücksichtigen, daß die in (C.45) auftretende Aggregationsmatrix $\mathbf{m}_{Aggregation}(i,j)$ nur die Indizes i und j besitzt. Bei der Programmierung ist darauf zu achten, daß die Matrix dann für alle k gültig ist.

$$\begin{aligned} \Delta \mathbf{m}_{FAM}(i, j, k) &= -\mathbf{h}_{FAM} \frac{\partial E_l}{\partial \mathbf{m}_{FAM}(i, j, k)} \\ &= -\mathbf{h}_{FAM} \frac{\partial E_l}{\partial z^*} \frac{\partial z^*}{\partial \mathbf{m}_{Akkumulation}(k)} \frac{\partial \mathbf{m}_{Akkumulation}(k)}{\partial \mathbf{m}_{Produkt}(i, j, k)} \frac{\partial \mathbf{m}_{Produkt}(i, j, k)}{\partial \mathbf{m}_{FAM}(i, j, k)} \end{aligned} \quad (C.44)$$

$$\Delta \mathbf{m}_{FAM}(i, j, k) = \mathbf{h}_{FAM} \left(z^{*,ref} - z^* \right) \left(\frac{s_C z_{m,Ck} \text{Nenner} - \text{Zähler}}{\text{Nenner}^2} \right) \left(\frac{1 - \mathbf{m}_{Akkumulation}(k)}{1 - \mathbf{m}_{Produkt}(i, j, k)} \right) \mathbf{m}_{Aggregation}(i, j) \quad (C.45)$$

Da die in den unscharfen assoziativen Karten $\mathbf{m}_{FAM}(i, j, k)$ eingetragenen Relevanzfaktoren nur Werte zwischen einschließlich 0 (Regel nicht gültig) und 1 (Regel vollständig gültig) haben sollten, ist bei der Berechnung der neuen Relevanzfaktoren nach (C.46) noch ein Begrenzer zu programmieren, der die Einhaltung dieses Intervalls sicherstellt.

$$\mathbf{m}_{FAM}(i, j, k)^{neu} = \mathbf{m}_{FAM}(i, j, k)^{alt} + \Delta \mathbf{m}_{FAM}(i, j, k) \quad (C.46)$$

Werden die Änderungen für die Einträge in $\mathbf{m}_{FAM}(i, j, k)$ gemäß (C.46) für alle Trainingsdatentripel $(x^*, y^*, z^{*,ref})_l$ mehrfach vorgenommen, so ist das Ergebnis meist keine konsistente Regelbasis mehr [3]. Die Regelbasis muß daher noch heuristisch auf Plausibilität überprüft werden [26], wobei Relevanzfaktoren unterhalb eines Schwellenwertes auf Null und Relevanzfaktoren nahe 1 auf 1 gesetzt werden können. Wegen dieser Schwierigkeit wird in [50] vorgeschlagen, die Änderung der Relevanzfaktoren auch nur für die Feinabstimmung des System-Übertragungsverhaltens zu verwenden, und die hauptsächliche Anpassung über die Eingangs- und Ausgangsmengen vorzunehmen.

Im folgenden werden die Anpassungen des Skalierungsfaktor s_A , der Werte $x_{m,Ai}$ und der Breiten a_{Ai} beschrieben. Dabei treten die Ableitungen der Zugehörigkeitsfunktionen nach diesen Parametern auf. Da man bei der Festlegung der unscharfen Eingangsmengen A_i für die linguistische Variable A die Wahl zwischen der Gauß-Funktion (C.1), der linksseitigen (C.2) und der rechtsseitigen (C.3) Randfunktion hat, sind die Ableitungen auch unterschiedlich. Bei der vereinfachenden Berechnung der Ableitungen (Formeln (C.48), (C.51) und (C.54)) bleiben diese Ableitung daher zunächst unverändert stehen und sind je nach Wahl der Zugehörigkeitsfunktion durch die oben genannten Ausdrücke zu ersetzen. In der folgenden Darstellung wird $x_{m,Ai}$ weiterhin als Mittenwert bezeichnet, für die

s-förmigen Randfunktionen ist diese Benennung unsinnig. Hier würde die Bezeichnung Begrenzungswert eher passen, worauf zur einfacheren Darstellung verzichtet wird.

Bei der Anpassung des Skalierungsfaktors s_A ist der Ausgabefehler E_l durch das gesamte Neuro-Fuzzy-System zurückzuführen, was zu dem unhandlichen Ausdruck (C.47) führt.

$$\begin{aligned} \Delta s_A &= -\mathbf{h}_{s_A} \frac{\partial E_l}{\partial s_A} \\ &= -\mathbf{h}_{s_A} \frac{\partial E_l}{\partial z^*} \frac{\partial z^*}{\partial \mathbf{m}_{\text{Akkum.}}(k)} \frac{\partial \mathbf{m}_{\text{Akkum.}}(k)}{\partial \mathbf{m}_{\text{Prod.}}(i,j,k)} \frac{\partial \mathbf{m}_{\text{Prod.}}(i,j,k)}{\partial \mathbf{m}_{\text{Aggreg.}}(i,j)} \frac{\partial \mathbf{m}_{\text{Aggreg.}}(i,j)}{\partial \mathbf{m}_A(i)} \frac{\partial \mathbf{m}_A(i)}{\partial s_A} \end{aligned} \quad (\text{C.47})$$

$$\Delta s_A = \mathbf{h}_{s_A} (z^{*,\text{ref}} - z^*) \left(\frac{s_C z_{m,Ck} \text{Nenner} - \text{Zähler}}{\text{Nenner}^2} \right) \left(\frac{1 - \mathbf{m}_{\text{Akkum.}}(k)}{1 - \mathbf{m}_{\text{Prod.}}(i,j,k)} \right) \mathbf{m}_{\text{FAM}}(i,j,k) \mathbf{m}_B(j) \frac{\partial \mathbf{m}_A(i)}{\partial s_A} \quad (\text{C.48})$$

Bei Berechnung des Ausdrucks (C.48) erhält man zunächst ein dreidimensionales Feld, in welchem die Änderungen des Skalierungsfaktors s_A stehen. Für alle Eingangsmengen A_i existiert jedoch nur ein einziger Skalierungsfaktor, sodaß eine Summation über alle i, j und k vorgenommen wird (C.49).

$$s_A^{\text{neu}} = s_A^{\text{alt}} + \sum \Delta s_A \quad (\text{C.49})$$

Ein ähnlicher Ausdruck wird für die Anpassung des Modalwertes $x_{m,Ai}$ in (C.50) erhalten, wobei lediglich die letzte Ableitung differiert.

$$\begin{aligned} \Delta x_{m,Ai} &= -\mathbf{h}_{x_{m,Ai}} \frac{\partial E_l}{\partial x_{m,Ai}} \\ &= -\mathbf{h}_{x_{m,Ai}} \frac{\partial E_l}{\partial z^*} \frac{\partial z^*}{\partial \mathbf{m}_{\text{Akkum.}}(k)} \frac{\partial \mathbf{m}_{\text{Akkum.}}(k)}{\partial \mathbf{m}_{\text{Prod.}}(i,j,k)} \frac{\partial \mathbf{m}_{\text{Prod.}}(i,j,k)}{\partial \mathbf{m}_{\text{Aggreg.}}(i,j)} \frac{\partial \mathbf{m}_{\text{Aggreg.}}(i,j)}{\partial \mathbf{m}_A(i)} \frac{\partial \mathbf{m}_A(i)}{\partial x_{m,Ai}} \end{aligned} \quad (\text{C.50})$$

$$\Delta x_{m,Ai} = \mathbf{h}_{x_{m,Ai}} (z^{*,\text{ref}} - z^*) \left(\frac{s_C z_{m,Ck} \text{Nenner} - \text{Zähler}}{\text{Nenner}^2} \right) \left(\frac{1 - \mathbf{m}_{\text{Akkum.}}(k)}{1 - \mathbf{m}_{\text{Prod.}}(i,j,k)} \right) \mathbf{m}_{\text{FAM}}(i,j,k) \mathbf{m}_B(j) \frac{\partial \mathbf{m}_A(i)}{\partial x_{m,Ai}} \quad (\text{C.51})$$

Bei Anwendung der Rechenvorschrift (C.51) erhält man als Ergebnis ein dreidimensionales Feld. Für die Änderung der Modalwerte $x_{m,Ai}$ wird aber lediglich ein n -dimensionaler Vektor benötigt, weshalb man für ein i eine

Summation über alle j und k vornimmt. Ausgedrückt wird dies durch das Summationszeichen in (C.52).

$$x_{m,Ai}^{neu} = x_{m,Ai}^{alt} + \sum \Delta x_{m,Ai} \quad (C.52)$$

Auch bei der Berechnung der Änderung der Breiten der Eingangsmengen A_i wird zunächst ein dreidimensionales Feld erhalten (C.54), welches durch Summation über alle j und k auf einen n -dimensionalen Vektor zurückgeführt wird (C.55).

$$\begin{aligned} \Delta \mathbf{a}_{Ai} &= -\mathbf{h}_{aAi} \frac{\partial E_l}{\partial \mathbf{a}_{Ai}} \\ &= -\mathbf{h}_{aAi} \frac{\partial E_l}{\partial z^*} \frac{\partial z^*}{\partial \mathbf{m}_{Akkum.}(k)} \frac{\partial \mathbf{m}_{Akkum.}(k)}{\partial \mathbf{m}_{Prod.}(i,j,k)} \frac{\partial \mathbf{m}_{Prod.}(i,j,k)}{\partial \mathbf{m}_{Aggreg.}(i,j)} \frac{\partial \mathbf{m}_{Aggreg.}(i,j)}{\partial \mathbf{m}_A(i)} \frac{\partial \mathbf{m}_A(i)}{\partial \mathbf{a}_{Ai}} \end{aligned} \quad (C.53)$$

$$\Delta \mathbf{a}_{Ai} = \mathbf{h}_{aAi} \left(z^{*,ref} - z^* \right) \left(\frac{s_C z_{m,Ck} \text{Nenner} - \text{Zähler}}{\text{Nenner}^2} \right) \left(\frac{1 - \mathbf{m}_{Akkum.}(k)}{1 - \mathbf{m}_{Prod.}(i,j,k)} \right) \mathbf{m}_{FAM}(i,j,k) \mathbf{m}_B(j) \frac{\partial \mathbf{m}_A(i)}{\partial \mathbf{a}_{Ai}} \quad (C.54)$$

$$\mathbf{a}_{Ai}^{neu} = \mathbf{a}_{Ai}^{alt} + \sum \Delta \mathbf{a}_{Ai} \quad (C.55)$$

Die in den Formeln (C.47) bis (C.55) aufgezeigten Rechenschritte lassen sich problemlos auf die Anpassung der Eingangsmengen B_j der linguistischen Variable B übertragen. Für die Änderungen der Modalwerte $y_{m,Bj}$ und der Breiten \mathbf{a}_{Bj} wird dann entsprechend für jedes j über alle i und k summiert.

In [26] und [45] wird angesprochen, daß es bei der Anwendung des Lernverfahrens zur Neubestimmung der Eingangs-Fuzzy-Mengen zu gegenläufigen Änderungen (Elemente in den Feldern (C.51) und (C.54)) kommen kann. Mit den vorgenommenen Summationen über alle Änderungen, die eine unscharfe Eingangsmenge betreffen, wird aber sichergestellt, daß ein linguistischer Term auch nach der Anpassung nur durch eine unscharfe Menge repräsentiert wird. Dennoch kann nicht verhindert werden, daß die Parameteradaptionen unter Umständen am Ende des Adaptionprozesses zu veränderten Zugehörigkeitsfunktionen führen, die nicht unbedingt mit ihren verbalen Bezeichnungen übereinstimmen. Dadurch kann auch die Interpretierbarkeit der Regelbasis beeinträchtigt werden.

Bei der Wahl der Lernraten h für die Skalierungsfaktoren, Mittenwerte und Breiten der Zugehörigkeitsfunktionen sowie für die Regelbasiseinträge ist

man auf ein Vorgehen nach Versuch und Irrtum angewiesen. Da im Sinne einer einfachen Interpretierbarkeit des Neuro-Fuzzy-Systems keine Skalierung der Eingabewerte x^* , y^* oder der Ausgabewerte $z^{*,ref}$ auf ein definiertes Intervall von z.B. -1 bis +1 vorgenommen wird, ist die Wahl der Lernraten auch von der Größenordnung dieser Werte abhängig.

Das in Anhang C vorgestellte Neuro-Fuzzy-System verdient seinen Namen nach den Definitionen eines solchen Systems in [3, 7, 26, 46, 47, 50]. Es soll jedoch nicht verschwiegen werden, daß in [45] daraufhingewiesen wird, daß das hier beschriebene System in keinem seiner Teile ein künstliches neuronales Netz wie in Anhang B enthält, und es damit eher als adaptives Fuzzy-Logik-System zu beschreiben wäre.

Verwendete Symbole Anhang C

a	Breite der Zugehörigkeitsfunktion
Δa	Korrektur der Breite a
A	Linguistische Variable für Basisvariable x
A_i	Teilmenge der Grundmenge X
B	Linguistische Variable für Basisvariable y
B_j	Teilmenge der Grundmenge Y
C	Linguistische Variable für Basisvariable z
C_k	Teilmenge (Singleton) der Grundmenge Z
E_l	Fehler der Ausgangsvariable z^*
E_{fkt}	Zwischengespeicherter Wert
$Exponent$	Zwischengespeicherter Wert
i, j, k	Laufindizes
k, k_1, k_2	Konstanten
m	Anzahl der Teilmengen (Terme) B_j
h_{sC}	Lernrate für Skalierungsfaktor linguistische Variable C
h_{z_m, C_k}	Lernrate für Position der Ausgangssingletons z_{m, C_k}
h_{FAM}	Lernrate für Relevanzfaktoren der unscharfen assoziativen Karten
h_{sA}	Lernrate für Skalierungsfaktor linguistische Variable A
h_{x_m, A_i}	Lernrate für Modal- bzw. Begrenzungswerte x_{m, A_i}
h_{aA_i}	Lernrate für Breite der Zugehörigkeitsfunktion von A_i
h_{sB}	Lernrate für Skalierungsfaktor linguistische Variable B
h_{y_m, B_j}	Lernrate für Modal- bzw. Begrenzungswerte y_{m, B_j}
h_{aB_j}	Lernrate für Breite der Zugehörigkeitsfunktion von B_j

$\mathbf{m}_{A_i}(x)$	Zugehörigkeitsgrad des Elements x zur Teilmenge A_i
$\mathbf{m}_{\text{Aggregation}}(i,j)$	Aggregationsmatrix, entstanden aus den Vektoren $\mu_A(i)$, $\mu_B(j)$
$\mathbf{m}_{\text{Akkumulation}}(k)$	Akkumulationsvektor, enthält den Grad der Entschlossenheit bezüglich der Schlußfolgerung k
$\mathbf{m}_{B_j}(y)$	Zugehörigkeitsgrad des Elements y zur Teilmenge B_j
$\mathbf{m}_{C_{k,mod}}(z)$	Modifizierte Ausgangsmenge C_k über der Ausgangsvariable z
$\mathbf{m}_{C, res}(z)$	Resultierende Ausgangsmenge C , entstanden aus der Zusammenfassung der k modifizierten Ausgangsmengen $C_{k,mod}$
$\mathbf{m}_{FAM}(i,j,k)$	Unschärfe assoziative Karten, dreidimensionales Feld der Relevanzfaktoren aller $i \times j \times k$ Regeln
$\mathbf{m}_{\text{Produkt}}(i,j,k)$	dreidimensionales Produktfeld, entstanden aus der elementweisen Multiplikation von $\mathbf{m}_{\text{Aggregation}}(i,j)$ und $\mathbf{m}_{FAM}(i,j,k)$
n	Anzahl der Teilmengen (Terme) A_i
<i>Nenner</i>	Zwischengespeicherter Wert
o	Anzahl der Teilmengen (Terme) C_k
s_A	Skalierungsfaktor der linguistischen Variable A
s_B	Skalierungsfaktor der linguistischen Variable B
s_C	Skalierungsfaktor der linguistischen Variable C
Δs	Korrektur des Skalierungsfaktors
x	Element der Grundmenge X
x^*	Eingangsgröße des Fuzzy-Logik-Systems (Element von X)
x_{m,A_i}	Modalwert oder Begrenzungswert der Zugehörigkeitsfunktion
$\Delta x_{m,A_i}$	Korrektur des Modal- oder Begrenzungswerts x_{m,A_i}
X	Grundmenge
y	Element der Grundmenge Y
y^*	Eingangsgröße des Fuzzy-Logik-Systems (Element von Y)
y_{m,B_j}	Modalwert oder Begrenzungswert der Zugehörigkeitsfunktion
$\Delta y_{m,B_j}$	Korrektur des Modal- oder Begrenzungswerts y_{m,B_j}
Y	Grundmenge
z	Element der Grundmenge Z
z^*	Ausgangsgröße des Fuzzy-Logik-Systems (Element von Z)
$z^{*,ref}$	Sollwert der Ausgangsgröße z^*
z_{m,C_k}	Position des Ausgangssingletons
$\Delta z_{m,C_k}$	Korrektur der Position des Ausgangssingletons z_{m,C_k}
Z	Grundmenge
<i>Zähler</i>	Zwischengespeicherter Wert

Anhang D: Genetische Algorithmen [24-32]

Algorithmus, String-Darstellung und Startpopulation

Der genetische Algorithmus ist ein Optimierungsalgorithmus, der auf evolutionären Prinzipien beruht [13]. Er wird modelliert nach den Konzepten der biologischen Evolution, und versucht so, die Art wie sich biologische Systeme an ihre Umgebung anpassen zu imitieren [10, 24]. Neben der Evolutionsstrategie, die aus der Natur übertragen wird [25], bedient sich der genetische Algorithmus auch der biologischen Fachausdrücke. So wird ein Parametersatz, der die Lösung zu einem Optimierungsproblem ist [25], als Individuum bezeichnet, das als String repräsentiert wird. Mehrere Strings werden in einer Population zusammengefaßt. Die Durchläufe des Algorithmus werden als Generationen bezeichnet.

Die meisten mathematischen Optimierungsverfahren arbeiten in der Regel nur bei Vorliegen idealisierter Annahmen wie der Linearität der Problemstellung und der Differenzierbarkeit des zu optimierenden Problems zufriedenstellend [4]. Läßt sich die Qualität einer Problemlösung als Güte in Abhängigkeit von den Parametern des Parametersatzes (Strings) darstellen, so wird der Schwierigkeitsgrad eines Optimierungsproblems wesentlich durch die Charakteristik der Güte als Funktion der Parameter bestimmt [4]. Der Vorteil eines genetischen Algorithmus ist, daß er keine niedrig-dimensionalen, differenzierbaren und kontinuierlichen Gütefunktionen benötigt [10, 29].

Im Gegensatz zu anderen Verfahren arbeitet der genetische Algorithmus nicht nur mit einem Parametersatz (einer möglichen Lösung) und versucht diesen anhand der Gütefunktion zu optimieren, sondern mit einer Population aus n möglichen Parametersätzen [17, 30]. Der genetische Algorithmus handhabt die n Parametersätze der Population simultan, so daß diese Informationen aus den verschiedenen Parameterbereichen der Gütefunktion liefern [13]. Dabei benötigt der genetische Algorithmus lediglich den Wert der Gütefunktion an den Punkten, die durch die Parametersätze beschrieben werden, darüberhinausgehende Informationen wie z. B. den Gradienten der Gütefunktion an diesen Punkten sind nicht notwendig [13]. Dies ist einer der Gründe warum der Algorithmus auch bei zerklüfteten Topologien der Gütefunktion zuverlässig arbeitet.

Der Algorithmus besteht aus folgenden Schritten: (1) der Bildung der Startpopulation, (2) der Bestimmung der Qualität der Lösungen, welche zu den Parametersätzen gehören, (3) der Reproduktionsoperation, (4) der Rekombinations- bzw. Kreuzungsoperation und (5) der Mutationsoperation. Die Schritte 2 bis 5 werden zyklisch wiederholt.

Die Operationen Reproduktion, Rekombination und Mutation haben sowohl algorithmische als auch zufällige Eigenschaften [24, 30]. Der genetische Algorithmus kann daher als ein stochastisches, nicht-deterministisches Optimierungsverfahren [13, 17, 26] bezeichnet werden. Die folgende Abbildung 63 zeigt den Ablauf des Algorithmus. Auf die einzelnen Schritte wird im folgenden eingegangen.

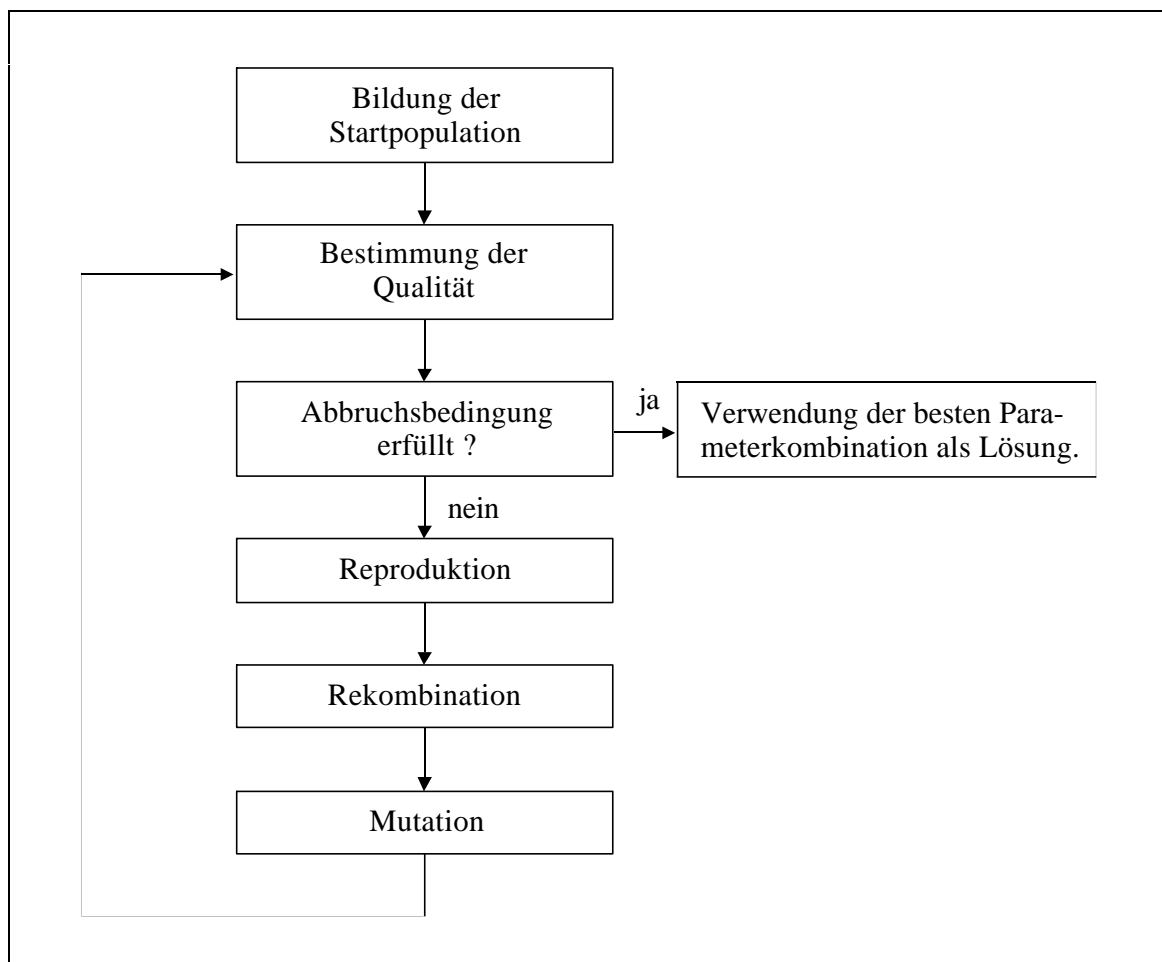


Abb. 62 Genetischer Algorithmus

Damit ein genetischer Algorithmus die n Parametersätze (Individuen) der Population bearbeiten kann, müssen diese als String (D.1) darstellbar sein [13, 24]. Ein String ist ein Zeilenvektor, in dem die Parameter des Opti-

mierungsproblems nacheinander angeordnet sind. Diese Form der Darstellung ist notwendig, da die genetischen Operatoren auf diesen Typ der Repräsentation zugeschnitten sind [17]. Jeder String ist eine mögliche Lösung des Optimierungsproblems [13].

$$x_{i,j} = (x_{i,1}, x_{i,2}, \dots, x_{i,j}, \dots, x_{i,m}) \quad (\text{D.1})$$

Zu Beginn des Verfahrens müssen n Parametersätze für die Startpopulation erzeugt werden. Liegen Vorkenntnisse über den möglichen Ort des Optimums der Gütefunktion vor, so können diese bei der Wahl der Startpopulation mit einfließen. Liegen hingegen keine Kenntnisse vor, so wird zweckmäßigerweise eine im ganzen zulässigen Parameterraum gleichverteilte Startpopulation gewählt [4].

Die Anzahl n der in der Population vertretenen Strings ist unter dem Einfluß der beiden gegenläufigen Anforderungen Zeitaufwand und weitläufige Erfassung des Parameterraums festzulegen.

Eine große Anzahl von Individuen in der Population bedeutet die simultane Betrachtung vieler Parametersätze, was den (Rechen-)Zeitaufwand pro Generation erhöht [17]. Es besteht die Gefahr, daß die Suche dann eher einem Durchprobieren aller möglichen Parameterkombinationen gleicht (*exhaustive search*) [32]. Die Konvergenz des Verfahrens auf ein Optimum hin wird verschlechtert, so daß auch der Zeitaufwand insgesamt erhöht wird [25].

Weil aber viele Parametersätze getestet werden, ist die Diversität in der Population und damit auch die Wahrscheinlichkeit der Konvergenz auf das globale Optimum höher, als wenn nur eine geringe Anzahl Individuen betrachtet wird [17]. Der genetische Algorithmus begründet demnach seine Sicherheit zum Auffinden des globalen Optimums auch in der hohen Anzahl an überprüften Individuen [30]. Ist die Anzahl hingegen viel zu klein, kann das Erreichen des globalen Optimums verhindert werden [32]. Ferner ist zu berücksichtigen, daß die Menge aller Individuen der einzige Informationsspeicher ist, über den der genetische Algorithmus verfügt [25].

Reproduktions-, Rekombinations- und Mutationsoperation

Im ersten Schritt des sich wiederholenden Zyklus wird zunächst die Qualität von jedem Individuum i der Population bestimmt. Die Qualität eines Individuums ist ein Maß dafür wie gut die Lösung des Optimierungsproblems ist, die durch den jeweiligen Parametersatz repräsentiert wird.

Der genetische Algorithmus arbeitet jedoch nicht direkt mit der Qualität eines Individuums, sondern mit seiner Güte (*fitness*). Die Güte eines Individuums wird über eine Gütefunktion (D.2) aus der Qualität abgeleitet. Die Existenz einer geeigneten Gütefunktion ist die Voraussetzung für die Einsetzbarkeit eines genetischen Algorithmus. Die Gütefunktion stellt den einzigen Kontakt zwischen dem Algorithmus und dem Optimierungsproblem dar. Der eigentliche genetische Algorithmus ist problemunabhängig [10, 13, 17, 26].

$$f_i = f(\text{Qualität}(x_i)) \quad (\text{D.2})$$

Ziel des genetischen Algorithmus ist es die mittlere Güte der n Individuen in der Population zu erhöhen, d.h. er arbeitet an einem Maximierungsproblem. Ist das zu lösende Optimierungsproblem auch ein Maximierungsproblem, so kann die Qualität eines Individuums direkt als Güte betrachtet werden. Es ist jedoch darauf zu achten, daß die Güte keine negativen Werte annimmt [30]. Gegebenenfalls ist die Qualität durch Addition einer Konstanten in eine geeignete Güte zu überführen.

Ist das zu lösende Optimierungsproblem aber ein Minimierungsproblem so muß eine andere Gütefunktion verwendet werden. Nimmt die Qualität der Individuen nur positive Werte an, so kann (D.3) als Funktion verwendet werden [10, 13]. Kann hingegen keine Aussage über das Vorzeichen der Qualität gemacht werden, so kann (D.4) als Gütefunktion angewendet werden. Sie gewährleistet das die Güte immer positiv ist [10, 25].

$$f_i = \frac{1}{\text{Qualität}(x_i)} \quad (\text{D.3})$$

$$f_i = \text{MAX}_i(\text{Qualität}(x_i)) - \text{Qualität}(x_i) \quad (\text{D.4})$$

Die Gütefunktion f besitzt erheblichen Einfluß, da die Geschwindigkeit, mit der der Algorithmus konvergiert, von der gewählten Beziehung abhängig ist [13]. Diskriminiert die Gütefunktion sehr stark zwischen Individuen geringer Qualität und solchen hoher Qualität, so kommt es unter Umständen zu einem schnellen Verlust an Diversität in der Population und zu der vorzeitigen Konvergenz auf ein suboptimales Optimum [27]. Man spricht in diesem Fall von einem hohen evolutionären Druck.

Auf der anderen Seite kann ein schwacher evolutionärer Druck (mangelnde Diskriminierung durch die Gütefunktion) die Suche uneffektiv werden lassen [27], da der Algorithmus nicht in die Lage versetzt wird, zwischen guten und schlechten Individuen zu unterscheiden.

Um eine zusätzliche Differenzierung zwischen den Güten der einzelnen Individuen vornehmen zu können, existieren die zwei Verfahren Potenzgesetz-Skalierung (*power law scaling* [27]) und lineare Skalierung (*linear scaling* [27, 30])

Bei der Skalierung nach dem Potenzgesetz (D.5) wird die skalierte Güte f_i' aus der ursprünglichen Güte f_i gebildet, indem f_i mit einem Exponenten $k > 1$ potenziert wird. Üblicherweise liegt der Wert des Exponenten k zwischen den Werten 1.005 und 2 [27].

$$f_i' = f_i^k \quad (\text{D.5})$$

Bei der linearen Skalierung (D.6) steht die skalierte Güte f_i' in einer linearen Beziehung zu der Güte f_i . Die Parameter k_1 und k_2 werden unter den Bedingungen gewählt, daß die mittlere Güte auf sich selbst abgebildet wird ($f'_{ave} = f_{ave}$), und daß das Individuum mit der geringsten Güte f_{min} eine skalierte Güte $f'_{min} = 0$ erhält [30].

$$f_i' = k_1 f_i + k_2 \quad (\text{D.6})$$

Die Güten der einzelnen Individuen ist die einzige Information die ein genetischer Algorithmus verarbeitet, darüber hinausgehende Informationen wie die Gradienten der Gütefunktion werden nicht benötigt [13, 17].

Nachdem die Qualität bzw. die Güte der einzelnen Individuen festgestellt ist, stellt sich die Frage, ob das beste Individuum schon eine Lösung repräsentiert, die für das gegebene Optimierungsproblem als ausreichend angesehen werden kann. Es ergibt sich hier die Notwendigkeit, eine Abbruchbedingung des Algorithmus zu formulieren.

Mit zunehmender Generationenzahl wird die mittlere Güte der Population, als auch die Güte des besten Individuums zunehmen. Allerdings nimmt die erzielte Güteverbesserung mit zunehmender Generationenzahl meist ab. Der Zeitaufwand, der zu einer weiteren Güteverbesserung betrieben werden muß, kann recht hoch werden. Er läßt sich bei Inkaufnahme schlechterer Lösungen begrenzen [25]. Einen allgemeinen Ratschlag, wie die Abbruch-

bedingung eines genetischen Algorithmus formuliert werden muß, kann es nicht geben, da im Ablauf des Verfahrens immer weitere Verbesserungen erwartet werden können, diese mit zunehmender Konvergenz jedoch auch seltener werden. Die Abbruchbedingung muß daher mit den Anforderungen, die an den Zeitaufwand und an die Güte der Lösung gestellt werden, abgestimmt werden [25].

Als mögliche Abbruchbedingungen werden in [27] das Erreichen einer maximalen Generationenanzahl, das Erreichen einer maximalen Anzahl an Funktionsbestimmungen und das Nichtauftreten einer weiteren signifikanten Verbesserung genannt. Während sich über eine der ersten beiden Bedingungen der Zeitaufwand in engen Grenzen halten läßt, ist bei der letzteren Abbruchbedingung die Veränderung der Lösungsgüte entscheidend, so daß der Zeitaufwand erheblich werden kann.

Es wird gehofft, daß bei Abbruch des Verfahrens das beste Individuum eine Lösung nahe der optimalen Lösung repräsentiert [27]. Da der genetische Algorithmus ein stochastisches Verfahren ist, besteht jedoch keine Garantie, daß er die optimale Lösung auch findet [10].

Wird die Lösungsqualität des besten Individuums für noch nicht ausreichend erachtet folgt als nächster Schritt die Reproduktionsoperation (Selektion). Die Reproduktionsoperation hat zwei Aufgaben zu erfüllen. Einerseits soll das Verfahren die an guten Individuen gewonnen Informationen über die Qualitätsfunktion weiterverwenden, andererseits soll eine vorzeitige Konvergenz auf ein womöglich suboptimales Extremum durch Aufrechterhalten der Diversität in der Population verhindert werden [17].

Gute Strings, Individuen hoher Güte, sollen in der Folgegeneration tendenziell öfter vertreten sein als schlechte. Allerdings sollen schlechte Lösungen nicht ganz aus der Population verschwinden, da sie zur Diversität beitragen [13, 17, 26, 31]. Eine Methode, welche diesen Anforderungen entspricht, ist die Rouletterad-Selektion. Dazu werden zunächst die relativen Gütwerte f_i^{rel} der Strings ermittelt:

$$f_i^{rel} = \frac{f_i'}{\sum_i f_i'} \quad (D.7)$$

Anschließend wird aus den relativen Gütwerten f_i^{rel} ein Rouletterad mit dem Umfang 1 erstellt [13, 25]. Strings hoher relativer Güte belegen auf

diesem Rouletterad große Kreisausschnitte, während Strings mit niedriger Güte nur einen kleineren Kreisausschnitt zugeteilt bekommen. Auf dieses drehende Rouletterad wird eine Kugel geworfen, und es wird derjenige String in die neue Population übernommen, auf dessen Kreissegment die Kugel zu liegen kommt. Auf dem Rechner realisiert wird die Rouletterad-Selektion, indem jedem Individuum i die kumulierte Selektionswahrscheinlichkeit q_i (D.8) zugeordnet wird.

$$q_i = \sum_{k=1}^i f_k^{rel} \quad (D.8)$$

Die Individuen, welche in die Population der neuen Generation übernommen werden sollen, werden nun so bestimmt, daß eine Zufallszahl $r \in [0, 1]$ erzeugt wird, und anhand dieser Zufallszahl die Auswahl getroffen wird. Ist $r < q_1$, kommt das Individuum 1 in die neue Population. Sonst wird das Individuum i ausgewählt, für das gilt: $q_{i-1} < r \leq q_i$. Dieser Vorgang wird n -mal wiederholt, bis die Anzahl der Individuen in der Kindpopulation der Größe der Elternpopulation entspricht.

Jeder String kommt mit einer Wahrscheinlichkeit in der neuen Population vor, die seiner relativen Güte entspricht [10, 13]. Gute Strings bleiben eher in der Population und werden sogar noch vervielfacht, weniger gute fallen eher aus der Population heraus. Das Rouletterad-Verfahren erlaubt es so auch schlechten Strings in der Folgegeneration vertreten zu sein [13].

Zusätzlich zur Rouletterad-Selektion gibt es das Verfahren der Elitisierung [13, 27]. Elitisierung bedeutet, der beste String der Population wird in jedem Fall in die Population der folgenden Generation übernommen. Ferner wird er dort vor dem Einfluß des Kreuzungsoperators und des Mutationsoperators geschützt. Das Verfahren der Elitisierung kann kontraproduktiv sein. Wenn der beste String in der Folgegeneration zufällig nur einmal existiert und geschützt ist, können die guten Parameter seines Parametersatzes nicht über den Kreuzungsoperator verwendet werden [13], und so unter Umständen zu noch besseren Strings führen.

Das Rouletterad-Verfahren kann ersetzt werden durch rangbasierte Methoden (*rank-based-methods* [10, 17]), durch die Methode der *stochastic remainder selection* [13, 27] oder durch ein *crowding factor model* [27]. Reproduktionsverfahren, die mit der rangbasierten Methode arbeiten, verletzen allerdings das Schema-Theorem [27]. Andererseits verhindern sie eine schädliche vorzeitige Konvergenz des Verfahrens [10].

Für die Rekombinations- oder Kreuzungsoperation werden zwei Strings a und b per Zufall ausgewählt. Jeder String wird mit der Rekombinationswahrscheinlichkeit p_{Rekomb} für diese Operation herangezogen. Für die Kreuzungsoperation stehen zwei Operatoren zur Verfügung.

Bei dem 1-Punkt-Kreuzungsoperator werden die Strings a und b nach der Position h zerschnitten (D.9) und gekreuzt, d.h. ab der Position $h+1$ werden an den Teilstring a die folgenden Parameter des Teilstrings b angehängt und umgekehrt (D.10).

$$\begin{aligned} x_{a,j} &= (x_{a,1}, x_{a,2}, \dots, x_{a,h}, x_{a,h+1}, \dots, x_{a,m}) \\ x_{b,j} &= (x_{b,1}, x_{b,2}, \dots, x_{b,h}, x_{b,h+1}, \dots, x_{b,m}) \end{aligned} \quad (D.9)$$

$$\begin{aligned} x_{a,j}^{neu} &= (x_{a,1}, x_{a,2}, \dots, x_{a,h}, x_{b,h+1}, \dots, x_{b,m}) \\ x_{b,j}^{neu} &= (x_{b,1}, x_{b,2}, \dots, x_{b,h}, x_{a,h+1}, \dots, x_{a,m}) \end{aligned} \quad (D.10)$$

Anstelle des 1-Punkt-Kreuzungsoperators kann auch der 2-Punkt-Kreuzungsoperator verwendet werden. Auch er wählt zufällig zwei Strings a und b aus, benötigt dann allerdings zwei Positionen g und h , um die beiden Strings zu kreuzen. Bei dem 2-Punkt-Kreuzungsoperator wird bei beiden Strings ein mittleres Stück zwischen den Positionen g und h ausgeschnitten (D.11) und zwischen den Strings getauscht (D.12).

$$\begin{aligned} x_{a,j} &= (x_{a,1}, x_{a,2}, \dots, x_{a,g-1}, x_{a,g}, \dots, x_{a,h}, x_{a,h+1}, \dots, x_{a,m}) \\ x_{b,j} &= (x_{b,1}, x_{b,2}, \dots, x_{b,g-1}, x_{b,g}, \dots, x_{b,h}, x_{b,h+1}, \dots, x_{b,m}) \end{aligned} \quad (D.11)$$

$$\begin{aligned} x_{a,j}^{neu} &= (x_{a,1}, x_{a,2}, \dots, x_{a,g-1}, x_{b,g}, \dots, x_{b,h}, x_{a,h+1}, \dots, x_{a,m}) \\ x_{b,j}^{neu} &= (x_{b,1}, x_{b,2}, \dots, x_{b,g-1}, x_{a,g}, \dots, x_{a,h}, x_{b,h+1}, \dots, x_{b,m}) \end{aligned} \quad (D.12)$$

Die Stelle h bzw. die Stellen g und h der Kreuzungsoperation können entweder fest vorgegeben oder zufällig bestimmt werden [26]. Damit die Populationsgröße n durch die Operation nicht verändert wird, werden die alten (Eltern-)Individuen a und b durch die neu erzeugten (Kind-)Individuen ersetzt.

Das Zerschneiden und Zusammenfügen durch den Kreuzungsoperator führt zu einem Austausch von Segmenten, so daß die Strings in der neuen Population Charakteristiken von beiden Elternstrings aufweisen [13]. Möglicherweise werden gute Parameter an unterschiedlichen Positionen in

einem einzigen String zusammengebracht [13]. Der Kreuzungsoperator sorgt in gewisser Weise für einen Informationsaustausch zwischen den Individuen [13, 27]. Es wird erhofft, daß einer der neuen Strings eine bessere Qualität aufweist als der beste Elternstring, da die Anteile von zwei Elternstrings (Parametersätzen) neu zusammengesetzt worden sind [13].

Die Wahl der Rekombinationswahrscheinlichkeit p_{Rekomb} ist wieder ein Kompromiß zwischen gegenläufigen Tendenzen. Ist sie zu gering, werden neue Individuen hauptsächlich durch den Mutationsoperator erzeugt. Die Konvergenz in Richtung des Optimums ist gering [17, 25]. Ist die Rekombinationswahrscheinlichkeit zu hoch, besteht die Gefahr, daß auch gute Strings durch den Kreuzungsoperator zerstört werden [25], sodaß sich der genetische Algorithmus in einem suboptimalen Maximum verfängt, daß durch eine zufällig nicht beeinträchtigte suboptimale Parameterkombination repräsentiert wird [17].

Der Einfluß des Kreuzungsoperators ist zu Beginn des Optimierungsvorgangs am größten. Er führt dazu, daß vorteilhafte Parameter aus guten Parametersätzen (selektiert durch die Reproduktionsoperation) zusammengebracht werden [17]. Mit zunehmender Generationenzahl, wenn bereits wenige Parametersätze mehrfach in der Population vorkommen, geht der Einfluß des Kreuzungsoperators zurück, da dann hauptsächlich gleiche Strings gekreuzt werden [25] und auf diese Weise keine neuen Individuen erzeugt werden können.

Im Gegensatz zur Kreuzungsoperation arbeitet die Mutationsoperation mit nur einem Individuum [27]. Mutation ist eine zufällige Veränderung an einer zufällig ausgewählten Position h in einem String a (D.13) [13]. Jede Position $j=1..m$ in jedem String $i=1..n$ wird mit einer Wahrscheinlichkeit von p_{Mut} für die Mutationsoperation ausgewählt. In [13] wird vorgeschlagen, den Parameter $x_{a,h}$ durch einen zufälligen Wert zu ersetzen, der aus demselben Wertebereich stammt wie die Werte der Anfangspopulation. Dagegen wird in [4] im Zusammenhang mit einem geänderten Reproduktionsverfahren ein abweichender Mutationsoperator beschrieben. Er addiert zu jedem Stringelement $x_{a,h}$ eine mittelwertfreie, normalverteilte Zufallszahl mit der Streuung s_h , wobei die Streuung mit fortschreitender Generationenzahl verkleinert wird.

$$x_{a,j}^{neu} = (x_{a,1}, x_{a,2}, \dots, x_{a,h}^*, \dots, x_{a,m}) \quad (\text{D.13})$$

Die Mutationsoperation führt dazu, daß das Individuum a einen neuen Punkt im Parameterraum belegt [13, 17]. Dadurch wird die Diversität der Population, welche durch das Selektionsverfahren vermindert wurde, erhöht und u.U. wird ein Stehenbleiben des genetischen Algorithmus in einem lokalen Extremum verhindert [13, 27].

Für die Wahl der Mutationswahrscheinlichkeit p_{Mut} sind einige Grundüberlegungen zu berücksichtigen. Ist p_{Mut} zu hoch, so wird die Konvergenz des Verfahrens behindert. Auch gute String werden stark beeinträchtigt oder zerstört. Der Parameterraum wird zwar erschöpfend abgesucht, allerdings wird viel Zeit auf die Bewertung wahrscheinlich schlechter Individuen verwendet (der Mutationsoperator wirkt ungerichtet) [17, 25]. Ist die Mutationswahrscheinlichkeit p_{Mut} zu niedrig, besteht die Gefahr der vorzeitigen Konvergenz auf ein suboptimales Maximum [17], da weder die Reproduktionsoperation, noch die Kreuzungsoperation neue Werte innerhalb Parametersätze erzeugen können. Der Parameterraum wird nur ungenügend abgetastet. Mit zunehmender Generationenzahl besteht bei zu geringer Mutationswahrscheinlichkeit die Möglichkeit, daß der genetische Algorithmus in ein einfaches Hill-Climbing-Verfahren übergeht [25].

Der Einfluß des Mutationsoperators bleibt über die fortschreitende Generationenzahl konstant. Die Wichtigkeit des Operators nimmt jedoch zu, da der Kreuzungsoperator als Operator zur Schaffung neuer Individuen (Parametersätze) bei hoher Generationenzahl an Bedeutung verliert. Ein genetischer Algorithmus, der lediglich auf Reproduktion und Mutation basiert, wäre bereits ein funktionsfähiges Optimierungsverfahren [27].

Das Rouletterad-Verfahren überführt bevorzugt Individuen hoher Güte in die nachfolgende Generation. Kann der Einfluß der Operatoren Kreuzung und Mutation als nicht zu groß angesehen werden, sollte die Population der nachfolgenden Generation eine höhere durchschnittliche Güte aufweisen [13]. Der genetische Algorithmus verbessert also eher die durchschnittliche Güte der Population als die Güte eines einzelnen Individuums [26]. Damit erfüllt der Algorithmus die beiden widersprüchlichen Anforderungen, Konvergenz des Verfahrens auf eine optimale Lösung und ein weites Absuchen des Parameterraumes zu gewährleisten [27].

Genetische Algorithmen besitzen eine garantierte Konvergenz. Dabei muß nicht notwendigerweise die optimale Lösung gefunden werden, da es sich um ein stochastisches Verfahren handelt [26]. Da der Algorithmus jedoch viele

mögliche Lösungen gleichzeitig überprüft, ist das Finden einer Lösung in der Nähe des Optimums äußerst wahrscheinlich [17]. Ein Hill-Climbing-Verfahren dagegen orientiert sich nur an einer Parameterkombination, die es zu optimieren gilt, und vermeidet so ein Absuchen des Parameterraumes [27]. Die Parameter, die Einfluß auf die Leistungsfähigkeit eines genetischen Algorithmus nehmen, sind zusammengefaßt: Die Populationsgröße n , die Gütefunktion f , die Rekombinationswahrscheinlichkeit p_{Rekomb} und die Mutationswahrscheinlichkeit p_{Mut} [17]. Für die Abschätzung dieser Parameter fehlen allerdings noch gute Heuristiken [27], so daß man bei ihrer Wahl meist auf Versuch und Irrtum angewiesen ist.

Binäre Darstellung und Schema Theorem

Bis zu diesem Punkt wurde lediglich die Darstellung behandelt, in der ein String ein Vektor aus Fließkommazahlen $x_{i,j}$ ist, wobei jede Fließkommazahl ein einzelner Parameter des zu optimierenden Problems ist. Klassische genetische Algorithmen hingegen arbeiten mit binären Strings, wobei alle Operatoren auf diese binäre Darstellung abgestimmt werden [27]. Meist wird diese Darstellung als binärer String bevorzugt [10, 17].

Um einen binären String zu konstruieren, wird jeder Parameter x_k^{real} in eine binäre Zahl x_k^{bin} übersetzt. Es muß angegeben werden, zwischen welchem maximalen Wert $x_{k,max}$ und welchem minimalen Wert $x_{k,min}$ der Wert von x_k^{real} definiert ist. Ferner muß die geforderte Genauigkeit als Anzahl der signifikanten Stellen a_k vorgegeben werden. Anhand dieser Vorgaben läßt sich ermitteln, über wieviele Stellen m_k die binäre Zahl verfügen muß (D.14), damit der Parameter x_k^{real} mit der geforderten Genauigkeit dargestellt werden kann.

$$2^{m_k} - 1 \geq (x_{k,max}^{real} - x_{k,min}^{real}) \cdot 10^{a_k} \quad (D.14)$$

Bei der Erzeugung der n Strings $x_{i,j}^{bin}$ der Anfangspopulation wird folgendermaßen vorgegangen. Für jeden Fließkommamaparameter x_k^{real} (mit $k=1..o$) wird eine binäre Zahl x_k^{bin} mit m_k Stellen erzeugt, anschließend werden die binären Zahlen x_k^{bin} aneinandergelagert, um den gesamten String $x_{i,j}^{bin}$ der Länge $m = \sum m_k$ zu bilden ($i = 1..n, j = 1..m$). Für die Übersetzung der binären Darstellung in die Fließkommadarstellung der einzelnen Parameter x_k^{real} wird der binäre String $x_{i,j}^{bin}$ in die o binären Zahlen x_k^{bin} zerschnitten (deren Länge bekannt ist) und es wird für jeden Parameter x_k^{real} die Beziehung (D.15) angewendet.

$$x_k^{real} = x_k^{real} + dec(x_k^{bin}) \cdot \frac{x_{k,max}^{real} - x_{k,min}^{real}}{2^{m_k} - 1} \quad (D.15)$$

Die Anzahl der Bits m_k , durch die ein Parameter x_k^{real} kodiert wird, gibt die Genauigkeit vor, mit der der optimale Parameterwert von dem genetischen Algorithmus ermittelt werden kann. Es muß demnach vorher abgeschätzt werden, welche Genauigkeit in der Lösung gefordert wird.

Bei der Darstellung als binärer String ist darüber hinaus gewährleistet, daß der Wert des Parameters x_k^{real} nicht aus dem vorgegebenen Intervall von $x_{k,min}$ bis $x_{k,max}$ herauswandert. Dies kann bei der Fließkommadarstellung und der Anwendung des Mutationsoperators nach [4] der Fall sein. Verfügt man nur über geringe Kenntnisse, wie der Suchraum geeignet einzugrenzen ist, so kann dieser Effekt durchaus erwünscht sein.

Die Rekombinations- bzw. Kreuzungsoperation wird mit denselben Operatoren ausgeführt, welche bereits für die Darstellung der Strings in Fließkommaform beschrieben wurden. Der Kreuzungsoperator besitzt bei binärer Stringdarstellung die Fähigkeit zur Erzeugung ganz neuer Parameter x_k^{real} innerhalb des Parametersatzes, da er die binären Zahlen x_k^{bin} auch in der Mitte trennen und neu kombinieren kann. Diesbezüglich wird in [31] festgestellt, daß der Kreuzungsoperator zwar auf Strings in Fließkommadarstellung angewendet werden kann; da der Operator bei dieser Darstellungsart jedoch nur an den Parametergrenzen wirkt, ist der Einfluß des Kreuzungsoperators beschränkt, und die Suche für jeden einzelnen Parameter hängt dann hauptsächlich vom Mutationsoperator ab. Die Operatoren des genetischen Algorithmus wurden ursprünglich für binäre Strings entwickelt und verlieren daher an Wirkung, wenn sie direkt auf Strings übertragen werden, die in Fließkommadarstellung kodiert sind [17].

Die Mutationsoperation vereinfacht sich bei Strings in binärer Darstellung zu einer Bit-Flip-Operation [25]. Dabei wird an jeder Position $x_{i,j}^{bin}$ mit einer Mutationswahrscheinlichkeit von p_{Mut} eine Änderung des Wertes von 0 auf 1 oder umgekehrt vorgenommen.

Weitere Operatoren, die bei der Verwendung von binären Strings und für besondere Problemstellungen zur Verfügung stehen, sind der Inversionsoperator und der Operator Duplizieren und Löschen [17].

Der Inversionsoperator wählt zufällig einen binären String $x_{a,j}$ aus und kehrt die Reihenfolge der Bits zwischen den beiden zufällig bestimmten

Positionen h und $h + g$ um (D.16). Anschließend wird der Ausgangsstring in Population durch den neuen String ersetzt.

$$\begin{aligned} x_{a,j} &= (x_{a,1}, x_{a,2}, \dots, x_{a,h-1}, x_{a,h}, x_{a,h+1}, \dots, x_{a,h+g}, x_{a,h+g+1}, \dots, x_{a,m}) \\ x_{a,j}^{neu} &= (x_{a,1}, x_{a,2}, \dots, x_{a,h-1}, x_{a,h+g}, x_{a,h+g-1}, \dots, x_{a,h}, x_{a,h+g+1}, \dots, x_{a,m}) \end{aligned} \quad (D.16)$$

Was der Mutationsoperator für das einzelne Bit ist, ist der Inversionsoperator für Schemata. Er trägt dazu bei, daß der genetische Algorithmus nicht auf eine suboptimale Lösung hereinhält, nur weil zufällig einige Schemata, die für die suboptimale Lösung bestimmend sind, überhand genommen haben [27]. Wie der Mutationsoperator erhöht der Inversionsoperator die Diversität innerhalb der Population.

Neben der oben gezeigten binären Stringdarstellung, die der üblichen Binärzahldarstellung entspricht, existiert die binäre Stringdarstellung nach Gray (*Gray-scale binary encoding* [10]). Dieser Art der Darstellung liegt die Überlegung zugrunde, daß sich bei der Repräsentation nach Gray der Wert des kodierten Parameters x_k^{real} beim Wertewechsel eines Bits ausgelöst durch den Mutationsoperator im Mittel geringfügiger ändert als bei der klassischen Binärrepräsentation.

Ein Grund für die Darstellung eines Parametersatzes als binärer String ist, daß das binäre Alphabet eine maximale Anzahl an Schemata pro Ziffer bietet im Vergleich zu den übrigen Darstellungstechniken [17, 27]. Ein Schema ist formal eine Ähnlichkeitsschablone, die aus fixen Zeichenfolgen und Platzhaltern besteht [26]. In (D.17) ist beispielhaft ein Schema für einen String der Länge $m = 8$ dargestellt.

$$x_{s,j} = (*, *, *, x_{s,4}^{fix}, *, *, x_{s,6}^{fix}, *) \quad \text{mit } j = 1..8 \quad (D.17)$$

Ein Schema wird beschrieben durch seine definierende Länge und seine Ordnung. Die definierende Länge eines Schemas ist die Distanz zwischen der ersten und der letzten vorgegebenen Position x^{fix} eines Schemas. Ein Schema mit einer einzelnen fest vorgegebenen Position hat eine definierende Länge von null [27]. Die Ordnung eines Schemas ist die Anzahl der fest vorgegebenen Positionen x^{fix} [13, 17]. Das Schema (D.17) besitzt demnach eine definierende Länge von 3 und eine Ordnung von 2.

Die Reproduktionsoperation überführt bevorzugt Individuen mit überdurchschnittlicher Güte in die Population der nachfolgenden Generation. Dies bewirkt nicht nur, daß Strings hoher Güte in der neuen Population mehrfach vorkommen, sondern auch, daß die in ihnen vorkommenden guten Parametern vermehrt auftreten [13]. Mit den Strings werden jedoch nicht nur die einzelnen Parameter überführt, sondern auch Parameterschemata. Gute Schemata besitzen eine höhere Wahrscheinlichkeit in der neuen Population vertreten zu sein, u.U. sogar mehrfach, als weniger gute Schemata. Der genetische Algorithmus handhabt bei der Reproduktionsoperation jedoch nicht nur wenige spezielle Schemata, sondern alle möglichen Schemata parallel. Diese Eigenschaft wird als impliziter Parallelismus (*implicit parallelism*) bezeichnet [13, 27]. Das Prinzip des impliziten Parallelismus gilt auch bei einer Fließkommadarstellung der Parameter und hängt nicht davon ab, ob eine binäre Repräsentation verwendet wird [27].

Die nach der Reproduktion auf die neue Population angewendeten Operatoren Rekombination und Mutation können ein Schema beeinträchtigen. Der kombinierte zerstörerische Effekt der genetischen Operatoren auf ein Schema ist weniger signifikant, wenn das Schema eine kurze definierende Länge und eine kleine Ordnung besitzt [27]. So wird ein Schema kurzer definierender Länge vom Rekombinationsoperator nur mit geringer Wahrscheinlichkeit gespalten [13, 27, 30]. Daß ein Schema kleiner Ordnung vom Mutationsoperator angegriffen wird, ist auch eher unwahrscheinlich [13]. Darüber hinaus wird für die Mutationswahrscheinlichkeit p_{Mut} meist ein recht kleiner Wert gewählt, was den Einfluß der Mutationsoperation ebenfalls einschränkt [30]. Diese Überlegungen werden im Schema-Theorem zusammengefaßt. Es besagt, daß kurze Schemata, mit niedriger Ordnung und überdurchschnittlicher Güte eine exponentiell zunehmende Anzahl Nachkommen in den folgenden Generationen besitzen [27, 30].

Kurze Schemata, mit niedriger Ordnung und überdurchschnittlicher Güte werden als aufbauende Blöcke (*building blocks*) bezeichnet [30]. Zusammen mit dem Schema-Theorem geht man davon aus, daß eine optimale Lösung des zu optimierenden Problems dadurch gefunden wird, daß sich der String, welcher diese optimale Lösung repräsentiert, mit fortschreitender Generationenzahl aus solchen sich überlappenden aufbauenden Blöcken zusammensetzt.

Verwendete Symbole Anhang D

a	Für die Mutationsoperation ausgewählter String
a_k	Genauigkeit des Fließkommawertes x_k^{real}
a, b	Für die Rekombinationsoperation ausgewählte Strings
f_i	Güte des Individuums (Strings) i , Gütefunktion
$f\checkmark$	Skalierte Güte des Individuums (Strings) i
$f_{i,rel}$	Relative Güte des Individuums (Strings) i in bezug auf die Gesamtgüte der Population
g, h	Positionen in den Strings a und b , an denen der Zwei-Punkt-Kreuzungsoperator schneidet
h	Position im String a , an der der Mutationsoperator wirkt; Position in den Strings a und b , an denen der Ein-Punkt-Kreuzungsoperator schneidet
i	Laufindex über alle n Individuen (Strings)
j	Laufindex über alle m Elemente eines Strings
k	Konstante bei der Potenzgesetzskalierung
k_1, k_2	Konstanten bei der linearen Skalierung
n	Anzahl Individuen in der Population
m	Anzahl der Elemente eines Strings
m_k	Anzahl binärer Stellen zum Erreichen der Genauigkeit von a_k
p_{Mut}	Mutationswahrscheinlichkeit
p_{Rekomb}	Rekombinationswahrscheinlichkeit
q_i	Kumulierte Selektionswahrscheinlichkeit des Individuums i für die Rouletterad-Selektion
r	Zufallszahl aus dem Intervall von 0 bis 1 einschließlich
x_k^{real}	String k in Fließkommadarstellung
x_k^{bin}	String k in binärer Darstellung
$x_{k,max}$	Maximaler Fließkommawert, den ein Element in x_k^{real} annimmt
$x_{k,min}$	Minimaler Fließkommawert, den ein Element in x_k^{real} annimmt
$x_{i,j}$	Individuum (bzw. String) i mit Indizierung des Elements j