

# **Studien zur verbesserten Ausnutzung des Informationsgehaltes von Multisensorsystemen**

Inauguraldissertation

zur

Erlangung des Doktorgrades  
der Naturwissenschaften  
der Justus-Liebig-Universität Gießen  
Fachbereich Physik

vorgelegt von  
Thorsten Rühl  
aus Homberg/Ohm

Institut für Angewandte Physik  
Justus-Liebig-Universität Gießen  
Oktober 2001

Dekan: Prof. Dr. A. Beutelspacher

1. Berichterstatter: Prof. Dr. D. Kohl

2. Berichterstatter: Prof. Dr. A. Bunde

Tag der mündlichen Prüfung: 14.12.2001

# Inhaltsverzeichnis

Zusammenfassung .....	1
1. Einleitung .....	2
2. Grundlagen .....	3
2.1 Multisensorsysteme: Einführung in die Problematik .....	4
2.2. Sensoren zum Nachweis von Gasen in Luft .....	6
2.2.1 Metalloxid-Halbleitersensoren .....	6
2.3 Signalverarbeitung .....	14
2.3.1 Mustererkennung .....	14
2.3.2 Merkmalgenerierung, Merkmalselektion und -extraktion .....	14
2.3.3 Beurteilung der Merkmalsgüte .....	17
2.3.4 Klassifikations- und Projektionsalgorithmen .....	21
2.3.5 Visualisierung hochdimensionaler Daten .....	32
2.3.6 Fuzzy Logik .....	38
2.3.7. Nichtlineare Optimierungsverfahren .....	39
3. Experimentelles .....	43
3.1. Grundlagen zur Früherkennung von Bränden an Bord von Luftfahrzeugen .....	44
3.1.1 Vorbereitungen .....	44
3.1.2 Meßreihen .....	47
3.2. Experimentelle Grundlagen zur Detektion von Lösungsmitteln unterhalb UEG .....	48
3.2.1 Erzeugung definierter Lösungsmittelkonzentrationen .....	48
3.2.2 Gassensoren .....	50
3.2.3 Versuchsreihen .....	52
4. Ergebnisse .....	53
4.1 Bestimmung der optimalen Heizspannung .....	54
4.2 Sensorverhalten bei Angeboten von CO und H <sub>2</sub> .....	55
4.3 Verhalten der Sensoren bei Brand- und Störgrößenmessungen .....	56
4.4 Sensorsignale bei Klimakammermessungen .....	58
4.5 Lösungsmittelnachweis mit Halbleitersensoren .....	58
4.5.1 Ermittlung der optimalen Betriebstemperaturen .....	58
4.5.2 Feuchtequerempfindlichkeit .....	59
4.5.3 Response des Sensor-Arrays für verschiedene Gasangebote .....	60
4.5.4 Teilspezifitäten der Einzelsensoren .....	61
4.5.5 CO <sub>2</sub> -Querempfindlichkeit .....	62
4.6 Lösungsmittelnachweis mit massensensitiven Sensoren .....	63
4.6.1 Reaktion auf Gasangebot .....	63
4.6.2 Feuchtequerempfindlichkeit .....	64
4.6.3 CO <sub>2</sub> -Querempfindlichkeit .....	65
4.6.4 Zusammenfassung der Ergebnisse .....	65
5. Diskussion .....	66
5.1 Vorgehen zur Auswahl geeigneter Gassensoren für die Branddetektion .....	67
5.2 Vorteile eines Kombimelders gegenüber einem optischen Detektor .....	71
5.2.1 Einfluß rauchhemmender Materialien auf Detektionssicherheit und - geschwindigkeit .....	71

5.2.2	Unterdrückung von staub- und aerosolinduzierten Fehlalarmen .....	72
5.2.3	Zusammenfassung: Vorteile eines Brandmeldesystems mit Gassensoren .....	73
5.3.	Maßnahmen zur verbesserten Nutzung des Informationsgehaltes von Sensorsystemen	74
5.3.1	Nanostrukturierung von SnO <sub>2</sub> -Sensoren zur Steigerung der Nachweisempfindlichkeit .....	74
5.3.2	Temperaturmodulierte Sensorbetriebsweise zur verbesserten Ausnutzung der sensorischen Information .....	77
5.3.3	Streßtests zur Verifikation der Robustheit von Merkmalen .....	81
5.3.4	Mathematische Verfahren zur verbesserten Nutzung des Informationsgehaltes von Sensorsystemen .....	87
5.4	Erkennung kritischer Zustände in Dynamischen Systemen .....	122
5.4.1	Log-periodische Oszillationen als Frühindikatoren für Phasenübergänge .....	122
5.4.2	Beispiel: Nasdaq-Korrektur 2000 .....	124
6.	Literatur .....	126
7.	Anhang .....	I
7.1	VISOM .....	I

## Zusammenfassung

Im Rahmen der vorliegenden Arbeit wurden Methoden zur Erkennung kritischer Zustände mit Hilfe gassensorischer Systeme in z.T. sicherheitsrelevanten Umgebungen untersucht. Im Vordergrund steht die Integration der sensorischen Hardware mit Methoden der modernen Datenanalyse zu leistungsfähigen Diagnosesystemen.

Eines der wesentlichen Ziele der Untersuchungen bestand darin, Möglichkeiten zur Steigerung des Informationsgehaltes von Sensorsystemen zu entwickeln und diese in realen Systemen umzusetzen.

Hierzu wurden entlang der kompletten Informationsverarbeitungskette, beginnend mit der Erzeugung elektrischer Signalmuster auf Seite der sensorischen Hardware bis hin zu Ausgabe komplexer Systemzustandsbeschreibungen durch das User-Interface, bislang ungenutzte Potentiale auf die Möglichkeit zur Erschließung hin untersucht bzw. einer vereinheitlichten Behandlung zugänglich gemacht.

Ein besonderes Augenmerk galt den Eigenschaften halbleitender Metalloxid-Gassensoren. Zum einen wurden Möglichkeiten zur Steigerung der Nachweisempfindlichkeit durch mikro- bzw. nanostrukturierte Gassensoren untersucht. Ein weiterer Untersuchungsschwerpunkt galt der Betriebsweise dieser Sensoren. Durch eine zeitlich gesteuerte Veränderung der Temperatur der gassensitiven Schicht wurde eine physikalische Effektmodulation erzeugt, die einem Informationsmultiplexing eines Einzelsensors gleichkommt („virtueller Multisensor“). Da die Umkehrabbildung der meßbaren Effekte hin zur physikalischen Ursache in der Regel nicht eindeutig analytisch ableitbar ist, wurden Algorithmen zur Extraktion entscheidungsrelevanter Informationen aus z.T. stark verrauschten sensorischen Daten entwickelt.

Bei der Beurteilung des Informationsgehaltes durch Merkmalsextraktoren werden solche Merkmale für relevant erachtet und zur weiteren Verarbeitung selektiert, deren Zusammenhang mit den nachzuweisenden physikalischen Ursachen aufgrund vorher festgelegter Gütebeurteilungsfunktionen am wahrscheinlichsten ist.

Die zeitliche Stabilität dieser Merkmale sowie die Robustheit gegenüber Ausreißern sind für den dauerhaften Einsatz der so entstandenen Systeme von entscheidender Bedeutung und wurden mit voneinander unabhängigen Methoden untersucht. Dazu gehören neben Langzeituntersuchungen mit „echten“ Sensoren die gezielte Variation der Versuchsbedingungen (Streßtests) sowie die Simulation von Exemplarstreuungen und Alterungseffekten mit Hilfe von Computermodellen.

Schließlich kommt den Algorithmen zur Mustererkennung bzw. Funktionsapproximation eine tragende Rolle zu. Letztere läßt sich immer als Optimierungsproblem formulieren, so daß neuere, EA-ähnliche Algorithmen mit in die Diskussion einbezogen wurden.

Schließlich wurden im Rahmen dieser Arbeit zwei Softwarepakete entwickelt, um immer wiederkehrende Prozeßschritte zu beschleunigen. Hierbei handelt es sich zum einen um MSMEV (merkmalselektierendes Mustererkennungsverfahren), das im Rahmen eines mehrstufigen Merkmalselektionsprozesses einen GRNN-basierten Klassifikator selbständig aufbaut, zum anderen um VISOM, eine Software zur Visualisierung hochdimensionaler Merkmalsvektoren und Klassifikatoren sowie zur Datenclustering, Visualisierung und Klassifikation mit Hilfe von Self-Organizing Maps.

## 1. Einleitung

Im Verlauf der letzten Jahre wurden am Institut für Angewandte Physik der Justus-Liebig-Universität Giessen zahlreiche Gassensorsysteme zur Erkennung von Risikozuständen für verschiedene Anwendungsgebiete entwickelt. Das Spektrum der Anwendungen reicht von der Detektion kritischer Wasserstoffkonzentrationen in Kernkraftwerken bis hin zur Branderkennung an Bord von Passagierflugzeugen.

Am Beginn der Entwicklung eines solchen Systems steht die Auswahl geeigneter Sensoren. Diese dürfen z.B. in explosionsgefährdeten Umgebungen keine Zündquellen darstellen. Eine Analyse der bei einem zu detektierenden Ereignis freiwerdenden Gase sowie die im Normalbetrieb des Systems auftretenden Hintergrundgase und Umgebungsbedingungen ergeben die wichtigsten Randbedingungen bei der Sensorauswahl.

Vor der verbesserten Nutzung der sensorischen Information stehen Maßnahmen zur Steigerung der vom Sensor gelieferten Datenqualität. Dies betrifft zum einen die prinzipielle untere Nachweisgrenze eines Sensors, die z.B. mit Hilfe von Nanostrukturierungsmaßnahmen deutlich herabgesetzt werden kann. Zum anderen kann der Informationsgehalt des Sensors mit Hilfe dynamischer Betriebsweisen deutlich erhöht werden. Hierzu bieten sich Temperaturwechselverfahren zur physikalischen Effektmodulation an. Die Reaktion von Gasen an heißen Halbleiter-Sensoroberflächen ist von einer Vielzahl charakteristischer Effekte geprägt (Cracken von Molekülen, Adsorption und Desorption von Fragmenten), die durch einen periodisch wiederkehrenden Heizzyklus quasi „zeitaufgelöst“ werden. Die Muster im Signalverlauf können sodann mit Hilfe verschiedener mathematischer Verfahren zum Rückschluß auf die beteiligten Gase genutzt werden.

Merkmalsextraktionsverfahren werden zum Aufspüren der relevanten Merkmale im Signalverlauf herangezogen, ein Identifizieren der relevanten Einflußfaktoren „von Hand“ ist in der Mehrzahl der in der Praxis auftretenden Fälle nicht möglich.

Mustererkennungsverfahren sollen auf Basis der herausgefilterten Merkmale eine Kategorisierung der sensorischen Daten ermöglichen. Je nach Anwendungsfall genügen Aussagen über eine gewisse Klassenzugehörigkeit oder aber eine quantitative Beschreibung. Für letzteren Fall werden allgemeine Funktionsapproximatoren wie Neuronale Netze oder verwandte Algorithmen eingesetzt.

Eine zentrale Stütze für den Entwickler komplexer Informationsverarbeitungssysteme stellen Visualisierungsverfahren dar. Das menschliche Denken ist zum größten Teil durch den visuellen Wahrnehmungsapparat geprägt. Nur ein System, daß im wahrsten Sinne des Wortes „einsehbar“ ist, hat die Chance, vom Anwender zur Übernahme komplexer Überwachungsfunktionen eingesetzt zu werden.

Der Erfolg des Gesamtprozesses hängt letzten Endes entscheidend von der Qualität der Teilprozesse und dem Gelingen der Integration derselben zu einem komplexen Informationsverarbeitungssystem ab. Die Informationskette erstreckt sich vom Sensor als Rohdatenlieferanten (elektronisches Signalmuster) bis hin zur Ausgabe einer hochaggregierten Information (Kabelbrand im Cockpit, austretende Explosiva im Chemikalienlager).

In diesem Sinne handelt es sich bei der vorliegenden Arbeit um einen ersten Ansatz, hierfür ein einheitliches Rahmenwerk zu schaffen.

## 2. Grundlagen

### Übersicht

2.1 Multisensorsysteme: Einführung in die Problematik .....	4
2.2 Sensoren zum Nachweis von Gasen in Luft .....	6
2.2.1 Metalloxid-Halbleitersensoren .....	6
2.2.1.1 Das Kristallvolumen .....	7
2.2.1.2 Die Kristalloberfläche .....	8
2.1.1.2.1 Adsorptions- und Desorptionsprozesse auf der Kristalloberfläche .....	9
2.2.1.2.1.1 Physisorption .....	9
2.2.1.2.1.2 Ionosorption .....	9
2.2.1.2.1.3 Chemisorption .....	10
2.2.1.2.1.4 Desorption .....	11
2.2.1.2.1.5 Adsorptions-Isothermen .....	11
2.2.1.2.1.5.1 Die Henry-Isotherme .....	11
2.2.1.2.1.5.2 Die Langmuir-Isotherme .....	11
2.2.1.2.1.5.3 Die Freundlich-Isotherme .....	12
2.2.1.3 Polykristalline Schichten .....	12
2.2.1.3.1 Thermoionische Emission über Schottky-Barrieren .....	12
2.2.1.3.2 Ohmsche Leitungsmechanismen .....	13
2.3 Signalverarbeitung .....	14
2.3.1 Mustererkennung .....	14
2.3.2 Merkmalgenerierung, Merkmalselektion und -extraktion .....	15
2.3.3 Beurteilung der Merkmalsgüte .....	17
2.3.3.1 Überlappungsgrad .....	17
2.3.3.2 Klassifikationsrate bei vorgegebenem Klassifikator .....	21
2.3.4 Klassifikations- und Projektionsalgorithmen .....	21
2.3.4.1 Hauptkomponentenanalyse .....	21
2.3.4.2 Sammon Nonlinear Mapping (NLM) .....	23
2.3.4.3 Radiale-Basisfunktionen-Netze (RBF-Netze) .....	24
2.3.4.4 Lernende Vektorquantisierung (LVQ) .....	25
2.3.4.4.1 LVQ1 .....	25
2.3.4.4.2 OLVQ1 .....	26
2.3.4.4.3 LVQ2.1 .....	26
2.3.4.4.4 LVQ3 .....	27
2.3.4.4.5 Allgemeine Hinweise zur Anwendung von LVQ .....	27
2.3.4.5 Self-Organizing Maps .....	28
2.3.4.6 MLP zur Datenkompression/Dimensionsreduktion .....	32
2.3.5 Visualisierung hochdimensionaler Daten .....	32
2.3.5.1 Explorative Datenanalyse .....	32
2.3.5.2 Techniken zur Visualisierung selbstorganisierender Karten .....	33
2.3.5.2.1 Projektion des Neuronengitters auf den Eingaberaum .....	33
2.3.5.2.2 Vektorlagekarten/Schnitte durch die Ebenen des Netzes .....	33
2.3.5.2.3 Visualisierung der Klassenverteilung .....	33
2.3.5.2.4 Unified Distance Matrizen .....	33
2.3.5.3 Visualisierung zur Akquisition von Regelwissen .....	35
2.3.6 Fuzzy Logik .....	38
2.3.6.1 Unscharfe Mengen .....	38
2.3.6.2 Linguistische Variablen .....	38
2.3.6.3 Fuzzifikation der Eingangssignale .....	39
2.3.6.4 Approximatives Schließen .....	39
2.3.6.5 Defuzzifikation der Ausgangsgrößen .....	39
2.3.7 Nichtlineare Optimierungsverfahren .....	39
2.3.7.1 Simulated Annealing .....	40
2.3.7.2 Threshold Accepting .....	41
2.3.7.3 Sintflut-Algorithmus .....	42
2.3.7.4 Record-to-Record-Travel .....	42

## 2.1 Multisensorsysteme: Einführung in die Problematik

Den einfachsten Fall eines Multigassensensorsystems stellt ein Zweisensorsystem dar, das in einem Gemisch aus zwei verschiedenen Gasen operiert und bei dem eine lineare Abhängigkeit der Sensorsignale ( $S_1, S_2$ ) bezüglich der Konzentration der beiden Gase ( $C_1, C_2$ ) vorliegt:

$$S_1 = a_{11} * C_1 + a_{12} * C_2$$

$$S_2 = a_{21} * C_1 + a_{22} * C_2$$

Sind die Querempfindlichkeitsvektoren ( $a_{i1}, a_{i2}$ ) der beiden Gassensoren linear unabhängig, so kann man mit Hilfe dieses Systems beide Gase in exakter Konzentration nachweisen. Ein solches System kann näherungsweise mit  $H_2$  und CO als nachzuweisenden Gasen und zwei elektrochemischen CO-Sensoren unterschiedlicher  $H_2$ -Querempfindlichkeit realisiert werden (wobei in der Praxis allerdings noch eine Temperaturabhängigkeit der  $a_{ij}$  zu berücksichtigen ist). Halbleitersensoren besitzen nichtlineare Kennlinien, so daß sich die Konzentrationbestimmung für einzelne Gase nicht mehr durch das Lösen eines linearen Gleichungssystems bewerkstelligen läßt. Der obigem Beispiel zugrundeliegende Gedanke kann allerdings auf diesen Fall übertragen werden. Anstelle der linear unabhängigen Querempfindlichkeitsvektoren ( $a_{i1}, \dots, a_{in}$ ), die mit einem Konzentrationsvektor ( $C_1, \dots, C_n$ ) multipliziert werden, treten nun linear unabhängige Antwortfunktionen  $f_i(a_{i1}(C_1, \dots, C_n), \dots, a_{in}(C_1, \dots, C_n), C_1, \dots, C_n)$ , deren analytische Beschreibung allerdings nur für sehr einfache Systeme (geringe Anzahl anwesender Gase) möglich sein wird.

In den meisten Applikationen geht das zu bewältigende Komplexitätsniveau allerdings deutlich über den soeben skizzierten Fall hinaus.

Die folgende Übersicht gibt die wichtigsten Faktoren mit ihren möglichen Ausprägungen wieder, die für die Komplexität eines Sensorsystems bestimmend sind:

- Art der zu treffenden Aussage: stetig/diskret/zusammengesetzt
- Aggregationsgrad der Aussage: hoch/niedrig
- Zahl der Eingänge/Basisinformation: 1 / mehr als 1
- Einfluß von Störfaktoren (z.B. Störgase, Temperatur, Feuchte etc.): hoch/gering
- Reproduzierbarkeit des Sensorverhaltens: hoch/niedrig
- Exemplarstreuungen der Sensoren: hoch/niedrig
- Inhärentes Rauschen der Sensorhardware: hoch/niedrig
- Konstanz der Umgebungsbedingungen: hoch/niedrig

Im einfachsten Fall soll die Ausprägung einer einzelnen physikalische Größe gemessen werden, so daß die Aufgabe des Sensorsystems im wesentlichen darin besteht, diese durch einen passenden Sensor in eine meßbare Größe zu überführen. Am Beispiel eines einzelnen in Luft nachzuweisenden Gases würde dies bedeuten, einen Sensor zu finden, der auf dieses Gas auf irgendeine Weise „anspricht“, seine Kennlinie zu bestimmen und das vom Sensor ausgegebene Signal mit dieser Kennlinie zu verrechnen. Treten in der Applikation keine weiteren Gase auf und ist das Verhalten des Sensors reproduzierbar, so ist das Problem an dieser Stelle bereits gelöst.

Am anderen Ende der Skala befindet sich z.B. die Detektion eines Schwelbrandes mit Hilfe eines Multisensorsystems. Die Erkennung erfolgt über den Nachweis brandcharakteristischer

Gase in brandcharakteristischen Konzentrationsverhältnissen und –Verläufen. Neben diesen Brandgasen treten auch im „Normalzustand“ eine Reihe von weiteren Gasen auf, die eine Sensorreaktion hervorrufen, aber nicht ad-hoc als Störgase identifizierbar sind.

Anhand dieses Beispiels lassen sich bereits die essentiellen Komponenten eines Multisensorsystems identifizieren und die damit auftretenden Probleme charakterisieren: Häufig genügt nicht der Nachweis eines einzelnen Gases. Dies führt dazu, daß statt eines einzelnen Sensors mehrere eingesetzt werden müssen, die auf unterschiedliche Gase in unterschiedlicher Weise reagieren. Somit wird, zumindest prinzipiell, die rechnerische Identifikation eines Gases ermöglicht.

Allerdings ist die Empfindlichkeitsmatrix eines solchen Sensor-Arrays wiederum von der Konzentration abhängig, was dazu führt, daß zur vollständigen Behandlung des Problems ein umfassendes Kennfeld aufgenommen werden muß (Lookup-Tabelle oder Neuronales Netz). Bei 5 Gasen in 10 Konzentrationsstufen würde dies bereits das Aufnehmen von 105 Meßpunkten bedeuten, um das System in Betrieb nehmen zu können bzw. es exakt nachzukalibrieren.

Eine weitere Schwierigkeit bei der Informationsextraktion besteht darin, daß das Verhalten der Sensoren produktions- und alterungsbedingt starken Schwankungen unterworfen sein kann.

Darüber hinaus erschwert der Faktor Zeit das Auffinden einfacher Lösungen: zum einen müssen in einem Mehrsensorarray die unterschiedlichen Ansprechzeiten der Einzelsensoren berücksichtigt werden, zum andern ist auch der zeitliche Ablauf des zu detektierenden Ereignisses nicht immer exakt derselbe und soll dennoch innerhalb nur schlecht definierbarer Grenzen wiedererkannt werden.

Aus den skizzierten Randbedingungen ergeben sich die folgenden Schritte bei der Entwicklung eines neuen Sensorsystems mit den entsprechenden Anforderungen an den Entwickler:

1. Gewinnung von Sensordaten: Hierzu gehören Kenntnisse über Sensorgrundlagen wie Betriebsweisen, Wechselwirkung der Stoffe auf der Oberfläche, Querempfindlichkeiten, Anforderungen an das Gesamtsystem (Dichte der Platzierung, autonome oder interagierende Subsysteme,...).
2. Strukturierung der Daten: Hierzu gehört der Aufbau der Informationsverarbeitungskette bestehend aus den Komponenten Rohdatengewinnung, Merkmalsbildung, Merkmalsextraktion, Aufbau von Referenzbibliotheken und Klassifikatoren.
3. Strukturierung des Entscheidungsprozesses: Ist die Zielfunktion symmetrisch oder asymmetrisch (unterschiedliche Behandlung von Fehlalarmen 1. und 2. Art)? Hierzu gehört auch die Hierarchisierung der Entscheidungsstruktur: nur die auf der jeweiligen Entscheidungsstufe relevante Information wird in die Teilentscheidungsprozesse aufgenommen.
4. Aggregation der Gesamtentscheidung: Kombination der Teilentscheidungen (aus hierarchischer Klassifikation) mit quantitativen Aussagen.
5. Systemtest: Überprüfung von Systemparameteränderungen (veränderte Schwellwerte) und Änderungen in den Eingangsdaten (Sensordrift, Rauschen), Optimierung von Auswertearithmen mit Surrogatdaten.

## 2.2 Sensoren zum Nachweis von Gasen in Luft

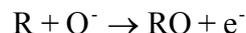
### 2.2.1 Metalloxid-Halbleitersensoren

Halbleitende Gassensoren eignen sich zum Nachweis von oxidierenden und reduzierenden Gasen, da sie auf deren Anwesenheit mit einer meßbaren Änderung ihrer elektrischen Leitfähigkeit reagieren. Verglichen mit anderen Sensortypen, wie z.B. elektrochemischen Gassensoren, sind sie sehr preisgünstig herzustellen und verfügen über eine nahezu unbegrenzte Lebensdauer. Ein einzelner Halbleiter-Gassensor ist jedoch nur bedingt zum selektiven Nachweis bestimmter Gase geeignet, da er gegenüber praktisch allen oxidierenden und reduzierenden Gasen Querempfindlichkeiten aufweist. Durch eine geeignete Wahl seiner Herstellungs- und Betriebsparameter lassen sich die Querempfindlichkeiten zu einzelnen Gasen jedoch so beeinflussen, daß durch die Kombination verschieden präparierter Sensoren zu einem Array mit nichtidentischen Querempfindlichkeitsmatrixelementen ein selektiver Nachweis von Gasbestandteilen möglich wird.

Die gassensitiven Schichten der im Rahmen dieser Arbeit verwendeten Sensoren bestehen aus nichtstöchiometrischem Zinndioxid, dem zur Zeit verbreitetsten Sensormaterial. Die n-leitenden Eigenschaften dieses Metalloxid-Halbleiters ergeben sich aus der Donatorwirkung der Sauerstofflückenstellen im Kristallgitter. Seine elektrische Leitfähigkeit hängt neben den geometrischen Abmessungen von der Ladungsträgerbeweglichkeit und der Ladungsträgerdichte ab. Während letztere über acht Größenordnungen nahezu linear vom Partialdruck der zu messenden Substanz abhängt, ändert sich die Ladungsträgerbeweglichkeit in diesem Bereich nur geringfügig.

Sauerstoffmoleküle der Umgebungsluft werden zunächst durch schwache Van-der-Waals-Kräfte an der Oberfläche des Halbleiters physisorbiert, ohne dessen elektrische Eigenschaften zu verändern. Bei der anschließend möglichen Chemisorption wird der Sauerstoff dissoziiert und über eine Dipolbindung an den Halbleiter gebunden, so daß diesem Elektronen entzogen werden. Die Leitfähigkeit des Halbleiters ist somit antiproportional zur Dichte der an der Halbleiteroberfläche chemisorbierten Sauerstoffatome und nimmt daher mit steigendem Sauerstoffpartialdruck in der Luft ab. Das thermodynamische Gleichgewicht zwischen diesem und dem Sauerstoffdefizit im  $\text{SnO}_{2-x}$ -Kristallgitter stellt sich jedoch erst oberhalb einer Temperatur von ca. 280°C ein, darunter sind die Sauerstofflückenstellen quasi „eingefroren“.

Die reduzierenden Gase (R), für deren Nachweis man sich interessiert, reagieren an der Halbleiter-Oberfläche mit den chemisorbierten Sauerstoffatomen:



Die frei gewordenen Ladungsträger werden an das Leitungsband des Halbleiters zurückgegeben, während das Reaktionsprodukt thermisch von der Halbleiter-Oberfläche desorbiert wird. Der Leitwert an der Halbleiter-Oberfläche steigt wieder an. Solange Sauerstoff und reduzierende Gase in der Umgebung des Halbleiters vorhanden sind, sind Chemisorption und Reduktion kontinuierlich ablaufende Prozesse. Bei gleichbleibendem Sauerstoffpartialdruck in der Luft ist der elektrische Leitwert des Halbleiters also proportional zur Konzentration reduzierender Gase in der Atmosphäre und kann daher zu deren Nachweis verwendet werden.

Da die beschriebenen Prozesse nur die Oberfläche des Halbleiters betreffen und tiefere Schichten nur marginal beeinflussen, werden Halbleiter-Gassensoren entweder aus porösem Sintermaterial oder in Dünnschichttechnik angefertigt, um ein möglichst großes Verhältnis von Oberfläche zu Volumen zu erzielen.

Bei reduzierenden Gasen sind die Korngrenzen des gesinterten polykristallinen  $\text{SnO}_2$ , sowie Oberflächenadsorbate für die Leitfähigkeitsänderung maßgebend, wogegen bei oxidierenden

Gasen im wesentlichen die Grenzfläche  $\text{SnO}_2$ -Metallkontakt-Gasphase verantwortlich ist [Wei 92]. Eine eingehende Behandlung der Oberflächenprozesse, die für die Detektion von reduzierenden Gasen mit  $\text{SnO}_2$ -Sensoren ausgenutzt werden, findet sich in [Koh 89].

Zur Erhöhung der Reaktionsgeschwindigkeiten und zur Selektivitätsbeeinflussung werden Katalysatoren eingesetzt. So läßt sich z.B. die Empfindlichkeit gegenüber CO durch die Dotierung des Halbleitermaterials mit Thoriumdioxid verbessern, während eine Dotierung mit Silber die Empfindlichkeit gegenüber  $\text{H}_2$  erhöht. Neben der Dotierung des Halbleiters ist dessen Betriebstemperatur ein wichtiger Parameter zur Steuerung der Selektivität gegenüber einzelnen Gasen.

Bei länger wählender Lagerung können an der Oberfläche des Halbleiters Luftverunreinigungen und Wassermoleküle adsorbiert werden, die erst bei hohen Temperaturen und längerer Betriebszeit (Stunden bis Tage) wieder vollständig desorbieren.

Auch während des Betriebs kann es zu Störungen kommen, deren Auswirkungen sich meist in einem Driften des Sensorsignals oder in Schwankungen der Sensitivität manifestieren und deren häufigste Ursachen aus Veränderungen von Sensortemperatur, Sauerstoffpartialdruck oder Luftfeuchtigkeit bestehen. Die in der Umgebungsluft vorhandenen Wassermoleküle können an der Halbleiter-Oberfläche unter Bildung von Hydroxylgruppen und Wasserstoff dissoziieren. Dabei wird die OH-Gruppe an der Halbleiter-Oberfläche chemisorbiert, während die Reaktion des Wasserstoffs mit einem der chemisorbierten Sauerstoffatome zur Bildung einer zweiten, ebenfalls chemisorbierten Hydroxylgruppe führen kann. Als Folge findet eine Freisetzung elektrischer Ladungsträger an der Oberfläche des Halbleiters statt. Ausgehend von einer endlichen Anzahl freier Oberflächenplätze, an denen Sauerstoffatome oder Hydroxylgruppen chemisorbiert werden können, führt deren Absättigung mit Hydroxylgruppen zu einer Abnahme der Sensorempfindlichkeit. Zur Desorption (bzw. Nichtadsorption) von Hydroxylgruppen muß der Sensor entweder permanent bei hohen Temperaturen betrieben (sofern sich dies mit der gewünschten Zielgasselektivität vereinbaren läßt) oder in regelmäßigen Abständen ausgeheizt werden.

### 2.2.1.1 Das Kristallvolumen

Bei den zum Gasnachweis einsetzbaren Metalloxiden wie  $\text{SnO}_2$ ,  $\text{TiO}_2$  oder  $\text{Ga}_2\text{O}_3$  handelt es sich um Halbleiter, deren n-leitende Eigenschaft aus einem Sauerstoffdefizit im Kristallgitter, dem sog. *Bulk*, herrührt, d.h. die Fehlstellen wirken als Elektronendonatoren.

Bei hohen Temperaturen stellt sich ein thermodynamischer Gleichgewichtszustand zwischen dem Sauerstoffpartialdruck der umgebenden Atmosphäre und den Gitterfehlstellen ein, so daß die Stöchiometrie des Kristalls grundsätzlich vom  $\text{O}_2$ -Gehalt seiner Umgebung abhängig ist.

Bei niedrigen Temperaturen stellt sich das thermodynamische Gleichgewicht jedoch so langsam ein, daß die Fehlstellen im Gitter quasi „eingefroren“ bleiben (beim  $\text{SnO}_2$  bis ca.  $280^\circ\text{C}$ , beim  $\text{Ga}_2\text{O}_3$  bis ca.  $1000^\circ\text{C}$ ).

Der Ladungstransport läßt sich beim  $\text{SnO}_2$  durch freie Elektronen [Zem 88], beim  $\text{Ga}_2\text{O}_3$  durch ein Polaronenmodell beschreiben [Fle92], [Kru98].

### 2.2.1.2 Die Kristalloberfläche

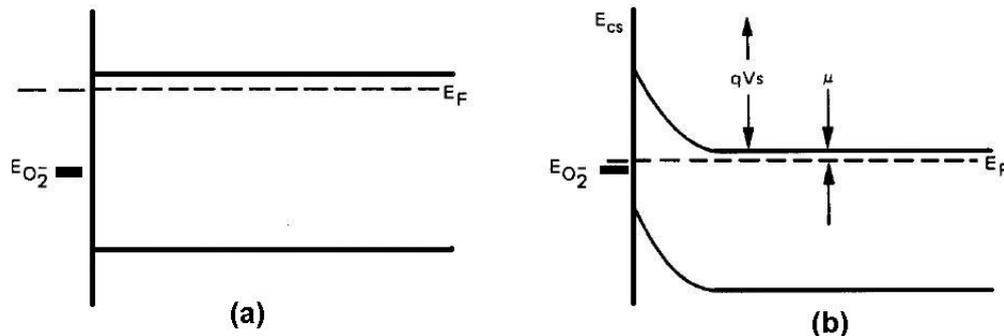
Die Störung der Periodizität an der Oberfläche des Kristalls führt zu lokalisierten, diskreten Energieniveaus, die u.U. auch in der verbotenen Zone liegen können [Mad89].

Betrachtet man diese Oberflächenzustände für sich alleine (unbesetzte Akzeptorzustände, besetzte Donatorzustände), so ist deren Fermi-Energie nicht identisch mit der im Innern des Kristalls. Es kommt daher zu einem Ladungsaustausch zwischen den Oberflächenzuständen und dem Bulk. Dabei bilden sich lokalisierte Oberflächenladungen. Zusätzlich adsorbiert Umgebungssauerstoff auf der Halbleiteroberfläche, was zu einem weiteren Ladungsaustausch führt.

Es entsteht eine Oberflächenladung der Stärke  $qN_s$  (mit  $q$  = Elementarladung und  $N_s$  = Anzahl besetzter Adsorptionsplätze auf der Oberfläche), die einen an Ladungsträgern verarmten Bereich dicht unter der Oberfläche induziert, die sog. Verarmungsrandschicht. Die entstehende Bandverbiegung lässt sich mit Hilfe der Poisson-Gleichung beschreiben

$$\frac{\partial^2 \Phi}{\partial x^2} = -\frac{\rho}{\epsilon_r \epsilon_0} \quad (2.1)$$

, wobei  $\rho$  die Ladungsdichte im Bulk,  $x$  den Abstand zur Oberfläche und  $\Phi$  das elektrische Potential repräsentiert. Diese Bandverbiegung führt letztlich zu einer Begrenzung der Sauerstoffadsorption an der Oberfläche: Sobald das Niveau der Oberflächenakzeptoren auf das elektrochemische Potential des Bulks abgesunken ist, ist keine Chemisorption mehr möglich. Dieser Gleichgewichtsprozess wird auch als „Fermi level-pinning“ bezeichnet. Das Energieniveau der Oberflächenzustände bestimmt somit die Fermienergie des Systems und limitiert den Wert von  $qV_s$  auf ca. 0,5 bis 1,0eV (s. Abb. 2.1).



**Abb.: 2.1:** Verarmungsrandschicht an der Oberfläche eines n-Halbleiters nach Adsorption eines Elektronenakzeptors (Sauerstoff) [Mad 89].

Durch zweimaliges Integrieren der Poissongleichung lässt sich die Höhe der sog. Schottky-Barriere bestimmen:

$$V_B = \frac{q N_d \lambda^2}{2 \epsilon_r \epsilon_0} \quad (2.2)$$

Dabei ist  $N_d$  die Donatordichte und  $\lambda$  die Dicke der Verarmungsrandschicht. Ersetzt man in der Lösung der Poisson-Gleichung  $N_d \lambda = N_s$ , so erhält man die Schottky-Approximation für das Oberflächenpotential:

$$V_B = \frac{q N_s^2 \lambda^2}{2 \epsilon_r \epsilon_0 N_d} = V_s \quad (2.3)$$

Eine Limitierung von  $qV_s$  führt somit zu einer Limitierung von  $N_s$ , also einem Bedeckungslimit. Sauerstoff erzeugt ein maximales Oberflächenpotential von  $qV_s \approx 1\text{eV}$ , seine maximale Bedeckung liegt bei  $N_D = 10^{12}$  bis  $10^{13} \text{ cm}^{-2}$  (Weisz-Limit) [Hei 88], [Mad 89]. Mit Werten von  $\epsilon_r \epsilon_0 \approx 10^{-12} \text{ F cm}^{-1}$  erhält man einen Wertebereich von  $\lambda \approx 100 - 1000\text{nm}$  für die Dicke der Verarmungsrandschicht.

### 2.1.1.2.1 Adsorptions- und Desorptionsprozesse auf der Kristalloberfläche

Die Adsorption von Akzeptoren oder Donatoren auf der Kristalloberfläche beeinflusst die Verarmungsrandschicht durch Extraktion von Ladungsträgern aus dem oder Injektion von Ladungsträgern in den Bulk.

Reduzierend wirkende Gase wie z.B. CO adsorbieren auf der Oberfläche und reagieren mit dem ionosorbierten Sauerstoff oder auch mit dem Gittersauerstoff. Die vorher vom Sauerstoff gebundenen Elektronen werden freigesetzt bzw. die Sauerstoff-Fehlstellen diffundieren in den Bulk, was letztlich zu einer Leitwerterhöhung führt [Koh 89].

#### 2.2.1.2.1.1 Physisorption

Mit Physisorption bezeichnet man einen Adsorptionsprozeß, bei dem sowohl die geometrische Struktur als auch die elektronischen Eigenschaften der freien Teilchen und der freien Oberfläche erhalten bleiben. Dieser Adsorptionsprozeß, kann durch elektrostatische Kräfte, Dipolkräfte oder Van-der-Waals-Kräfte ausgelöst werden.

Das Potential der wechselwirkenden Teilchen kann dabei mit Hilfe eines Zweiteilchenpotentials beschrieben werden, das sich aus einem anziehenden und einem abstoßenden Term zusammensetzt. Ersterer ist halbklassisch betrachtet auf elektrostatische Anziehung sowohl stationärer als auch zeitlich fluktuierender Ladungsverteilungen zurückzuführen, letzterer rührt aus dem Pauli-Prinzip her.

Das resultierende Zweiteilchenpotential kann mit Hilfe des Lennard-Jones Potentials beschrieben werden:

$$E_{pot} = E_{attr.} + E_{rep} \propto 4 \epsilon \left[ - \left( \frac{\sigma}{r} \right)^6 + \left( \frac{\sigma}{r} \right)^{12} \right] \quad (2.4)$$

, wobei  $\sigma$  den Abstand der wechselwirkenden Teilchen und  $\epsilon$  die Tiefe der Potentialmulde darstellt.

Eine stärkere (Bindungsenergie  $> 0,5\text{eV}$  pro Teilchen), chemische Bindung des Adsorbats an den Kristall liegt im Falle einer Ionosorption oder Chemisorption vor.

#### 2.2.1.2.1.2 Ionosorption

Bei der als Ionosorption bezeichneten Bindung wird das Adsorbat ionisiert, indem Ladung vom Leitungs- oder Valenzband des Festkörpers zum Adsorbat transferiert wird; die Bindung erfolgt über elektrostatische Kräfte.

Sauerstoff kann als  $O_2^-$ ,  $O^-$  oder  $O^{2-}$  adsorbiert werden, wobei letztere Konstellation sehr instabil ist und relativ hohe Oberflächenpotentiale zur Bindung benötigt. Ermöglicht wird die

Ionosorption dadurch, daß die Oberflächenzustände beim  $\text{SnO}_2$  unterhalb der Fermienergie des isolierten Zustandes (in Abwesenheit von Umgebungssauerstoff) liegen.

### 2.2.1.2.1.3 Chemisorption

Der Chemisorptionszustand ist durch eine Aktivierungsbarriere vom Physisorptionszustand getrennt. Als Unterscheidungskriterium für das Vorliegen des einen oder anderen Zustandes dient die Bindungsenergie zwischen Adsorbat und Oberfläche. Liegt diese oberhalb von  $0,5\text{eV}$  pro Teilchen, so spricht man von Chemisorption. Wechselwirkungen dieser Stärke können die räumliche Anordnung der Oberflächenatome beeinflussen. Bei höheren Temperaturen kann die Chemisorption von Molekülen zu deren Dissoziation an der Oberfläche führen. Abb. 2.2 zeigt schematisch ein Potentialdiagramm für eine solche dissoziative Chemisorption eines zweiatomigen Moleküls  $X_2$ . Nähert sich das  $X_2$ -Molekül der Oberfläche, so kommt es in einem Abstand  $z^{\text{phys}}$  zu einem Energieminimum von Anziehung und Abstoßung. Diese Stelle nennt man Physisorptionsposition von  $X_2$  (rechte Potentialkurve in Abb 2.2). Nach Aufwendung der Dissoziationsenergie  $E^{\text{diss}}$  kann man die beiden einzelnen Atome  $X$  der Oberfläche nähern (linke Kurve in Abb 2.2). Die beiden Atome werden dann in einem Abstand  $z^{\text{chem}}$  an der Oberfläche gebunden. Aus der Abbildung ist ersichtlich, daß die beiden Kurven sich in einem Punkt schneiden, der um  $\Delta E$  höher als die Energie eines ruhenden Teilchens liegt, welches unendlich weit entfernt ist. Die Chemisorption kann also nur dann aus dem Physisorptionszustand heraus erfolgen, wenn die entsprechende Aktivierungsenergie aufgebracht wird. Dies geschieht in der Regel durch thermische Aktivierung, weshalb bei tiefen Temperaturen fast ausschließlich Physisorption und keine Chemisorption stattfindet.

Während der Chemisorption kommt es zu Änderungen in der elektronischen Struktur der freien Moleküle und der Oberfläche. Die Moleküle gehen eine chemische Bindung mit den Oberflächenatomen ein. Kovalente Bindungen bilden sich bevorzugt zwischen Metallen und dem Adsorbat aus, während sich Ionenbindungen oft zwischen dem Adsorbat und Halbleitern oder Isolatoren ausbilden. Dadurch kommt es auch zu einer Änderung der Bindungen der Oberflächenatome untereinander. Die Chemisorptionsbindung ist meist nicht reversibel.

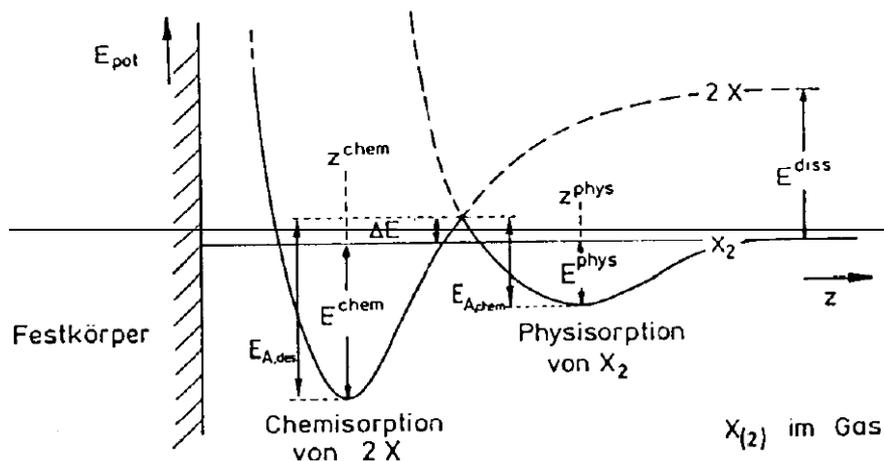


Abb. 2.2: Potentialdiagramm Chemisorption / Physisorption

#### 2.2.1.2.1.4 Desorption

Unter Desorption versteht man den Umkehrprozeß der Adsorption, d.h. das Aufbrechen chemischer Bindungen mit anschließender Entfernung der adsorbierten Teilchen von der Oberfläche. Sie kann aufgrund thermischer Anregung (Aufheizen der Probe auf Desorptionstemperatur) oder durch gezielte Anregung bestimmter elektronischer oder vibronischer Zustände erfolgen.

Bei der isothermen Desorption lösen sich die Teilchen von der Oberfläche bei konstanter Temperatur, getrieben durch einen Gradienten im chemische Potential zwischen Teilchen in der Gasphase und den adsorbierten Teilchen. Sie tritt z.B. dann auf, wenn ein zuvor adsorbiertes Gas plötzlich aus der Umgebung abgepumpt wird oder wenn die Oberfläche zunächst mit einem Molekularstrahl beschossen wurde und dieser unterbrochen wird.

Die Desorption von Adsorbaten kann auch durch nicht-thermische Prozesse ausgelöst werden, z.B. durch Photonen-, Elektronen- oder Ionenbeschuß sowie durch starke elektrostatische Felder ausgelöst werden.

#### 2.2.1.2.1.5 Adsorptions-Isothermen

Zur Beschreibung des Zusammenhangs zwischen der Oberflächen-Bedeckung  $\Theta$  eines Adsorbierten Gases und dessen Partialdruck in der Gasphase in der Nähe der Oberfläche stehen verschiedene Adsorptions-Isothermen zur Verfügung. Die wichtigsten Vertreter ihrer Art sind dabei:

##### 2.2.1.2.1.5.1 Die Henry-Isotherme

Die Henry Isotherme geht von einem linearen Zusammenhang zwischen Bedeckungsgrad  $\Theta$  und Partialdruck des Gases aus. Dieser Zusammenhang stellt für niedrige Bedeckungsgrade tatsächlich bereits eine sehr gute Näherung dar.

##### 2.2.1.2.1.5.2 Die Langmuir-Isotherme

Der Langmuir-Isotherme liegen folgende Annahmen zugrunde: Begrenzte Anzahl identischer Adsorptionsplätze, so daß eine maximale Bedeckung  $\Theta_{\max}$  auch bei steigendem Partialdruck nicht überschritten werden kann, Unabhängigkeit der Adsorptionsenergie vom Bedeckungsgrad sowie vernachlässigbare Wechselwirkung der Teilchen untereinander.

Für den Gleichgewichtsfall werden identische Adsorptions- und Desorptionsraten unterstellt, die neben dem Bedeckungsgrad  $\Theta$  nur von den temperaturabhängigen Adsorptions- bzw. Desorptionskonstanten  $k_{ads}$  und  $k_{des}$  sowie im Fall der Adsorption vom Partialdruck  $p$  abhängig seien:

$$k_{ads}(1 - \Theta)p = k_{des}\Theta \quad (2.5)$$

Löst man (2.5) nach der Bedeckung auf, so erhält man die Langmuir-Isotherme:

$$\Theta = \frac{k_{ads}p}{k_{ads}p + k_{des}} = \frac{Kp}{1 + Kp} \quad (2.6)$$

mit  $K = k_{ads}/k_{des}$ . Im Fall hoher Bedeckungsgrade wird also ein von K abhängiger Sättigungswert erreicht.

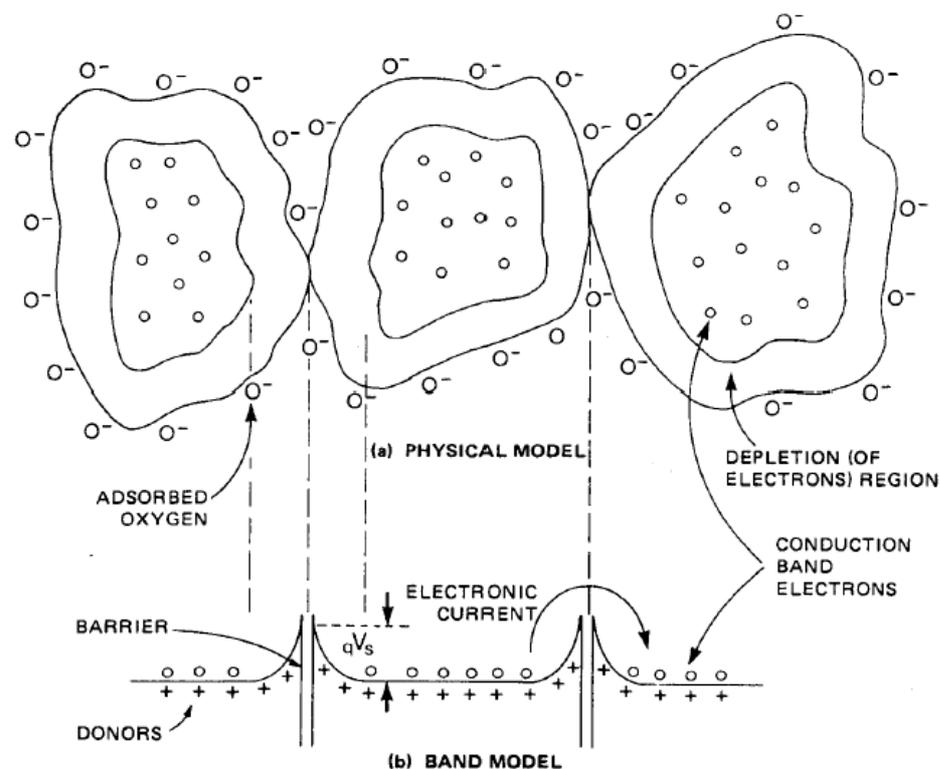
Aufgrund ihrer vereinfachenden Annahmen besitzt auch die Langmuir-Isotherme nur Näherungscharakter.

### 2.2.1.2.1.5.3 Die Freundlich-Isotherme

Die Freundlich-Isotherme wurde zunächst empirisch gefunden und kann als eine Überlagerung von Langmuir-Isothermen mit unterschiedlichen Adsorptionsenergien betrachtet werden. Die Adsorptionsenergien nehmen mit dem Logarithmus der Bedeckung ab. Die Freundlich-Isotherme besitzt keinen Sättigungswert für  $\Theta$ .

### 2.2.1.3 Polykristalline Schichten

Um ein möglichst günstiges Oberflächen zu Volumen-Verhältnis zu erzielen, werden Gassensoren meist nicht aus Einkristallen sondern aus polykristallinen Sinterstrukturen hergestellt, wie in Abb. 2.3 schematisch dargestellt:



**Abb 2.3:** Polykristallines  $\text{SnO}_2$  mit Verarmungsrandschicht an den Kornoberflächen

Je nach Versinterungsgrad wird die Leitfähigkeit des Gesamtsystems durch das Verhältnis zwischen Oberflächen- und Volumenleitfähigkeit (Randschichtmodell der Chemisorption) oder Korngrenzeneffekte bestimmt. Bei Dominanz von Korngrenzeneffekten lässt sich der Kontakt zwischen den Körnern entweder eher als ohmsch oder als Schottky-Diode betrachten.

### 2.2.1.3.1 Thermoionische Emission über Schottky-Barrieren

Die thermoionische Emission stellt den dominierenden Ladungstransportmechanismus bei Schottky-Barrieren dar. Eine weitere Art des Ladungstransports bei Doppel-Schottky-Barrieren wird durch den Tunneleffekt ermöglicht, der bei hochdotierten Materialien eine Rolle spielen kann.

Die Schottky-Approximation (Gl. 2.3) liefert den Zusammenhang zwischen Oberflächenladung durch adsorbierten Sauerstoff und der Höhe der sich daraus ergebenden Barrieren. Für den Fall der thermoionischen Emission erhält man für die Stromdichte (s. z.B. [Wei 80])

$$j \propto \exp\left(\frac{-q \cdot E_0}{k \cdot T}\right) \quad (2.7)$$

Zur Gasetektion wird die Variation der Breite der Verarmungsrandschicht an den Korngrenzen in Gegenwart von reduzierenden oder oxidierenden Gasen ausgenutzt, die ihrerseits zu einer Variation der Höhe der Energiebarrieren für freie Ladungsträger führt.

Wird bei Anwesenheit von Gas die Barrierenhöhe verringert, kommt es zu einem Anwachsen der Leitfähigkeit der gassensitiven Schicht. Die Höhe Nachweisempfindlichkeit von SnO<sub>2</sub> für viele Gase resultiert aus der exponentiellen Abhängigkeit der Korn-Korn-Leitfähigkeit von der Barrierenhöhe  $E_0$ .

### 2.2.1.3.2 Ohmsche Leitungsmechanismen

Bei stark versinterten Korn-Korn-Übergängen bilden sich zwischen den benachbarten Körnern sogenannte Necks aus, durch die eine ohmsche Stromleitung mit konstanter Ladungsträgerdichte stattfinden kann. Eine detaillierte Abhandlung findet sich z.B. in [Ulr 99].

### Morphologische Eigenschaften und Parameterstreuungen

Der in Bezug auf die erzielbaren Empfindlichkeiten vorteilhafte polykristalline Aufbau der gassensitiven SnO<sub>2</sub>-Schicht ist von einer Reihe weiterer Effekte begleitet, die sich auf das Sensorverhalten eher negativ auswirken. Dazu gehören neben einer oftmals beobachtbaren Grenzempfindlichkeit für Zielgase das Auftreten produktionsbedingter und zeitlicher Parameterstreuungen in den Sensorempfindlichkeiten. Eine ausführliche Studie findet sich in [Ulr 99].

## 2.3 Signalverarbeitung

### 2.3.1 Mustererkennung

Bei der Mustererkennung handelt es sich um ein Teilgebiet der künstlichen Intelligenz, dessen Zielsetzung in einer Nachahmung der enormen Erkennungsfähigkeit biologischer Vorbilder durch technische Systeme besteht. Im Gegensatz zu rein neuronalen Ansätzen läßt sie beliebig formulierte Algorithmen zur Lösung von Erkennungsproblemen zu.

Das zu erkennende Muster wird zunächst in die Form eines Merkmalvektors  $\mathbf{v}$  gebracht, dessen Komponenten aus Attributen unterschiedlichen Abstraktionsgrades bestehen können, um dann mit Hilfe des Erkennungsprozesses mit einer zugehörigen Bedeutung  $\omega_i$  aus der Menge der  $L$  möglichen Bedeutungen  $\Omega = \{\omega_1, \omega_2, \dots, \omega_i, \dots, \omega_L\}$  verknüpft zu werden.

Die Art und Weise, in der die Bedeutungsmenge  $\Omega$  numerisch zu repräsentieren ist, hängt von der Wahl des eingesetzten Klassifikators ab. Die in der vorliegenden Arbeit eingesetzten neuronalen Klassifikatoren arbeiten z.T. nach dem WTA-Prinzip (winner takes all), so daß die Verwendung eines Klassenindex ausreichend ist. Dabei wird jede der  $L$  Kategorien  $\omega_i$  aus  $\Omega$  nur durch ihren Index  $i$  repräsentiert.

Bei der Verwendung von Funktionsapproximatoren mit nur einem Ausgang ( $F: \mathbb{R}^n \rightarrow \mathbb{R}$ , z.B. MLP mit nur einem Ausgangsneuron) als Klassifikatoren würde die Verwendung von Klassenindizes jedoch zu unerwünschten Nachbarschaftsbeziehungen führen. Diese lassen sich durch die Verwendung orthogonaler Zielvektoren vermeiden. In diesem Fall besteht  $\Omega$  aus einer  $L \times L$ -Einheitsmatrix, deren Spalten die Zielvektoren darstellen. Konkret bedeutet dies ein Ausweichen auf einen Funktionsapproximator mit  $L$  Ausgängen ( $F: \mathbb{R}^n \rightarrow \mathbb{R}^L$ , d.h. z.B. MLP mit einem Ausgangsneuron pro Klasse).

Die Anzahl der für das Erlernen der gewünschten Abbildungsfunktion erforderlichen Beispiele hängt von der Qualität der Musterrepräsentation sowie der Komplexität der zu lösenden Aufgabe ab. Zur Überprüfung des Erkennungssystems verwendet man in der Regel sowohl die Lernstichprobe (Reklassifizierung oder Resubstitution) als auch eine unabhängig gebildete Teststichprobe (Generalisierung). Diese Vorgehensweise trägt der Tatsache Rechnung, daß ein zu stark an die Eigenheiten der Trainingsmenge angepaßtes Erkennungssystem einen Teil seiner Vorhersagefähigkeit einbüßt.

Wie auch in anderen Disziplinen üblich, werden in der Mustererkennung Begriffe verwendet, die ebenfalls eine allgemeinere, vorwissenschaftliche Bedeutung tragen und daher an dieser Stelle definiert werden sollen [Bot 95]:

#### **Definition 2.3.1.1:** *Objekt*

Ein *Objekt* ist ein abstraktes Modell eines realen Systems, das durch einen Satz meßbarer Modellparameter festgelegt wird.

#### **Definition 2.3.1.2:** *Merkmal*

Die Modellparameter nach Def. 3.1 heißen auch *Merkmale*. Diese repräsentieren charakteristische Eigenschaften von Objekten und wirken insofern unterscheidend. Sie ordnen die Objekte durch Bereichsbildung in einem Merkmalsraum, der durch sie aufgespannt wird.

**Definition 2.3.1.3:** *Cluster (Klasse), Klassifizierung*

Die Gruppierung von Objekten nach einem festgelegten Ähnlichkeitskriterium (Metrik) führt zu (Objekt-) *Clustern*. Der Vorgang der Eingruppierung heißt *Klassifizierung*. Cluster können in einem weiterführenden Prozeß wiederum als Objekte angesehen werden.

**Definition 2.3.1.4:** *Klassifikator*

Ein *Klassifikator* ist ein Algorithmus, der ein unbekanntes Objekt  $x_i$ , charakterisiert durch einen Merkmalsvektor  $v_i$ , einem oder im Falle unscharfer Klassifikatoren mehreren Clustern  $\omega_k$  zuordnet. Er zerlegt damit die Grundmenge der Objekte  $x_i$  in  $L$  Teilmengen.

**2.3.2 Merkmalgenerierung, Merkmalselektion und -extraktion**

Der Prozeß der Überführung realer Objekte in eine zur Mustererkennung geeignete mathematische Repräsentation nennt sich Merkmalbildung und umfaßt die Teilschritte Merkmalgenerierung und Merkmalselektion (bzw. Merkmalsextraktion). Unter Merkmalgenerierung soll im Rahmen der vorliegenden Arbeit die Erzeugung einer geeigneten Grundgesamtheit von Merkmalsvektoren verstanden werden. Im Zusammenhang mit Multisensorsystemen ist damit insbesondere die geeignete Aufbereitung der aufgezeichneten Sensorsignale und deren Anordnung zu Merkmalsvektoren innerhalb eines Datensatzes gemeint. Im Gegensatz zu den relativ leicht zu automatisierenden Prozessen der Merkmalselektion (bzw. Merkmalsextraktion) und Klassifikatorauswahl handelt es sich bei der Merkmalgenerierung um einen Vorgang, der stark von Intuition und Erfahrung bei der Wahl geeigneter Verfahren abhängt.

Zwischen Merkmalgenerierung und Klassifikation liegt eine systematische Dimensionsreduktion des Merkmalsraums. Diese dient neben einer Beschleunigung des Klassifikationsvorgangs in erster Linie der Beseitigung irrelevanter bzw. störender Information. Dabei ist zwischen zwei Gruppen von Verfahren zu unterscheiden:

Die erste Gruppe umfaßt Methoden zur Selektion entscheidungsrelevanter Merkmale aus einer vorgegebenen Menge generierter Merkmale (Merkmalselektion im engeren Sinne). Die Merkmalselektion dient der systematischen Beschränkung auf problemrelevante Merkmale zur Erhöhung der Erkennungssicherheit und der Aufwandsreduktion bei Merkmalsvektorbildung und Klassifikation. Die Güte der gefundenen Lösung hängt von der verwendeten Suchstrategie und dem Gütekriterium ab.

Bei der zweiten Gruppe von Verfahren (Merkmalsextraktion) findet die Dimensionsreduktion des Ausgangsraums durch eine mathematische Transformation in einen niederdimensionalen Merkmalsraum statt. Als Kriterien für eine solche Transformationsvorschrift dienen Signalrekonstruierbarkeit oder Strukturhaltung. Im Gegensatz zur Merkmalselektion gehen hier alle Komponenten der ursprünglichen Merkmalsvektoren ein, so daß bei der Merkmalbildung eine Aufwandserhöhung stattfindet. Die Merkmalsextraktion dient also ausschließlich der Entlastung des Klassifikators.

Im Rahmen dieser Arbeit kamen verschiedene algorithmische Merkmalselektionsverfahren zum Einsatz, die sich bezüglich der Suchstrategie und der Art ihres Merkmalgütebewertungssystems unterscheiden. Eine vollständige Durchsuchung des Merkmalsraums mit dem Ziel, die beste Merkmalskombination aus  $M$  Merkmalen auszuwählen würde die Überprüfung von

$$\sum_{i=1}^M \binom{M}{i} = 2^M - 1$$

Merkmalskombinationen erfordern. Bei einer Ausgangsmenge von z.B. 180 Merkmalen erhält man bereits über  $10^{54}$  Möglichkeiten. Eine solche Suche würde zwar alle Abhängigkeiten höherer Ordnung berücksichtigen, ist aber nicht mehr praktisch durchführbar.

Es lassen sich jedoch auch mit relativ einfachen Suchheuristiken gute Lösungen erzielen. Die im Rahmen dieser Arbeit eingesetzten hierarchischen Selektionsverfahren kommen bei der gleichen Merkmalsausgangsmenge im ungünstigsten Fall bereits mit  $1,6 \times 10^4$  Kombinationen aus. Läßt man einen vorzeitigen Abbruch der Suche beim Überschreiten einer vorgegebenen Merkmalsmindestgüte zu, so wird nochmals eine Aufwandsreduktion um bis zu vier Größenordnungen möglich.

Je nachdem, ob man die Suche mit dem vollständigen Merkmalsatz beginnt und diesen durch systematisches Weglassen der am wenigsten relevanten Merkmale verkleinert (top-down) oder den umgekehrten Weg beschreitet und das wichtigste Einzelmerkmal sukzessive durch weitere Merkmale ergänzt (bottom-up), spricht man von destruktiven (Sequential-Backward-Selection, SBS) bzw. konstruktiven (Sequential-Forward-Selektion, SFS) Verfahren.

Die Vorgehensweise sei exemplarisch anhand von SBS näher erläutert. Ausgehend von der kompletten Menge der Merkmale  $y^1$  wird durch probeweises Weglassen eines Merkmals  $v_l$  untersucht, welcher Art sein Einfluß auf eine vorher festzulegende Gütefunktion  $G$  ist. Dies geschieht durch Auswerten der Gütefunktion für das Komplement jedes der  $M$  Einzelmerkmale  $v_l$ . Endgültig entfernt wird dasjenige Merkmal, für dessen Probeweise Entfernung die Gütefunktion  $G(y^{i+1} = y^i \setminus v_k)$  ein Maximum bezüglich einer Merkmalszahl von  $M-1$  annimmt:

$$G(y^{i+1}) = G(y^i \setminus v_k) = \max_{i=1}^M G(y^i \setminus v_l) \quad (2.7)$$

Die Untersuchung wird mit der besten gefundenen Untermenge  $y^{i+1}$  fortgesetzt, bis nach  $M-1$  Schritten nur noch eine einelementige Untermenge  $y^M$  verbleibt. Ausgewählt wird schließlich diejenige Menge  $y^*$ , für die die Gütefunktion ihr globales Maximum erreicht:

$$G(y^*) = \max_{i=1}^M G(y^i) \quad (2.8)$$

SBS nimmt eine Einschränkung des Suchraums dadurch vor, daß es einmal entfernte Elemente auf den tieferen Rekursionsstufen nicht mehr berücksichtigt. Die gefundenen Merkmale, sowie deren ebenfalls durch SBS bestimmte optimale Anzahl können demzufolge mehr oder weniger stark vom tatsächlichen Optimum abweichen. Angesichts einer Aufwandsreduzierung von nahezu  $2^{M+1}$  zu  $M(M+1)$  (für große  $M$ ) gegenüber der vollständigen Suche sind die daraus resultierenden Nachteile allerdings zu vernachlässigen.

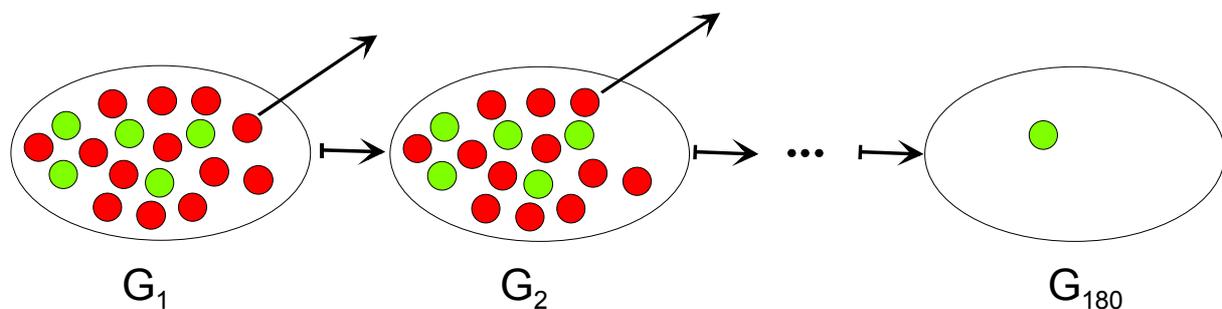


Abb 2.4: Schema der SBS

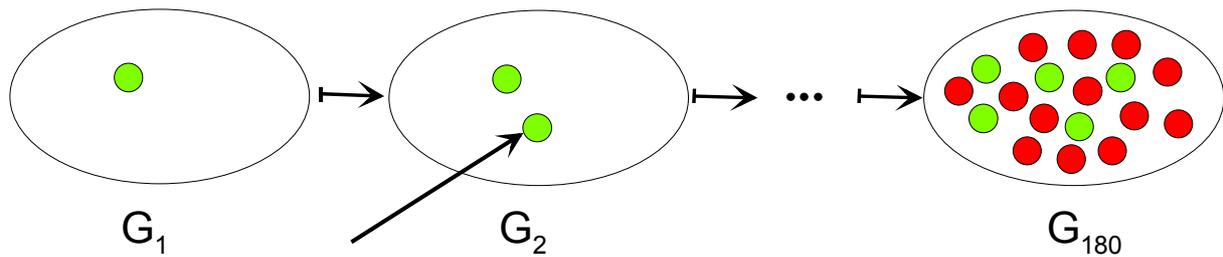


Abb 2.5: Schema der SFS

Die beschriebenen Suchheuristiken eliminieren nicht nur diejenigen Merkmale, die nicht zu einer Verbesserung der Unterscheidbarkeit verschiedener Klassen beitragen (also stark miteinander korrelierte Merkmale), sondern auch solche, die eine Trennbarkeit verschlechtern. Letzteres ist bei Variablen der Fall, die keinerlei mit den interessierenden Klassen übereinstimmende Clusterung aufweisen, so daß ihr Vorhandensein zu einer Abschwächung der Wirkung der entscheidungsrelevanten Variablen führt. Das Resultat der Suche hängt nicht zuletzt von der verwendeten Merkmalsgütemessungsfunktion ab. Beispiele für mögliche Gütefunktionen werden im folgenden Abschnitt (2.3.3) beschrieben.

### 2.3.3 Beurteilung der Merkmalsgüte

Um dem Ziel einer fehlerfreien Klassifikation gerecht zu werden, müssen die von den Merkmalvektoren gebildeten Klassengebiete leicht separierbar sein. Eine starke Überlappung der Cluster wirkt sich in jedem Fall störend aus. Daher stellt eine Gütemaß zur Bewertung des Überlappungsgrades ein wertvolles Hilfsmittel zur Beurteilung der Trennbarkeit verschiedener Klassen dar. Andererseits ist es denkbar, daß ein Klassengebiet sich zwar nicht mit anderen überlappt, dafür aber z.B. aus mehreren unzusammenhängenden Teilen besteht, die möglicherweise andere Klassengebiete umschließen oder allgemein eine Struktur aufweist, die die Fähigkeiten eines bestimmten Klassifikortyps überfordert (z.B. nichtlineare Klassentrennlinie bei linearem Klassifikator). Daher kann es zweckmäßig sein, den später einzusetzenden Klassifikator bei der Gütebewertung mit einzubeziehen.

#### 2.3.3.1 Überlappungsgrad

Die Bestimmung des Überlappungsgrades erfolgt durch Auszählung der  $k$  nächsten Nachbarn hinsichtlich eigener und fremder Klassenzugehörigkeit. Je nachdem, in welchem Maße weitere Informationen über Rangfolge und Distanzen in die Bewertungsgröße einfließen sollen, kann auf unterschiedlich rechenintensive Varianten zurückgegriffen werden [Kön 95]. Die Überprüfungs-kriterien werden dabei für jeden der  $N$  Merkmalvektoren  $\mathbf{v}_j$  angewandt und das Ergebnis zu einem Gütemaß gemittelt:

$$q_0 = \frac{1}{N} \sum_{j=1}^N \frac{\sum_{i=1}^k q_{NN_{j_i}} + \sum_{i=1}^k n_i}{2 \sum_{i=1}^k n_i} \quad (2.9)$$

mit

$$n_i = 1 - \frac{d_{NN_{ji}}}{d_{NN_{jk}}} \quad (2.10)$$

und

$$q_{NN_{ji}} = \begin{cases} n_i & : \omega_j = \omega_i \\ -n_i & : \omega_j \neq \omega_i \end{cases} \quad (2.11)$$

Dabei bezeichnen  $NN_{ji}$  und  $NN_{jk}$  den nächsten bzw. am weitesten entfernten nächsten Nachbarn von  $\mathbf{v}_j$ ,  $d_{NN_{ji}}$  und  $d_{NN_{jk}}$  geben die entsprechenden Distanzen zu diesem an. Die Klassenzugehörigkeit von  $\mathbf{v}_j$  sei  $\omega_j$ , die des  $i$ -ten nächsten Nachbarn  $\omega_i$ . Der Gewichtungsfaktor  $n_i$  bewirkt einen mit dem Rang abfallenden Einfluß der nächsten Nachbarn bei der Berechnung des Überlappungsgrades  $q_{NN_{ji}}$  (Null für den entferntesten nächsten Nachbarn  $NN_{jk}$ ). Je nach Klassenzugehörigkeit von  $NN_{ji}$  wird das Überlappingsmaß  $q_{0j}$  für den Vektor  $\mathbf{v}_j$ , um  $n_i$  erhöht oder reduziert, so daß  $q_{0j}$  Werte zwischen  $-\sum_{i=1}^k n_i$  und  $\sum_{i=1}^k n_i$  annehmen kann. Die Addition des Terms  $\sum_{i=1}^k n_i$  zu  $\sum_{i=1}^k q_{NN_{ji}}$  mit anschließender Division durch  $2\sum_{i=1}^k n_i$  dient einer Transformation des Wertebereichs auf das Intervall  $[0, 1]$ .

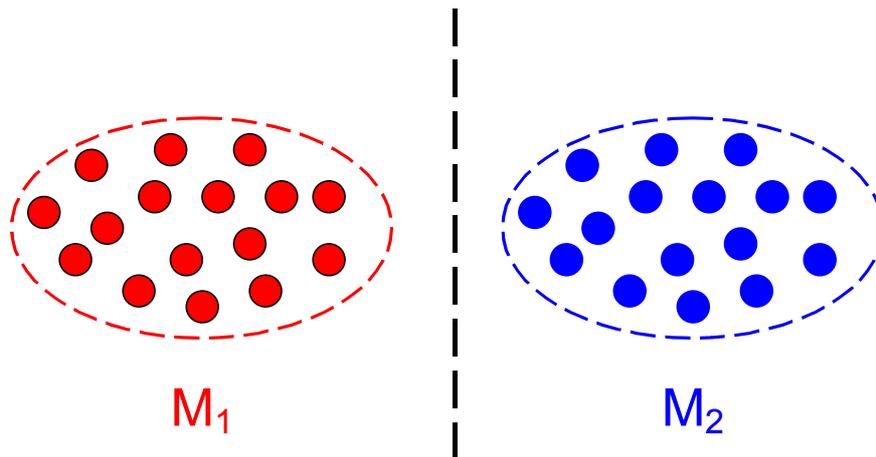
Eine Vereinfachung des Überlappingsmaßes entsteht durch den Verzicht auf die Gewichtungsfaktoren  $n_i$ . Sei  $\sum_{i=1}^k d_{NN_{ji}^{\omega_i}}$  die Summe aller Distanzen zu den nächsten Nachbarn von  $\mathbf{v}_j$ , für die  $\omega_j = \omega_i$  gilt, so erhält man ein Überlappingsmaß, bei dem immer noch die Rangfolge der  $k$  nächsten Nachbarn mit einfließt, allerdings nur noch derjenigen mit gleicher Klassenzugehörigkeit:

$$q_0' = \frac{1}{N} \sum_{j=1}^N \frac{\sum_{i=1}^k d_{NN_{ji}^{\omega_i}}}{\sum_{i=1}^k d_{NN_{ji}}} \quad (2.12)$$

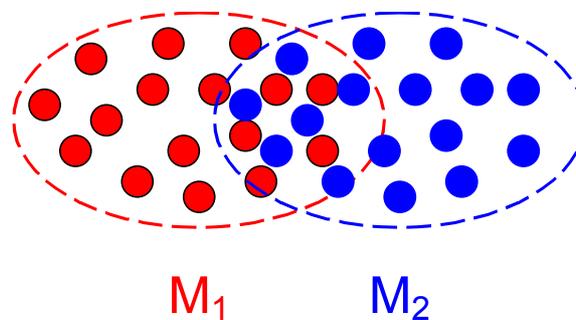
Ein völliger Verzicht auf die Berücksichtigung der Distanzen der  $k$  nächsten Nachbarn zu  $\mathbf{v}_j$  führt auf die einfachste Formulierung des Überlappingsgrades:

$$q_0'' = \frac{1}{N} \sum_{j=1}^N \frac{k^{\omega_j}}{k} \quad (2.13)$$

Dieses am wenigsten rechenzeitintensive Überlappingsmaß gibt also nur noch den prozentualen Anteil der Nachbarn gleicher Klassenzugehörigkeit  $k^{\omega_i}$  an der Menge der  $k$  nächsten Nachbarn an.

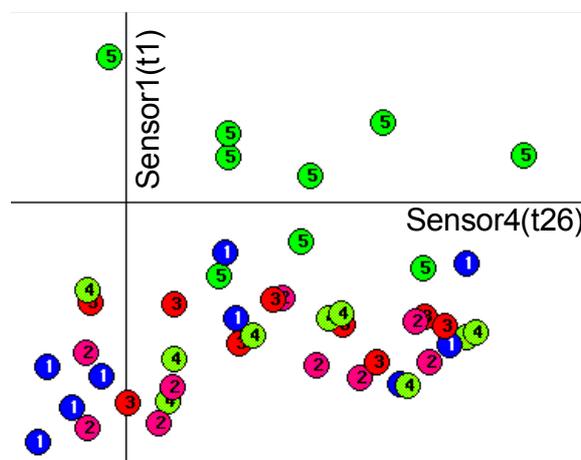


*Abb. 2.6: Leichte Separierbarkeit bei Überlappungsfreiheit*



*Abb. 2.7: Ungeeignete, weil zu Mengenüberlappung führende Merkmale*

Die folgenden beiden Abbildungen zeigen Beispiele für verschiedene Merkmalsgüten sensorischer Daten. Ein Merkmalspunkt entspricht einem Sensorwert zum Zeitpunkt  $t$ , die Zahlen spiegeln unterschiedliche Stoffklassen wieder:



*Abb. 2.8: Ungeeignete Merkmale (hoher Überlappungsgrad)*

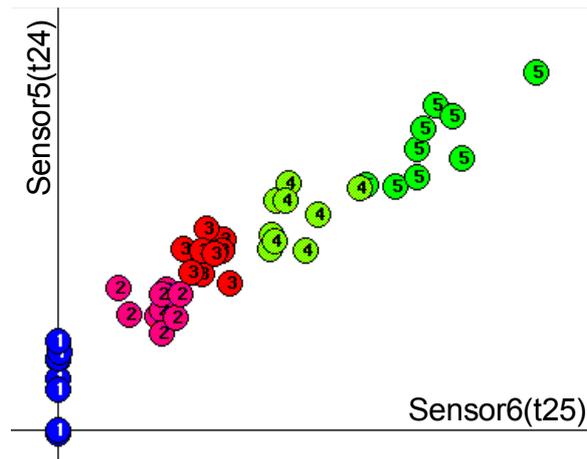


Abb. 2.9: Geeignete Merkmale (überlappungsfrei, 2D-Schnitt)

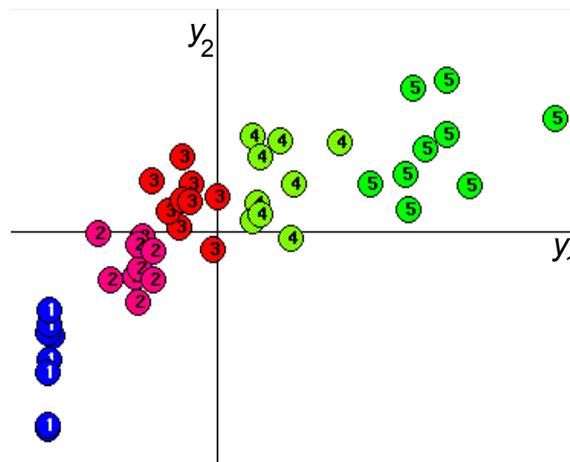


Abb. 2.10: Merkmale aus Abb 2.9 (Projektion mit NLM)

Bei multimodalen Verteilungen besteht die Gefahr, daß der Klassifikator überfordert wird, so daß es von Vorteil ist, diesen in die Merkmalsgütebeurteilung mit einzubeziehen.

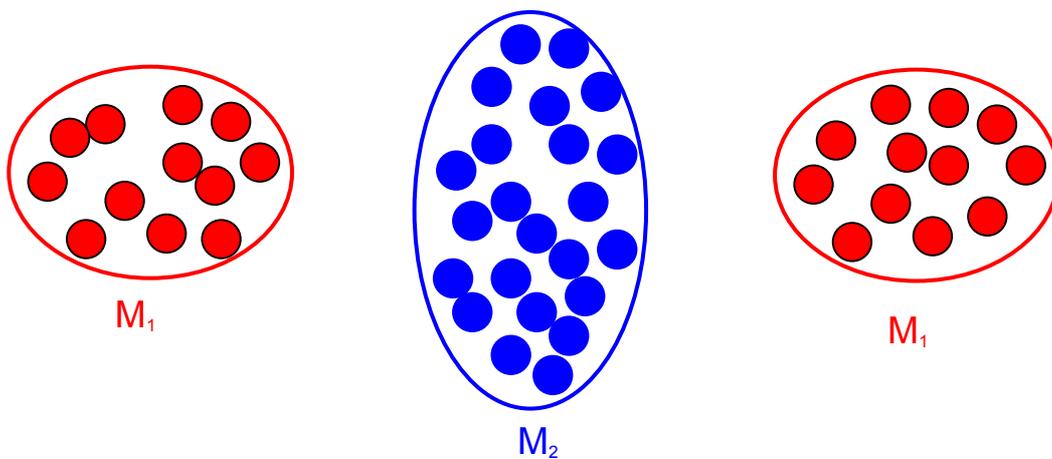


Abb. 2.11: Multimodale Verteilung

### 2.3.3.2 Klassifikationsrate bei vorgegebenem Klassifikator

Die Motivation für diese rechenintensivere Art der Merkmalsgütebewertung liegt in der Einbeziehung der Besonderheiten des Erkennungssystems. Für den Einsatz in einer SBS- bzw SFS-Selektionsstrategie eignet sich die Klassifikationsrate eines vorher festgelegten Klassifikators für eine hinreichend große Anzahl verschiedener Trainings- und Testmengen, die aus einer vorliegenden Datenbasis gebildet werden. Um sowohl Aussagen über Recall als auch Prediktion machen zu können, werden zunächst die zur Bildung der Trainingsmenge verwendeten Vektoren mit einer Wahrscheinlichkeit  $p$  aus der Gesamtmenge der Merkmalvektoren ausgewählt. Der mit dieser Teilmenge trainierte Klassifikator wird anschließend zur Klassifikation des gesamten Datensatzes verwendet. Diese Prozedur wird für jede Merkmalskombination  $m$  mal wiederholt. Zur Gütebestimmung kann entweder eine Mittelung der Klassifikationsraten über alle  $m$  Versuche durchgeführt oder das jeweils schlechteste Klassifikationsergebnis herangezogen werden (Worst-Case-Szenario). Die Klassifikationsrate wird dabei als das Verhältnis der korrekt klassifizierten Stichproben zur Gesamtstichprobenzahl definiert:

$$q_1 = \frac{\text{Korrekt klassifizierte Stichproben}}{\text{Gesamtstichprobenzahl}} \quad (2.14)$$

### 2.3.4 Klassifikations- und Projektionsalgorithmen

Transformationen in niederdimensionale Räume können zur visuellen Inspektion hochdimensionaler Datensätze eingesetzt werden. Zur Reduzierung des dabei auftretenden Informationsverlustes werden bestimmte Dateneigenschaften vorgegeben, die im Zielraum möglichst gut wiedergegeben werden sollen. Die „Erhaltungsgrößen“ der im folgenden beschriebenen Projektionsverfahren sind Varianz, Abstand und Topologie der Ausgangsdaten.

Projektionsalgorithmen dienen allgemein zur Merkmalsextraktion (s. 2.3.2). Bei ausreichend kleinen Zieldimensionen ( $\leq 3$ ) kann diese wiederum zur Visualisierung verwendet werden.

Bei den Klassifikationsalgorithmen unterscheidet man je nachdem, ob die Klassenzugehörigkeiten der Eingabevektoren beim Lernprozeß berücksichtigt werden oder nicht, zwischen überwacht (LVQ) und unüberwacht (SOM) lernenden Verfahren.

#### 2.3.4.1 Hauptkomponentenanalyse

Bei der Hauptkomponentenanalyse (auch Hauptachsentransformation, Principal-Component-Analysis, PCA) handelt es sich um eine häufig genutzte Technik zur Reduktion einer Vielzahl von Variablen auf einige wenige Komponenten mit dem Ziel einer Merkmalsextraktion oder zur Projektion hochdimensionaler Daten in die Ebene. Dazu nutzt sie die Redundanz aus, die in einer Korrelations- oder Kovarianzmatrix mit nichtdiagonalen Elementen ungleich Null zum Ausdruck kommt. Mit Hilfe der Hauptkomponentenanalyse werden aus der gegebenen Korrelations- oder Kovarianzmatrix sogenannte Hauptkomponenten extrahiert, zwischen denen keine Zusammenhänge bestehen, die also orthogonal zueinander sind. Je stärker die Korrelation zwischen den ursprünglichen Variablen ist, desto weniger Komponenten sind zur Erhaltung ihres Informationsgehaltes notwendig.

Ein  $n$ -dimensionaler Merkmalvektor  $\mathbf{x}_i = (x_{i1}, \dots, x_{in})$  diene zur Beschreibung einer Stichprobe (z.B. Leitwertänderungen von  $n$  Sensoren bei einem bestimmten Gasangebot). Nach der Durchführung von  $m$  Versuchen, lassen sich die  $m$  Merkmalsvektoren in einer sogenannten Beobachtungsmatrix zusammenfassen:

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_m \end{pmatrix} \quad (2.15)$$

Daraus läßt sich die Abweichungsmatrix  $\mathbf{M}$  durch komponentenweise Subtraktion der Spaltenmittelwerte berechnen:

$$\mathbf{M} = \mathbf{X} - \frac{1}{m}(\mathbf{1} \cdot \mathbf{1}^T \cdot \mathbf{X}), \quad \text{mit } \mathbf{1}^T = (1, \dots, 1) \quad (2.16)$$

Die Abweichungsquadrat- und -produktmatrix  $\mathbf{T}$  erhält man wie folgt:

$$\mathbf{T} = \mathbf{M}^T \cdot \mathbf{M} \quad (2.17)$$

$\mathbf{T}$  ist eine symmetrische Matrix, in deren Hauptdiagonale die Summen der Abweichungsquadrate stehen und deren nichtdiagonale Elemente von den Summen der Abweichungsprodukte der Variablen gebildet werden:

$$\mathbf{T} = \begin{pmatrix} t_{11} & \cdots & t_{1n} \\ \vdots & t_{jj} & \vdots \\ t_{n1} & \cdots & t_{nn} \end{pmatrix} \quad (2.18)$$

mit

$$t_{jj} = \sum_{i=1}^m (x_{ij} - \bar{x}_j)^2, \quad j = 1, \dots, n, \quad \text{und} \quad t_{jk} = \sum_{i=1}^m (x_{ij} - \bar{x}_j) \cdot (x_{ik} - \bar{x}_k), \quad k = 1, \dots, n \quad (2.19)$$

Daraus ergibt sich die Kovarianzmatrix  $\mathbf{S}$  zu

$$\mathbf{S} = \frac{1}{m-1} \mathbf{T} \quad (2.20)$$

Deren Eigenvektoren bilden den Transformationskern für die lineare Transformation in einen Zielraum, in dem die Merkmale dekorreliert sind, d.h. in dem nur die Hauptdiagonale der Kovarianzmatrix besetzt ist. Anschaulich wird durch die Transformation eine Drehung des Koordinatensystems in die Hauptachsen der Verteilung der Merkmale vorgenommen. Die Eigenwerte  $\lambda_i$  der Kovarianzmatrix sind die Varianzen der entsprechenden Hauptkomponenten. Für die Gesamtvarianz gilt

$$Sp(\mathbf{S}) = \sum_{i=1}^n \lambda_i \quad (2.21)$$

Der Anteil an der Gesamtvarianz, den eine Hauptkomponente  $y_i$  auf sich vereinigt, ergibt sich demnach zu

$$Var = \frac{\lambda_i}{\sum_{j=1}^n \lambda_j} \quad (2.22)$$

Für eine Transformation der ursprünglichen Daten in die Ebene wählt man nun die beiden Hauptkomponenten aus, deren Anteil an der Gesamtvarianz am größten ist. Die Dimensionsreduktion erfolgt also einfach durch Weglassen der weniger relevanten Eigenvektoren aus der Transformationsmatrix  $\mathbf{R}$ . Diese wird lediglich von den beiden Eigenvektoren  $\psi_1$  und  $\psi_2$  der Kovarianzmatrix gebildet, die zu den beiden größten Eigenwerten gehören:

$$\mathbf{R} = (\psi_1, \psi_2) \quad (2.23)$$

Die transformierte Beobachtungsmatrix

$$\mathbf{Y} = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_m \end{pmatrix} \quad (2.24)$$

mit den entsprechend transformierten Vektoren  $\mathbf{y}_i = (y_{i1}, \dots, y_{in})$  erhält man dann durch Rechtsmultiplikation der ursprünglichen Beobachtungsmatrix mit  $\mathbf{R}$ :

$$\mathbf{Y} = \mathbf{X} \cdot \mathbf{R} \quad (2.25)$$

#### 2.3.4.2 Sammon Nonlinear Mapping (NLM)

Während die Hauptachsentransformation eine lineare Projektion unter dem Kriterium der Varianzerhaltung durchführt, handelt es sich bei NLM [Sam 69] um eine nichtlineare dimensionsreduzierende Abbildung unter der Vorgabe bestmöglicher Erhaltung der Vektorabstände im Ursprungsraum. Sie wird überwiegend zur Visualisierung hochdimensionaler Datensätze verwendet und eignet sich aufgrund ihrer distanzerhaltenden Eigenschaft hervorragend zur optischen Überprüfung von Merkmalvektoren im Zusammenhang mit nächste-Nachbar-Klassifikatoren.

Der Grundgedanke zur Konstruktion der Abbildungsvorschrift ist außerordentlich simpel: Ausgehend von einer Zufallsinitialisierung der Zielvektoren in der Ebene wird eine Fehlerfunktion gebildet, die ein Maß für die Abweichung der Abstände jener von denen der Ausgangsvektoren darstellt.

Dieser Abbildungsfehler wird iterativ durch den Einsatz eines Gradientenabstiegsverfahrens minimiert. Dessen unerwünschte Eigenschaft, bereits in lokalen Minima der Fehlerfläche „stecken-zubleiben“ läßt allerdings ein Ausweichen auf modernere Verfahren wie zum Beispiel *Simulated Annealing* ratsam erscheinen.

Der Abbildungsfehler nach der  $m$ -ten Iteration beträgt für  $N$  Vektoren bei Verwendung der Euklidischen Norm:

$$E(m) = \frac{1}{c} \sum_{j=1}^N \sum_{i=j+1}^N \frac{(d_{X_{ij}} - d_{Y_{ij}}(m))^2}{d_{X_{ij}}} \quad (2.26)$$

mit

$$c = \sum_{j=1}^N \sum_{i=j+1}^N d_{X_{ij}} \quad (2.27)$$

Dabei bezeichnet  $d_{X_{ij}}$  den Abstand der  $M$ -dimensionalen Vektoren im Ursprungsraum und  $d_{Y_{ij}}(m)$  den Abstand der  $d$ -dimensionalen Repräsentanten nach der  $m$ -ten Iteration im Zielraum:

$$d_{X_{ij}} = \sqrt{\sum_{q=1}^M (x_{iq} - x_{jq})^2} \quad \text{und} \quad d_{Y_{ij}}(m) = \sqrt{\sum_{q=1}^d (y_{iq}(m) - y_{jq}(m))^2} \quad (2.28)$$

Der Gradient der Fehlerfunktion wird zur Ermittlung der neuen Koordinaten der  $N$  Repräsentantenvektoren  $y_i$  herangezogen:

$$y_{iq}(m+1) = y_{iq}(m) - \text{MF} \cdot \Delta y_{iq}(m) \quad (2.29)$$

mit

$$\Delta y_{iq}(m) = \frac{\frac{\partial E(m)}{\partial y_{iq}(m)}}{\left| \frac{\partial^2 E(m)}{\partial y_{iq}^2(m)} \right|} \quad (2.30)$$

Der *Magic-Factor* MF wurde von Sammon empirisch zu  $\text{MF} \approx 0.3$  bis  $0.4$  bestimmt [Sam 69] und stellt ein Äquivalent zur Lernrate neuronaler Algorithmen dar. Durch Bilden der partiellen Ableitungen der Fehlerfunktion erhält man

$$\frac{\partial E}{\partial y_{iq}} = -\frac{2}{c} \sum_{\substack{j=1 \\ j \neq i}}^N \left( \frac{d_{X_{ij}} - d_{Y_{ij}}}{d_{X_{ij}} d_{Y_{ij}}} \right) (y_{iq} - y_{jq}) \quad (2.31)$$

$$\frac{\partial^2 E}{\partial y_{iq}^2} = \frac{-2}{c} \sum_{\substack{j=1 \\ j \neq i}}^N \frac{1}{d_{X_{ij}} d_{Y_{ij}}} \left[ d_{X_{ij}} - d_{Y_{ij}} - \frac{(y_{iq} - y_{jq})^2}{d_{Y_{ij}}} \left( 1 + \frac{d_{X_{ij}} - d_{Y_{ij}}}{d_{Y_{ij}}} \right) \right] \quad (2.32)$$

Wie bereits erwähnt, kann das Gradientenverfahren zu einem Nebenminimum hin konvergieren, was sich in einem hohen Endwert des Abbildungsfehlers widerspiegelt. Sammon bezeichnet Scatter-Plots mit einem Abbildungsfehler von deutlich mehr als 10% als ungeeignet, um noch etwas über die tatsächliche Struktur der Ursprungsdaten auszusagen [Sam 69]. Zur Erzielung eines möglichst geringen Abbildungsfehlers ist es hilfreich, mit verschiedenen Anfangsinitialisierungen zu experimentieren. Bleibt der Fehler dennoch außergewöhnlich hoch, so läßt dies auf eine hohe *intrinsische* Dimensionalität der Daten schließen.

### 2.3.4.3 Radiale-Basisfunktionen-Netze (RBF-Netze)

RBF-Netze besitzen nur eine Schicht verdeckter Neuronen mit radialsymmetrischen Aktivierungsfunktionen, die die Basisfunktionen eines Funktionensystems zur Approximation mehrdimensionaler Funktionen anhand vorgegebener Stützstellen, die im einfachsten Fall von den Trainingsmustern vorgegeben werden, darstellen. Gegenüber anderen Netzwerkmodellen besitzen RBF-Netze folgende Vorteile:

- Die Aktivierungsfunktionen nehmen nur in der Nähe der Referenzpunkte große Werte an. Für Testmuster außerhalb des von den Trainingsmustern abgedeckten Bereiches liefert das Netz nur geringe Aktivierungen, im Gegensatz zu Netztypen mit sigmoiden

Aktivierungsfunktionen, bei denen sich außerhalb des trainierten Bereichs oftmals Fehlklassifikationen ergeben (lokale versus globale Approximation).

- Die einfache Struktur der RBF-Netze gestattet den Einsatz unkomplizierter Trainingsverfahren bis hin zur direkten (d.h. nichtiterativen) Berechnung der Gewichte. Dies führt gegenüber aufwendiger zu trainierenden Netzwerktypen (MLP mit Backpropagation) zu deutlich verkürzten Trainingszeiten sowie einer wirksamen Vermeidung von Nebenminima-Fallen.
- Der einschichtige Aufbau führt zu einer hohen Transparenz (Visualisierbarkeit!) des fertigen Netzes; die Wirkungsweise kann (gegebenenfalls nach Transformation aus einem höherdimensionalen Merkmalsraum) geometrisch dargestellt werden.

Auf einen speziellen Vertreter dieses Typs wird im Diskussionsteil näher eingegangen.

#### 2.3.4.4 Lernende Vektorquantisierung (LVQ)

Vektorquantisierung (VQ) ist ein klassisches Signal-Approximationsverfahren, das eine diskretisierte Näherung der Wahrscheinlichkeitsdichtefunktion  $p(\mathbf{x})$  einer stochastischen Größe  $\mathbf{x} \in \mathbb{R}^n$  mit Hilfe einer endlichen Anzahl sogenannter Codebook-Vektoren  $\mathbf{w}_i$  vornimmt.

Bei der von Teuvo Kohonen entwickelten *lernenden Vektorquantisierung* wird die Platzierung dieser Codebook-Vektoren anhand eines überwachten Lernverfahrens vorgenommen, was den Namen dieses Konzeptes motiviert.

Im Gegensatz zur unüberwachten Clusterung durch VQ, bei der eine Minimierung des Quantisierungsfehlers angestrebt wird, handelt es sich bei Kohonens lernenden Vektorquantisierungsverfahren um Methoden zur Bestimmung optimaler Entscheidungsgrenzen für Klassifikationsaufgaben. Je nachdem, nach welchen Regeln die Codebook-Vektoren des Klassifikators unter Berücksichtigung der Klassenverteilung im Eingaberaum platziert werden, unterscheidet man zwischen drei Grundverfahren (LVQ1, LVQ2.1, LVQ3) und einer Modifikation (OLVQ1).

##### 2.3.4.4.1 LVQ1

In der Lernphase werden dem Klassifikator sämtliche Merkmalvektoren der Trainingsmenge in beliebiger Reihenfolge präsentiert. Jeder Eingabevektor  $\mathbf{x}$  wird dabei parallel mit allen Codebook-Vektoren  $\mathbf{w}_i$  (Neuronen) verglichen. Ein Gewinnerneuron (Index  $c$ ) wird anhand eines vorher festgelegten Ähnlichkeitsmaßes (meist Euklidische Norm der Vektordifferenz  $\mathbf{x} - \mathbf{w}_i$ , im Falle normierter Eingabeneuronen auch Skalarproduktnorm  $\langle \mathbf{x} | \mathbf{w}_i \rangle$ ) ermittelt:

$$\|\mathbf{x} - \mathbf{w}_c\| = \min_i(\|\mathbf{x} - \mathbf{w}_i\|), \text{ bzw. } c = \arg \min_i(\|\mathbf{x} - \mathbf{w}_i\|) \quad (2.33)$$

Gleichung (2.33) wird auch bei der Klassifikation nach Abschluß des Trainings angewandt. Insofern handelt es sich bei LVQ um nächste-Nachbar-Klassifikatoren, allerdings mit dem Unterschied, daß nicht einfach alle Trainingsmuster abgespeichert werden, sondern eine zahlenmäßig begrenzte Stellvertretermenge anhand eines Lernverfahrens gebildet wird. Um dem Ziel eines minimalen Klassifikationsfehlers gerecht zu werden, wird von Kohonen folgende Vorgehensweise bei der Adaption der Codebook-Vektoren vorgeschlagen [Koh 95]:

Das Gewinnerneuron  $c$ , also das, dessen Gewichtsvektor  $\mathbf{w}_c$  dem Eingabevektor  $\mathbf{x}$  am ähnlichsten ist, wird je nach Klassenzugehörigkeit entweder durch Addition eines durch die Lern-

rate  $\alpha$  vorgegebenen Bruchteils des Differenzvektors  $\mathbf{x} - \mathbf{w}_c$  zu diesem hingezogen (gleiche Klasse) oder weiter abgestoßen (verschiedene Klassen):

$$\mathbf{w}_c(t+1) = \begin{cases} \mathbf{w}_c(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{w}_c(t)] & \text{falls Klasse}(\mathbf{w}_c) = \text{Klasse}(\mathbf{x}) \\ \mathbf{w}_c(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{w}_c(t)] & \text{falls Klasse}(\mathbf{w}_c) \neq \text{Klasse}(\mathbf{x}) \end{cases} \quad (2.34)$$

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) \quad \text{für alle } j \neq c \quad (2.35)$$

Die Lernrate  $\alpha(t) \in (0, 1)$  wird häufig so gewählt, daß sie ausgehend von einem nicht zu großen Startwert (kleiner als 0,1) im Laufe des Trainings linear auf Null abfällt.

#### 2.3.4.4.2 OLVQ1

Der ausgeschriebene Name dieses Verfahrens, Optimized-Learning-Rate LVQ1, weist bereits darauf hin, daß es sich um eine Weiterentwicklung von LVQ1 mit dem Ziel schnellstmöglicher Konvergenz handelt. Im Unterschied zu LVQ1 wird jedem Neuron eine eigene Lernrate zugewiesen, deren „optimaler“ Wert rekursiv aus den zeitlich vorangegangenen ermittelt wird:

$$\alpha_c(t) = \frac{\alpha_c(t-1)}{1 + s(t)\alpha_c(t-1)} \quad (2.36)$$

Dabei nimmt  $s(t)$  bei einer korrekten Klassifikation den Wert +1 an, im anderen Fall -1. Die Gewichtsadaptionen unterscheiden sich bis auf die individuellen Lernraten nicht von LVQ1:

$$\mathbf{w}_c(t+1) = \begin{cases} \mathbf{w}_c(t) + \alpha_c(t)[\mathbf{x}(t) - \mathbf{w}_c(t)] & \text{falls Klasse}(\mathbf{w}_c) = \text{Klasse}(\mathbf{x}) \\ \mathbf{w}_c(t) - \alpha_c(t)[\mathbf{x}(t) - \mathbf{w}_c(t)] & \text{falls Klasse}(\mathbf{w}_c) \neq \text{Klasse}(\mathbf{x}) \end{cases} \quad (2.37)$$

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) \quad \text{für alle } j \neq c \quad (2.38)$$

Da (2.36) auch ein Anwachsen der Lernrate gestattet, muß sichergestellt werden, daß dies nicht zu Werten größer als 1 führt. Diese Aufgabe muß vom einbettenden Algorithmus wahrgenommen werden, in der Implementierung von [Koh+95] wird er z.B. immer unterhalb des Ausgangswertes gehalten, so daß die Wahl relativ großer Startwerte (um 0,3) möglich ist.

#### 2.3.4.4.3 LVQ2.1

Im Unterschied zu den OLVQ1-Verfahren werden bei LVQ2.1 die *zwei* nächsten Nachbarn  $\mathbf{w}_i$  und  $\mathbf{w}_j$  des Eingabevektors ermittelt und nur dann eine Gewichtsadaption vorgenommen, wenn die folgenden drei Bedingungen zutreffen:

- $\mathbf{w}_i$  und  $\mathbf{w}_j$  gehören verschiedenen Klassen an ( $\omega_i \neq \omega_j$ )
- $\mathbf{x}$  gehört einer der beiden Klassen  $\omega_i$  oder  $\omega_j$  an
- $\mathbf{x}$  liegt in einem *Fenster* der relativen Breite  $b$  entlang der Mittelsenkrechten von  $\mathbf{w}_i$  und  $\mathbf{w}_j$

Seien  $d_i$  und  $d_j$  die euklidischen Abstände des Eingabevektors  $\mathbf{x}$  von den beiden nächsten Gewichtsvektoren  $\mathbf{w}_i$  und  $\mathbf{w}_j$ , dann ist Bedingung c) nach [Koh 95] genau dann erfüllt, wenn

$$\min\left(\frac{d_i}{d_j}, \frac{d_j}{d_i}\right) > s, \quad \text{mit} \quad s = \frac{1-b}{1+b} \quad (2.39)$$

gilt. Kohonen empfiehlt relative Fensterbreiten von  $b = 0,2$  bis  $b = 0,3$  [Koh 95]. Wenn die Bedingungen a) bis c) erfüllt sind, erfolgt eine Anpassung der Gewichtsvektoren der beiden nächsten Nachbarneuronen  $\mathbf{w}_i$  und  $\mathbf{w}_j$ , wobei in (2.40) eine Klassenzugehörigkeit von  $\mathbf{x}$  zu  $\omega_i$  angenommen wurde:

$$\begin{aligned} \mathbf{w}_i(t+1) &= \mathbf{w}_i(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \\ \mathbf{w}_j(t+1) &= \mathbf{w}_j(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{w}_j(t)] \end{aligned} \quad (2.40)$$

LVQ2.1 basiert auf der Idee einer Optimierung der Codebook-Vektor-Abstände von den Klassengrenzen, ohne die Verteilungsdichte der Eingabevektoren zu berücksichtigen. Es besteht keine Garantie bezüglich der Konvergenz des Verfahrens, weshalb es in erster Linie zum „Fine-Tuning“ (wenige Iterationen bei niedrig gewählter Lernrate) bereits vortrainierter Klassifikatoren eingesetzt wird.

#### 2.3.4.4.4 LVQ3

LVQ3 stellt eine Erweiterung von LVQ2.1 dar, bei der zur Wahrung der Eingabevektorverteilungsdichte auch die Codebook-Vektoren im Innern eines Klassengebietes verschoben werden (wie bei LVQ1). Wie bei LVQ2.1 werden die Gewichte der beiden am nächsten zu  $\mathbf{x}$  gelegenen Neuronen angepaßt, wenn die Bedingungen a) bis c) aus 2.3.4.4.3 erfüllt sind:

$$\begin{aligned} \mathbf{w}_i(t+1) &= \mathbf{w}_i(t) + \alpha(t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \\ \mathbf{w}_j(t+1) &= \mathbf{w}_j(t) - \alpha(t)[\mathbf{x}(t) - \mathbf{w}_j(t)] \end{aligned} \quad (2.41)$$

Gehören  $\mathbf{w}_i$  und  $\mathbf{w}_j$  beide derselben Klasse wie  $\mathbf{x}$  an, so werden beide im Unterschied zu LVQ2.1 ebenfalls angepaßt:

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \varepsilon\alpha(t)[\mathbf{x}(t) - \mathbf{w}_k(t)], \quad k \in \{i, j\} \quad (2.42)$$

Brauchbare Werte für  $\varepsilon$  wurden empirisch zu 0,1 bis 0,5 bestimmt [Koh+95]. Der optimale Wert hängt von der relativen Fensterbreite  $b$  ab und sollte für engere Fenster entsprechend kleiner gewählt werden. Die Modifikationen gegenüber LVQ2.1 bewirken, daß dieser Algorithmus selbststabilisierend ist, d.h. einmal optimal eingestellte Gewichtsvektoren werden bei fortgesetztem Lernen nicht mehr verändert.

#### 2.3.4.4.5 Allgemeine Hinweise zur Anwendung von LVQ

Die erzielbare Genauigkeit bei der Lösung einer Klassifikationsaufgabe mit LVQ hängt auf der Seite des Klassifikators von folgenden Faktoren ab:

- der Zahl der für jede Klasse eingesetzten Codebook-Vektoren, sowie deren Anfangsinitialisierung
- der Art des verwendeten Lernverfahrens, der richtigen Lernrate, sowie einem passenden Abbruchkriterium zur Beendigung des Lernvorgangs

Kohonen empfiehlt selbst im Falle ungleicher a priori Wahrscheinlichkeiten für jede Klasse mit der gleichen Anzahl von Codebook-Vektoren zu beginnen. Eine Obergrenze für deren Anzahl ist nach [Koh+95] nur durch die Begrenzungen der Computerhardware gegeben. Nach den Erfahrungen des Autors kann eine zu große Codebook-Vektoranzahl bei nichtidealen Trainingsdaten allerdings auch zum „Auswendiglernen“ von Artefakten führen. Um die Entstehung von sog. „toten“ Neuronen zu verhindern, also solchen, die aufgrund ihrer Lage von keinem der Eingabevektoren aktiviert werden, wählt man zweckmäßigerweise zur Anfangsinitialisierung Vektoren aus der Trainingsmenge aus.

Kohonen empfiehlt, das Klassifikatortraining immer mit OLVQ1 zu beginnen, das aufgrund der optimierten individuellen Lernraten im Schnitt nur 30 bis 50 Lernschritte pro Codebook-Vektor zum Erreichen seiner asymptotischen Erkennungsgenauigkeit benötigt. Häufig reicht dieses Training bereits aus, anderenfalls kann der Lernvorgang mit LVQ1, LVQ2.1 oder LVQ3 fortgesetzt werden.

Allerdings sollte das Training nicht über zu viele Iterationszyklen weitergeführt werden (Richtwert sind das 50- bis 200-fache der Anzahl der Codebook-Vektoren), da es sonst ähnlich wie im Falle einer zu großen Anzahl von Codebook-Vektoren dazu kommen kann, daß sich die Codebook-Vektoren zu sehr an die Trainingsmenge anpassen und die Erkennungsrate auf einer zur dieser disjunkten Testmenge wieder geringer wird (sog. „Übertrainieren“).

#### 2.3.4.5 Self-Organizing Maps

Die Self-Organising Maps (SOM, auch Self-Organizing Feature Maps, SOFM, Kohonen-Netze oder selbstorganisierende Karten) können als eine Erweiterung von LVQ unter Einbeziehung einer lokalen Nachbarschaftsbeziehung betrachtet werden, die dazu führt, daß benachbarte Neuronen auf ähnliche Reize reagieren. Das Konzept ist neurophysiologisch motiviert und orientiert sich an Beobachtungen der Organisation von Neuronenschichten im Cortex des Gehirns. Der Umstand, daß die Ausbildung dieses Ordnungszustandes nur auf der immanenten Struktur der eingehenden Signale basiert legitimiert den Terminus „selbstorganisierend“. Da eine Signalähnlichkeit der Eingangsmuster unter Berücksichtigung ihrer Verteilungsdichte in eine Lagenachbarschaft jeweils erregter Neuronen umgesetzt wird, kann die Karte zur topologieerhaltenden (nichtlinearen) Projektion hochdimensionaler Eingangsdaten in die Ebene eingesetzt werden. Eine solche Abbildung kann sehr gut zur Visualisierung und zum Auffinden von Ähnlichkeitsbeziehungen in einem hochdimensionalen Eingaberaum verwendet werden.

Die Neuronen der Kohonen-Netze sind meist in Form eines zweidimensionalen Gitters mit quadratischer oder hexagonaler Nachbarschaftsbeziehung angeordnet. Diese bewirkt, daß sich die Karte in der Trainingsphase wie ein elastisches Gitternetz verhält, bei dem Lageveränderungen eines Knotenpunktes auch die umliegenden Knoten beeinflussen.

Im Gegensatz zu den LVQ-Lernverfahren, bei denen die Klassenzugehörigkeiten der Trainingsvektoren im voraus bekannt sein müssen, vollzieht sich die Organisation der Kohonen-Karten unüberwacht. Auch hier werden beim Training die n-dimensionalen Eingabevektoren parallel mit allen Gewichtsvektoren des Netzes in einer vorher festgelegten Metrik verglichen und das Gewinnerneuron ermittelt:

$$\|\mathbf{x} - \mathbf{w}_c\| = \min_i(\|\mathbf{x} - \mathbf{w}_i\|), \text{ bzw. } c = \arg \min_i(\|\mathbf{x} - \mathbf{w}_i\|) \quad (2.43)$$

Allerdings wird nicht nur der Gewichtsvektor des Gewinnerneurons verändert, sondern auch die Gewichtsvektoren der im Gitter benachbarten Neuronen. Die Definition der Nachbar-

schaft erfolgt anhand der Gittertopologie, d.h. die tatsächlichen Abstände der Gewichtsvektoren im Eingaberaum spielen hier keine Rolle. Die Adaption der Gewichtsvektoren erfolgt gemäß

$$\mathbf{w}_i(t+1) = \mathbf{w}_i(t) + \alpha(t)h_{ci}(t)[\mathbf{x}(t) - \mathbf{w}_i(t)] \quad (2.44)$$

Dabei bezeichnet  $\alpha(t)$  wieder die in der Regel kontinuierlich mit der Zeit abnehmende Lernrate und  $h_{ci}(t)$  die über dem Neuronengitter definierte Distanzfunktion (neighborhood kernel), die für Form und Größe der Nachbarschaft verantwortlich ist. Meist beginnt man mit einer sehr großen Umgebung, die z.B. alle Neuronen des Gitters umfaßt, und reduziert diese mit fortschreitender Zeit. Ein Beispiel für eine häufig verwendete Distanzfunktionen ist die sog. Bubble-Funktion:

$$h_{cj,bubble}(t) = \begin{cases} 1 & \text{falls } j \in N_c(t) \\ 0 & \text{sonst} \end{cases} \quad (2.45)$$

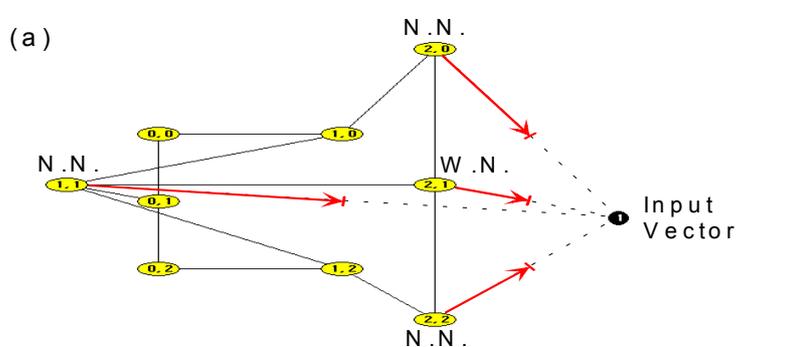
Hier gibt  $N_c(t)$  die Größe der momentanen Nachbarschaft um das Gewinnerneuron  $c$  herum an. Die zweite im Rahmen dieser Arbeit verwendete Funktion ist die normalisierte Gaußfunktion, deren Varianz  $\sigma(t)$  die Breite des Nachbarschaftskerns definiert:

$$h_{cj,gauss}(t) = \frac{1}{\sigma(t)\sqrt{2\pi}} \cdot e^{-\frac{\|\mathbf{r}_c - \mathbf{r}_j\|^2}{2\sigma^2(t)}} \quad (2.46)$$

Im Verlauf des Lernvorgangs (Größenordnung  $10^4$ - $10^6$  Iterationszyklen gemäß (2.44)) nimmt die Karte eine Approximation der Dichtefunktion der Eingangsvektoren durch die Gewichtsvektoren vor. Eine Aussage über die Qualität der Abbildung kann anhand des über alle  $N$  Eingabevektoren gemittelten Quantisierungsfehlers getroffen werden:

$$E = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{w}_c\| \quad (2.47)$$

Zur Illustration der Wirkungsweise von (2.44) ist in Abb 2.8 die Durchführung eines einzelnen Lernschrittes für ein  $3 \times 3$  Neuronen großes Netz mit Rechtecktopologie dargestellt. Die Zweidimensionalen Gewichtsvektoren des Netzes sind als Punkte im Ortsraum zu sehen. Die Lernrate für den dargestellten Adaptionsschritt beträgt  $\alpha = 0,5$ ; die Nachbarschaftsfunktion ist vom Typ „bubble“ (2.45) mit Nachbarschaftsradius 1 und erstreckt sich nur auf die vier unmittelbar verbundenen Nachbarn.

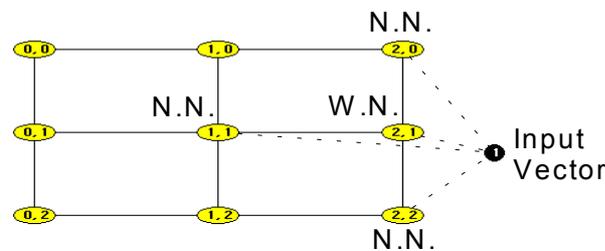


**Abb 2.8a:** Kohonen-Lernschritt, Teil I

Während der Trainingsphase wird dem Netz ein Eingabevektor präsentiert (Abb. 2.8a). Das Neuron mit dem geringsten Abstand zum Eingabevektor ist das Gewinner-Neuron (W.N.) mit

Indexpaar (2,1). Dessen Gewichtsvektor wird mit dem mit der Lernrate multiplizierten Differenzvektor (halber Abstand) zum Eingabevektor hingezogen. Die gewählte Nachbarschaftsfunktion vom Typ Bubble hat bis zum Rand den Wert 1, daher werden die zur unmittelbaren Nachbarschaft gehörenden Neuronen (N.N.) mit den Indexpaaren (2,0), (1,1) und (2,2) ebenfalls auf halbe Strecke zum Eingabevektor hingezogen.

(b)



**Abb 2.8b:** Kohonen-Lernschritt, Teil II

In Abb 2.8b) ist zu erkennen, wie die Nachbarschaftsbeziehung zwischen den Neuronen (man beachte, daß  $h$  über dem Neuronengitter, d.h. über den Indexpaaren und nicht über dem Raum der Eingabevektoren aufgespannt ist) dafür sorgt, daß der in Abb. 2.8a) noch vorhandene topologische Defekt beseitigt wurde. Die Regel

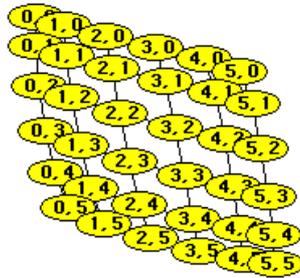
$$\text{Adaptionsvektor} = \text{Differenzvektor} \times \text{Lernrate} \times \text{Wert der Nachbarschaftsfunktion}$$

bewirkt, daß der Differenzvektor zu (1,1) durch den großen Wert der Nachbarschaftsfunktion maximal zur Korrektur beitragen kann (der reduzierende Faktor Lernrate betrifft alle Neuronen gleichermaßen).

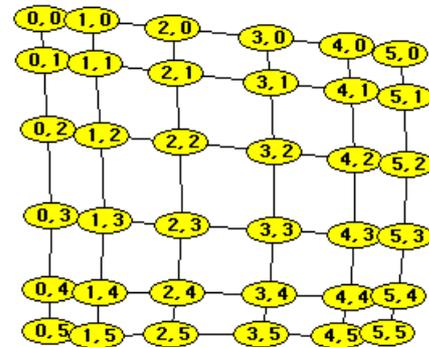
Die geordnete Entfaltung einer selbstorganisierenden Karte ist in Abb. 2.9 grafisch dargestellt. Der Nachbarschaftsradius spielt bei diesem Prozeß eine kritisch Rolle. Wird er am Anfang zu klein gewählt oder sinkt er zu schnell, so können sich zwar Teilbereiche des Netzes korrekt ordnen, die globale Ordnung des Netzes kann aber starke topologische Defekte aufweisen (Abb. 2.9f).



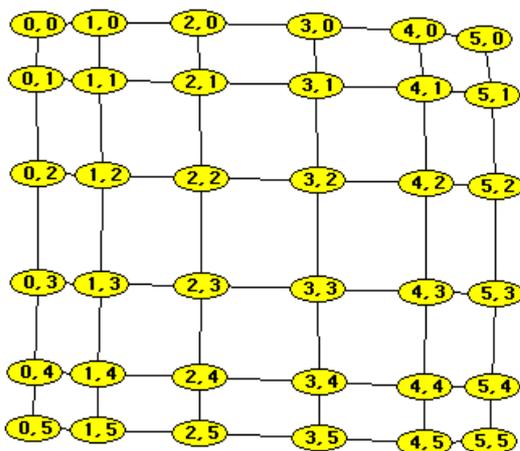
(a) 0 Iterationen



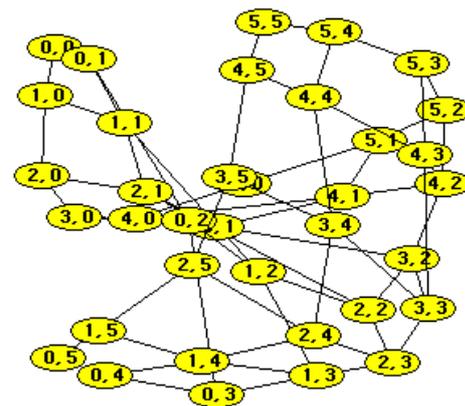
(b) 50 Iterationen



(c) 200 Iterationen



(d) 1000 Iterationen



(e) 10000 Iterationen



(f) SOM mit topologischem Defekt

**Abb. 2.9:** Zeitliche Entwicklung der Gewichtsvektoren der Neuronen einer quadratischen selbstorganisierenden Karte mit 6x6 Neuronen

Zur Vermeidung solcher Defekte sowie zur Beschleunigung des Lernvorgangs teilt man diesen meist in zwei Phasen auf. In der ersten, der Ordnungsphase, sorgt eine relativ große Lernrate in Verbindung mit einem großen Nachbarschaftsradius für eine rasche, defektfreie Entfaltung der Karte. In der zweiten, der Konvergenzphase, erhält das Netz die Gelegenheit, sich an die Feinheiten der Eingabevektorverteilung anzupassen. Dies geschieht durch die Wahl kleiner Nachbarschaftsradien und kleiner Lernraten.

Bei praktischen Anwendungen schließt sich an das unüberwacht erfolgte Training noch das sog. „Labeln“ der selbstorganisierenden Karte an (auch Kalibrierung genannt), um die Abbildung unbekannter Eingabemuster auf das Netz interpretieren zu können (z.B. für Klassifikationsaufgaben). Dazu muß ermittelt werden, auf welche Gebiete der Karte eine Anzahl vorher manuell (z.B. bezüglich Klassenzugehörigkeit) analysierter Trainingsvektoren abgebildet wird. Auf diese Art werden auf dem Neuronengitter Referenzpunkte geschaffen, die zur In-

terpretation unbekannter Daten durch Interpolation oder Extrapolation herangezogen werden können.

#### **2.3.4.6 MLP zur Datenkompression/Dimensionsreduktion**

Zur Durchführung einer Datenkompression unter der Vorgabe bestmöglicher Signalrekonstruierbarkeit können mit Erfolg Multilayer Perceptrons (MLP) eingesetzt werden. Durch die Präsentation identischer Eingangs- und Ausgangsvektoren in der Lernphase (z.B. mit Backpropagation) wird das Netz auf die Durchführung einer Identitätsabbildung trainiert. Eine im Sinne der Signalrekonstruierbarkeit verlustarme Dimensionsreduktion läßt sich nun dadurch erzielen, daß man in das Netz eine verdeckte Schicht einbringt, die genauso viele Neuronen aufweist, wie es der gewünschten Zieldimension entspricht. Durch diesen „Flaschenhals“ muß das Netz nun die Identitätsabbildung realisieren. Da der Lernprozeß auf eine Fehlerminimierung am Ausgang abzielt, werden in der verengten Schicht (die nach vollendetem Training zur Ausgangsschicht wird) genau jene Signalanteile extrahiert, die am stärksten zur Signalrekonstruierbarkeit beitragen.

Dieser Ansatz ist mit dem der Hauptkomponentenanalyse identisch, allerdings erlaubt die Verwendung eines MLP mit fünf (oder mehr) Schichten auch nichtlineare Transformationen.

### **2.3.5 Visualisierung hochdimensionaler Daten**

#### **2.3.5.1 Explorative Datenanalyse**

Der visuelle Wahrnehmungsapparat des Menschen ist in der Lage, selbst komplexe, einander ähnelnde Muster wiederzuerkennen und voneinander zu unterscheiden (z.B. Gesichtserkennung). Diese Fähigkeit kann bei der Klassifikation hochdimensionaler Datensätze von Nutzen sein, wenn es gelingt, die in Frage kommenden Muster möglichst repräsentativ auf einen Unterraum zu projizieren, der einer visuellen Inspektion zugänglich ist. Dabei können neben Informationen über die Separierbarkeit von Klassen anhand des Verlaufs ihrer Trennlinien sowie ihres Überlappungsgrades auch erste Eindrücke über die Auswirkungen von Parameteränderungen bei der Merkmalgenerierung gewonnen, sowie Ausreißer entdeckt werden.

Schließlich dient eine Visualisierung des Klassifikators dazu, dessen Eignung zur Lösung einer Erkennungsaufgabe zu beurteilen. Insbesondere im Zusammenhang mit realen Meßdaten, bei denen die Zeit der Aufnahme als Kriterium für die Bildung von Trainings- und Testmengen eine Rolle spielt (entspricht einer praxisnahen Vorgehensweise, da ein neuronaler Klassifikator immer mit Beispielen aus der Vergangenheit zur Voraussage künftiger Entscheidungen trainiert wird), können die Auswirkungen einer möglichen Drift auf das Klassifikationsergebnis direkt beobachtet werden.

Eine einfache Technik zur Visualisierung eines  $n$ -dimensionalen Datensatzes stellt seine Projektion über jeweils zwei Komponenten in die Ebene dar. Da es jedoch insgesamt  $n(n-1)/2$  derartige Seitenansichten gibt, werden bei großen  $n$  schnell die Grenzen der Übersichtlichkeit erreicht.

Daher versucht man, den Einfluß möglichst aller Komponenten in einer einzigen Projektion sichtbar zu machen, wobei die varianz-, distanz- oder topologieerhaltenden Transformationen aus 2.3.4.1, 2.3.4.2, 2.3.4.5 und 2.3.4.6 eingesetzt werden können. Da im Zusammenhang mit nächste-Nachbar-Klassifikatoren die Distanzen der Datenpunkte zu den Referenzvektoren von ausschlaggebender Bedeutung sind, eignet sich hier der Einsatz von 2.3.4.2 (NLM) in besonderer Weise.

## 2.3.5.2 Techniken zur Visualisierung selbstorganisierender Karten

### 2.3.5.2.1 Projektion des Neuronengitters auf den Eingaberaum

Die in 2.3.4.1, 2.3.4.2, 2.3.4.5 und 2.3.4.6 beschriebenen Transformationen können dazu verwendet werden, ein trainiertes Netz mitsamt der Punkte des Eingaberaumes in einer gemeinsamen Projektion darzustellen. Auf diese Weise wird sichtbar, in welchem Maße eine erfolgreiche Anpassung an die Verteilung der Eingabevektoren erfolgt ist.

### 2.3.5.2.2 Vektorlagekarten/Schnitte durch die Ebenen des Netzes

Hier betrachtet man das Kohonen-Netz als einen „Zahlenquader“, dessen horizontale Abmessungen durch Zahl und Anordnung der Neuronen bestimmt werden und dessen vertikale Ausdehnung durch die Anzahl  $n$  der Komponenten der Codebook-Vektoren gegeben ist. Jede der  $n$  Komponentenschichten kann dabei grafisch (z.B. mit Hilfe eines Farbverlaufs oder durch die Angabe der Absolutwerte der Einzelkomponenten) dargestellt werden. Auf diese Weise kann z.B. festgestellt werden, welche Komponenten der Eingabevektoren für eine Clusterbildung besonders signifikant sind.

### 2.3.5.2.3 Visualisierung der Klassenverteilung

Hierbei handelt es sich um eine „Draufsicht“ auf den in 2.3.5.2.2 beschriebenen Netzquader, bei dem die einzelnen Neuronen anhand ihrer Klassenzugehörigkeiten markiert sind. Diese Ansicht dient zur Kontrolle der vom Netz unüberwacht vorgenommenen Clusterung des Eingaberaums und kann zur Klassifikation unbekannter Daten verwendet werden. Voraussetzung für diese Darstellung ist eine vorhergehende Kalibrierung des Netzes.

### 2.3.5.2.4 Unified Distance Matrizen

Als Graustufen- oder 3-D-Grafik dargestellte Unified Distance Matrizen (kurz U-Matrizen) ermöglichen eine simultane Visualisierung von Neuronen-Topologie und Codebook-Vektor-Distanzen.

Gegeben sei ein Kohonen-Netz mit  $n \times m$  Neuronen, deren Lage im Gitter anhand eines Indexpaares eindeutig festgelegt ist:

...	...	x	...	...
...	(x-1,y-1)	(x,y-1)	(x+1,y-1)	...
y	(x-1,y)	(x,y)	(x+1,y)	...
...	(x-1,y+1)	(x,y+1)	(x+1,y+1)	...
...	...	...	...	...

Alle in der „Draufsicht“ mit einem Index gekennzeichneten Neuronen gehören zur unmittelbaren Nachbarschaft des Neurons  $(x, y)$ , aber nur die grau schattierten besitzen eine unmittelbare Verbindung zum zentralen Neuron. Zur Aufnahme der Abstandsinformationen wird die Karte „aufgeweitet“ und man erhält das Schema für die korrespondierende  $(2n+1) \times (2m+1)$  große U-Matrix:

...	...	...	...	...	...	...
...	$u_{i-2,j-2}$	$u_{i-2,j-1}$	$u_{i-2,j}$	$u_{i-2,j+1}$	$u_{i-2,j+2}$	...
...	$u_{i-1,j-2}$	$u_{i-1,j-1}$	$u_{i-1,j}$	$u_{i-1,j+1}$	$u_{i-1,j+2}$	...
...	$u_{i,j-2}$	$u_{i,j-1}$	$u_{i,j}$	$u_{i,j+1}$	$u_{i,j+2}$	...
...	$u_{i+1,j-2}$	$u_{i+1,j-1}$	$u_{i+1,j}$	$u_{i+1,j+1}$	$u_{i+1,j+2}$	...
...	$u_{i+2,j-2}$	$u_{i+2,j-1}$	$u_{i+2,j}$	$u_{i+2,j+1}$	$u_{i+2,j+2}$	...
...	...	...	...	...	...	...

Die Berechnung der (skalaren) Einträge  $u_{i,j}$  der U-Matrix aus den Codebook-Vektoren  $\mathbf{w}_{xy}$  der Kohonen-Karte geschieht wie folgt (Rechenvorschrift nur für doppelt gerahmten Bereich, Rest symmetrisch analog):

...	...	...	...
...	$u_{i,j} = \frac{1}{4}(u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j})$	$u_{i,j+1} = d(\mathbf{w}_{x,y}, \mathbf{w}_{x+1,y})$	...
...	$u_{i+1,j} = d(\mathbf{w}_{x,y}, \mathbf{w}_{x,y+1})$	$u_{i+1,j+1} = \frac{1}{2}[d(\mathbf{w}_{x,y}, \mathbf{w}_{x+1,y+1}), d(\mathbf{w}_{x+1,y}, \mathbf{w}_{x,y+1})]$	...
...	...	...	...

Die Mittelung über die vier direkten Nachbarn von  $u_{ij}$  zur Ermittlung eines Wertes für  $u_{ij}$  mutet zunächst etwas willkürlich an, da eine natürlichere Definition offensichtlich der Abstand des Neurons  $\mathbf{w}_{xy}$  zu sich selbst,  $u_{ij} = d(\mathbf{w}_{x,y}, \mathbf{w}_{x,y}) = 0$ , wäre. In der Tat unterliegt der Wert für  $u_{ij}$  einer gewissen Willkür; eine Mittelwertbildung vermeidet irreführende weiße Flecken in der Graustufendarstellung.

Ebenfalls mit einer gewissen Willkür behaftet ist der Wert für  $u_{i+1,j+1}$ . Zumindest dieser Makel tritt bei hexagonalen Karten nicht auf, da alle Nachbarn gleichberechtigt (d.h. direkt verbunden) sind:

...	...	...	...	...
...	$(x-1,y-1)$	$(x,y-1)$	$(x+1,y-1)$	...
...	$(x-1,y)$	$(x,y)$	$(x+1,y)$	...
...	$(x-1,y+1)$	$(x,y+1)$	$(x+1,y+1)$	...
...	...	...	...	...

Hier lautet die zugehörige U-Matrix:

...	...	...	...	...	...	...
$u_{i-2,j-2}$	$u_{i-2,j-1}$	$u_{i-2,j}$	$u_{i-2,j+1}$	$u_{i-2,j+2}$	...	...
...	$u_{i-1,j-2}$	$u_{i-1,j-1}$	$u_{i-1,j}$	$u_{i-1,j+1}$	$u_{i-1,j+2}$	...
...	$u_{i,j-2}$	$u_{i,j-1}$	$u_{i,j}$	$u_{i,j+1}$	$u_{i,j+2}$	...
...	$u_{i+1,j-2}$	$u_{i+1,j-1}$	$u_{i+1,j}$	$u_{i+1,j+1}$	$u_{i+1,j+2}$	...
$u_{i+2,j-2}$	$u_{i+2,j-1}$	$u_{i+2,j}$	$u_{i+2,j+1}$	$u_{i+2,j+2}$	...	...
...	...	...	...	...	...	...

mit

$$u_{i,j} = 1/6(u_{i-1,j} + u_{i-1,j+1} + u_{i,j-1} + u_{i,j+1} + u_{i+1,j} + u_{i+1,j+1})$$

$$u_{i,j+1} = d(\mathbf{w}_{x,y}, \mathbf{w}_{x+1,y})$$

$$u_{i+1,j} = d(\mathbf{w}_{x,y}, \mathbf{w}_{x,y+1})$$

$$u_{i,j+1} = d(\mathbf{w}_{x,y}, \mathbf{w}_{x+1,y+1})$$

Zu Visualisierungszwecken werden die in U-Matrizen enthaltenen Distanzinformationen meist in Graustufen oder Höheninformationen umgesetzt. Die dabei entstehende Landschaft aus Gebirgszügen und Tälern gibt einen sehr anschaulichen und schnellen Einblick in die Struktur der zum Training verwendeten Merkmaldaten: Täler weisen auf zusammenhängende Cluster hin, während hohe Gebirge zwischen ihnen auf eine leichte Trennbarkeit derselben schließen lassen.

### 2.3.5.3 Visualisierung zur Akquisition von Regelwissen

Die topologieerhaltende Abbildungseigenschaft selbstorganisierender Karten (ähnliche Daten werden auf benachbarte Neuronen abgebildet) kann in Verbindung mit der in 2.3.5.2.4 erläuterten Methode zur Neuronenabstandsvisualisierung dazu verwendet werden, verborgene Datenstrukturen sichtbar und als explizites Regelwissen verfügbar zu machen. Dazu interpretiert man die Codebook-Vektoren der trainierten Karte als Regelprämissen, während die Klassenzugehörigkeit des Gewinnerneurons den entsprechenden Konklusionsteil darstellt. Wird ein Neuron mit  $n$ -dimensionalem Gewichtsvektor  $\mathbf{w}$  und Klassenzugehörigkeit  $\omega_{\mathbf{w}}$  von einem Eingabevektor  $\mathbf{x}$  aktiviert, dann schließt man auf eine identische Klassenzugehörigkeit desselben. Regelbasiert läßt sich dieser Vorgang folgendermaßen beschreiben:

$$\text{WENN } [x_1 \approx w_1] \text{ UND } [x_2 \approx w_2] \dots \text{ UND } [x_n \approx w_n] \text{ DANN } [\omega_x = \omega_w]$$

Läßt man nichtsignifikante Komponenten weg (d.h. solche, deren Streuung innerhalb eines separierten Klassengebietes sich nur unwesentlich von der innerhalb der gesamten Netzebene unterscheidet), erhält man UND-Verknüpfungen mit weniger als  $n$  Prämissen. ODER-Verknüpfungen werden wie üblich durch zwei oder mehr UND-Verknüpfungen mit verschiedenen Prämissen und gleichem Konklusionsteil realisiert.

Die Vorgehensweise bei der „Wissensgewinnung“ sei exemplarisch anhand eines kleinen Beispiels erläutert. Die Aufgabe bestehe darin, eine regelbasierte Beschreibung von Lebensmitteln aus den Kategorien Fette & Öle, Fleisch, Fisch, Getreideprodukte, Gemüse, Früchte und Nüsse anhand von fünf aus einer Nährwert-Tabelle herausgegriffen Merkmalen zu finden [Wfd 90]. Bei diesen handelt es sich um den Gehalt an Eiweiß, Fett, Kohlenhydrate und Ballaststoffe in g pro 100g eßbare Substanz, sowie den Energiegehalt in kJ pro 100g Nahrungsmittel. Alle Merkmale wurden auf einen Wertebereich von [0,1] transformiert, wobei 0 für den geringsten Anteil und 1 für den höchsten Anteil innerhalb aller (insgesamt 56) ausgewählten Nahrungsmittel steht.

GRAF	CUCU PAPR TOMA	CHAM LETT SPIN		CODF	TROU	HARE CHIC VEAL BEEF
APRI CHER	ORAN STRA	REDC CABB		HERR	FILL PORK	HAM
POTA APPL BANA GRAP	PEAR	CAUL BEAN CARR ONIO		EEL	DUCK	MACK TUNA
		KIWI			MUTT	
BRE1	REYB BRE2					BUTT MARG MAYO
PAS1 RICE	OATS PAS2		PEAN HAZE ALMO	WALN		COCO OLIV SUNF LARD

Abb. 2.10: 6x7 Kohonen-Karte zur Nahrungsmittel-Clusterung

Abb. 2.10 zeigt ein mit diesen Daten trainiertes Kohonen-Netz mit 42 Neuronen. Die Karte faßt die Gruppen *Fleisch* und *Fisch* zusammen, ebenso *Früchte* und *Gemüse*. Diese können anhand der fünf ausgesuchten Merkmale nicht voneinander getrennt werden.

Die folgenden Abbildungen zeigen die korrespondierende U-Matrix (2D/3D-Visualisierung):

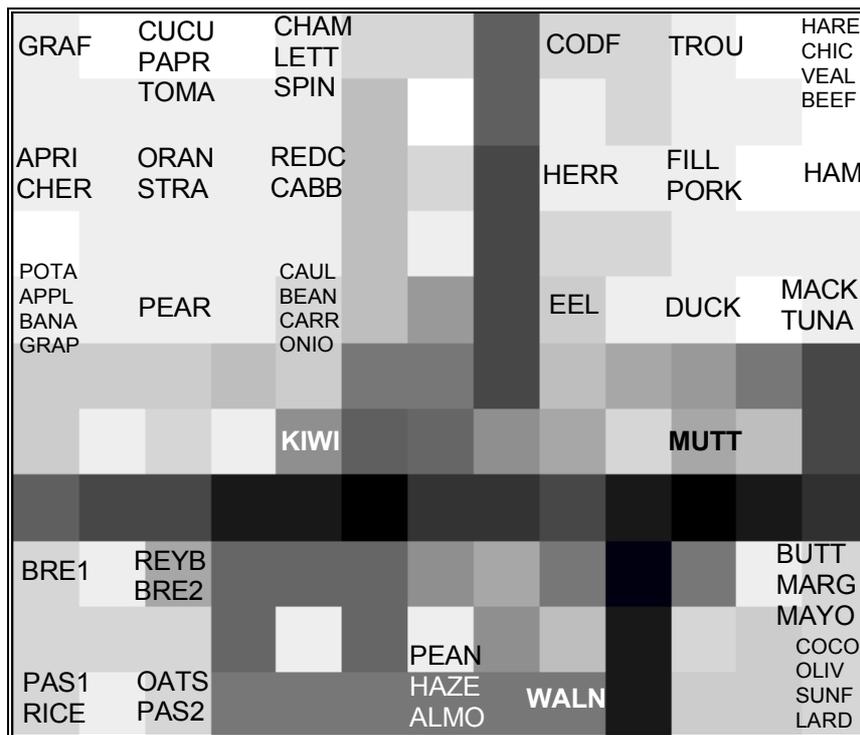


Abb. 2.11: U-Matrix von 2.10 in Graustufendarstellung

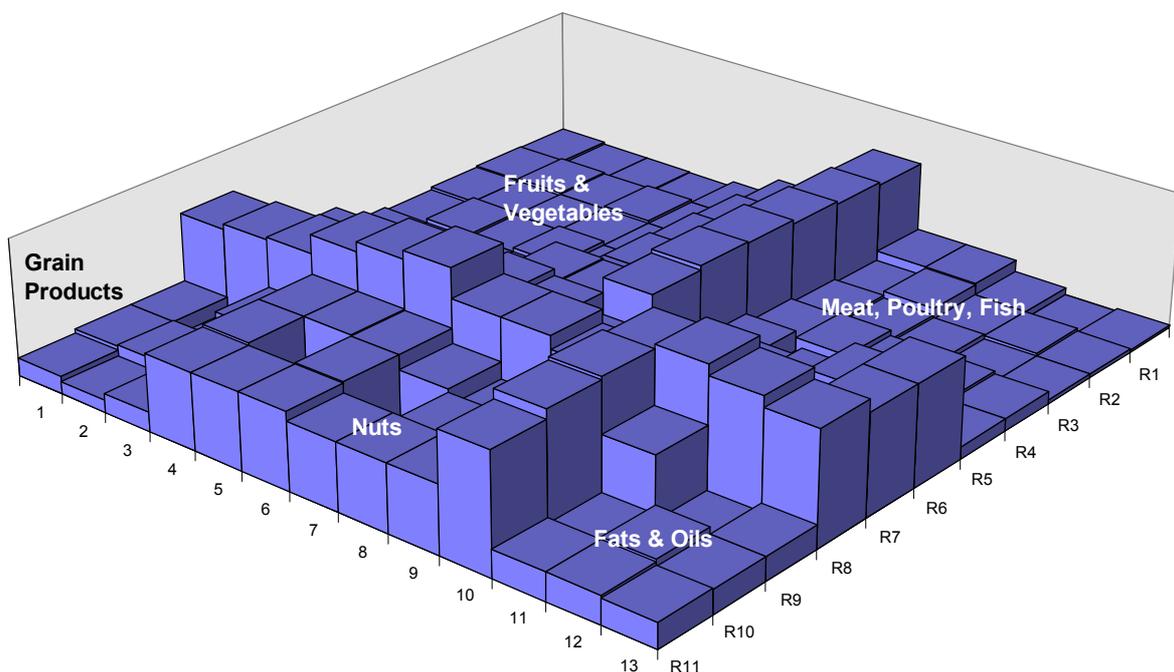


Abb. 2.12: U-Matrix von 2.10 in „Mauerdarstellung“

In der U-Matrix Darstellung Abb. 2.11 und Abb. 2.12 läßt sich die Bildung von fünf wohlseparierten Clustern erkennen. Die Kenntnis über deren Lage kann in Verbindung mit den fünf Ebenendarstellungen zur Regelableitung genutzt werden, z.B.:

WENN [*Eiweißgehalt = niedrig*] UND [*Fettgehalt = hoch*] UND [*Kaloriengehalt = hoch*]  
DANN [*Nahrungsmittelkategorie = Fette & Öle*]

Die Implementierung solcher linguistischer Regeln kann mit Hilfe eines Fuzzy-basierten Algorithmus erfolgen.

### 2.3.6 Fuzzy Logik

Die 1965 von Lofti Zadeh formulierte *Fuzzy-Logik* kann als eine Verallgemeinerung der klassischen Logik auf ein Kontinuum von Wahrheitswerten aufgefaßt werden. Sie ermöglicht die Überwindung der mit der dualen Logik untrennbar verbundenen Einschränkungen, wie z.B. das Auftreten von Paradoxien oder die Unvereinbarkeit von Vollständigkeit und Widerspruchsfreiheit. Im Zusammenhang mit dieser Arbeit ist Fuzzy-Logik in zweierlei Hinsicht von Interesse: Sie bietet erstens einen Ausweg aus dem Dilemma, daß sich hohe Komplexität nicht mit hoher Präzision vereinen läßt (die Beschreibung eines Schmelbrandes anhand eines Satzes von Differentialgleichungen erscheint weder sinnvoll noch durchführbar) und gestattet zweitens eine straightforward Übertragung des in linguistischer Form vorliegenden Expertenwissens auf eine mathematische Beschreibungsebene.

In seiner allgemeinen Bedeutung bezeichnet der Begriff Fuzzy-Logik eine ganze Familie möglicher Logiken, die sich in der Definition ihrer logischen Operatoren unterscheiden. Die Schlüsselkonzepte eines fuzzy-basierten Regel- bzw. Entscheidungssystems werden im folgenden kurz beschrieben. Eine ausführliche Darstellung findet sich z.B. in [Bot 95].

#### 2.3.6.1 Unscharfe Mengen

Im Gegensatz zu klassischen Mengen können die Elemente einer unscharfen Menge  $M$  dieser nur *teilweise* angehören. Mathematisch wird dies durch den Wert einer sog. *Zugehörigkeitsfunktion*  $\mu_M$  ausgedrückt, mit deren Hilfe sich die (kontinuierlichen) Zugehörigkeitsgrade zur Menge  $M$  (0 für keine, 1 für volle Zugehörigkeit) angeben lassen. Die klassischen Mengen sind in diesem Konzept als Sonderfall enthalten - ihre Zugehörigkeitsfunktionen zeichnen sich dadurch aus, daß sie nur die Werte 0 und 1 annehmen können.

#### 2.3.6.2 Linguistische Variablen

Linguistische Variablen stellen die Basis für das „Rechnen mit Worten“ (Zadeh) dar. Die Werte, die sie annehmen können, werden als linguistische Terme bezeichnet und sind meist der Umgangssprache entlehnt (z.B. linguistische Variable „Temperatur“ mit den Termen „hoch“ und „niedrig“). Linguistische Variablen werden mit Hilfe von Zugehörigkeitsfunktionen in Form von Verteilungsfunktionen über einer (numerischen) Basisvariablen mathematisch repräsentiert.

#### 2.3.6.3 Fuzzifikation der Eingangssignale

Dieser Terminus bezeichnet die Transformation der (scharfen) numerischen Eingangssignale auf den (unscharfen) linguistischen Wertebereich. Dabei wird jeder scharfen Eingangsgröße

ein Zugehörigkeitswert zu jedem der linguistischen Terme zugewiesen. Dies geschieht dadurch, daß die Werte der entsprechenden Zugehörigkeitsfunktionen über dem aktuellen Wert der Basisvariablen abgelesen werden.

#### 2.3.6.4 Approximatives Schließen

Um Aussagen der Form WENN [...] UND/ODER [...] DANN [...] modellieren zu können, müssen die entsprechenden logischen Operatoren zur Verfügung gestellt werden, z.B. Min-Operator für UND und Max-Operator für ODER. Der Vorgang des approximativen Schließens besteht aus zwei Schritten. Im ersten (Aggregation) wird bestimmt, zu welchem Grad die Prämisse der Regel erfüllt ist, im zweiten (Komposition) wird dieser Gültigkeitsgrad dazu verwendet, den der Konsequenz zu ermitteln. Bei Standard-Inferenzmethoden geschieht dies durch Gleichsetzen der Erfüllung der Konsequenz mit der der Prämisse.

#### 2.3.6.5 Defuzzifikation der Ausgangsgrößen

Die Rückwandlung der unscharfen Inferferenzergebnisse  $B(y)$  in scharfe Ausgangsgrößen  $y_0$  wird als „Defuzzifikation“ bezeichnet. Zur Bestimmung von  $y_0$  wird oft der Abszissenwert des Schwerpunktes der Fläche unterhalb der Zugehörigkeitsfunktion  $\mu_B(y)$  herangezogen:

$$y_0 = \frac{\int_Y y \mu_B(y) dy}{\int_Y \mu_B(y) dy} \quad (2.48)$$

In der Praxis wird (2.48) zur Rechenzeiterparnis meist durch eine einfache Summenformel approximiert.

#### 2.3.7. Nichtlineare Optimierungsverfahren

Die im folgenden beschriebenen Methoden sind, ähnlich wie Evolutionäre Algorithmen, als naturanalog zu bezeichnen. Simulated Annealing beruht allerdings nicht auf dem Prozeß der Evolution, sondern bedient sich des Abkühlens und Auskristallisierens einer Schmelze als natürliches Vorbild. Bei Threshold Accepting, Sintflut-Algorithmus und Record-to-Record Travel handelt es sich Modifikationen des Simulated Annealing mit vereinfachten Kontrollstrukturen. Alle vier Methoden folgen einem gemeinsamen Ablaufschema:

Zunächst wird (meist stochastisch) ein Ausgangsdatensatz als 0-te Iteration der angestrebten Lösung erzeugt. Aus diesem wird eine neue Lösung generiert, die sich von der vorigen nur geringfügig unterscheidet, d.h. die neue Lösung liegt in einer vorab definierten Nachbarschaft der alten Lösung.

Beide werden nun hinsichtlich ihrer Zielfunktionswerte verglichen und eine Entscheidung hinsichtlich der Akzeptabilität der neuen Lösung getroffen. Erfüllt die neue Lösung das Akzeptanzkriterium, so ersetzt diese die alte Lösung und der Vorgang wird mit der neuen Lösung als Ausgangsmenge wiederholt. Anderenfalls wird die neue Lösung verworfen und die Suche ausgehend von der alten fortgesetzt.

Das Akzeptanzkriterium (Selektionskriterium) ist so formuliert, daß grundsätzlich auch begrenzte Lösungsverschlechterungen toleriert werden. Mit Fortschreiten des Optimierungsprozesses werden jedoch die Qualitätsanforderungen sukzessive erhöht. Zur Sicherstellung endlicher Laufzeiten werden Abbruchkriterien definiert.

### 2.3.7.1 Simulated Annealing

Simulated Annealing wurde unabhängig voneinander durch Kirkpatrick et. al [Kir 83] sowie von Cerny [Cer 85] als Optimierungsverfahren für kombinatorische Problemstellungen vorgeschlagen. Das Verfahren beruht auf dem Vorbild des physischen Abkühlungsprozesses einer Schmelze zum Festkörper. Ein Beispiel wäre die Bildung eines gleichmäßigen Kristalls aus der geschmolzenen Substanz. In der Schmelze können sich die Moleküle praktisch frei bewegen. Je weiter die Temperatur abnimmt, desto mehr wird ihre Bewegungsfreiheit eingeschränkt. Erfolgt das Abkühlen hinreichend langsam, so wird auf jeder Temperaturstufe ein thermisches Gleichgewicht erreicht. Die Substanz gelangt so schließlich in eine Grundstruktur mit minimalem Energieniveau, wie etwa bei einem Kristall. Bei zu raschem Abkühlen verfestigt sich die Substanz zwar ebenfalls, jedoch in einer Struktur mit höherem Energieniveau. Wichtig ist hier, daß ein System niedriger Temperatur durchaus in einem energetisch hohen Zustand sein kann.

Befindet sich das System im thermischen Gleichgewicht, dann ist die Wahrscheinlichkeit  $p_T(a)$ , daß sich das System bei einer Temperatur  $T$  in Zustand  $a$  befindet, abhängig vom Energieniveau  $E_a$  dieses Zustandes und gegeben durch die Boltzmann-Verteilung:

$$p_T(a) = \frac{\exp\left(\frac{-E_a}{k \cdot T}\right)}{\sum_{b \in S} \exp\left(\frac{-E_b}{k \cdot T}\right)} \quad (2.49)$$

Dabei ist  $S$  die Menge aller möglichen Zustände des Systems und  $k$  die Boltzmann-Konstante. Bereits 1953 hatten Metropolis et. al. [Met 53] ein stochastisches Simulationsverfahren zur Beschreibung der Strukturentwicklung einer Schmelze bei gegebener Temperatur vorgeschlagen: Angenommen, das System befindet sich zu einem Zeitpunkt  $t$  im Zustand  $a$  mit Energieniveau  $E_a$ . Durch eine geringfügige zufällige Veränderung von  $a$  erhält man einen neuen Vorschlag  $b$  für den Zustand des Systems zum Zeitpunkt  $t+1$ . Ob der neue Zustand akzeptiert wird oder nicht, hängt von der Differenz  $\Delta E = E_b - E_a$  der Energieniveaus von  $a$  und  $b$  ab. Ist  $\Delta E \leq 0$ , so wird der neue Zustand akzeptiert. Für den Fall  $\Delta E > 0$  wird  $b$  nicht automatisch verworfen, sondern mit der Wahrscheinlichkeit

$$p = \exp\left(\frac{E_a - E_b}{k \cdot T}\right) \quad (2.50)$$

akzeptiert. Wird eine hinreichend große Anzahl von Iterationen durchgeführt, so stellt diese Akzeptanzregel sicher, daß sich das System auf jedem Temperaturniveau zum thermischen Gleichgewicht hinbewegt.

Dieses sogenannte Metropolis-Verfahren dient im Simulated Annealing dazu, eine Sequenz von Lösungen für kombinatorische Optimierungsprobleme zu generieren. Dabei werden folgende Analogien zum physischen Abkühlen benutzt:

- Lösungen des Optimierungsproblems korrespondieren zu Systemzuständen (Konfigurationen) beim physischen Abkühlen.

- Die Zielfunktion  $F$  entspricht dem Energieniveau.
- Die Suche nach möglichst guten Lösungen entspricht der Suche nach einem Systemzustand minimaler Energie.
- Die Temperatur  $T$  wird zu einem Kontrollparameter der Optimierungsmethode.

Simulated Annealing kann als eine iterative Anwendung des Metropolis-Verfahrens auf sinkenden Werten des Kontrollparameters  $T$  aufgefaßt werden.

Von entscheidender Bedeutung für die Performance von SA ist der gewählte Abkühlungsplan (*annealing schedule*). Hierbei wird festgelegt, in welchen Schritten der Kontrollparameter  $T$  abgesenkt werden soll und wie viele Iterationen dabei auf den einzelnen Stufen durchzuführen sind.

Der ursprüngliche Vorschlag von Kirkpatrick et al. für einen Abkühlungsplan sieht einen Anfangswert für  $T_0$  vor, der so hoch ist, daß praktische jede Lösung akzeptiert wird. Dadurch wird vermieden, daß sich der Optimierungsprozeß schon frühzeitig auf einen bestimmten Bereich des Suchraumes beschränkt. Im weiteren Optimierungsverlauf wird  $T$  dann mittels einer proportionalen Abkühlungsfunktion der Form  $T_{t+1} = \alpha \cdot T_t$  abgesenkt. Typische Werte für die Konstante  $\alpha$  liegen zwischen 0,8 und 0,99. So erhält man eine asymptotische Annäherung an den Minimalwert  $T_{\min} = 0$ . Weitere Abkühlungspläne finden sich z.B. in [Nrc 92].

Die Akzeptanzregel nach Metropolis et al. bewirkt, daß SA auch Lösungsverschlechterungen toleriert, wobei die Annahmewahrscheinlichkeit um so höher ist, je größer der Wert von  $T$  ist und je geringer die Lösungsverschlechterung ausfällt. Dieser Mechanismus bewirkt, daß die Barrieren, die ein lokales Minimum umgeben, zur Auffindung besserer Lösungen überwunden werden können. Mit fortschreitendem Optimierungsprozeß wird  $T$  immer kleiner und damit die Akzeptanzwahrscheinlichkeit für Lösungsverschlechterungen. Bei  $T=0$  werden nur noch Verbesserungen zugelassen.

Langsames Abkühlen führt zu sehr guten Ergebnissen, aber auch zu langen Rechenzeiten. Schnelles Abkühlen verkürzt die Rechenzeit, führt im allgemeinen aber auch zu schlechteren Ergebnissen. Für praktische Anwendungen muß daher ein trade-off zwischen akzeptabler Rechenzeit und Güte der erzielten Lösung gefunden werden.

### 2.3.7.2 Threshold Accepting

Threshold Accepting (TA), Sinflut-Algorithmus (SiA) und Record-to-Record Travel (RR) basieren auf Simulated Annealing, enthalten jedoch einige Vereinfachungen. Die Algorithmen wurden am IBM Forschungsinstitut in Heidelberg unter Vorgabe folgender Zielsetzungen entwickelt:

- einfache Implementierung,
- einfache Parametrisierung,
- geringe Rechenzeiten,
- möglichst vergleichbare oder bessere Lösungsqualität als bei Simulated Annealing.

Der Hauptunterschied zu Simulated Annealing liegt in der Ausgestaltung der Akzeptanzregel für neu generierte Lösungen. Wenn die neue Lösung gegenüber der alten den Zielfunktionswert verschlechtert, muß eine Annahmewahrscheinlichkeit berechnet und schließlich, mittels einer Zufallszahl, die Akzeptanzentscheidung getroffen werden. Diese Vorgehensweise ist beim Simulated Annealing sehr rechenzeitaufwendig und bildet den Ansatzpunkt für Vereinfachungen in TA, SI und RR ([DUE 90], [DUE 93]).

TA akzeptiert jede neue Lösung, soweit sie „nicht viel schlechter“ ist als die alte Lösung. Das Ausmaß der erlaubten Verschlechterung hängt von einem Schwellenwert (threshold)  $T$  ab, der im Verlauf des Optimierungsprozesses sukzessive bis auf Null verringert wird. Dadurch reduziert sich die jeweils in einem Schritt zulässige Verschlechterung allmählich, bis schließlich nur noch Verbesserungen akzeptiert werden. Wie bei allen Optimierungs-Heuristiken ist es zweckmäßig, die beste während des Optimierungsprozesses bisher gefundene Lösung separat festzuhalten und immer dann zu aktualisieren, wenn eine noch bessere Lösung gefunden wird. Auf diese Weise wird sichergestellt, daß die bislang beste gefundene Lösung erhalten bleibt und als Endergebnis zur Verfügung steht.

Auch hier kann die Wahl einer neuen probeweisen Lösung aus der Nachbarschaft der alten Lösung einerseits stochastisch oder aber auch deterministisch erfolgen werden, also nach Maßgabe eines festgelegten Plans ohne Zufallseinflüsse. Wichtig ist, daß ein permanentes Zirkulieren zwischen einzelnen Lösungen möglichst vermieden wird.

Wie beim Simulated Annealing für die Temperatur  $T$  muß auch beim Threshold Accepting ein „Abkühlungsplan“ zum gezielten Herunterfahren des Steuerparameters  $T$  festgelegt werden.

### 2.3.7.3 Sintflut-Algorithmus

Beim Sintflut-Algorithmus und Record-to-Record-Travel handelt es sich um weitere Vereinfachungen des Threshold-Accepting [Due 93]. Der Sintflutalgorithmus läßt sich am besten erläutern, wenn man ihn als Maximierungsproblem formuliert und anhand der Metapher eines Wanderers beschreibt, der ohne Karte (d.h. ohne Kenntnis des globalen Maximums) versucht, einen möglichst hohen Berggipfel zu erreichen. Die Reise des Wanderers wird von einem sintflutartigen Regen beleitet (Namensgebung!), der ihn in immer höhere Regionen zwingt.

Im Gegensatz zum Threshold Accepting ist nun nicht mehr das Ausmaß der momentanen Lösungsänderung von Bedeutung (Lösungsverbesserung wird mit  $p=1$  akzeptiert, Lösungsver-schlechterung nur bis zu einem gewissen Schwellwert  $T$  mit  $p=1$  sonst mit  $p=0$ ), sondern ein geforderter Minimalwert der Zielfunktion, der „Wasserstand“, der im Laufe des Prozesses immer weiter ansteigt.

Die Erhöhung Wasserstandes  $W$  sollte nicht mit konstanter Rate erfolgen, sondern sollte zunächst sehr stark und dann immer geringfügiger heraufgesetzt werden, um die Chance zu erhöhen, in der Endphase noch Lösungsverbesserungen zu finden.

### 2.3.7.4 Record-to-Record-Travel

Beim Record-to-Record-Travel wird jede neue Lösung akzeptiert, die besser oder „nur unwesentlich schlechter“ als die beste bisher gefundene Lösung (=Record) ist. Der Terminus „nur unwesentlich schlechter“ wird dabei in Form einer maximal erlaubten Abweichung  $A_{\max}$  implementiert. Im Gegensatz zum Threshold-Accepting wird für die Akzeptanzentscheidung nicht die *vorhergehende* Lösung zur Entscheidung herangezogen, sondern die *bisher beste* (Record). Die Abweichung  $A_{\max}$  bleibt im Laufe des Optimierungsprozesses konstant.

## 3. Experimentelles

### Übersicht

3.1. Grundlagen zur Früherkennung von Bränden an Bord von Luftfahrzeugen .....	44
3.1.1 Vorbereitungen.....	44
3.1.1.1 Vorauswahl der Sensoren.....	44
3.1.1.2 Aufbau und Bestückung der Branddetektoren .....	45
3.1.1.3 Gasmischapparatur zur Erzeugung definierter Gasgemische.....	45
3.1.2 Meßreihen.....	47
3.1.2.1 Gasmischermessungen zur Bestimmung der optimalen Heizspannung.....	47
3.1.2.2 Gasmischermessungen zur Bestimmung der Sensor-Kennlinien.....	47
3.1.2.3 Brand- und Störgrößenmessungen .....	47
3.1.2.4 Klimakammermessungen .....	48
3.2. Experimentelle Grundlagen zur Detektion von Lösungsmitteln unterhalb UEG .....	48
3.2.1 Erzeugung definierter Lösungsmittelkonzentrationen .....	48
3.2.1.1 Bereitstellung von Lösungsmitteln in der Gasphase .....	49
3.2.1.2 Erzeugung von Lösungsmittelkonzentrationen deutlich unter UEG.....	49
3.2.2 Gassensoren.....	50
3.2.2.1 Halbleitergassensoren.....	50
3.2.2.2 Massensensitive Sensoren .....	51
3.2.3 Versuchsreihen .....	52

### 3.1. Grundlagen zur Früherkennung von Bränden an Bord von Luftfahrzeugen

Mit freundlicher Unterstützung der Firma AOA Gauting wurden Untersuchungen zur Erweiterung der bislang zur Brand-Detektion in Flugzeugen eingesetzten Sensorik (Streulichtmel- der zur Rauch-Detektion) um eine geeignete Gas-Sensorik durchgeführt. Ziel dieser Untersu- chungen war die Erreichung folgender Verbesserungen gegenüber den derzeit existierenden Systemen:

- Brände sollen bereits im Frühstadium, d.h. in der Entstehungsphase vor Einsatz der Rauch- und Wärmeentwicklung anhand charakteristischer Gasemissionen erkannt werden.
- Brände sollen auch dann entdeckt werden können, wenn sie vom Detektor durch wärme- dämmende bzw. luftzirkulationsbehindernde Materialien abgeschirmt sind. Gase gelangen über Diffusion zum Detektor und sind somit als Nachweismedium besser als Rauch bzw. Wärme geeignet.
- Partikel/Aerosole, die nicht Produkte einer Verbrennung sind, sollen keine Fehlalarme auslösen.

Bei dem zu entwickelnden Sensorsystem soll es sich um einen kombinierten Detektor mit Streulicht- und Gassensorik handeln. Noch vorhandene Schwächen des bisherigen Systems sollen durch die erweiterte Sensorik eliminiert werden.

#### 3.1.1 Vorbereitungen

##### 3.1.1.1 Vorauswahl der Sensoren

Der Einsatz unter luftfahrtspezifischen Umgebungsbedingungen stellt neue Anforderungen an die verwendeten Gassensoren: die auftretenden Umgebungsbedingungen hinsichtlich Druck und Temperatur der umgebenden Luft können erheblich von denen einer typischen bodenge- bundenen Anwendung abweichen.

Eine breite Basis von Gassensoren für die gesamte Versuchsreihe sollte sicherstellen, daß Sensoren, die unter "Normalbedingungen" eine eher durchschnittliche Performance aufwei- sen, zu früh zugunsten solcher aufgegeben werden, die unter diesen Bedingungen außeror- dentlich gut arbeiten, sich unter geänderten Umgebungsbedingungen jedoch als untauglich erweisen sollten.

Folgende Kriterien wurden bei der Sensor-Vorauswahl berücksichtigt:

- Verfügbarkeit
- Preis
- geringe Baugröße
- geringe Leistungsaufnahme
- einfache Ansteuerung/Abfrage
- Sensitivität/Selektivität für die Brandindikator-Gase CO und H<sub>2</sub>
- geringe Sensitivität für Störgrößen (Druck, Temperatur, andere Gase...)

Es wurden Halbleiter-Gassensoren dreier verschiedener Hersteller ausgewählt:

Sensortypen von UST:

- GGS 1000 (Zielgas: CO)

- GGS 2015 (Zielgas: CO)
- GGS 6000 (Zielgas: H<sub>2</sub>)
- GGS 7000 (Zielgas: Lösungsmittel, NO)

Sensortypen von FIS:

- SB AQ 4 (Zielgas: Zigarettenrauch)
- SB-19 (Zielgas: H<sub>2</sub>)

Sensortypen von Taguchi:

- TGS 800 (Zielgas: Luftverunreinigungen)
- TGS 812 (Zielgas: CO)
- TGS 813 (Zielgas: brennbare Gase)

### 3.1.1.2 Aufbau und Bestückung der Branddetektoren

Die in den Versuchsreihen eingesetzten Sensorsysteme zur Branddetektion, im folgenden kurz „BIAP“ genannt (Brandmelder IAP) basieren auf dem Rauchmelder FD 9476 der Fa. AOA Gauting (optischer Streulichtmelder, Drucksensor, zwei Temperatursensoren), welche zusätzlich mit je drei Gassensoren bestückt wurden (Tabelle 3.1):

BIAP #	Sensor	Heizspannung [mV]
1	UST 7000	3000
	UST 6000	5000
	UST 7000	3000
2	FIS 19	900
	FIG 800	5000
	UST 1000	1900
3	FIS 4	900
	FIG 812	5000
	UST 2015	2500
4	FIS 19	900
	FIG 813	5000
	FIG 800	5000
5	UST 2015	2500
	UST 6000	5000
	UST 7000	3000
6	FIS 4	900
	FIG 812	5000
	UST 7000	3000
7	FIS 19	900
	UST 6000	5000
	UST 7000	3000

*Tabelle 3.1: Bestückung der Sensoreinheiten*

### 3.1.1.3 Gasmischapparatur zur Erzeugung definierter Gasmische

Für die Versuchsreihen wurde eine PC-gesteuerte Gasmischanlage verwendet, die die Erzeugung definierter Gaskonzentrationen, Durchflußgeschwindigkeiten und Luftfechtigkeiten gestattet.

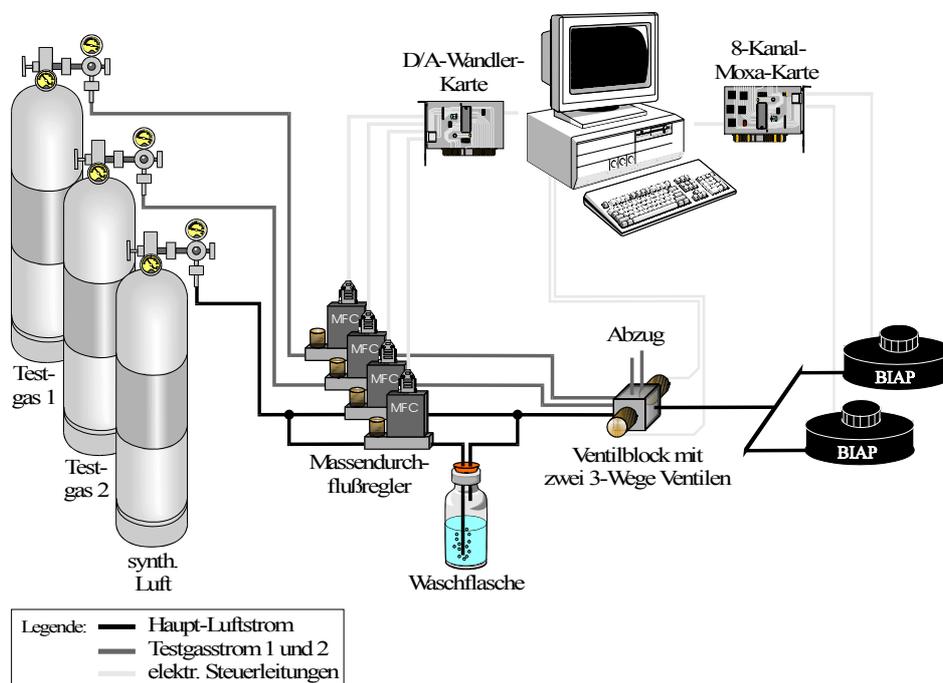
Die Gasmischanlage besteht aus insgesamt 4 programmierbaren Massendurchflußreglern (MFC), welche die Flüsse von Prüfgas und synthetischer Luft (80% N<sub>2</sub> und 20% O<sub>2</sub>) regeln. Zwei dieser Massendurchflußregler (je 0 – 100 l/h) steuern den Strom der synthetischen Luft, wobei ein Teilstrom durch eine Waschflasche mit destilliertem Wasser geleitet wird. Damit ist es möglich durch Variation des Verhältnisses der beiden Teilströme verschiedene Feuchtigkeitswerte zwischen 10% und annähernd 90% einzustellen. Die anderen beiden Durchflußregler (je 0 – 1,2 l/h) steuern zwei voneinander unabhängige Ströme von Prüfgasen.

Die Ströme der Prüfgase können in einem Ventilblock mit zwei Dreiwegeventilen dem Strom der synthetischen Luft zugemischt oder in einen Abzug geleitet werden. Diese Ventile sind in einen speziell konstruierten Edelstahlblock integriert, so daß sich sehr kurze Gaswege und demzufolge schnelle Gaswechselzeiten ergeben.

Aus diesem Grund werden auch die Prüfgase erst unmittelbar vor den BIAPs dem Luftstrom beigemischt. Einen weiteren Beitrag für kurze Gaswechselzeiten liefert das Verbinden der BIAPs über einen kurzen Schlauch mit dem Ventilblock.

In der folgenden Abbildung sieht man den schematischen Aufbau der Gasmischanlage im Labor. Die oben beschriebene Gasmischanlage erlaubt es, Meßreihen mit verschiedenen Gaskonzentrationen und relativen Luftfeuchten durchzuführen.

Zur Datenerfassung während wurde eine Windows95/NT-taugliche Software entworfen und implementiert. Diese liest die Ausgaben der Geräte seriell ein (maximal 8 Geräte über eine serielle Schnittstellenkarte mit Interrupt-Sharing)) und stellt die Signale der Einzelsensoren grafisch dar. Alle Meßwerte werden zur weiteren Verarbeitung abgespeichert.



**Abb. 3.1:** Schematische Darstellung der verwendeten Gasmischanlage

### 3.1.2 Meßreihen

#### 3.1.2.1 Gasmischermessungen zur Bestimmung der optimalen Heizspannung

An den Sensortypen UST1000, UST2015 und UST6000 wurde die Heizspannung zur Detektion der vorgesehenen Zielgase optimiert. Diese sind für die einzelnen Typen:

Sensor	Zielgas
UST 1000	CO
UST 2015	CO
UST 6000	H <sub>2</sub>

Hierzu wurden die Signaländerungen der Sensoren bei Angeboten von jeweils 20ppm CO bzw. H<sub>2</sub> mit verschiedenen Heizspannungen gemessen.

Dabei wurden bei den Sensoren zunächst die Heizspannungen eingestellt und nach einer Einlaufphase von ca. 30 min ein Gasangebot von 15 min. eingeschaltet. Die relative Luftfeuchte im Gasstrom betrug dabei 50 %.

#### 3.1.2.2 Gasmischermessungen zur Bestimmung der Sensor-Kennlinien

Es wurden die Signaländerungen der untersuchten Sensortypen bei Angebot der Gase CO und H<sub>2</sub>, welche als Zielgase bei der Branderkennung betrachtet werden sollten, aufgezeichnet.

Die Gasangebote wurden bei einem Gesamtfluß von 50l/h mit einer relativen Feuchte von 50% bei 25°C angeboten. Es wurde nacheinander 10ppm CO, 20ppm CO, 5ppm H<sub>2</sub> und 15ppm H<sub>2</sub> für jeweils 400s angeboten.

Neben Aussagen über Sensitivität und Selektivität eines Sensortyps geben die aufgezeichneten Kennlinien Aufschluß über weitere wichtige Kriterien zur Beurteilung eines Sensors wie z. B. das Ausmaß von Exemplarstreuungen.

#### 3.1.2.3 Brand- und Störgrößenmessungen

Hierzu wurden die Branddetektoren kopfüber in einer Höhe von 2,30 m über dem Boden in einem 3m hohen Raum installiert. Die Messungen wurden auf einem Tisch mit einer Höhe von 0,8 m durchgeführt, was einem Abstand von 1,50 m zwischen zu untersuchendem Material und Branddetektoren entspricht. Um bei einigen Versuchen die Entfernung zwischen Material und Branddetektoren auf 1 m zu reduzieren, wurde zusätzlich noch ein 0,5 m hohes Gestell auf den Tisch gestellt.

Bei den Brandversuchen wurde Baumwolle (offener Brand), Papier (offener Brand), Silikon (offener Brand), Teflon (Schwelbrand), Polyether (offener Brand) und Sitzschaumstoff der Concorde (offener Brand) verwendet. Hierbei wurden Messungen mit sowie ohne Rauchfang durchgeführt. Als Rauchfang diente ein Stoffvorhang, der über die Branddetektoren gehängt wurde. Dieser soll den Einfluß rauchbehindernder Materialien in der Flugzeugkabine (z.B. Vorhänge zwischen Business- und Economy-Class) simulieren.

Als Störgrößen wurden Haarspray, Zigarettarauch, Babypuder, Rasierwasser, Feuchttücher sowie die Lösungsmittel Aceton, Isopropanol und Ethanol (in vielen Reinigungsmitteln vorhanden) angeboten.

#### **3.1.2.4 Klimakammermessungen**

Die Messungen wurden in einer Klimakammer der Firma AOA Gauting durchgeführt. In dieser können Temperaturen von  $-40^{\circ}\text{C}$  bis  $+80^{\circ}\text{C}$  eingestellt und der Druck von 200mbar bis 1100mbar geregelt werden. Die Signale wurden über eine serielle Schnittstelle ausgelesen.

Insgesamt wurden drei verschiedene Messungen durchgeführt, bei denen die Umgebungstemperatur sowie der Druck variiert wurden. Zusätzlich wurde während einer Messung noch ein Papierbrand in der Klimakammer gezündet.

In der ersten Messung wurde bei einer konstanten Umgebungstemperatur von  $25^{\circ}\text{C}$  der Druck erst schnell (innerhalb von 5 min) und danach noch einmal langsam (innerhalb von 80 min) von 1100mbar auf 200mbar reduziert und wieder von 200mbar auf 1100mbar erhöht.

Die zweite Messung umfaßt bei konstanter Umgebungstemperatur von  $25^{\circ}\text{C}$  eine Druckminderung von 950mbar auf 200mbar. Danach wurde der Druck wieder erhöht auf 500mbar und ein Papierbrand gezündet. Nach Beendigung des Brandes wurde der Druck erhöht auf 950mbar und noch einmal auf 500mbar abgesenkt.

In der dritten Messung wurde zunächst bei konstanter Außentemperatur ( $25^{\circ}\text{C}$ ) der Druck von 1100mbar auf 200mbar abgesenkt und wieder auf 900mbar erhöht. Bei einem konstanten Druck von 900mbar wurde dann die Umgebungstemperatur auf  $-40^{\circ}\text{C}$  abgesenkt, danach auf  $+80^{\circ}\text{C}$  erhöht und wieder auf  $+25^{\circ}\text{C}$  erniedrigt. Das gleiche Temperaturprofil wurde daraufhin noch einmal bei einem konstanten Druck von 600mbar wiederholt.

### **3.2. Experimentelle Grundlagen zur Detektion von Lösungsmitteln unterhalb UEG**

In Zusammenarbeit mit der Firma Merck wurde am Institut für Angewandte Physik (IAP) der Justus-Liebig-Universität Gießen eine Machbarkeitsstudie zur Entwicklung eines Multisensorsystems zum Nachweis von Lösungsmitteln deutlich unterhalb UEG durchgeführt.

Zunächst sollte hierzu die prinzipielle Eignung verschiedener Metalloxid Halbleitersensoren sowie massensensitiver Sensoren zum Nachweis ausgewählter organischer Lösungsmittel untersucht werden. Dabei sollte u.a. ermittelt werden, inwieweit massensensitive Sensoren für den Bereich industrieller Meßtechnik eingesetzt werden können.

#### **3.2.1 Erzeugung definierter Lösungsmittelkonzentrationen**

Die im folgenden aufgeführten Lösungsmittel sollen bei einem etwaigen Austritt im Chemikalienlager deutlich unterhalb der jeweiligen unteren Explosionsgrenze detektiert werden:

Lösungsmittel	Kategorie	Eigenschaften	Summenformel
<b>Benzol</b>	arom. Lösungsmittel	leichtentzündlich, giftig, krebserregend, starkes Nervengift (sowohl über Atmung als auch über Hautresorption)	$C_6H_6$
<b>Diethylether</b>	alip. Ether	hochentzündlich; kann bei Anwesenheit von Licht + $O_2$ explosionsfähige Peroxide bilden	$C_4H_{10}O$ ( $C_2H_5-O-C_2H_5$ )
<b>tert-Butylmethylether</b>	alip. Ether	leichtentzündlich	$C_5H_{12}O$
<b>1,2-Propylenoxid</b> (= Propylenether)	alip. Ether	hochentzündlich, giftig, krebserregend	$C_3H_6O$
<b>2-Methylbutan</b> (= Isopentan)	alip. Lösungsmittel	leichtentzündlich	$C_5H_{12}$

*Tab. 3.2: Eigenschaften der untersuchten Lösungsmittel*

### 3.2.1.1 Bereitstellung von Lösungsmitteln in der Gasphase

Die insgesamt fünf Chemikalien wurden von der Firma Merck in Flüssigphase zur Verfügung gestellt. Das Abfüllen der Lösungsmittel in Gasflaschen wurde am IAP vorgenommen. Die in die Flaschen injizierten Flüssigkeitsmengen wurden dabei so bemessen, daß die Flaschenkonzentrationen im Bereich 300 - 700ppm liegen. Auf diese Weise können die gewünschten Konzentrationen mit Hilfe der im IAP vorhandenen Gasmischanlagen hergestellt werden.

Zur Konzentrationsbestimmung wurde ein Fourier-Transform-Infrarot-Spektrometer (FTIR) mit photoakustischem Detektor verwendet. Für alle fünf Lösungsmittel wurde mit Hilfe einer  $\mu$ l-Spritze jeweils soviel Flüssigkeit in eine Luftvolumen von 5 Litern injiziert, daß sich nach vollständigem Verdampfen eine Konzentration vom 500ppm ergab. Die so erhaltenen Spektren dienen als Referenzspektren für alle weiteren Messungen.

Um Fehler durch Ungenauigkeiten bei der oben beschriebenen Prozedur auszuschließen, wurden später fünf Fertiggemische zu je 500ppm von Messer Griesheim bezogen und bei den weiteren Messungen verwendet.

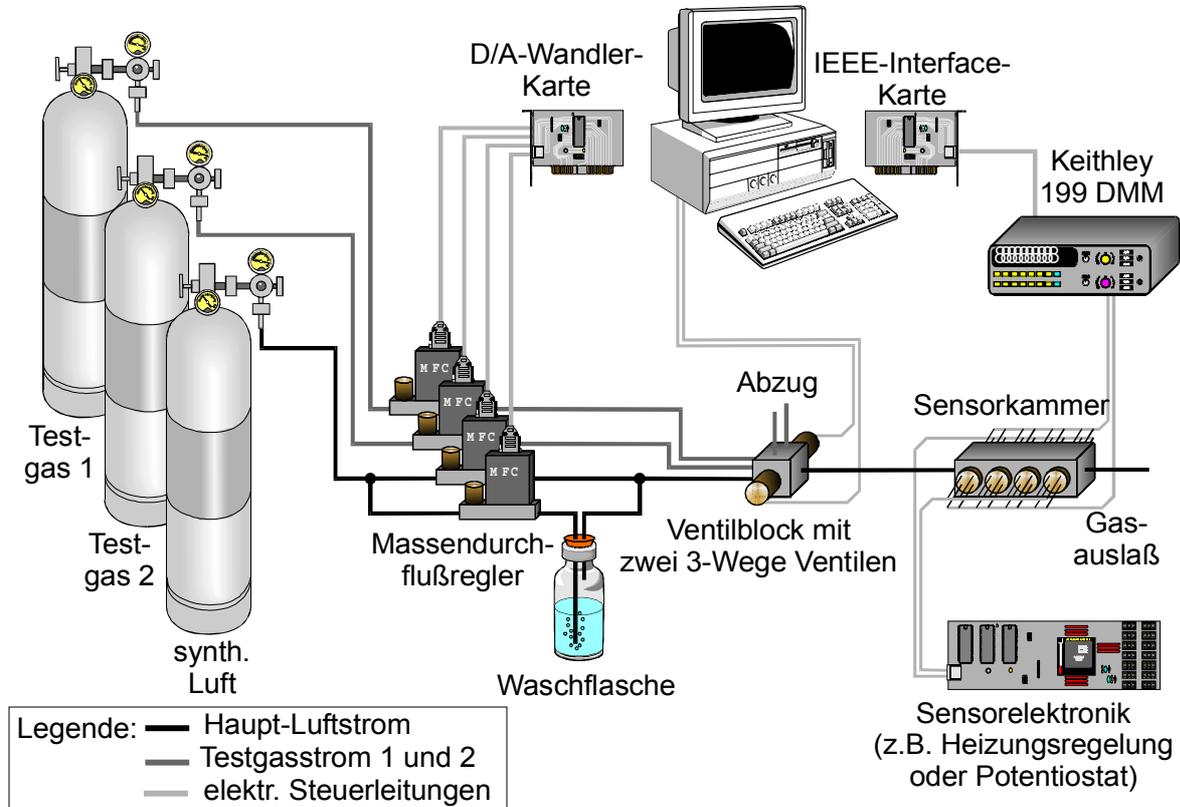
### 3.2.1.2 Erzeugung von Lösungsmittelkonzentrationen deutlich unter UEG

Zur Erzeugung variabler Lösungsmittelkonzentrationen wurde wieder eine computergesteuerte Gasmischanlage verwendet (s. Abb. 3.2). Diese mischt das zu untersuchende Gas in einen Trägergasstrom aus synthetischer Luft, so daß sich über die Steuerung der Teilströme beliebige Gaskonzentrationen erzeugen lassen. Der Vorteil von Gasströmen gegenüber stationären Gasmischungen liegt in den deutlich kürzeren Gaswechselzeiten.

Die Gasmischanlage wurde so konzipiert, daß man durch eine Aufspaltung des Trägergasstromes (synthetische Luft) in einen 100% gesättigten und einen trockenen Teilstrom eine stufenlose Regelung der Luftfeuchte vornehmen kann. Durch zwei zusätzliche Massflow-

controller kann neben dem gewünschten Zielgas eine weitere Störgaskomponente (hier CO<sub>2</sub>) zugemischt werden.

Der Gesamtfluß wurde auf 10l/h eingestellt. Die Lösungsmittelkonzentrationen wurden im Bereich von 2,5pp bis 10ppm in Schritten von 2,5ppm eingestellt. Die relative Luftfeuchtigkeit betrug dabei jeweils 30% im ersten und 70% im zweiten Meßzyklus.



*Abb. 3.2: Lösungsmittelangebot mit computergesteuerter Gasmischanlage*

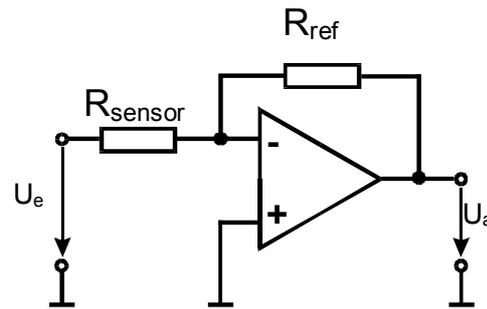
### 3.2.2 Gassensoren

#### 3.2.2.1 Halbleitergassensoren

Bei den für die Untersuchung eingesetzten Halbleitergassensoren wurde zunächst auf kommerziell erhältliche Sensoren zurückgegriffen:

Hersteller	Modell	Applikation gemäß Herstellerangabe
FIGARO	TGS 822	Lösungsmitteldetektion
FIGARO	TGS 2620	Lösungsmitteldetektion
UST	GG5 1000	Breitbandsensor
UST	GG5 3000	Sensor f. Kohlenwasserstoffe
UST	GG5 7000	Stickoxide u. Alkohol

Bei den Halbleiter-Gassensoren wurde eine konstante Eingangsspannung von 0,5Volt an die nachfolgend abgebildete Verstärkerschaltung angelegt:



*Abb. 3.3: Signalverstärkung*

Zum Ausgleich der unterschiedlich starken Leitwertänderungen bei den verschiedenen Lösungsmitteln wurden die Referenzwiderstände  $R_{\text{ref}}$  der einzelnen Sensoren entsprechend angepaßt.

### 3.2.2.2 Massensensitive Sensoren

Die im Rahmen dieser Arbeit eingesetzten massensensitiven Sensoren sind sogenannte **Bulk Acoustic Wave** (kurz BAW) Sensoren. Hierbei handelt es sich um Schwingquarze, die (wie in einer Quarzuhr) auf elektrischem Wege zu Oszillationen angeregt werden. Wenn sich Gasmoleküle auf einem solchen schwingenden Quarz anlagern, so nimmt dessen effektive Masse zu und seine Resonanzfrequenz verringert sich. Die Frequenzänderung ist proportional zur Massenzunahme und kann somit zum Gasnachweis herangezogen werden. Um diesen Effekt zu verwirklichen, wird der Schwingquarz mit einer absorptionsfähigen Polymerschicht versehen. Die verwendete Schicht hat maßgeblichen Einfluß auf die Selektivität und Sensitivität des Sensors. Das Polymer muß in der Lage sein, das zu detektierende Gas aufzunehmen. Allgemein gilt: je ähnlicher sich Polymer und aufzunehmendes Gas bezüglich ihrer chemischen und physikalischen Eigenschaften sind, desto besser kann das Polymer das Gas absorbieren.

In Zusammenarbeit mit der Universität Marburg wurden verschiedene Schichten, Schichtdicken sowie Beschichtungstechniken für den Nachweis von organischen Lösungsmitteln erprobt.

Ein Frequenzzähler zum Auslesen der 30MHz Quarze wurde von der Fa. SICAN (Hannover) entwickelt und dem IAP als Prototyp zur Verfügung gestellt. Das eingesetzte BAW Modul erlaubt ein paralleles Auslesen von bis zu 6 Schwingquarzen sowie das Protokollieren der aktuellen Temperatur- und Feuchtwerte.

Für die Messungen wurden 3 Schwingquarze mit unterschiedlich dicken Schichten von Poly(p-xylylen) (PPX), sowie Schichten aus Polystyrol, Poly( $\alpha$ -phenyl-para-xylylen), Polymethylmethacrylat und Polycarbonat eingesetzt

### 3.2.3 Versuchsreihen

Die angestrebte Nachweisgrenze wurde ursprünglich auf 1 Liter Lösungsmittel im Lagerinnenvolumen festgelegt, d.h. 1l verdampftes Lösungsmittel in 11.540.000 l Luft. Dies entspricht den folgenden Konzentrationen:

Lösungsmittel	ppm gefordert	UEG	UEG / 10	Zündtemperatur
Benzol	21,9 ppm	1,2 Vol. %	1200 ppm	555 °C
Diethylether	18,6 ppm	1,7 Vol. %	1700 ppm	170 °C
tert-Butylmethylether	16,3 ppm	1,6 Vol. %	1600 ppm	460 °C
1,2-Propylenoxid	27,75 ppm	1,9 Vol. %	1900 ppm	430 °C
2-Methylbutan	16,4 ppm	1,3 Vol. %	1300 ppm	420 °C

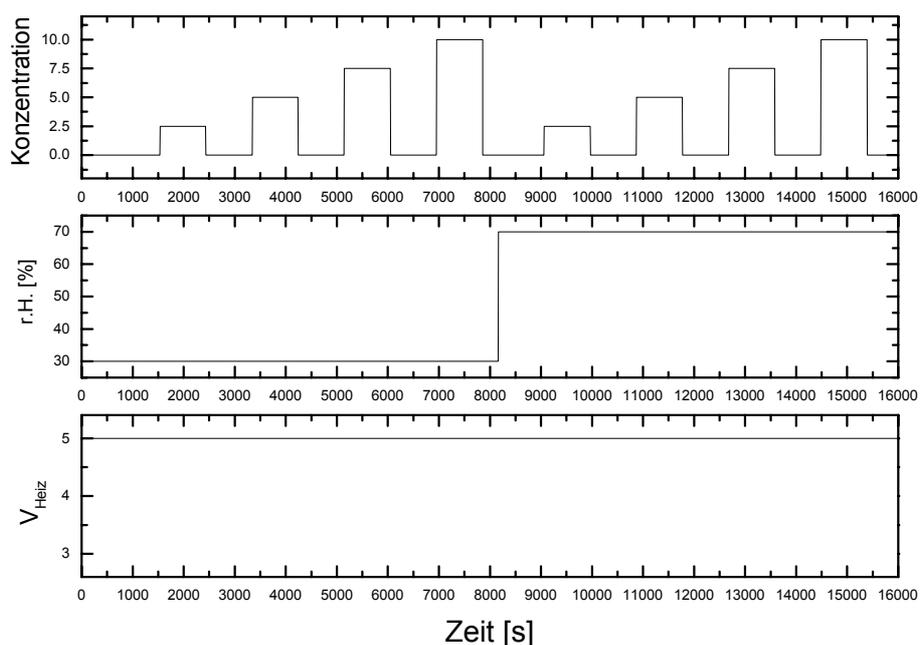
Um das Nachweisverhalten der Sensoren im unteren ppm-Bereich zu untersuchen, wurden Konzentrationen ab 2,5ppm angeboten. Neben der Gaskonzentration wurden zwei weitere Parameter variiert:

- die relative Feuchte (Wertebereich 30% und 70%)
- die Heizspannung (Wertebereich 3,0V bis 5,0V in Schritten von 0,5V)

Für jede fest eingestellte Heizspannung wurde folgendes Konzentrationsprofil angeboten (s. Abb. 3.3):

- 4 Gaspulse von 2,5ppm bis 10,0ppm in Schritten von 2,5ppm von jeweils 900s Dauer und 900s Pause bei 30% r.H.
- 4 Gaspulse von 2,5ppm bis 10,0ppm in Schritten von 2,5ppm von jeweils 900s Dauer und 900s Pause bei 70% r.H.

Zusätzlich wurde die CO<sub>2</sub>-Querempfindlichkeit durch Gasangebote von 500ppm bzw 1000ppm CO<sub>2</sub> ermittelt. Die BAW-Messungen verliefen in analoger Weise, lediglich der Konzentrationsbereich wurde höher gewählt.



**Abb. 3.3:** Konzentrationsprofil

## 4. Ergebnisse

### Übersicht

4.1 Bestimmung der optimalen Heizspannung .....	54
4.2 Sensorverhalten bei Angeboten von CO und H <sub>2</sub> .....	55
4.3 Verhalten der Sensoren bei Brand- und Störgrößenmessungen .....	56
4.4 Sensorsignale bei Klimakammermessungen .....	58
4.5 Lösungsmittelnachweis mit Halbleitersensoren .....	58
4.5.1 Ermittlung der optimalen Betriebstemperaturen .....	58
4.5.2 Feuchtequerempfindlichkeit .....	59
4.5.3 Response des Sensor-Arrays für verschiedene Gasangebote .....	60
4.5.4 Teilspezifitäten der Einzelsensoren .....	61
4.5.5 CO <sub>2</sub> -Querempfindlichkeit .....	62
4.6 Lösungsmittelnachweis mit massensensitiven Sensoren .....	63
4.6.1 Reaktion auf Gasangebot .....	63
4.6.2 Feuchtequerempfindlichkeit .....	64
4.6.3 CO <sub>2</sub> -Querempfindlichkeit .....	65
4.6.4 Zusammenfassung der Ergebnisse .....	65

#### 4.1 Bestimmung der optimalen Heizspannung

Aus den in 3.1.2.1 beschriebenen Messungen wird für jeden Sensor die relative Signaländerung bei 20ppm CO (dgCO), die relative Signaländerung bei 20ppm H<sub>2</sub> (dgH<sub>2</sub>) und das Verhältnis der relativen Signaländerungen (dgCO/dgH<sub>2</sub>) für jede Heizspannung bestimmt. Die optimale Heizspannung eines CO-Sensors findet man, wenn das Verhältnis der relativen Signaländerungen (dgCO/dgH<sub>2</sub>) maximal und für einen H<sub>2</sub>-Sensor minimal ist. Die folgenden beiden Tabellen führen jeweils die Meßergebnisse auf:

Spalte 1: Heizspannung in mV

Spalte 2: rel. Signaländerung bei 20ppm CO

Spalte 3: rel. Signaländerung bei 20ppm H<sub>2</sub>

Spalte 4: Verhältnis der rel. Signaländerungen

UST1000:

$U_{\text{heiz}}$ [mV]	dgCO	dgH <sub>2</sub>	dgCO/dgH <sub>2</sub>
1800	0,05	0,22	0,22727
1900	0,046	0,21	0,21905
2000	0,031	0,13	0,23846
2250	--	0,05	--
2500	--	0,05	--

UST2015:

$U_{\text{heiz}}$ [mV]	dgCO	dgH <sub>2</sub>	dgCO/dgH <sub>2</sub>
2000	0,045	--	--
2250	0,052	0,033	1,57576
2500	0,11	0,06	1,83333
3000	0,085	0,089	0,95506
3500	0,12	0,19	0,63158

Für den weiteren Betrieb der CO- bzw. H<sub>2</sub>-Sensoren wurden schließlich folgende Heizspannungen ausgewählt:

Typ	Heizspannung [mV]
UST1000	1900
UST2015	2500

#### 4.2 Sensorverhalten bei Angeboten von CO und H<sub>2</sub>

Die folgende Tabelle beschreibt zusammenfassend das Verhalten der untersuchten Sensortypen bei Angeboten der Gase CO und H<sub>2</sub>, welche als Zielgase bei der Branderkennung betrachtet werden sollten.

Sensortyp	Bemerkungen
UST 1000	Einzelexemplar sehr geringe Empfindlichkeit auf CO im BIAP (vorgesehenes Zielgas für diesen Sensortyp)
UST 2015	2 Exemplare geringe Empfindlichkeit auf CO im BIAP (vorgesehenes Zielgas für diesen Sensortyp) sehr große Exemplarstreuungen in den norm. LW-Änderungen bzgl. CO und H <sub>2</sub>
UST 6000	3 Exemplare hohe Empfindlichkeit auf H <sub>2</sub> (Zielgas), keine Querempfindlichkeit zu CO mittlere Exemplarstreuungen in den norm. LW-Änderungen bzgl. CO und H <sub>2</sub>
UST 7000	4 Exemplare sehr geringe Empfindlichkeit auf CO, geringe Empfindlichkeit auf H <sub>2</sub> große Streuungen in den absoluten LW-Änderungen, dafür sehr geringe Exemplarstreuungen in den norm. LW-Änderungen bzgl. CO und H <sub>2</sub>
FIS 4	2 Exemplare hohe Empfindlichkeit auf H <sub>2</sub> (Zielgas), keine Querempfindlichkeit zu CO geringe Exemplarstreuungen in den norm. LW-Änderungen bzgl. und H <sub>2</sub>
FIS 19	3 Exemplare hohe Empfindlichkeit auf H <sub>2</sub> (Zielgas), keine Querempfindlichkeit zu CO kleine Exemplarstreuungen in den norm. LW-Änderungen bzgl. und H <sub>2</sub>
TGS 800	2 Exemplare geringe Empfindlichkeit auf CO, mittlere Empfindlichkeit auf H <sub>2</sub> große Unterschiede in den norm. LW-Änderungen bzgl. CO
TGS 812	2 Exemplare geringe Empfindlichkeit auf CO, hohe Empfindlichkeit auf H <sub>2</sub> sehr große Unterschiede in den norm. LW-Änderungen bzgl. CO
TGS 813	Einzelexemplar sehr geringe Empfindlichkeit auf CO im BIAP, geringe Empfindlichkeit auf H <sub>2</sub>

### 4.3 Verhalten der Sensoren bei Brand- und Störgrößenmessungen

Generell weisen Halbleiter-Gassensoren auf SnO<sub>2</sub>-Basis deutliche Exemplarstreuungen auf. Diese sind jedoch bei den verschiedenen Sensortypen unterschiedlich stark ausgeprägt: Die beiden FIS Sensoren weisen eine geringere Exemplarstreuung auf, als die UST- bzw. Figaro-Sensoren.

Im Folgenden findet sich eine Zusammenfassung der Ergebnisse zur Signaländerung der Sensortypen in den Brand- und Störgrößenmessungen.

Sensortyp	Bemerkungen
opt. Melder	<p>Signaländerungen bei Rauchfang sind merklich schwächer als bei Versuchen ohne.</p> <p>nachfolgende Ergebnisse bei Versuchen ohne Rauchfang</p> <ul style="list-style-type: none"> <li>- deutliche bis starke Signaländerung bei Baumwoll- und Polyether, sehr starke Signaländerung bei Silikonbrand</li> <li>- deutliche bis starke Signaländerung bei Zigarettenrauch und Haarspray</li> <li>- keine bis schwache Reaktion auf Lösungsmittel und Teflonbrand</li> <li>- sehr starke Signaländerung bei Puder</li> </ul>
UST 1000	<p>Einzelexemplar</p> <ul style="list-style-type: none"> <li>- Leitwert-Zunahme bei Papier-, Baumwoll- und Silikonbrand sowie Zigarettenrauch und Haarspray</li> <li>- Leitwert-Abnahme bei Brand von Polyether und Angebot von Aceton</li> <li>- keine bis schwache Reaktion bei Teflonbrand sowie Versuchen mit Lösungsmitteln (außer Aceton)</li> </ul>
UST 2015	<p>2 Exemplare</p> <ul style="list-style-type: none"> <li>- starke Leitwert-Zunahme bei Papier-, Baumwollbrand, deutliche Zunahme auch bei Silikon und Sitzmaterial</li> <li>- Leitwert-Zunahme bei den Lösungsmitteln Isopropanol und Ethanol (auch beim Rasierwasser)</li> <li>- starke Leitwert-Zunahme bei Zigarettenrauch</li> <li>- sehr starke Leitwert-Zunahme bei Haarspray</li> <li>- beide Sensoren weisen vergleichbares Verhalten auf (außer bei Polyether)</li> </ul>
UST 6000	<p>3 Exemplare</p> <ul style="list-style-type: none"> <li>- starke bis sehr starke Leitwert-Zunahme bei Papier-, Baumwoll- und Silikonbrand</li> <li>- deutliche Zunahme bei Polyether, Sitzmaterial, Teflon und Haarspray</li> <li>- sehr starke Leitwert-Zunahme bei Zigarettenrauch</li> <li>- keine bis sehr schwache Reaktion bei Versuchen mit Lösungsmitteln</li> <li>- hohe Vergleichbarkeit der Einzelexemplare</li> </ul>

UST 7000	<p>4 Exemplare</p> <ul style="list-style-type: none"> <li>- deutliche bis starke Leitwert-Zunahme bei Papier-, Baumwoll- und Silikonbrand</li> <li>- deutliche Abnahme bei Polyether jedoch deutliche Zunahme bei Sitzmaterial</li> <li>- starke Leitwert-Zunahme bei Zigarettenrauch, deutliche bis starke Zunahme bei Haarspray</li> <li>- keine bis schwache Reaktion bei Versuchen mit Teflon und Lösungsmitteln</li> </ul> <p>große Streuungen im Verhalten der Einzelexemplare</p>
FIS 4	<p>2 Exemplare</p> <ul style="list-style-type: none"> <li>- starke bis sehr starke Signaländerung bei Papier-, Baumwoll- und Silikonbrand</li> <li>- sehr starke Signaländerung bei Zigarettenrauch</li> <li>- keine bis schwache Reaktion bei Versuchen mit Lösungsmitteln und Polyether</li> <li>- hohe Vergleichbarkeit der beiden Exemplare</li> </ul>
FIS 19	<p>3 Exemplare</p> <p>Verhalten ähnlich FIS 4</p> <ul style="list-style-type: none"> <li>- keine bis schwache Reaktion bei Versuch mit Sitzmaterial, Teflon und Haarspray</li> <li>- hohe Vergleichbarkeit der Exemplare</li> </ul>
TGS 800	<p>2 Exemplare</p> <ul style="list-style-type: none"> <li>- starke bis sehr starke Leitwert-Zunahme bei Papier-, Baumwoll- und Silikonbrand bei beiden Exemplaren</li> <li>- deutliche Zunahme bei Versuch mit Ethanol (beide Exemplare)</li> <li>- uneinheitliches Verhalten bei sonstigen Versuchen</li> <li>- geringe Vergleichbarkeit der Exemplare</li> </ul>
TGS 812	<p>2 Exemplare</p> <ul style="list-style-type: none"> <li>- insgesamt deutliche bis sehr starke Signaländerungen in nahezu allen Versuchen (sowohl Brand als auch Nicht-Brand)</li> <li>- hohe Empfindlichkeit auf Lösungsmittel</li> <li>- starke bis sehr starke Leitwert-Zunahme bei Papier-, Baumwoll-, Silikon- und Polyetherbrand sowie bei Brand von Sitzmaterial</li> <li>- deutliche Zunahme bei Teflonbrand</li> <li>- sehr starke Leitwert-Zunahme bei Zigarettenrauch und Haarspray</li> </ul> <p>gute Vergleichbarkeit der beiden Exemplare</p>

TGS 813	Einzelexemplar - deutliche bis starke Leitwert-Zunahme bei allen Brandversuchen mit Papier, Baumwolle und Silikon - starke bis sehr starke Leitwert-Zunahme bei Zigarettenrauch, Rasierwasser und Haarspray
---------	---

#### 4.4 Sensorsignale bei Klimakammermessungen

Bezüglich Temperatur- und Druckabhängigkeit lassen sich die eingesetzten Sensoren in zwei Kategorien einteilen.

Die Auswirkungen des Umgebungsdruckes im untersuchten Druckbereich von 200-1100hPa auf die Sensoren FIS 4 und FIS 19 ist vernachlässigbar gering.

FIS 19 zeigt darüber hinaus fast keine Abhängigkeit des Sensorgrundleitwertes von der Umgebungstemperatur.

Bei allen anderen Sensoren (FIGARO und UST) wirkt sich eine Druck- sowie Temperaturänderung signifikant auf die Signalgröße aus. Diese ist beim UST 6000 am geringsten und beim FIGARO 812 am stärksten zu beobachten.

Grundsätzlich hängen die Sensorgrundleitwerte von der Umgebungstemperatur in der Weise ab, daß steigende Temperaturen im Halbleiter zu einem erhöhten Grundleitwert führen (PTC-Halbleiter). Leider konnte in den Vergleichsmessungen von Temperatur- und Spannungsgeregelten UST-Sensoren keine Abnahme der Temperaturabhängigkeit durch die zu diesem Zeitpunkt implementierte Temperaturregelung festgestellt werden:

Bei einem Vergleich der UST 7000 Sensoren mit und ohne Temperaturregelung (mit: Box 1 und 5, ohne: Box 6 und 7) ist kein signifikanter Unterschied erkennbar. Die Sensorgrundleitwerte folgen in beiden Fällen dem Verlauf der Umgebungstemperatur, d.h. der Grundleitwert der temperaturgeregelten Sensoren verändert sich ebenfalls bei einer Variation der Umgebungstemperatur. Dies bedeutet, daß die Temperaturregelung nicht ausreicht, um die Temperaturvariation der Umgebung zu kompensieren.

Da die temperaturbedingten Signaländerungen teilweise in der Größenordnung der bei einem Schwelbrand zu erwartenden liegen, kann im Bedarfsfall eine rechnerische Kompensation derselben notwendig werden (Kennfeldaufnahme).

#### 4.5 Lösungsmittelnachweis mit Halbleitersensoren

##### 4.5.1 Ermittlung der optimalen Betriebstemperaturen

Bei folgenden Heizspannungswerten ist die Empfindlichkeit auf das jeweilige Zielgas am größten:

Sensor	Benzol	Diethylether	tert-Butyl.	1,2-Propylen.	2-Methyl.
UST 1000	3,5V	3,5V	3,0V-3,5V	3,5V	3,0V
UST 3000	3,0V	3,5 - 4,0V	4,5V	3,0V-4,0V	3,0V
UST 7000	3,5V	4,5 - 5,0V	5,0V	4,5V	4,0V
TGS 2620	4,0V	4,0V	3,5V	4,0V	4,0V
TGS 822	4,0 - 4,5V	4,0 - 4,5V	3,5V	3,5V-4,0V	4,0V

Zur Erreichung möglichst hoher Querempfindlichkeiten auf alle Gase werden folgende Betriebsspannungen festgelegt:

Sensor	$V_{\text{Heiz}}$
UST 1000	3,5V
UST 3000	3,5V
UST 7000	4,5V
TGS 2620	4,0V
TGS 822	4,0V

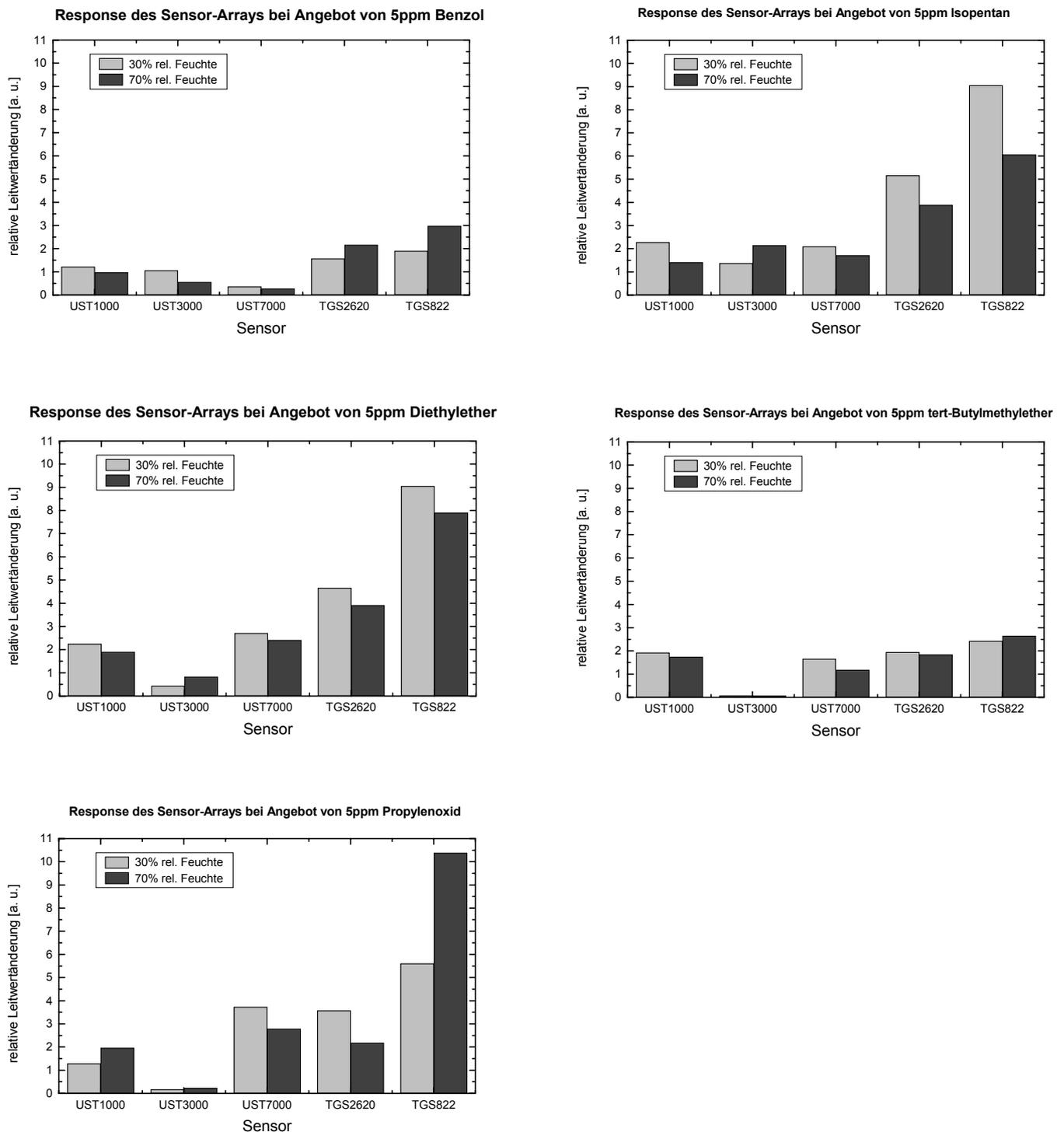
#### 4.5.2 Feuchtequerempfindlichkeit

Bei diesen Heizspannungen betrug die relative Leitwertänderung bei einem Gasangebot von 5ppm gegenüber Umgebungsluft:

	Benzol		Diethylether		tert-Butyl.		1,2-Propylen.		2-Methyl.	
	30% r.h.	70% r.h.	30% r.h.	70% r.h.	30% r.h.	70% r.h.	30% r.h.	70% r.h.	30% r.h.	70% r.h.
UST 1000	1.23	0.98	2.25	1.90	1.29	1.97	2.28	1.41	1.93	1.75
UST 3000	1.06	0.56	0.43	0.83	0.17	0.23	1.38	2.15	0.08	0.07
UST 7000	0.36	0.28	2.71	2.41	3.74	2.79	2.10	1.71	1.66	1.19
TGS 2620	1.57	2.16	4.67	3.92	3.58	2.19	5.17	3.89	1.95	1.85
TGS 822	1.90	2.98	9.06	7.90	5.61	10.39	9.05	6.06	2.43	2.65

### 4.5.3 Response des Sensor-Arrays für verschiedene Gasangebote

Die fünf zu detektierenden Lösungsmittel können anhand des charakteristischen Signalmusters des Sensor-Arrays eindeutig identifiziert werden:



**Abb. 4.1:** Response des Sensor-Arrays bei Gasangebot

Der Einfluß der Luftfeuchte hängt von der gewählten Betriebstemperatur des Halbleitersensors sowie vom nachzuweisenden Gas ab.

### 4.5.4 Teillektivitäten der Einzelsensoren

Ermöglicht wird die eindeutige Identifizierung des Lösungsmittels anhand der unterschiedlichen Teillektivitäten der eingesetzten Sensoren, wie man anhand der folgenden Abbildungen erkennen kann:

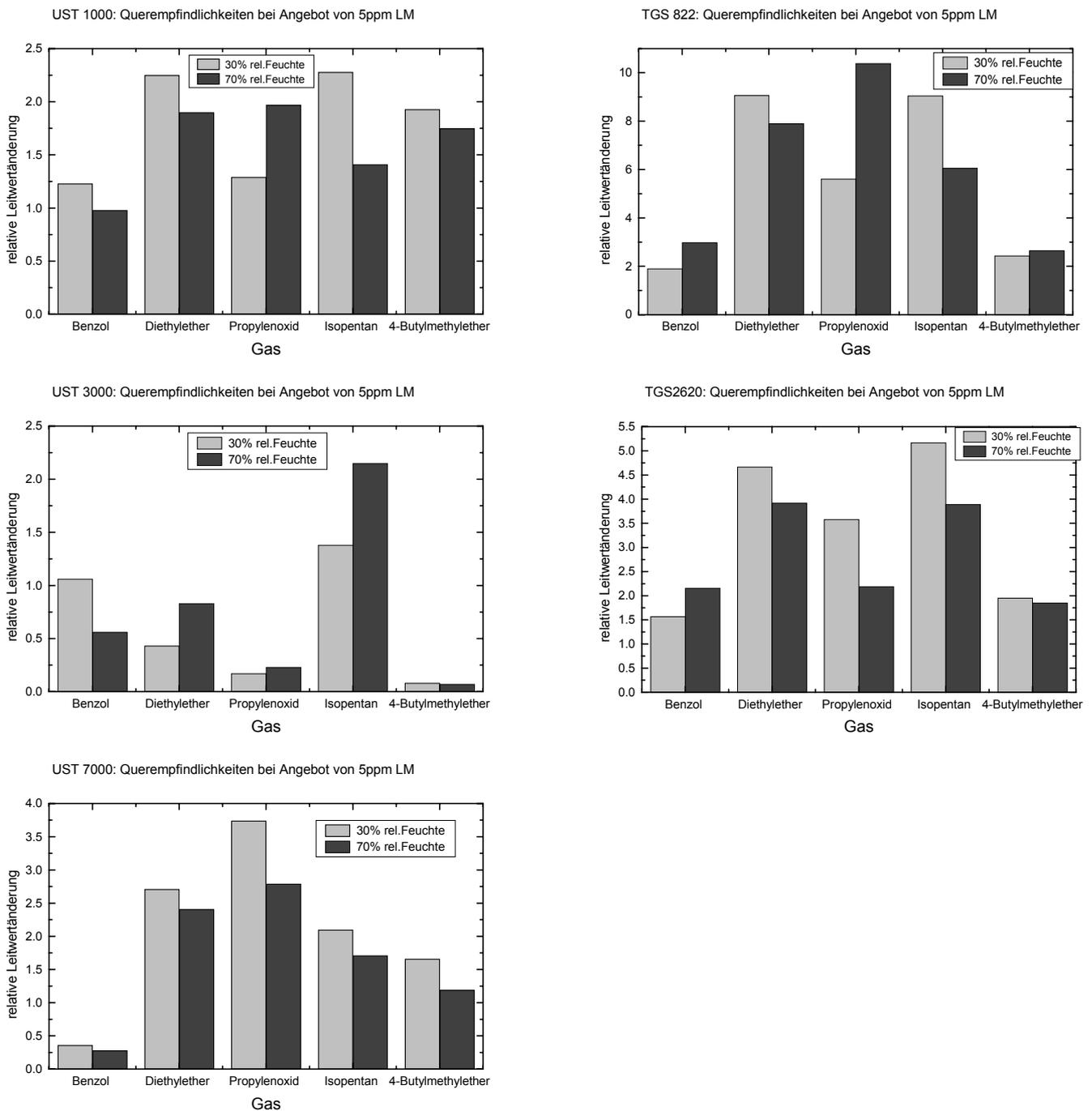
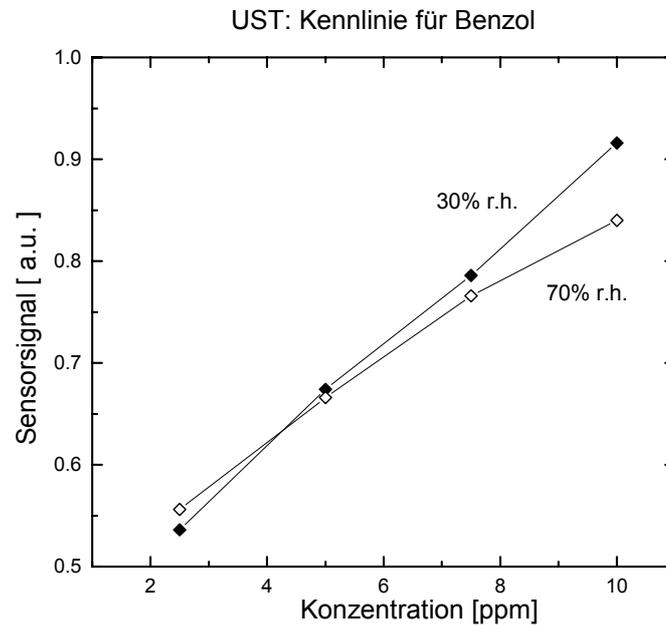


Abb. 4.2: Querempfindlichkeiten der Sensoren

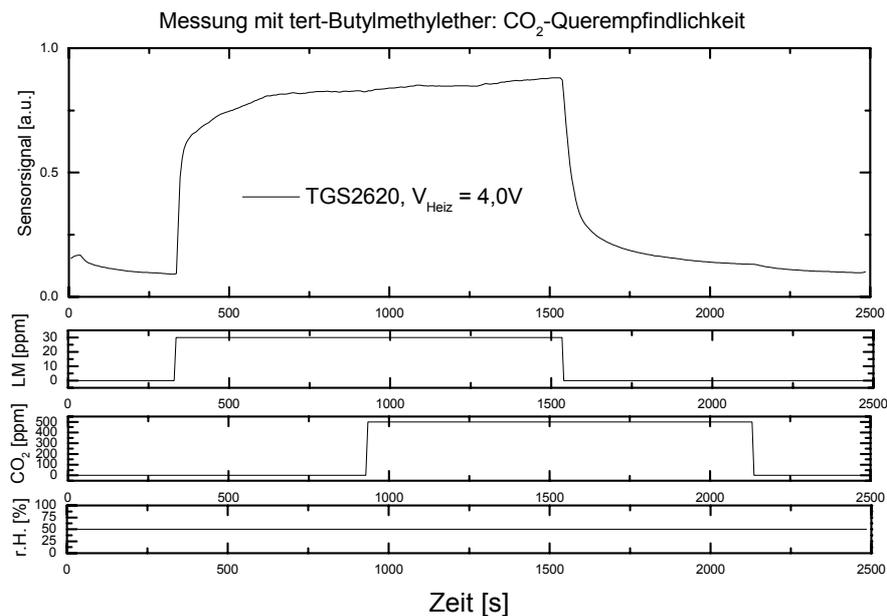
Ist das anwesende Lösungsmittel mit Hilfe des (eindeutigen) Sensor-Array-Signalmusters identifiziert, kann anhand der Sensorkennlinien eine Konzentrationsbestimmung vorgenommen werden:

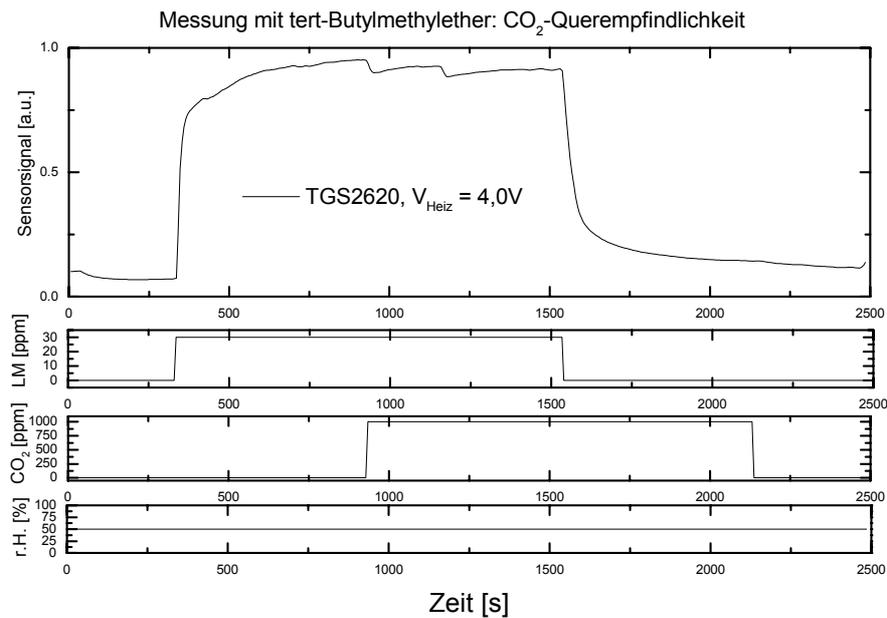


*Abb 4.3: Sensor-Kennlinie*

#### 4.5.5 CO<sub>2</sub>-Querempfindlichkeit

Es wurden zwei CO<sub>2</sub>-Hintergrundgaskonzentrationen angeboten: 500ppm und 1000ppm (s. Abb. 4.4). Die Halbleitersensoren UST 1000, UST 3000, TGS 822 und TGS 2620 zeigen keine bzw. im Fall des UST 3000 eine vernachlässigbar geringe CO<sub>2</sub>- Querempfindlichkeit im vorgegebenen Konzentrationsbereich. Lediglich der Sensor UST 7000 reagiert bei gleichzeitigem Lösungsmittelangebot mit einer leichten Leitwertabnahme. 500ppm CO<sub>2</sub> führen im Fall von tert-Butylmethylether zu einer Minderanzeige von ca. 20%.





*Abb 4.4: Angebot von 500 bzw 1000ppm CO<sub>2</sub>*

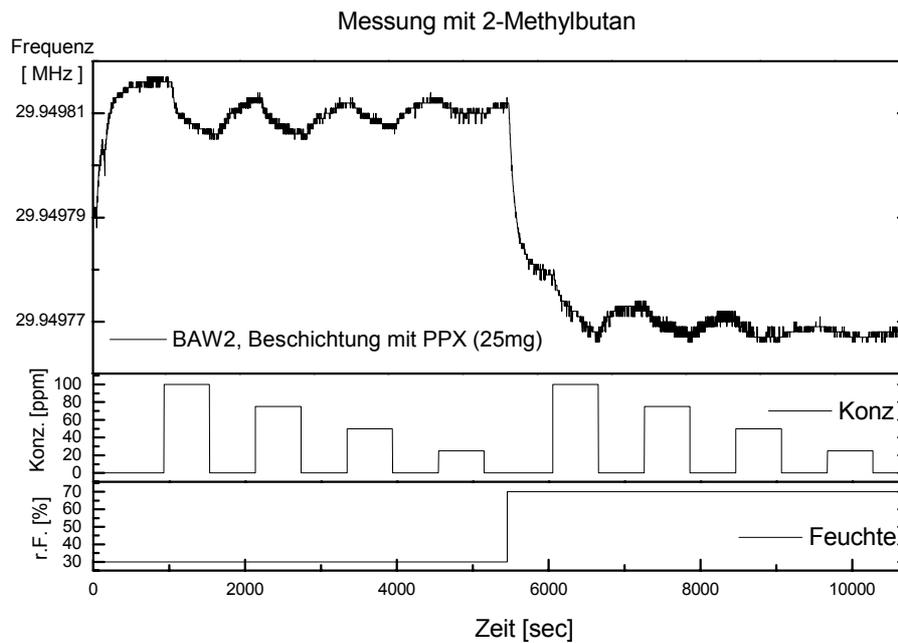
#### 4.6 Lösungsmittelnachweis mit massensensitiven Sensoren

Es wurden zwei Substrat-Typen unterschiedlicher Hersteller verwendet. Typ 1 wurde mit von Poly(p-xylylen) (PPX) beschichtet. Die Beschichtungen wurden mit Hilfe eines Pyrolyseverfahrens am Institut für Makromolekulare Chemie der Philipps-Universität Marburg vorgenommen. Die Sensoren laufen stabil, zeigen aber gegenüber den im Rahmen dieser Untersuchung vermessenen Lösungsmitteln nur geringe Querempfindlichkeiten.

Die Substrate vom Typ 2 wurden mit Polystyrol, Poly( $\alpha$ -phenyl-para-xylylen), Polymethylmethacrylat (PMMA), Poly-L-Lactid und Polycarbonat mittels Tauchverfahren beschichtet. Diese Polymere sollten für die interessierenden Lösungsmittel besser geeignet sein. Leider ergaben sich massive Probleme bei der Inbetriebnahme dieser Sensoren. Keiner war nach dem Beschichtungsvorgang zu einem Schwingvorgang im Bereich der ursprünglichen Eigenfrequenz von 30MHz zu bewegen. Lediglich der mit Poly-L-Lactid beschichtete Sensor konnte neben den drei PPX-BAWs für die Messungen verwendet werden.

##### 4.6.1 Reaktion auf Gasangebot

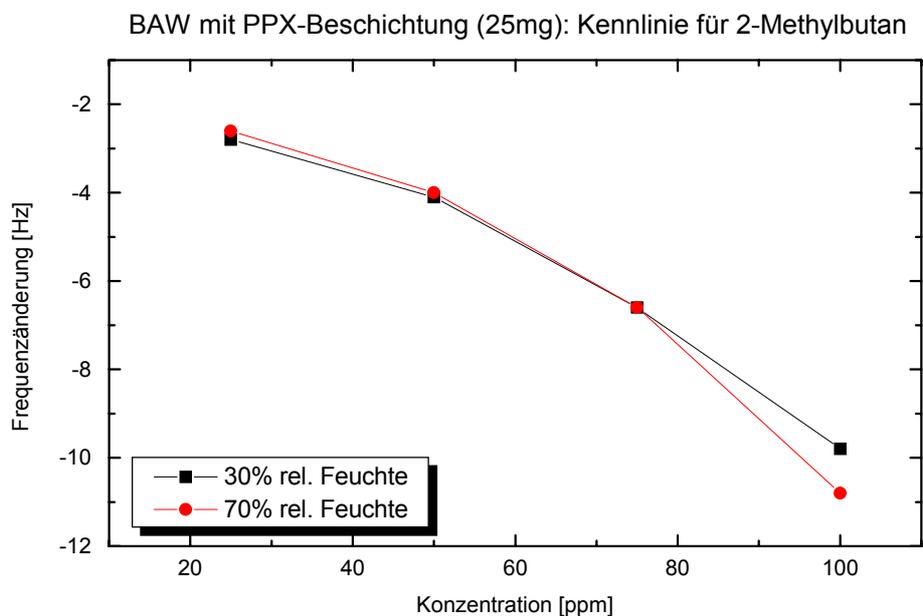
Abb. 4.5 zeigt eine Messung mit einem der PPX-beschichteten BAWs an 2-Methylbutan. Das Umschalten der relativen Luftfeuchtigkeit von 30% auf 70% führt zu einer relativ starken Frequenzerniedrigung (verglichen mit den von den Lösungsmittelangeboten verursachten).



*Abb. 4.5: 2-Methylbutan-Messung mit PPX-BAW*

#### 4.6.2 Feuchtequersensitivität

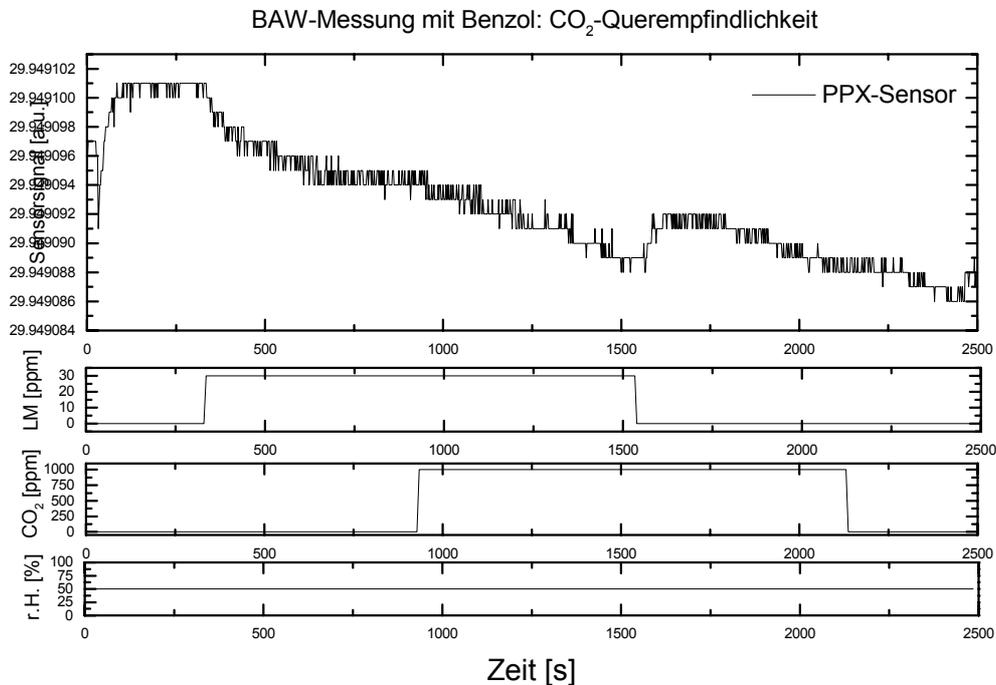
Die vom Lösungsmittelangebot verursachte Frequenzänderung, die als Sensorsignal verwendet wird, unterliegt jedoch nur einem geringen Feuchteinfluß, wie man der in Abb. 4.6 dargestellten Kennlinie entnehmen kann:



*Abb. 4.6: PPX-BAW-Kennlinie für 2-Methylbutan*

### 4.6.3 CO<sub>2</sub>-Querempfindlichkeit

Es wurde eine CO<sub>2</sub>-Hintergrundgaskonzentration von 1000ppm einem Lösungsmittelangebot von 50ppm Benzol überlagert (s. Abb. 4.7). Eine meßbare CO<sub>2</sub>-Querempfindlichkeit liegt bei keiner der getesteten Polymerschichten vor.



*Abb. 4.7: Angebot von 1000ppm CO<sub>2</sub>*

### 4.6.4 Zusammenfassung der Ergebnisse

Im Rahmen der am Institut für Angewandte Physik durchgeführten Machbarkeitsstudie wurde mit Hilfe fünf kommerzieller Halbleitersensoren ein Sensor-Array zum selektiven Nachweis der Lösungsmittel Benzol, Diethylether, tert-Butylmethylether, 1,2-Propylenoxid sowie 2-Methylbutan unter Laborbedingungen aufgebaut.

Bei den Versuchsreihen wurde ein Nachweis von Einzelgasen in Luft (keine Gemische) vorgenommen. Die aufgeführten Lösungsmittel können in einem Konzentrationsbereich bis unter 5ppm nachgewiesen werden.

Die Querempfindlichkeit des Gesamtsystems ist gegenüber CO<sub>2</sub> im vorgegebenen Konzentrationsbereich bis 1000ppm vernachlässigbar. Der Einfluß der Luftfeuchtigkeit kann auf rechnerischem Wege eliminiert werden.

Der Prozeß der Lösungsmitteldetektion vollzieht sich zweistufig. Anhand des vom Sensor-Array ausgegebenen Signalmusters wird zunächst das Lösungsmittel identifiziert. In einem zweiten Schritt wird mit Hilfe der Kennlinien der für das jeweilige Lösungsmittel am besten geeigneten Sensoren eine Konzentrationsbestimmung vorgenommen.

Bei den massensensitiven Sensoren sind vor einem industriellen Einsatz weitere Entwicklungsarbeiten erforderlich. Ihre Empfindlichkeit reicht derzeit noch nicht an das Niveau kommerziell verfügbarer Halbleitersensoren heran. Weiteres Entwicklungspotential ist in der Verwendung neuer Polymerschichten mit adäquaten Aufbringungstechniken zu sehen.

## 5. Diskussion

### Übersicht

5.1 Vorgehen zur Auswahl geeigneter Gassensoren für die Branddetektion.....	67
5.2 Vorteile eines Kombimelders gegenüber einem optischen Detektor.....	71
5.2.1 Geringer Einfluß rauchhemmender Materialien auf die Detektionssicherheit und - geschwindigkeit.....	71
5.2.2 Unterdrückung von staub- und aerosolinduzierten Fehlalarmen.....	72
5.2.3 Zusammenfassung: Vorteile eines Brandmeldesystems mit Gassensoren.....	73
5.3. Maßnahmen zur verbesserten Nutzung des Informationsgehaltes von Sensorsystemen..	74
5.3.1 Nanostrukturierung von SnO <sub>2</sub> -Sensoren zur Steigerung der Nachweisempfindlichkeit .....	74
5.3.2 Temperaturmodulierte Sensorbetriebsweise zur verbesserten Ausnutzung der sensorischen Information.....	77
5.3.3 Streßtests zur Verifikation der Robustheit von Merkmalen.....	81
5.3.3.1 Robustheitstest durch Variation der Versuchsbedingungen.....	81
5.3.3.2 Stabilitätsuntersuchung in Langzeitversuchsreihen.....	84
5.3.3.3 Surrogatdaten zur gezielten Untersuchung des Einflusses von Parameterschwankungen.....	85
5.3.4 Mathematische Verfahren zur verbesserten Nutzung des Informationsgehaltes von Sensorsystemen.....	87
5.3.4.1 Akquisition von Regelwissen durch SOM.....	87
5.3.4.1.1 Anwendungsbeispiel: Zuverlässige und fehlalarmsichere Detektion von Braunkohleschmelbränden.....	87
5.3.4.1.2 Extraktion von Expertenwissen aus einer Datenbasis.....	88
5.3.4.1.3 Umsetzung des akquirierten Wissens in ein regelbasiertes Fuzzy- Expertensystem.....	92
5.3.4.2 Berücksichtigung komplexer Zielfunktionen.....	93
5.3.4.3 Bayes Strategie zur Mustererkennung.....	97
5.3.4.3.1 Parametrische Verfahren.....	99
5.3.4.3.1.1 Bayes Normalverteilungsklassifikator und Maximum-Likelihood Normalverteilungsklassifikator.....	99
5.3.4.3.1.2 Mahalanobis Abstandsklassifikator.....	99
5.3.4.3.1.3 Euklidischer Abstandsklassifikator.....	99
5.3.4.3.2 Nichtparametrische Verfahren.....	99
5.3.4.3.2.1 Parzen-Fenster-Klassifikator.....	99
5.3.4.3.2.2 Nächste-Nachbar Verfahren.....	100
5.3.4.3.2.3 Probabilistische Neuronale Netze (PNN) und General Regression Neural Network (GRNN).....	101
5.3.4.3.2.3.1 Probabilistische Neuronale Netze (PNN).....	101
5.3.4.3.2.3.2 Das General Regression Neural Network (GRNN).....	104
5.3.4.3.2.3.3 Beurteilung.....	108
5.3.4.4 Merkmalsextrahierendes Mustererkennungsverfahren (MSMEV).....	111
5.3.4.4.1 Einfluß der freien Parameter.....	113
5.3.4.4.2 Zerlegung von Entscheidungsprozessen.....	116
5.4 Erkennung kritischer Zustände in Dynamischen Systemen.....	122
5.4.1 Log-periodische Oszillationen als Frühindikatoren für Phasenübergänge.....	122
5.4.2 Beispiel: Nasdaq-Korrektur 2000.....	124

### 5.1 Vorgehen zur Auswahl geeigneter Gassensoren für die Branddetektion

In die abschließende Bewertung flossen die Daten aus den Gasmischermessungen sowie aus 19 Versuchen mit Brand- und Störereignissen ein. Jedes Ereignis wurde von 7 Geräten, die am gleichen Ort installiert waren, mit je einem Rauchmelder und drei Halbleiter-Gassensoren registriert.

#### Auswahlkriterien

Bei der Auswahl geeigneter Sensoren müssen folgende Faktoren berücksichtigt werden:

1. Sensitivität: Wie gut spricht der Sensor auf das jeweilige Zielgas an?
2. Selektivität: Wie stark sind die Querempfindlichkeiten gegenüber Störgasen?
3. Unempfindlichkeit gegenüber variierenden Umgebungsparametern: Druck, Feuchte, Temperatur
4. Reproduzierbarkeit innerhalb eines Exemplares: Wie ähnlich verhält sich ein Sensor zu sich selbst, wenn er dem gleichen Ereignis mehrmals ausgesetzt wird?
5. Exemplarstreuungen: wie ähnlich verhalten sich die Sensoren eines Typs unter identischen Bedingungen (gleiches Gasangebot)?

#### Selektionsverfahren

Die eingesetzten Selektionsalgorithmen untersuchen, mit welchen Sensoren eine möglichst geringe Fehlklassifikationsrate zu erzielen ist. Als Fehlklassifikation wurde sowohl eine fälschliche Brandmeldung als auch das Nichterkennen eines Brandes gewertet.

Zur Bewertung wurden in dieser Stufe nur die (statischen) Sensorendwerte verwendet. Zeitliche Ableitungen wurden nicht berücksichtigt.

Als Sensorsignale wurden die Sensorleitwerte in drei verschiedenen Aufbereitungsstufen herangezogen:

- a) absolute Höhe des Sensorleitwertes
- b) absolute Änderung des Sensorleitwertes gegenüber dem Grundleitwert
- c) relative Änderung des Sensorleitwertes gegenüber dem Grundleitwert

Die Untersuchungen zeigen, daß die Eignung der Größen a) bis c) vom Sensortyp abhängt (s. Anforderung 4.)

Zur Beurteilung der Güte der Sensorsignale wurde das im folgenden näher beschriebene Prinzip angewendet:

Mit Hilfe der durchgeführten Versuchsbrände und simulierten Störereignisse (z.B. Haarspray) wurde eine Signalmuster-Referenzbibliothek angelegt. Neu gemessene Ereignisse werden mit den abgespeicherten Signalmustern verglichen und anhand der Ähnlichkeit zu diesen klassifiziert. Je ähnlicher das aktuelle Signalmuster einem der abgelegten ist, desto stärker wird das neue Ereignis diesem zugeordnet. Das Klassifikationsergebnis ist eine gewichtete Mittelung über die Klassenzugehörigkeiten der ähnlichsten Signalmuster.

Das Verfahren setzt implizit die Anwendbarkeit des starken Kausalitätsprinzips ("ähnliche Ursachen führen zu ähnlichen Wirkungen") voraus. Aus der Untersuchung nichtlinearer dynamischer Systeme geht hervor, daß bereits kleinste Abweichungen in den Ursachen große

Auswirkungen auf das Ergebnis haben können. Dem Auffinden von in diesem Sinne stabilen Ursache-Wirkungs-Beziehungen kommt daher eine entscheidende Bedeutung zu.

Bei der Halbleiter-Gassensorauswahl wurden zwei verschiedene Selektionsstrategien angewendet:

1. Aus der Menge der insgesamt 9 untersuchten Halbleiter-Gassensortypen wurde der beste nach dem oben aufgeführten Auswahlkriterium ausgewählt (27 Merkmale durch 3 verschiedene Aufbereitungsmethoden, weitere Merkmalerhöhung durch Quotientenbildung, z.B. Signal UST 6000 / Signal UST 1000 als Maß für das H<sub>2</sub> zu CO Verhältnis). In einem mehrstufigen Auswahlprozeß wird das beste gefundene Einzelkriterium mit demjenigen ergänzt, das das Klassifikationsergebnis am stärksten verbessert.

Ab einer gewissen Merkmalsanzahl wird das Klassifikationsergebnis wieder schlechter und der Vorgang kann abgebrochen werden.

2. Die Menge der einzusetzenden Sensoren/Merkmale wird vorher festgelegt. Es wird die beste 1er, 2er, etc Kombination ausgewählt

Die Vorgehensweisen 1 und 2 unterscheiden sich dahingehend, daß in 1 der beste Einzelsensor ausgewählt wird, und die Aufgabe der weiteren Sensoren darin besteht, dessen Schwächen auszugleichen. Verfahren 2 wählt solche Sensoren aus, die in Kombination besonders gut geeignet sind, d.h. auch Kombinationen aus solchen, die keine guten Einzelsensoren sind.

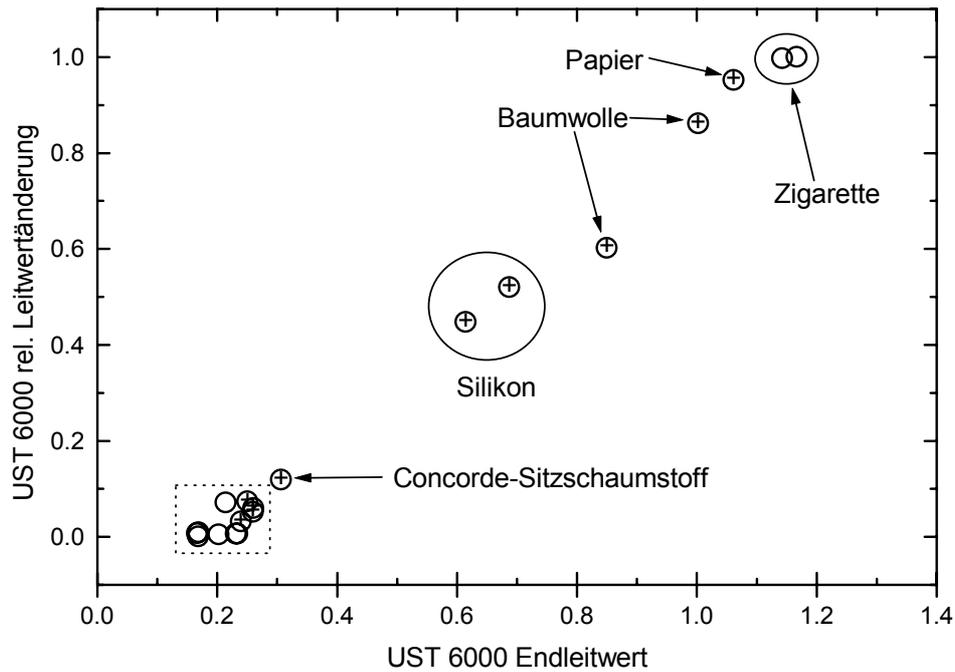
Folgende Sensoren wurden von den Selektionsalgorithmen unter Normalbedingungen bevorzugt:

H <sub>2</sub> -Sensoren	CO-Sensoren	sonstige Sensoren
UST 6000, FIS 4, FIS 19	TGS 813, TGS 812	(UST 7000)

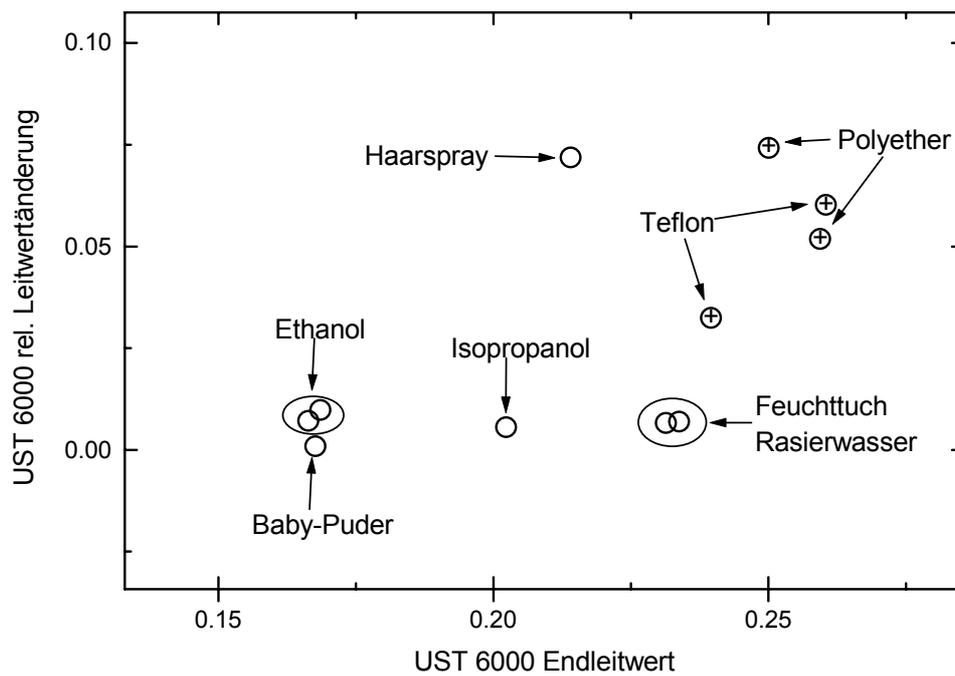
Berücksichtigt man zusätzlich die Ergebnisse der Klimakammeruntersuchungen, so erscheint die Kompensierbarkeit der beim UST 7000 auftretenden Signalbeeinflussungen allerdings fraglich.

Als hinreichend unempfindlich gegenüber klimatischen "Störungen" erweist sich der UST 6000, der bereits als Einzelsensor in der Lage ist, mit einer referenzbibliotheksbasierten Signalverarbeitung alle durchgeführten Versuchsbrände zuverlässig zu detektieren. Er ist resistent gegenüber allen Störereignissen mit Ausnahme des Zigarettenrauchs (s. Anhang).

Folgende Abbildung zeigt die untersuchten Ereignisse im Klassifikationsraum:



**Abb. 5.1:** Klassifikationsebene UST 6000



**Abb. 5.2:** Ausschnittsvergrößerung aus Abb. 5.1 (gestricheltes Rechteck)

Zur Erhöhung der Detektionssicherheit kann ein weiterer Sensor mit unterschiedlicher Charakteristik, wie z.B. TGS 813 (CO-Sensor) herangezogen werden. Die Achsen der folgenden zwei Abbildungen werden durch den projizierenden Algorithmus (Sammon Nonlinear Mapping, kurz NLM, abstandserhaltend) vorgegeben:

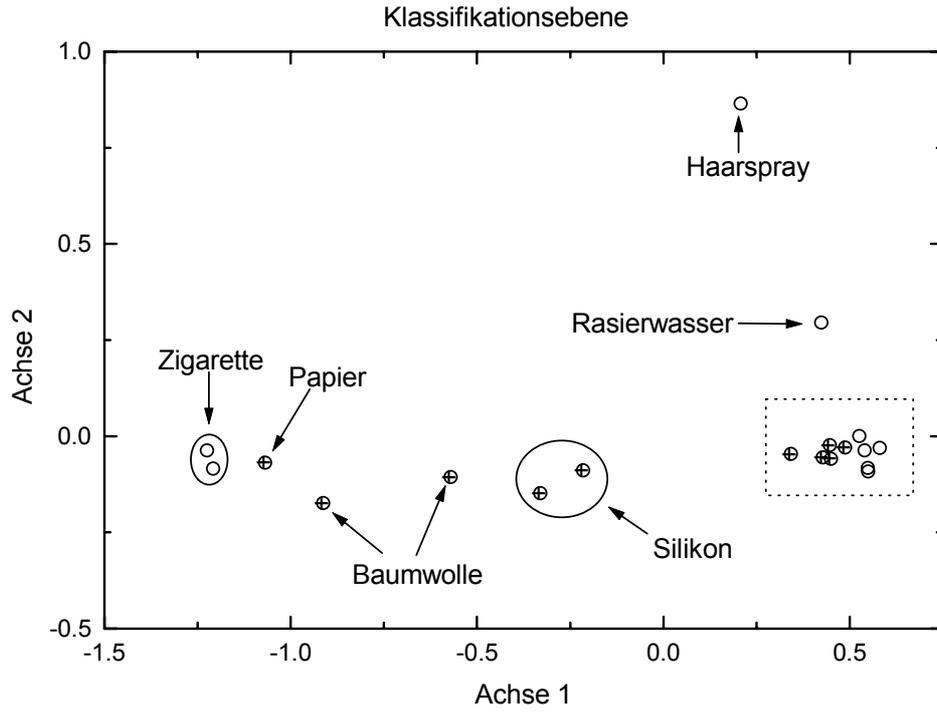


Abb. 5.3: Klassifikationsebene UST 6000 und TGS 813

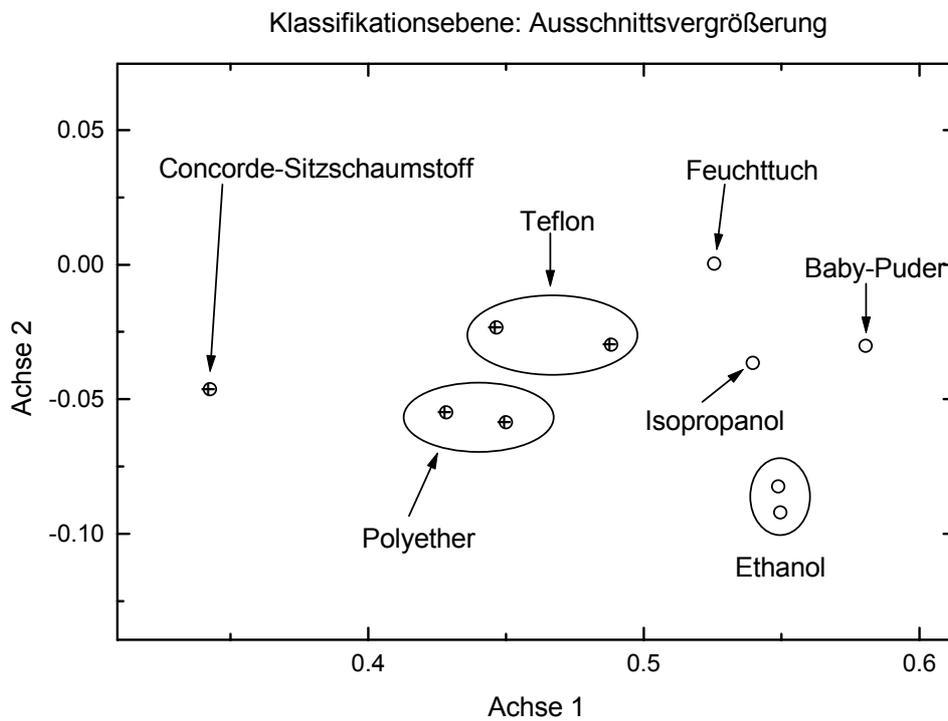
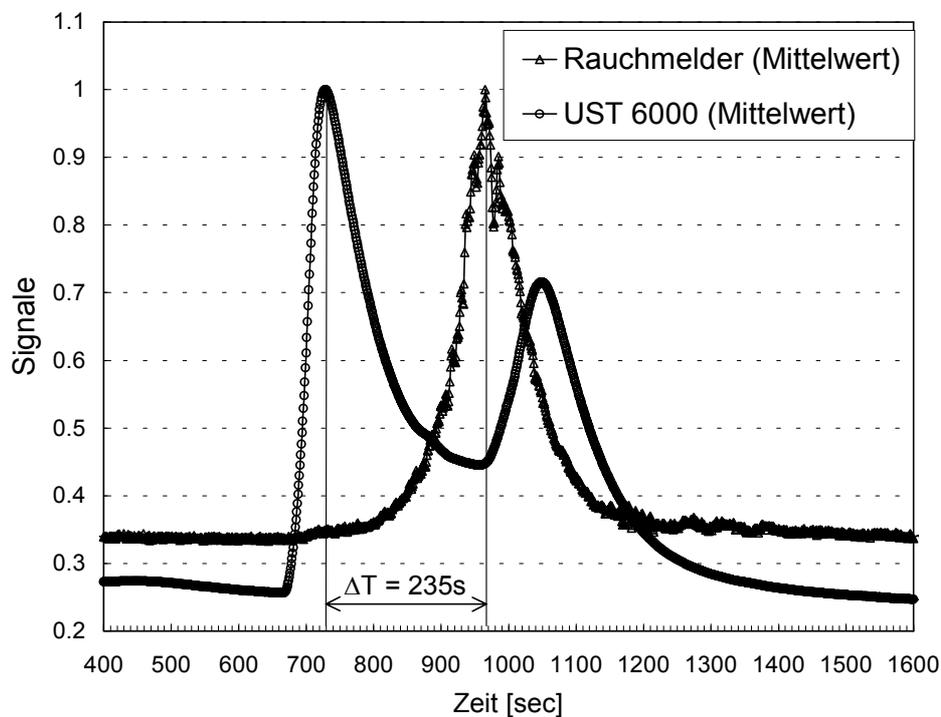


Abb. 5.4: Ausschnittsvergrößerung aus Abb. 5.3 (gestricheltes Rechteck)

## 5.2 Vorteile eines Kombimelders gegenüber einem optischen Detektor

### 5.2.1 Geringer Einfluß rauchhemmender Materialien auf die Detektionssicherheit und -geschwindigkeit

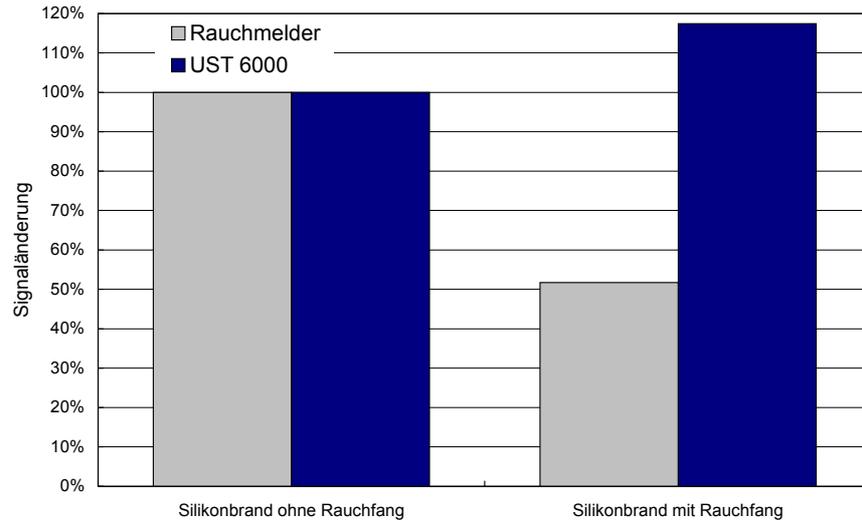
Brände sollen auch dann entdeckt werden können, wenn sie vom Detektor durch wärmedämmende bzw. luftzirkulationsbehindernde Materialien abgeschirmt sind. Gase gelangen über Diffusion zum Detektor und sind somit als Nachweismedium besser als Rauch bzw. Wärme geeignet. Am Beispiel eines Silikonschelbrandes mit und ohne Rauchfang vor dem Detektor soll der unterschiedlich starke Einfluß eines Rauchfängers (z.B. Vorhang) auf die Rauchmelder bzw., stellvertretend für die Halbleiter-Gassensoren, die Sensoren vom Typ UST 6000 gezeigt werden:



*Abb. 5.5: Einfluß eines Rauchfängers auf die Ansprechzeit*

Brände können durch Gasmelder bereits im Frühstadium, d.h. in der Entstehungsphase vor Einsatz der Rauch- und Wärmeentwicklung, anhand charakteristischer Gasemissionen erkannt werden.

*Abb. 5.5* zeigt, daß die Behinderung der offenen Rauchanströmung durch einen Rauchfang zu einem um fast 4 Minuten verzögerten Ansprechen der Rauchmelder gegenüber den Halbleiter-Gassensoren führt (Daten: Silikonbrand mit Rauchfänger).



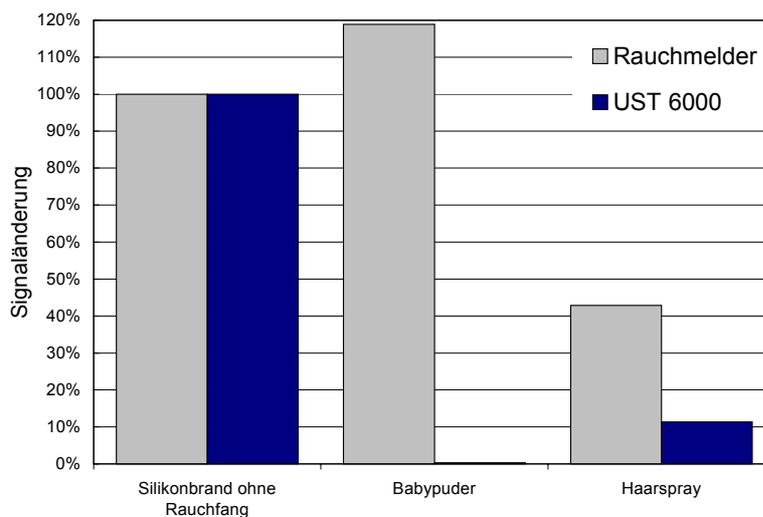
*Abb. 5.6: Einfluß eines Rauchfängers auf die Signalstärke*

Die Signaländerungen wurden für den Versuchsbrand ohne Rauchfang auf 100% normiert. Der 2. Versuchsbrand mit Rauchfang war von stärkerer Intensität (Signaländerung des UST 6000 stärker als beim ersten Brand). Dennoch erreicht die Signaländerung der Rauchmelder nur noch knapp über 50% des ursprünglichen Wertes.

### 5.2.2 Unterdrückung von staub- und aerosolinduzierten Fehlalarmen

Fehlalarme durch Partikel/Aerosole, die nicht Produkte einer Verbrennung sind, lösen bei Halbleitersensoren keine Fehlalarme aus.

In der folgenden Abbildung sind die Signaländerungen der Sensoren vom Typ UST 6000 denen der Rauchmelder gegenübergestellt. Die Skalierung der Signale wurde aus *Abb. 5.6* übernommen.



*Abb. 5.7: Einfluß von Haarspray und Babypuder auf die Detektoren*

Deutlich zu erkennen ist der starke Einfluß von Babypuder auf das Rauchmeldersignal; der Halbleiter-Gassensor weist keine Querempfindlichkeit auf. Die Signaländerung bei Einwir-

kung von Haarspray beträgt beim Rauchmelder mehr als 40% derjenigen bei einem Silikon-schwelbrand, die des Halbleiter-Gassensors beträgt demgegenüber nur knapp über 10%.

### 5.2.3 Zusammenfassung: Vorteile eines Brandmeldesystems mit Gassensoren

#### *Detektionssicherheit:*

Selbst komplizierteste Raumgeometrien können mit Gasmeldern zuverlässig überwacht werden, da weder ein optischer Kontakt zum Brandherd noch eine unbehinderte Strömung (Partikeltransport bei Rauchmeldern) vorausgesetzt werden muß.

#### *Falschalarmsicherheit:*

Durch die mathematische Verknüpfung mehrerer unabhängiger Brandparameter (basierend auf den gemessenen Gaskonzentrationen und deren Verlauf) läßt sich die Fehlalarmrate gegenüber einfachen schwellwertbasierten Systemen stark reduzieren.

#### *Detektionsgeschwindigkeit*

Die Gesamtzeit, die von der Entstehung bis zur Erkennung eines Brandes vergeht, setzt sich aus drei Zeitspannen zusammen:

#### 1. Die Zeit, die vergeht, bis eine brandcharakteristische Größe entsteht

Rauchmelder detektieren Aerosole, die typischerweise erst bei offenen Bränden auftreten. Gase hingegen werden bereits in einem früheren Stadium des Brandes emittiert. Im Falle eines unter Sauerstoffmangel vor sich hin schwelenden Brandes kann der daraus erzielte zeitliche Vorsprung einige Minuten bis Stunden zugunsten des Gasmelders betragen.

#### 2. Die Zeit, die vergeht, bis diese brandcharakteristische Größe braucht, um zum Sensor zu gelangen

Beim Transport durch Strömung unterscheiden sich Gasmelder in diesem Punkt nicht wesentlich von Rauchmeldern. Wird die Strömung allerdings behindert (z.B. durch einen Vorhang), gelangen die Gase per Diffusion immer noch zum Sensor, während die makroskopischen Partikel im ungünstigsten Fall vollständig zurückgehalten werden -> der Rauchmelder ist blind, der Gasmelder spricht immer noch an.

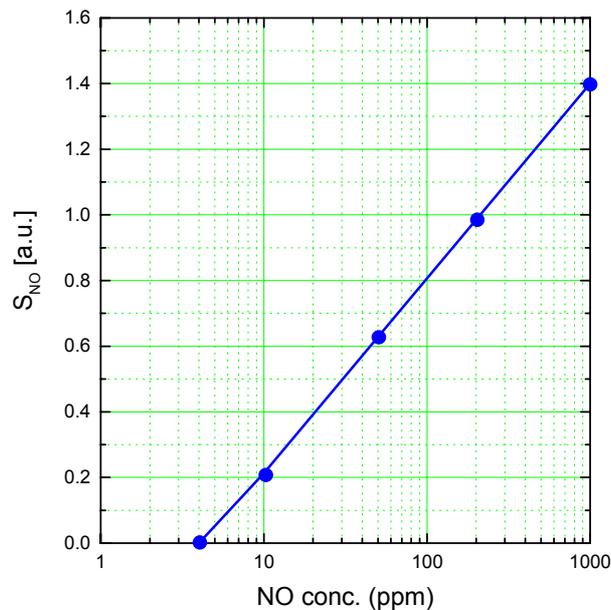
#### 3. Die Ansprechzeit der Sensoren

Die eigentliche Ansprechzeit der Sensoren liegt beim Gasmelder im Bereich weniger Sekunden. Bei einem sich langsam ausbreitenden Schwelbrand kann diese Zeitspanne vernachlässigt werden.

### 5.3. Maßnahmen zur verbesserten Nutzung des Informationsgehaltes von Sensorsystemen

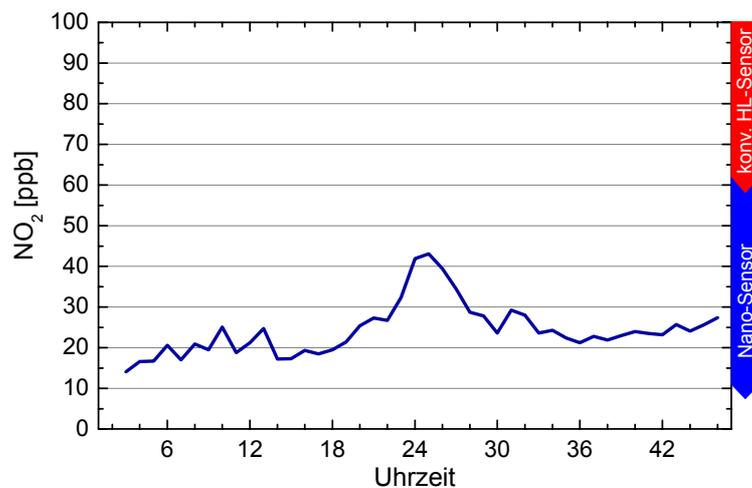
#### 5.3.1 Nanostrukturierung von SnO<sub>2</sub>-Sensoren zur Steigerung der Nachweisempfindlichkeit

Konventionelle Halbleitersensoren weisen eine Grenzempfindlichkeit auf, die typischerweise im unteren ppm-Bereich liegt. Die folgende Abb. 5.8 zeigt exemplarisch die Kennlinie eines TiO<sub>2</sub>-Sensors bei Angebot von NO.



**Fig. 5.8:** TiO<sub>2</sub>-Sensorkennlinie für NO [Sat 94]

Konzentrationen von weniger als 4 ppm NO liegen unterhalb der Nachweisgrenze. Ein Vordringen in diesen Konzentrationsbereich würde einen Einsatz von Halbleitersensoren in Anwendungsfeldern ermöglichen, der bislang kostenintensiven Analyseverfahren vorbehalten war.



**Abb. 5.9:** NO<sub>2</sub>-Tageskonzentrationsverlauf in Gießen, Meßverfahren: Gasphasen-Chemolumineszenz [Quelle : Hessische Landesanstalt für Umwelt (HLfU)]

Eine interessante Möglichkeit zur Herstellung von Hochempfindlichkeitssensoren besteht in der Verwendung von Körnern mit einem Durchmesser unterhalb der doppelten Debye-Länge  $L_D$

$$L_D = \sqrt{\frac{\varepsilon_0 \varepsilon_r k_B T}{e^2 n_b}} \quad (5.1)$$

Körner dieser Größe sind i.a. komplett an Ladungsträgern verarmt, die Bänder verlaufen flach über den gesamten Kornradius, d.h. es gibt keine Bandverbiegungseffekte an der Oberfläche. In diesem sogenannten Flachbandfall kann man die einzelnen Körner näherungsweise als digital schaltende Elemente auffassen, so daß das Leitverhalten des Sensors davon abhängt, ob und wie viele Pfade leitender Körner zwischen den auslesenden Elektroden ausgebildet werden.

Die folgenden Abbildungen zeigen ein vereinfachtes Gassensormodell, dessen Oberfläche aus einer einlagigen Körnerschicht „hinreichend kleiner“ Körner besteht. Bei Anwesenheit von Umgebungssauerstoff und Abwesenheit von reduzierenden Gasen sollten alle Körner einen hochohmigen Zustand annehmen. Bei Adsorption reduzierender Gase werden die betreffenden Körner „leitend“ geschaltet:

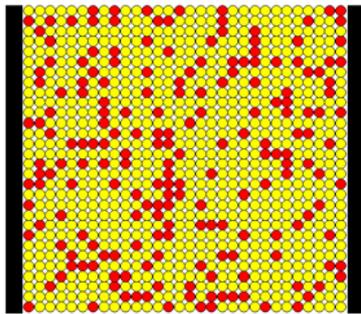


Abb. 5.10a:  $p=0,2$

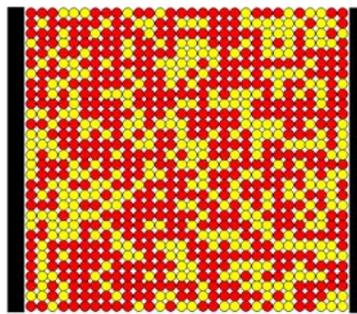


Abb. 5.10b:  $p=0,5928$

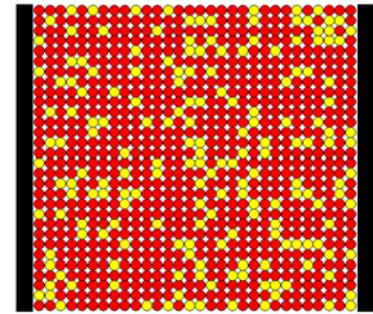


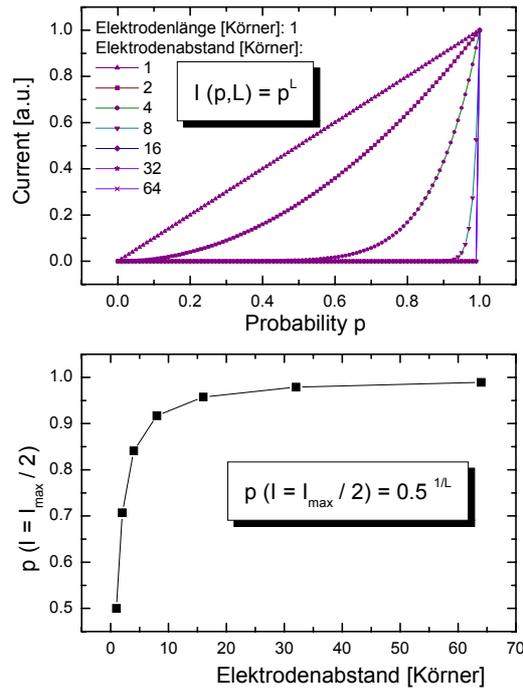
Abb. 5.10 c:  $p=0,8$

In Abb. 5.10a sind 20% der Körner durch die Anwesenheit reduzierender Gase leitend geschaltet. Da dieser Bruchteil unterhalb der Perkolationsschwelle liegt, können sich keine leitenden Strompfade zwischen den Elektroden des „Sensors“ (schwarze Balken) ausbilden.

Oberhalb der Perkolationsschwelle bilden sich bei zunehmender Bedeckung zunehmend mehr leitende Pfade aus; die Leitfähigkeit des Sensors verhält sich proportional zur Gasbedeckung.

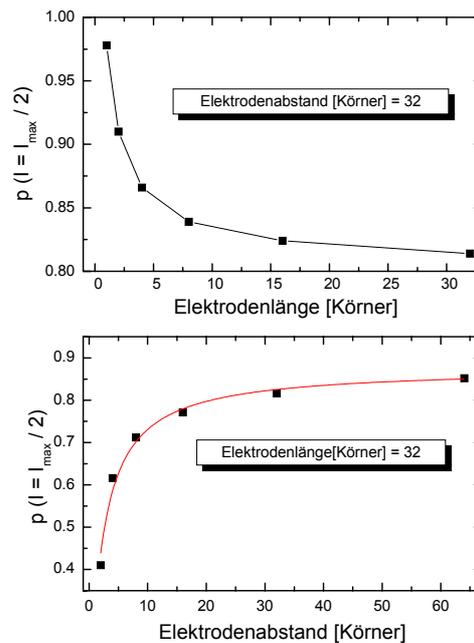
Um den erforderlichen Mindestbedeckungsgrad für das Ansprechen des Sensors zu senken, kann man sog. *finite-size* Effekte ausnutzen: Im Grenzfall einer nur einlagigen Körnerschicht zwischen den beiden Elektroden kann die untere Ansprechschwelle des Sensors beliebig klein gemacht werden, wenn die Anzahl der Körner in dieser einlagigen Schicht nur groß genug ist und die Ansprechempfindlichkeit des auslesenden ADCs hoch genug ist:

Die folgende Abbildung zeigt die Strom-Besetzungsdichte-Kennlinie eines „Ein-Korn-Elektrodenabstand“-Sensors für verschiedene Elektrodenlängen (a) bzw. die Wahrscheinlichkeit, daß ein „Ein-Korn-Elektrodenbreite“-Sensor bei unterschiedlicher Anzahl von Körnern leitend geschaltet wird (b):

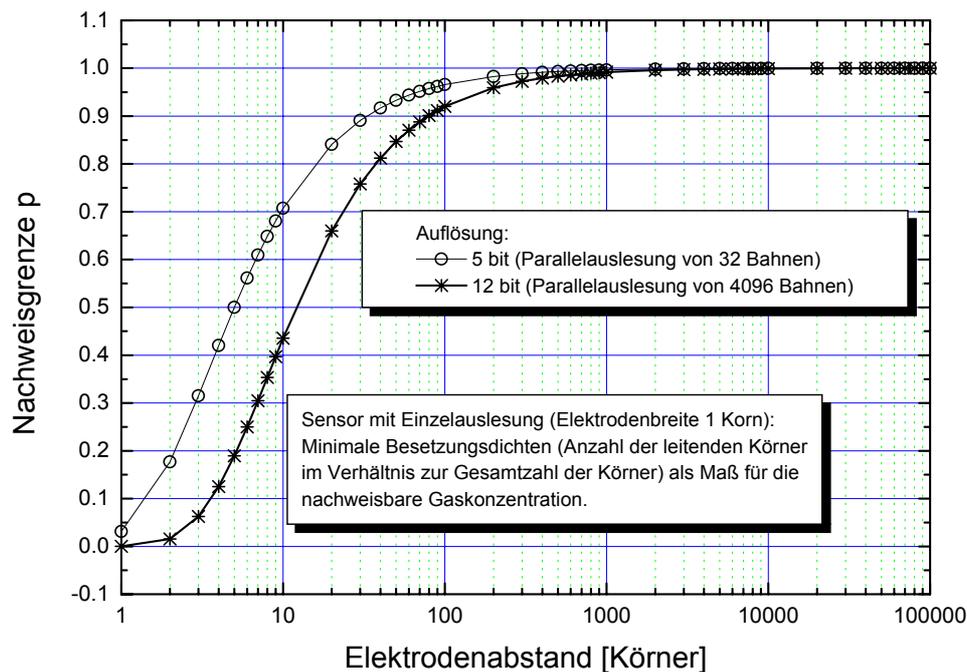


**Abb. 5.11 a):** Ein-Korn-Elektrodenabstand-Sensor bzw. **b)** Ein-Korn-Elektrodenbreite-Sensor

Die folgende Abbildung zeigt die Ergebnisse einer Simulationsrechnung, bei der zunächst der Elektrodenabstand festgehalten und die Elektrodenlänge variiert wurde (jeweils in Einheiten von „Körnern“) und anschließend bei konstant gehaltener Elektrodenlänge der Elektrodenabstand variiert wurde. Auf der Ordinate ist jeweils die Besetzungsdichte aufgetragen, die nötig ist, um einen Stromfluß zu erzeugen, der der Hälfte des maximalen Flusses entspricht ( $I=I_{\max}$   $\Leftrightarrow$  alle Körner leitend).



**Abb. 5.12:** Simulationsergebnisse



**Abb. 5.13:** Nachweisgrenze in Abhängigkeit vom Elektrodenabstand

Abb. 5.13 zeigt erforderlichen Besetzungsdichten (proportional zur Nachweisgrenze) in Abhängigkeit vom Elektrodenabstand eines Sensor mit „ein-Korn“-Elektrodenbreite: Bei gleichzeitigem Auslesen von 32 solcher Elektrodenbahnen (obere Kurve) und Parallelauslesung von 4096 solcher Bahnen (untere Kurve).

Um die Nachweisgrenze abzusenken, muß der Elektrodenabstand gesenkt werden. Bei einem Elektrodenabstand (bisheriger Stand der Technik) von  $2\mu\text{m}$  liegen 200 Körner mit Kornradius 10 nm zwischen den Elektroden. Eine Reduktion des Elektrodenabstandes auf  $0,1\mu\text{m}$  mit Hilfe von Nanostrukturierungstechnologien würde bei 24bit-Auslesung aufgrund der beschriebenen Effekte sowohl die Nachweisgrenze als auch den Dynamikbereich des Sensors um einen Faktor 5 steigern.

### 5.3.2 Temperaturmodulierte Sensorbetriebsweise zur verbesserten Ausnutzung der sensorischen Information

Wie im Kapitel Sensorgrundlagen dargestellt, lassen sich Halbleitersensoren nur bedingt auf ein einzelnes Zielgas einstellen, d.h. alle reduzierenden und oxidierenden Gase verursachen Leitwertänderungen im gassensitiven Material. Mit Hilfe von Katalysatoren und durch die Wahl optimierter Betriebstemperaturen lassen sich die Teilselektivitäten allerdings beeinflussen.

Bei der Wahl einer geeigneten Betriebstemperatur geht man in der Regel so vor, daß man verschiedene Meßreihen mit den zu detektierenden Zielgasen sowie den potentiell zu erwartenden Störgasen durchführt. Zum Aufbau eines Sensorarrays wird man sodann so verfahren, daß man für jeden Sensor die Temperatur so einstellt, daß er für eines der Zielgase besonders empfindlich, für die anderen sowie für die Störgase besonders unempfindlich ist, wobei man als Kenngröße üblicherweise die Leitwertänderung relativ zum Grundleitwert wählt.

Bei kleinen Systemen, d.h. solchen mit wenigen zu überwachenden Gasen und wenigen Sensoren, lassen sich so mit relativ geringem Aufwand zuverlässige Überwachungssysteme konzipieren.

Je mehr Gase gleichzeitig überwacht werden sollen, desto eher ist man darauf angewiesen, Mehrfachauswertungen für jeden Sensor vorzunehmen. Dazu macht man sich die Tatsache zunutze, daß verschiedene Gase in Abhängigkeit ihrer Molekularstruktur und ihrer funktionellen Gruppen ein unterschiedliches Adsorptions- und Desorptionsverhalten auf der Sensoroberfläche aufweisen und einen unterschiedlichen Effekt auf die Leitfähigkeit der gassensitiven Schicht ausüben.

Fährt man die Temperatur der Sensoroberfläche gezielt in Zyklen herauf und wieder herunter, so erhält man eine Folge unterschiedlicher Adsorptions- und Desorptionsvorgänge auf der Sensoroberfläche, die sich aufgrund der zeitlichen Hintereinanderanordnung zur Detektion der beteiligten Komponenten ausnutzen läßt.

Zur Herstellung von Gassensoren, die unter industriellen Bedingungen eingesetzt werden sollen, werden typischerweise kommerziell erhältliche Sensoren verwendet, um gewisse Mindestanforderungen bezüglich Verfügbarkeit und Reproduzierbarkeit im Ansprechverhalten gewährleisten zu können. Die Hersteller dieser Sensoren garantieren jedoch nicht, daß sich Sensoren, die z.B. für ein Zielgas X optimiert wurden, bezüglich ihrer Querempfindlichkeiten auf das Störgas Y ebenfalls gleich verhalten.

Aus diesem Grund ist es in praktischen Fällen nahezu ausgeschlossen, ein Sensorsystem „auf dem Reispfannkuchen“ zu optimieren, sondern man wird empirische Optimierungsverfahren wie das weiter unten beschriebene MSMEV einsetzen, um ein Zeit- und Sensorexemplar-unabhängiges System zu entwickeln.

Als besonders leistungsstark hat sich die Kombination dieses KI-Systems mit temperaturmodulierten Halbleitersensoren erwiesen. Voraussetzung auf der Hardwareseite ist dabei eine ausreichend starke thermische Kopplung von Heizer und gassensitiver Schicht sowie eine nutzbare Temperatur-Widerstandskennlinie des Heizers, wie man sie beispielsweise bei PT-100 Heizern vorfindet. Die Sensoren der UST GmbH erfüllen diese Voraussetzung.

Die Temperatur der Metalloxidsensoren wird dabei in einem Bereich variiert, der ein möglichst breites Spektrum an Oberflächenreaktionen abdeckt. Im unteren Temperaturbereich,  $T_{lo}$ , würde sich bei hinreichend langer Dauer ein Gleichgewicht einstellen, bei dem eine Vielzahl der Umgebungsgase auf der Oberfläche adsorbiert. Auf dem Weg zur oberen Temperatur,  $T_{hi}$ , desorbieren die Adsorbate bei charakteristischen Temperaturen, wobei es bei länger-kettigen Molekülen zu komplexeren Prozessen wie Aufcracken und Neuadsorption kommen kann. Hält man die Temperatur lange genug auf hohem Niveau konstant, wird die Oberfläche weitgehend „freigeheizt“.

Der zu wählende Temperaturzyklus hängt von den nachzuweisenden Gasen ab. Im einfachsten Fall wird eine stetig wechselnde Rampe  $T_{lo}$ ,  $T_{hi}$ ,  $T_{lo}$  usw. durchfahren. Die Wahl der Zeitdauer für jede Temperaturphase ist dabei so zu wählen, daß das Sensorsignal näherungsweise Sättigungsverhalten zeigt; da die Länge des Temperaturzyklus zugleich das zeitliche Auflösungsverhalten des Sensorsystems begrenzt, wird man einen Kompromiß zwischen vollständigem Erreichen der Gleichgewichtszustände und ausreichender Zeitauflösung des Sensorsystems einstellen.

Die folgenden Grafiken zeigen ein Anwendungsbeispiel, in dem es um den Nachweis von Pestiziden in geringen Konzentrationen ging. Zwar gibt es eine Vielzahl von Pestiziden, die sich in ihrer chemischen Struktur unterscheiden. Eine Selektivität auf Einzelsubstanzebene ist im vorliegenden Fall jedoch nicht gefordert bzw. nicht sinnvoll. Vielmehr stand im Vorder-

grund, die Gruppe „Pestizide“ zu identifizieren, d.h. sie von harmlosen Substanzen unterscheiden zu können.

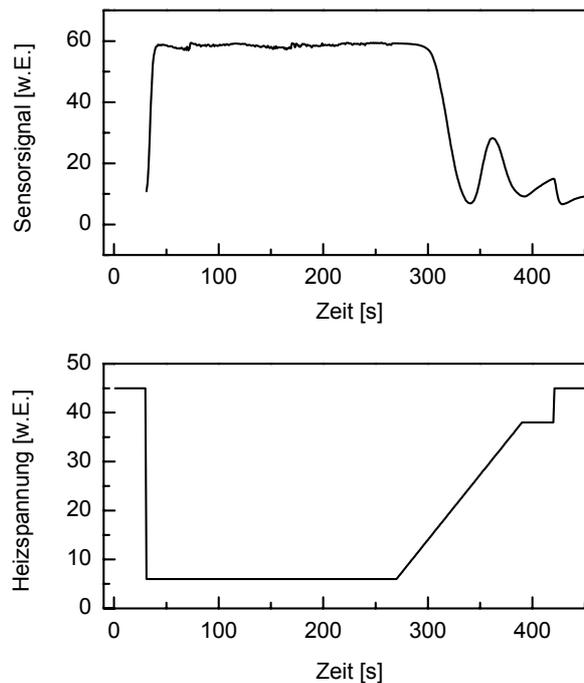
All diesen Stoffen gemeinsam ist eine hoher Dampfdruck. Dies ist für die Anwendung von besonderer Bedeutung, damit die Substanzen nach dem Besprühen über einige Zeit auf den Pflanzen bleiben. Diese Eigenschaft wurde bei der Konzeption des verwendeten Temperaturzyklus ausgenutzt:

Lange Phase niedriger Temperatur, um ein Adsorbieren der Stoffe auf der Sensoroberfläche zu ermöglichen. Die Temperatur beträgt hier ca. 60 °C. Es findet keine nennenswerte Umsetzung der Stoffe an der Sensoroberfläche statt.

Nach 4 Minuten wird der Sensor hochgeheizt. Bei diesem Prozeß findet dann nach und nach eine Reaktion der Stoffe statt.

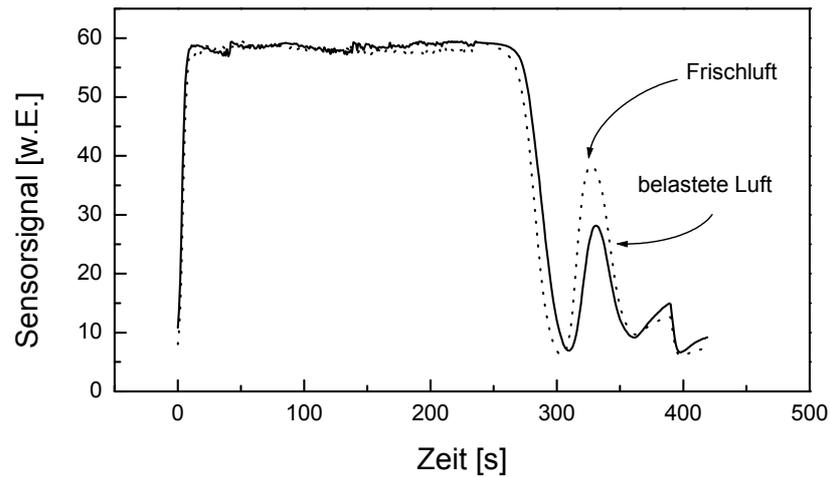
Am Ende erfolgt ein Ausheizen auf einer noch höheren Temperatur, um die Sensoroberfläche wieder „frei“ zu bekommen. Insbesondere beim Hochheizen und beim Übergang auf die Ausheiztemperatur kann man charakteristische Kurvenverläufe sehen.

Die folgende Abbildung zeigt den Verlauf der Heizspannung sowie das korrespondierende Sensorsignal:



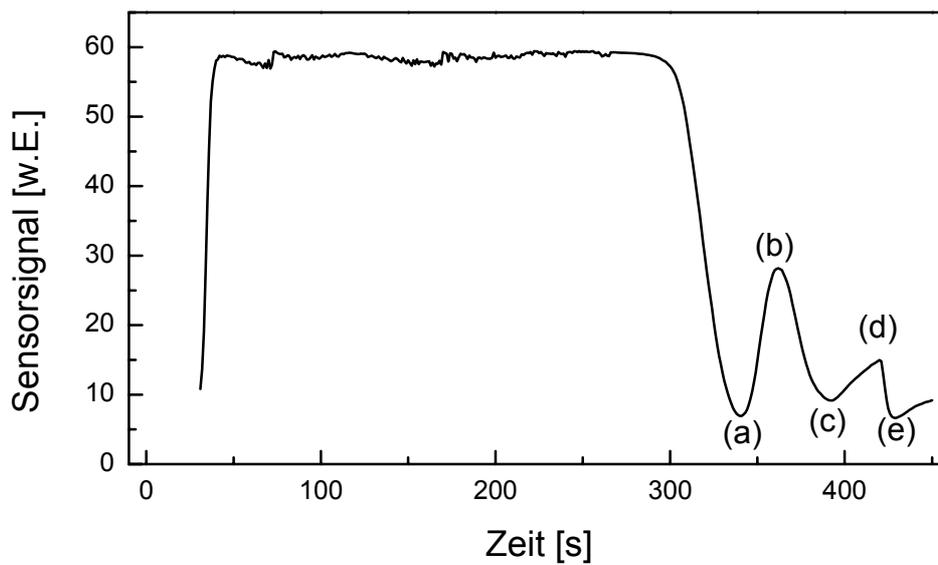
*Abb 5.14: Zyklus mit 400s Länge*

In der folgenden Abbildung sieht man den Unterschied in den Sensorsignalkurven bei belasteter und unbelasteter Luft, insbesondere die unterschiedlich stark ausgeprägte Peakhöhe nach ca. 325 s.



*Abb 5.15: Signalabweichungen bei belasteter/unbelasteter Luft*

Zur Quantifizierung dieser potentiellen Unterscheidungsmerkmale wurden folgende Kriterien konstruiert (Merkmalsvorauswahl):



*Abb. 5.16: Merkmalsvorauswahl*

Folgende Merkmale gehen als Ausgangsmenge in den Selektionsprozeß ein:

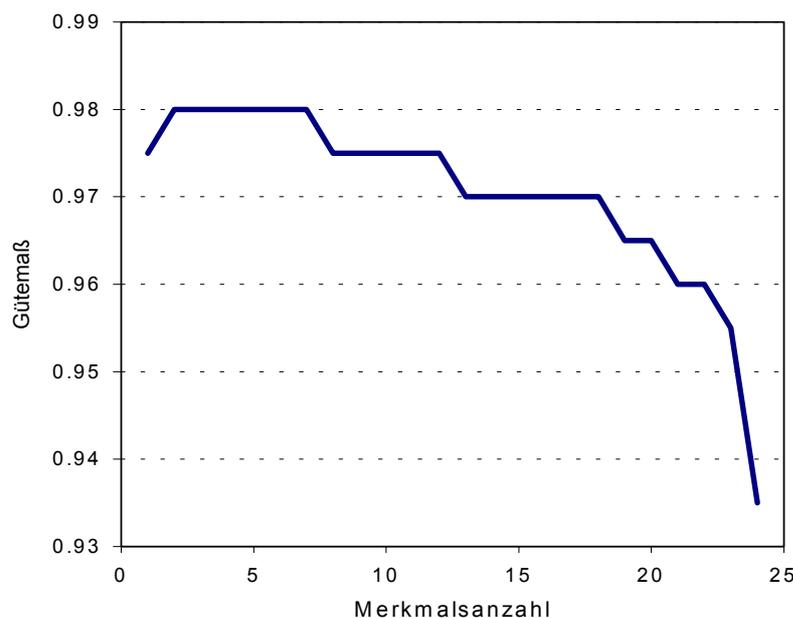
- |  |                         |
|--|-------------------------|
| 1. Steigung ( $t=30 \dots t=200$ )       | 13. Differenz (c) - (a) |
| 2. Signalthöhe $t=115$ - (a)             | 14. Quotient (c) / (a)  |
| 3. Differenz - Signal für $t = 236-231$  | 15. Differenz (d) - (a) |
| 4. Quotient - Signal für $t = 236 / 231$ | 16. Quotient (d) / (a)  |
| 5. Differenz (b) - (a)                   | 17. Differenz (e) - (a) |

- |                         |                         |
|-------------------------|-------------------------|
| 6. Quotient (b) / (a)   | 18. Quotient (e) / (a)  |
| 7. Differenz (c) - (b)  | 19. Differenz (d) - (b) |
| 8. Quotient (c) / (b)   | 20. Quotient (d) / (b)  |
| 9. Differenz (d) - (c)  | 21. Differenz (e) - (b) |
| 10. Quotient (d) / (c)  | 22. Quotient (e) / (b)  |
| 11. Differenz (e) - (d) | 23. Differenz (e) - (c) |
| 12. Quotient (e) / (d)  | 24. Quotient (e) / (c)  |

Um Drifteffekte im Signal weitestgehend zu eliminieren, wurden statt der absoluten Sensorsignale Differenzen bzw. Quotienten verwendet.

Die Auswahl der am stärksten mit dem tatsächlichen Gasangebot (Pestizide in geringer Konzentration) korrelierenden Merkmale geschieht mit dem in 5.3.4.4 beschriebenen MSMEV. Als Gütemaß wurde der Überlappungsgrad verwendet.

Die folgende Abbildung zeigt den Verlauf der Gütefunktion in Abhängigkeit von der Merkmalsanzahl:



*Abb. 5.17: Wert der Gütefunktion in Abhängigkeit von der Merkmalsanzahl*

### 5.3.3 Streßtests zur Verifikation der Robustheit von Merkmalen

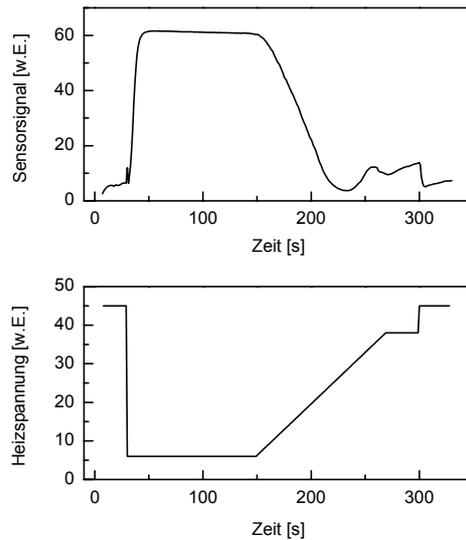
#### 5.3.3.1 Robustheitstest durch Variation der Versuchsbedingungen

Die vom MSMEV getroffene Merkmalsauswahl gilt streng genommen nur für die im Labor gewonnenen Meßreihen. Eine Aussage über die Stabilität der gefundenen Merkmale unter

veränderten Bedingungen, wie z.B. bei Messungen im Feldeinsatz, kann daraus noch nicht abgeleitet werden.

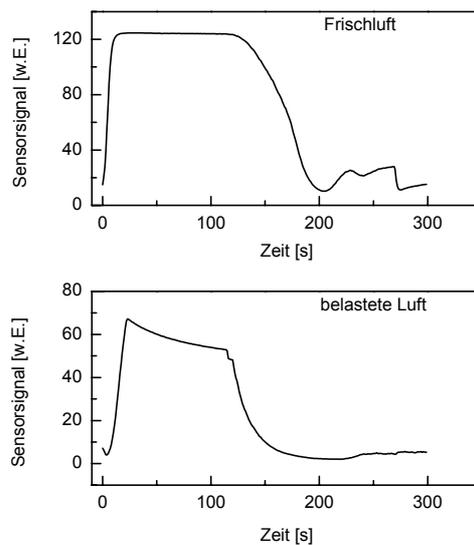
Um Aussagen über die Stabilität der Merkmale unter veränderten Bedingungen gewinnen zu können, wurden die Versuchsbedingungen variiert. Zu diesem Zweck wurde der Temperaturzyklus geändert (Verkürzung von 420 auf 300 Sekunden) und weitere Gase angeboten (u.a. CO und CH<sub>4</sub>).

Die folgende Abbildung zeigt den Verlauf des Sensorsignals in Abhängigkeit von der sich ändernden Heizspannung unter Umgebungsluft:



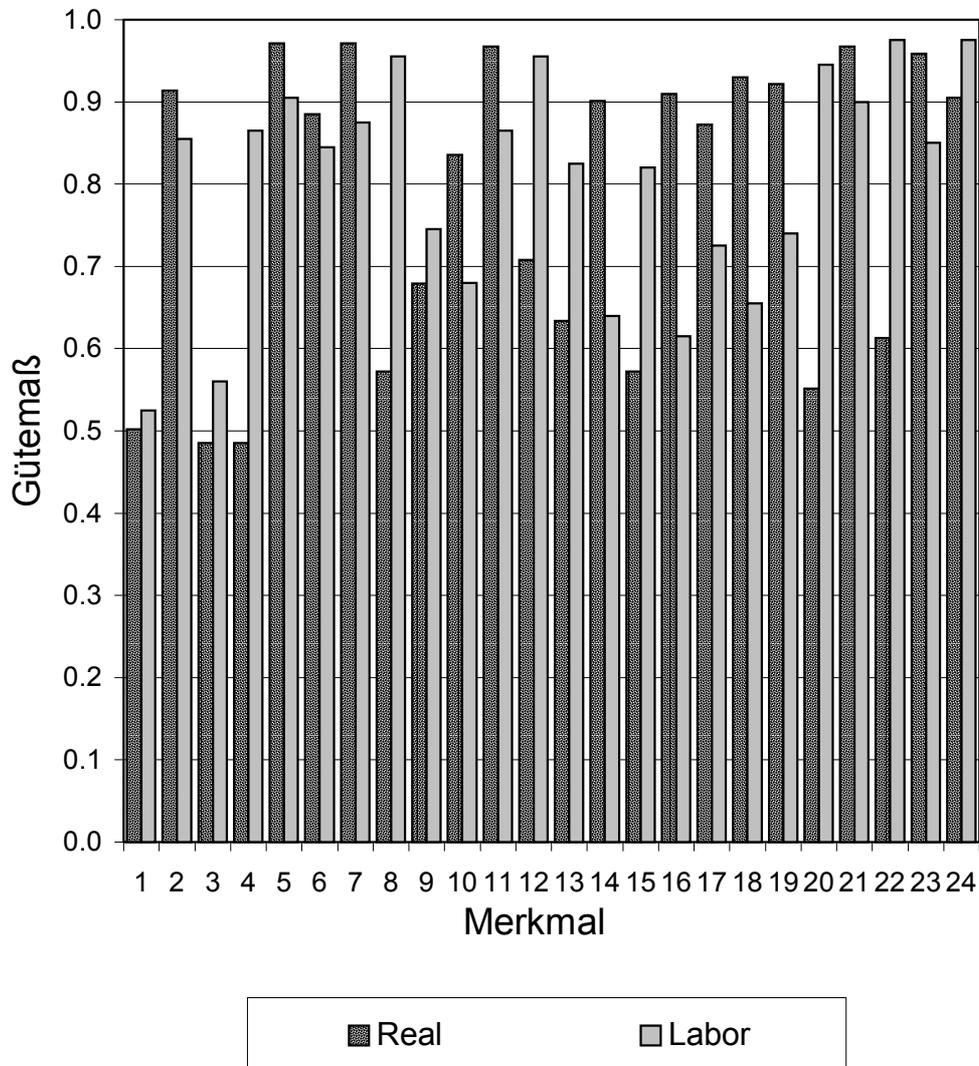
**Abb. 5.18:** Sensorsignal bei Umgebungsluft (oben), Heizspannung (unten)

Die Veränderung des ungestörten Signalverlaufs bei Gasangebot ist aus der folgenden Abbildung zu ersehen:



**Abb. 5.19:** Signalverlauf bei Frischluff und belasteter Luft

Wertet man beide Versuchsreihen aus, so zeigt sich, daß es Merkmale gibt, die in jeweils *einer* der beiden Versuchsanordnungen eine sehr hohe Güte aufweisen, in der anderen jedoch nicht. Mit anderen Worten: unter Variation der Versuchsbedingungen kommt es zu einem starken Einbruch der Merkmalsgüte.



**Abb. 5.20:** Merkmalsgüte für die Versuchsreihen „Real“ und „Labor“

Als robust sind demnach nur solche Merkmale zu bezeichnen, die unter beiden Versuchsbedingungen einen hohen Gütewert aufweisen, z.B. Merkmal 5 ( $:= (b)-(a)$ ) oder Merkmal 24 ( $:= (e)/(c)$ ). Andere Merkmale, wie z.B. #8 ( $:= (b)/(c)$ ) oder #16 ( $:= (d)/(a)$ ), die nur in einem der beiden Fälle einen hohen Gütewert aufweisen sind demnach als instabil zu verwerfen.

### 5.3.3.2 Stabilitätsuntersuchung in Langzeitversuchsreihen

Neben den Variationen in der Sensorbetriebsweise zur Simulation von Alterungs- und Exemplarstreuungseffekten dienen „echte“ Langzeituntersuchungen dazu, Aussagen insbesondere über die Zeitstabilität der Merkmalsgüte zu gewinnen. In Langzeitmessungen mit Pellistoren wurde die Stabilität sog. Responsefaktoren auf verschiedene Zielgase untersucht [Kie 99]. Bei diesen handelt es sich um die Reaktion des Sensors auf ein standardisiertes Ereignis, wie z.B. das Sensorsignal bei Gasangebot entsprechend 50% UEG, 1 Volumen% (konstante Volumenkonzentration) oder  $10\text{g/m}^3$  (konstante Massenkonzentration).

In Langzeitmessungen sollte ermittelt werden, ob diese Merkmale zeitstabil sind. Exemplarisch wurde der Responsefaktor gegenüber Propan bei einer Konzentration von 50% UEG untersucht. Für alle Stoffe wurden in festen Zeitabständen Kalibrierkurven aufgenommen; zu Beginn ca. 1-2 Wochen, später 4-6 Wochen [Kie 99].

Die Ergebnisse sind in den folgenden Abbildungen wiedergegeben:

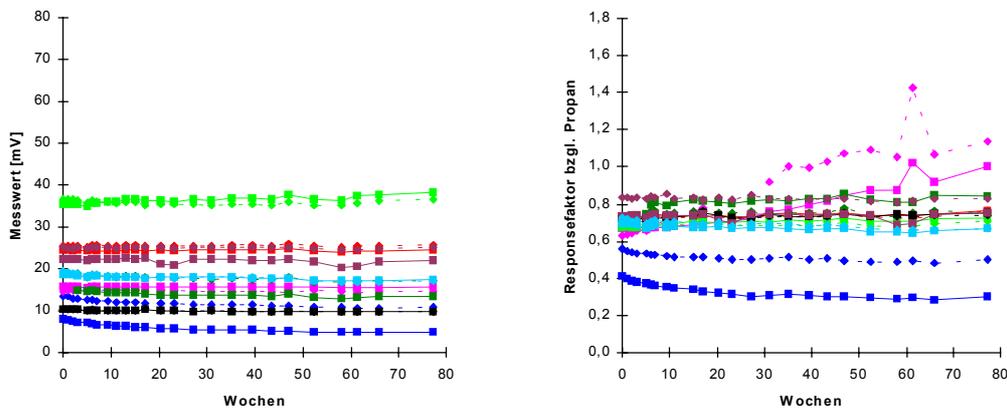


Abb. 5.21: Sensorsignale gegenüber *n*-Hexan (a) Responsefaktor bzgl. Propan (b) [Kie 99]

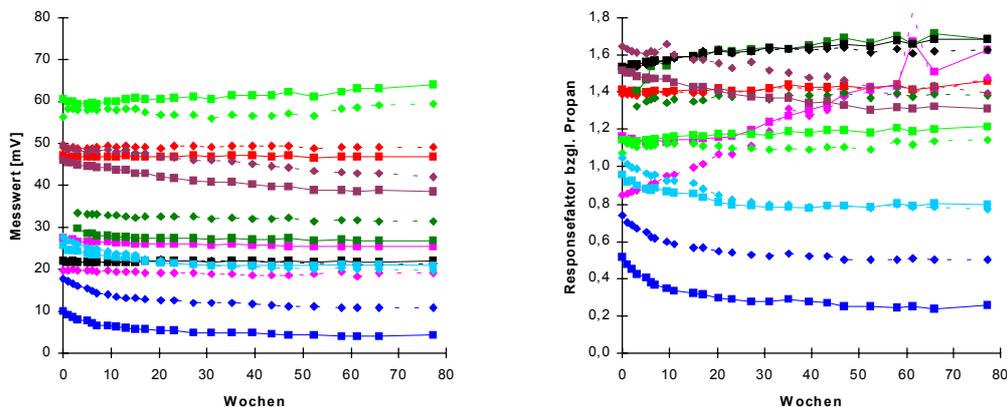
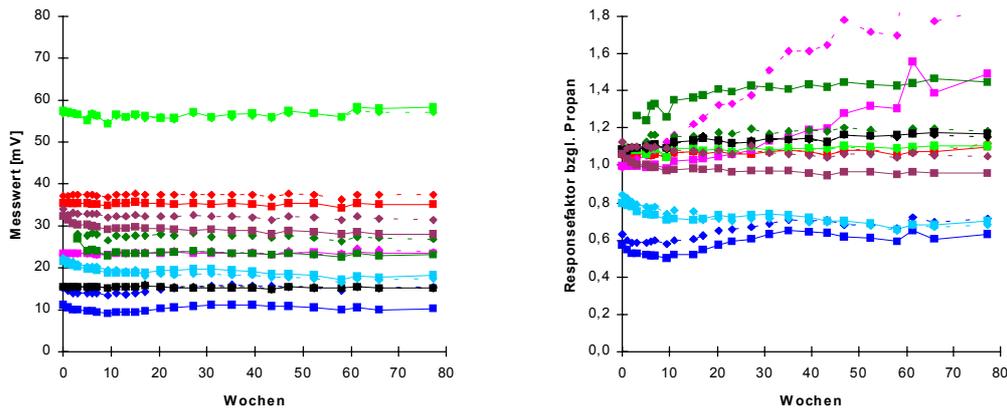


Abb. 5.22: Sensorsignale gegenüber Methanol (a) Responsefaktor bzgl. Propan (b) [Kie 99]



**Abb. 5.23:** Sensorsignale gegenüber Aceton (a) Responsefaktor bzgl. Propan (b) [Kie 99]

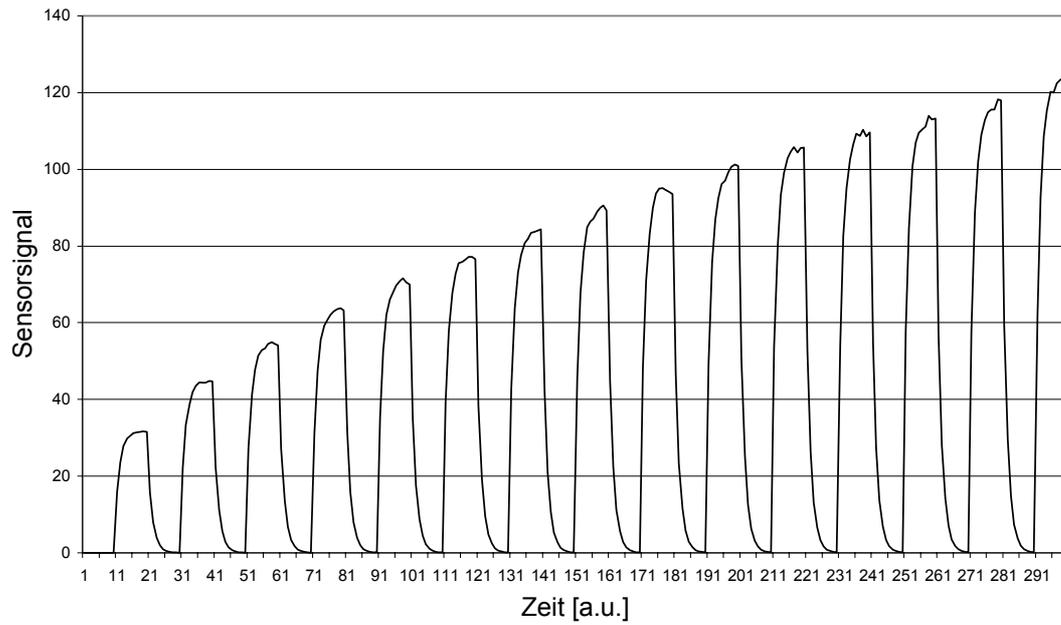
Aus Abb. 5.21-5.23 wird ersichtlich, daß zwar die Sensitivität gegenüber den eigentlichen Zielgasen annähernd stabil bleibt, daß der Responsefaktor gegenüber Propan aber je nach Sensortyp mitunter sehr großen Veränderungen unterliegt. Die angestrebte Intention, Propan als (ungefährliches) Gas zu Sensorkalibration einzusetzen, ist daher offensichtlich mit Risiken verbunden, die erst nach längerer Zeit sichtbar werden, d.h. die zum Zeitpunkt  $t$  vorhandenen Querempfindlichkeiten gegenüber Propan sind im Gegensatz zu den Zielgasempfindlichkeiten nicht stabil; ein Sensorsystem, das auf einer Konstanz der Querempfindlichkeitsverhältnisse aufbaut, würde nach einiger Zeit versagen.

### 5.3.3.3 Surrogatdaten zur gezielten Untersuchung des Einflusses von Parameterschwankungen

Während die in den letzten beiden Abschnitten beschriebenen Verfahren den Vorteil aufweisen, Aufschlüsse über das „real“ zu erwartende Ausmaß von Parameterschwankungen zu liefern, so besitzen sie doch den Nachteil, daß ein gezieltes „Einstellen“ einzelner Parameter am Sensor nicht bzw. nur mit sehr hohem Aufwand zu bewerkstelligen ist. Hat man jedoch mit Hilfe dieser Methoden die potentiellen Intervallgrenzen der einzelnen Parameter ausgelotet, so kann man mit Hilfe künstlich erzeugter Datensätze sehr präzise das Sensorverhalten für das dazwischenliegende Kontinuum ausloten und gegebenenfalls das Sensorverhalten jenseits der in der Praxis beobachtbaren Grenzen untersuchen.

Hierzu wurde das Übertragungsverhalten des Sensors bei Gasangebot durch folgendes vereinfachtes Modell simuliert. Der Leitwert  $G_S$  des Sensors hängt dabei wie folgt von der Gaskonzentration  $C$  ab:  $G_S = A C^\alpha$ , wobei  $A$  und  $\alpha$  sensorspezifische Konstanten sind, die zu Simulation von Exemplarstreuungen, Alterungseffekten und Sensorrauschen mit einem zeitabhängigen Störterm überlagert werden. Das resultierende Grundsignal wird anschließend zur Simulation des mit Trägheit behafteten Ansprechverhaltens des Sensors einer Tiefpaßfilterung unterzogen.

Das Ergebnis einer solchen „Messung“ bei unterschiedlich hohen „Gaskonzentrationen“ ist in der folgenden Abbildung zu sehen:



*Abb. 5.24: Simulierte Meßreihe*

Eine solche „Meßreihe“ weist die wesentlichen charakteristischen Eigenschaften einer „echten“ Messung auf und kann zur Konstruktion komplexer Sensorarrays verwendet werden.

### 5.3.4 Mathematische Verfahren zur verbesserten Nutzung des Informationsgehaltes von Sensorsystemen

#### 5.3.4.1 Akquisition von Regelwissen durch SOM

##### 5.3.4.1.1 Anwendungsbeispiel: Zuverlässige und fehlalarmsichere Detektion von Braunkohleschwelbränden

Ein typischer Braunkohlenschwelbrand entwickelt sich ohne nennenswerte Flammen-, Rauch- und Wärmeentwicklung, was seine Detektion mit herkömmlichen Brandmeldesystemen, die auf der Registrierung eben dieser Sekundäreffekte basieren, erschwert.

Die unvollständige Verbrennung breitet sich sehr langsam und oftmals unbemerkt auf der Oberfläche einer Braunkohlenstaubschicht aus, wobei allerdings jederzeit die Gefahr besteht, daß z.B. ein auftretender Luftzug zur Entfachung eines offenen Feuers oder gar einer Staubexplosion führen kann. Eine frühzeitige Erkennung von Schwelbränden wird daher angestrebt. Diese kann dadurch erfolgen, daß man die Gasemissionen des unvollständigen Verbrennungsvorgangs mit Hilfe von Sensoren registriert und sie von permanenten Hintergrundgaskonzentrationen diskriminiert.

Bei der Verbrennung von Kohle unter Sauerstoffmangel wird neben Kohlenmonoxid auch der in der Kohle gebundene Wasserstoff freigesetzt. Das Wissen um das Auftreten dieser beiden Gase läßt sich in Verbindung mit der Kenntnis der schwelbrandtypischen Konzentrationen zu einer Schwellenwertfestlegung für die Signalhöhen der entsprechenden Halbleitersensoren verwenden.

Allerdings ist die einfache Festlegung eines Schwellenwertes für z.B. das Signal des CO-Sensors nicht dazu geeignet, einen Schwelbrand zuverlässig von anderen Ereignissen zu unterscheiden. Eine Schwellenwertfestlegung stellt immer einen Kompromiß zwischen hoher Nachweisempfindlichkeit und geringer Fehlalarmrate dar. Eine Auswertung der Signalhöhenhäufigkeiten in der Bekohlungsanlage eines Braunkohlenkraftwerkes hat gezeigt, daß u.a. durch den Transport frisch gebrochener Kohle und die Freisetzung von Lösungsmitteldämpfen bei Reinigungs- und Reparaturarbeiten im Normalbetrieb der Anlage durchaus Signalhöhen erreicht werden, die von vergleichbarer Größenordnung wie die bei einem Schwelbrand auftretenden sind. Analoge Überlegungen gelten für den H<sub>2</sub>-Sensor. Die getrennte Schwellenwertfestlegung für nur einen einzelnen Sensor würde entweder zu unakzeptabel häufigen Fehlalarmen oder einer zu geringen Nachweisempfindlichkeit führen. Günstiger sieht bereits eine UND-Verknüpfung zweier Schwellenwerte aus, da das gemeinsame Auftreten von hohen H<sub>2</sub>-Sensor- und CO-Sensor-Signalen deutlich seltener vorkommt.

Im Idealfall würde ein Brand jedoch durch eine Merkmalskombination charakterisierbar sein, deren Auftreten im Normalbetrieb nahezu ausgeschlossen ist. Aus diesem Grund werden weitere potentiell zur Unterscheidung geeignete Größen herangezogen: die Trends der Sensorsignale und die Fluktuationen derselben.

Somit steht eine Grundgesamtheit von insgesamt neun Merkmalen zur Verfügung, mit deren Hilfe aufgrund von Erfahrungswissen eine Beschreibung der Ausnahmesituation „Brand“ erfolgen kann:

$$S_{H_2}, S_{CO}, S_{NO_x}, S_{H_2,grad}, S_{CO,grad}, S_{NO_x,grad}, S_{H_2,\sigma}, S_{CO,\sigma}, S_{NO_x,\sigma}$$

### 5.3.4.1.2 Extraktion von Expertenwissen aus einer Datenbasis

Zur Generierung einer Regelbasis nach dem in 2.3.5.3 beschriebenen Verfahren werden die in 5.2.4.1.1 beschriebenen Größen aus einer bereits vorliegenden Datenbasis verwendet. Dazu wird ein typischer Braunkohlenschwelbrand herausgegriffen und die 9-dimensionalen Merkmalvektoren so skaliert, daß sie die gleiche Spannweite und somit die gleiche maximale Differenz pro Komponente aufweisen. Dadurch ist gewährleistet, daß beim anschließenden Training eines Kohonen-Netzes mit euklidischer Norm als Ähnlichkeitsmaß keine der Komponenten bevorzugt wird.

Der zum Training verwendete Datensatz wird in die vier Bereiche „Normalzustand“, „beginnender Brand“, „Brand“ und „abklingender Brand“ unterteilt, wobei die Übergänge natürlich fließend sind, so daß die „harte“ Unterteilung einer gewissen Willkür unterliegt.

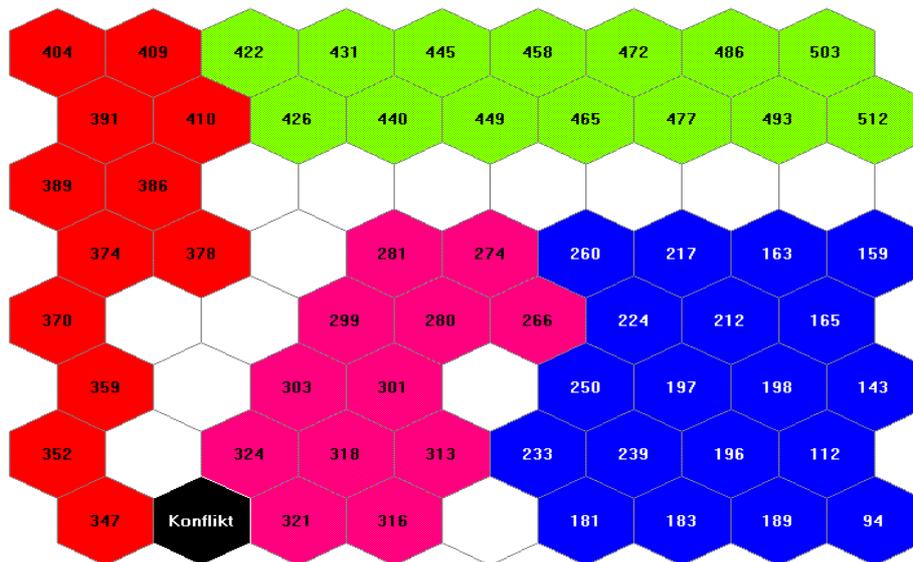
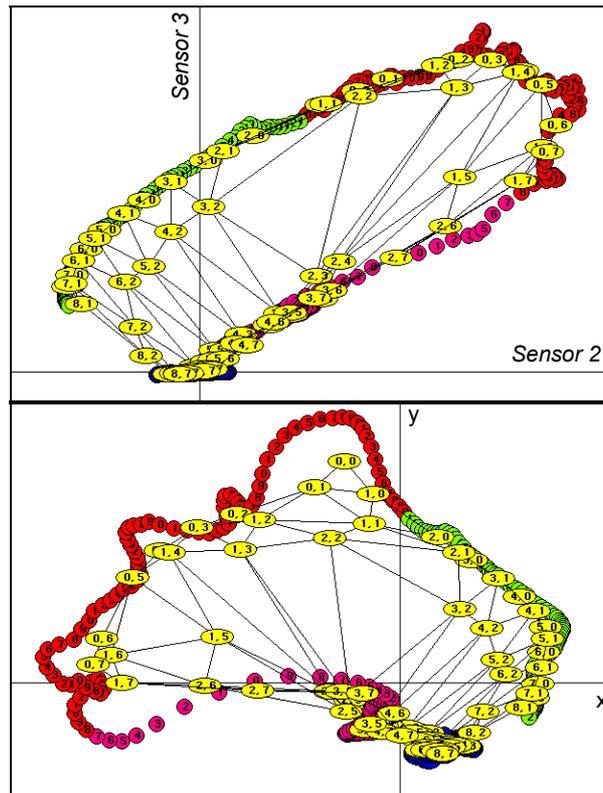


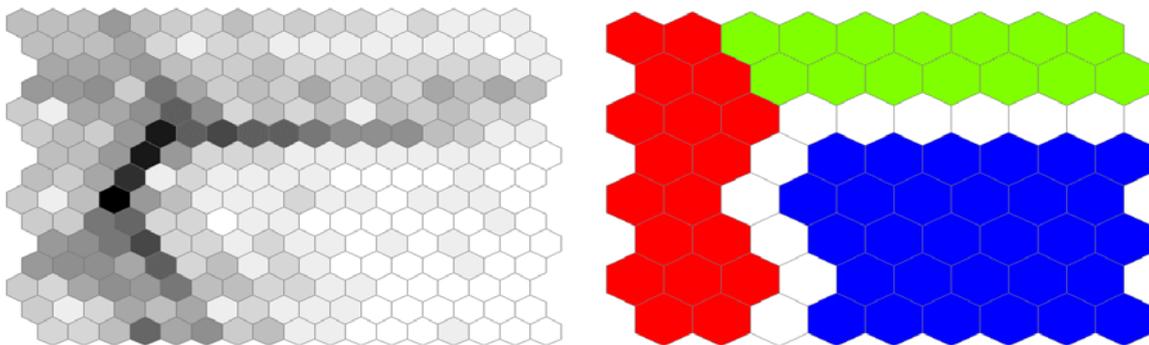
Abb. 5.25: Schwelbranddatenclustering mit Kohonen-Karte

Dies spiegelt sich in Abb. 5.25 in dem als Konflikt markierten Neuron wieder, welches sowohl durch Merkmalvektoren der Kategorie „beginnender Brand“ als auch durch solche der Klasse „Brand“ aktiviert wird. Die Zahlen auf den Neuronen geben den alle 30s hochgezählten internen Zyklenzähler des jeweils zuletzt aktivierenden Merkmalvektors wieder. Auf diese Weise wird die zeitliche Entwicklung des Brandes sichtbar. Die am IAP entwickelte und hier eingesetzte Software VISOM [Rüh 96] bietet darüber hinaus die Möglichkeit, die Aktivierung der einzelnen Neuronen des Netzes in realer zeitlicher Abfolge darzustellen, so daß Bewegungen innerhalb begrenzter Gebiete (z.B. „Normalzustand“) proportional zu ihrer tatsächlichen Häufigkeit wahrgenommen werden können. Das gezeigte Netz mit einer Größe von 9×8 Neuronen und hexagonaler Topologie zeigte in Bezug auf eine unüberwachte Clusterbildung das beste Ergebnis.



**Abb. 5.26:** 9x8x9-Hexagonal-Netz aus Abb. 5.25 a) von 2 Sensoren aufgespannte Ebene, b) NLM-Projektion  $\mathbb{R}^9 \rightarrow \mathbb{R}^2$

Eine Darstellung des Netzes im Eingaberaum ist in Abb. 5.26 zu sehen. Abb. 5.26 a) zeigt die von den Sensoren 2 ( $\text{NO}_x$ -Sensor) und 3 (CO-Sensor) aufgespannte Ebene. Gut erkennbar sind die Neuronen im Innenbereich, (3,2) bis (7,2), die in der Darstellung von Abb. 5.25 als nichtangeregte Neuronen in der dritten Reihe (Reihenindex 2) wiederzufinden sind, da sie für keinen der Datenpunkte nächste Nachbarn darstellen. Abb. 5.26 b) zeigt das gleiche Netz nach einer Projektion von Daten und Codebook-Vektoren mittels NLM. Hierbei wird erkennbar, daß das Netz trotz ausgedehnter Konvergenzphase beim Training nicht allen „Windungen“ der beim Brand durchlaufenen Trajektorie gefolgt ist.



**Abb. 5.27:** a) Graustufendarstellung der Codebook-Vektor-Abstände, b) Neulabelung aufgrund von a)

Die in Abb. 5.27a) als Graustufengrafik wiedergegebene U-Matrix zeigt, daß das Netz nur zwischen drei Bereichen sauber trennen kann, deren Grenzen als dunkle „Mauern“ erkennbar sind. Dieser Eigenheit wurde beim Neulabeln der Neuronen Rechnung getragen; es verbleiben drei Klassen, deren Verteilung in Abb. 5.27 b) dargestellt ist. Der Bereich unten rechts enthält die dem „Normalzustand“ zugeordneten Codebook-Vektoren. Die Neuronen auf der linken

Seite werden im Verlauf eines Brandes aktiviert, und zwar zunächst die im unteren Bereich, dann die im oberen. Beim Abklingen des Brandes werden schließlich die verbleibenden Codebook-Vektoren im oberen rechten Teil des Netzes aktiviert.

Ein Vorteil von trainierten Kohonen-Karten bei der Branderkennung besteht darin, daß ein Alarm nicht zwangsläufig dann ausgelöst werden muß, wenn die aktuell aktivierten Neuronen in die Kategorie „Brand“ fallen, sondern man diese Triggerung erst nach dem Durchlaufen eines vorausgehenden Stadiums vom Typ „Brandfrühstadium“ zulassen kann. Damit schließt man all diejenigen Ereignisse aus, die sich aktuell zwar so wie eine Brand verhalten (bezüglich aller 9 Komponenten), sich den Neuronen der Kategorie „Brand“ auf einer nicht-brandtypischen Trajektorie genähert haben. Andererseits unterscheiden sich Brände hinsichtlich ihrer Größe und Entfernung zum Detektor, so daß die Verwendung scharfer Werte, wie sie in den Codebook-Vektoren vorliegen, nicht sinnvoll erscheint.

Einen geeigneten Kompromiß stellt ein Fuzzy-System dar, daß zwei Ausgangskategorien liefert: „Brandfrühstadium“ und „Brand“. Dieses läßt sich in einen Meta-Algorithmus einbetten, der im Falle „Brand“ nur dann einen Alarm auslöst, wenn vorher eine Freischaltung durch das Ereignis „Brandfrühstadium“ erfolgt ist.

Daher soll innerhalb des vom Netz sauber abgetrennten Bereiches „Brand“ eine Differenzierung hinsichtlich seiner Stadien vorgenommen werden. Der im Netz „unten“ (Neuronen mit höherem 2. Indexwert) liegende Teil repräsentiert dabei den neuen Zustand „Brandfrühstadium“, der andere „Brand“.

Ein Blick auf die Komponentenebenen des Netzes gestattet nun die linguistische Beschreibung der einzelnen Cluster. Mit Blick auf das später zu generierende Fuzzy-System soll die Kategorie „Normalzustand“ die Standardausgabe des Klassifikators darstellen, so daß nur nach Regeln zur Beschreibung der Klasse „Brand“, bzw. „Brandfrühstadium“ gesucht wird. Als geeignet zur Beschreibung einer Klasse gelten dabei jeweils die Komponenten, die innerhalb derselben nur geringfügig, insgesamt jedoch sehr stark variieren.

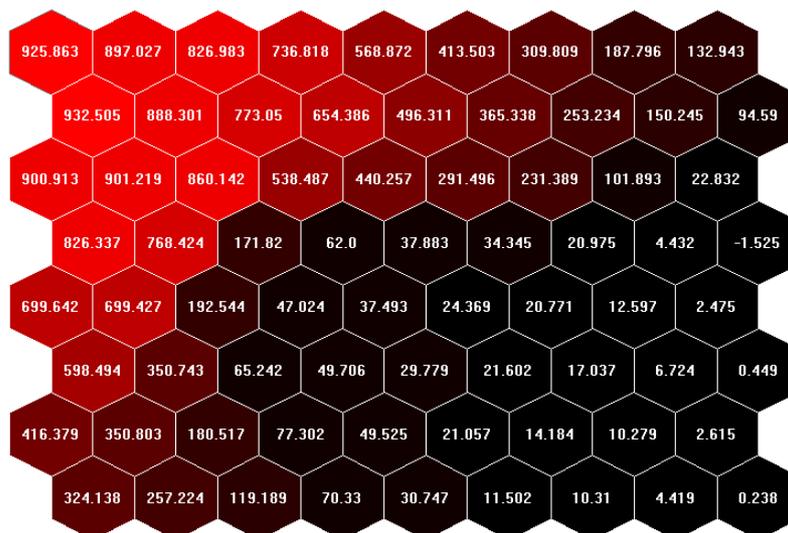
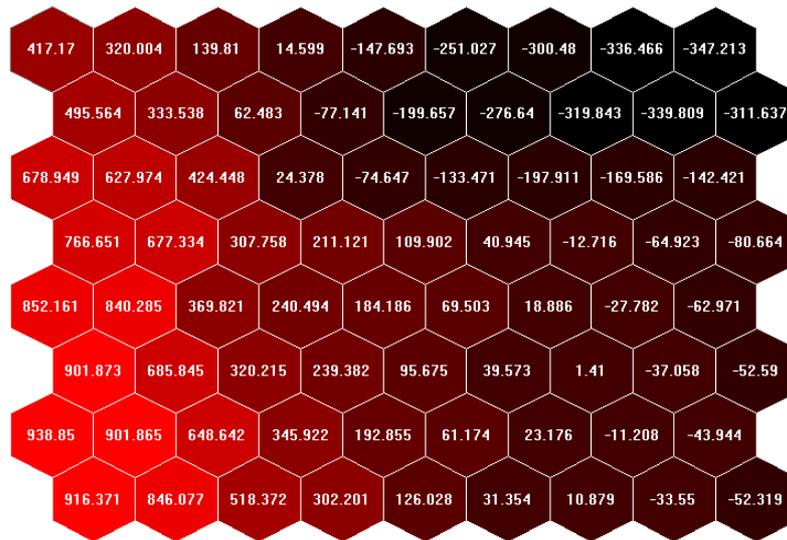


Abb. 5.28a:  $S_{H_2}$ -Ebene des Netzes



**Abb. 5.28b:**  $S_{CO}$ -Ebene des Netzes

In *Abb. 5.28 a)* und *b)* sind die zu den Komponenten  $S_{H_2}$ ,  $S_{CO}$  korrespondierenden Ebenen des Netzes dargestellt. Der Wert der jeweiligen Codebook-Vektor-Komponenten (Spannweite  $\equiv$  1000) ist zum einen als absoluter Zahlwert ablesbar, zum anderen ist jede Ebene mit einem Farbverlauf von schwarz (Minimalwert der Ebene) nach hell (Maximalwert der Ebene) versehen.

Begnügt man sich zunächst mit zwei linguistischen Termen (*hoch* und *niedrig*) für jede linguistische Variable ( $S_{H_2}$  bis  $S_{NO_x, \sigma}$ ), so kann eine direkte Umsetzung des Farbeindrucks in jene vorgenommen werden (hell  $\equiv$  *hoch*, schwarz  $\equiv$  *niedrig*). Gemäß *Abb. 5.27 b)* repräsentieren die mittelgrau markierten Neuronen im linken Teil des Netzes den Bereich „Brand“.

*Abb. 5.28 a)* ( $S_{H_2}$ -Ebene) zeigt im oberen Teil dieses Bereichs, der einem fortgeschrittenen Brandstadium entspricht, eine gleichmäßige Graufärbung an. Diese Beobachtung kann als erste Bedingung in die Beschreibungsregel für „Brand“ eingehen. Analog werden die anderen Ebenen auf ihre Brauchbarkeit zur Beschreibung des Stadiums „Brand“ hin untersucht. Für die Klasse „Brandfrühstadium“ verfährt man ähnlich, nur daß jetzt der untere Teil des in *Abb. 5.27 b)* mittelgrau markierten Bereichs betrachtet wird.

Die linguistischen Regeln zur Beschreibung der Zustände „Brandfrühstadium“ und „Brand“ lauten demnach wie folgt:

- 1) WENN ( $NO_x$ -Sensor = *hoch*) UND ( $NO_x$ -Anstieg = *hoch*) UND ( $NO_x$ -Fluktuation = *hoch*) UND (CO-Anstieg = *hoch*) DANN ([Wahrscheinlichkeit für] Brandfrühstadium [= *hoch*])
- 2) WENN ( $H_2$ -Sensor = *hoch*) UND ( $NO_x$ -Sensor = *hoch*) und (CO-Sensor = *hoch*) und ( $H_2$ -Anstieg = *hoch*) und (CO-Fluktuation = *hoch*) DANN ([Wahrscheinlichkeit für] Brand [= *hoch*])

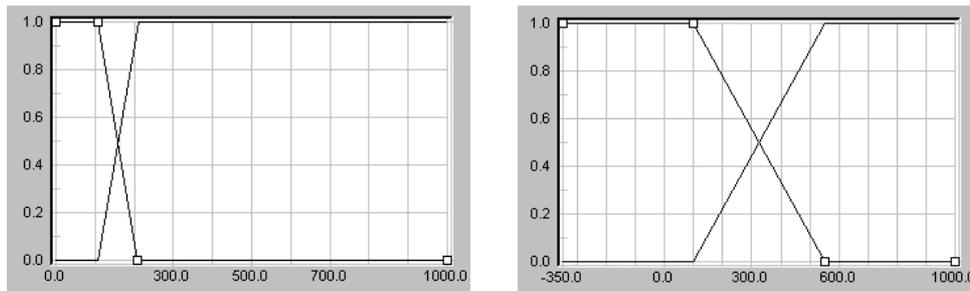
Ein hohes  $NO_x$ -Sensorsignal ist also für beide Brandphasen charakteristisch, während die Variable  $H_2$ -Fluktuation nicht zur Klassenunterscheidung beiträgt (*Abb. 5.28 b)*): nur ein kleiner Unterbereich von „Brand“ ist hellgrau).

Diese beiden Regeln werden durch eine Metaregel verknüpft:

- 3) WENN („Vorhergehender Zustand“ = „Brandfrühstadium“) UND („Aktueller Zustand“ = „Brand“) DANN „Alarm“

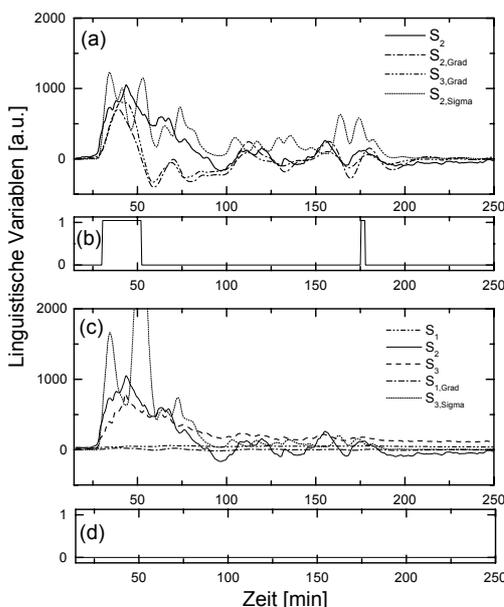
### 5.3.4.1.3 Umsetzung des akquirierten Wissens in ein regelbasiertes Fuzzy-Expertensystem

Die Aufteilung der linguistischen Variablen in die Terme hoch und niedrig ist in *Abb. 5.29* am Beispiel von  $H_2$  und CO zu sehen. Dabei wurde der Wertebereich der jeweiligen Basisvariablen zunächst dreigeteilt. Das untere Drittel hat dabei die volle Zugehörigkeit zur Eigenschaft „niedrig“, das obere die volle Zugehörigkeit zu „hoch“. Das mittlere Drittel wurde symmetrisch so aufgeteilt, daß je eine Z-Typ und eine S-Typ Standardzugehörigkeitsfunktion entsteht. Die Übergänge sind dabei so gewählt, daß sich die Summe der beiden Zugehörigkeitsfunktionen im gesamten Intervall zu 1 addiert.

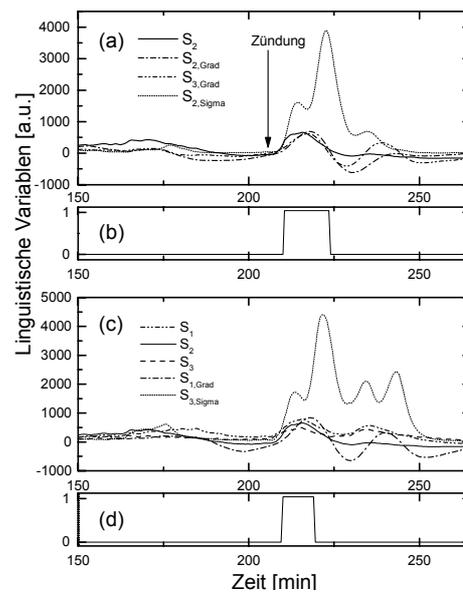


**Abb. 5.29:** Zugehörigkeitsfunktionen der Linguistischen Variablen  $S_{H_2}$  und  $S_{CO}$

Mit dieser Aufteilung und den drei Regeln aus **5.2.4.1.2** konnten bereits 8 von 9 Versuchsbränden detektiert werden. Man beachte, daß dieses Ergebnis auf der Basis einer rein sprachlichen Beschreibung des Problems erzielt wurde. Die Nichtdetektion des einen Brandes war auf ein äußerst schwaches Signal des  $H_2$ -Sensors zurückzuführen. Nachdem der Term *hoch* für das  $H_2$ -Signal dahingehend modifiziert wurde, daß bereits bei einem Wert der Basisvariablen von 200 die volle Zugehörigkeit vorliegt (s. *Abb. 5.29 a*), wurde auch dieser Brand richtig erkannt. Die Formulierung der Regeln selbst mußte nicht geändert werden.



**Abb. 5.30:** Versuchsschwelbrand



**Abb. 5.31:** Störereignis (verdampfendes Lösungsmittel)

Abb. 5.30 zeigt einen ausgewählten Versuchsschmelbrand. Im oberen Teil sind die Entscheidungsvariablen für „Brandfrühstadium“, sowie die Ausgangsvariable selbst zu sehen, im unteren Teil die Variablen für „Brand“, sowie „Brand“ selbst.

Der Grundgedanke bei der Systementwicklung war eine hinreichend präzise Beschreibung der Situation Brand, in der Hoffnung, daß Störereignisse nicht zum selben Signalmuster führen. Ein alternativer Ansatz würde darin bestehen, diese gleich in das Training der Kohonen-Karte zu integrieren. Die Fehlalarmsicherheit des Systems muß im vorliegenden Fall noch anhand typischer Störereignisse überprüft werden.

Abb. 5.31 zeigt die Reaktion des Systems auf verdampfendes Propanol. Zwar wird die Ausgangsvariable „Brandfrühstadium“ (Abb. 5.31b) aufgrund der Lösungsmittelsensoraktivität und der Querempfindlichkeit des CO-Sensors auf den Wahrheitswert TRUE gesetzt, da dies aber nicht für die Ausgangsvariable „Brand“ geschieht, tritt kein Fehlalarm auf.

### 5.3.4.2 Berücksichtigung komplexer Zielfunktionen

#### Wahl einer geeigneten Kostenfunktion

Die Grundproblematik beim Entwurf eines Sensorsystems entspricht der eines Hypothesentests. Die These  $H_0$  entspricht dabei der Annahme, daß der Normalzustand des zu beobachtenden Systems vorliegt (kein Feuer, kein Gas etc.).  $H_1$  entspreche dem Ereignisfall. Prinzipiell können also vier Ereignis-Sensorreaktions-Kombinationen auftreten:

	$H_0$	$H_1$
Sensoroutput: $H_0$	<b>kein Fehler</b>	<b>Fehler 1. Art</b>
Sensoroutput: $H_1$	<b>Fehler 2. Art</b>	<b>kein Fehler</b>

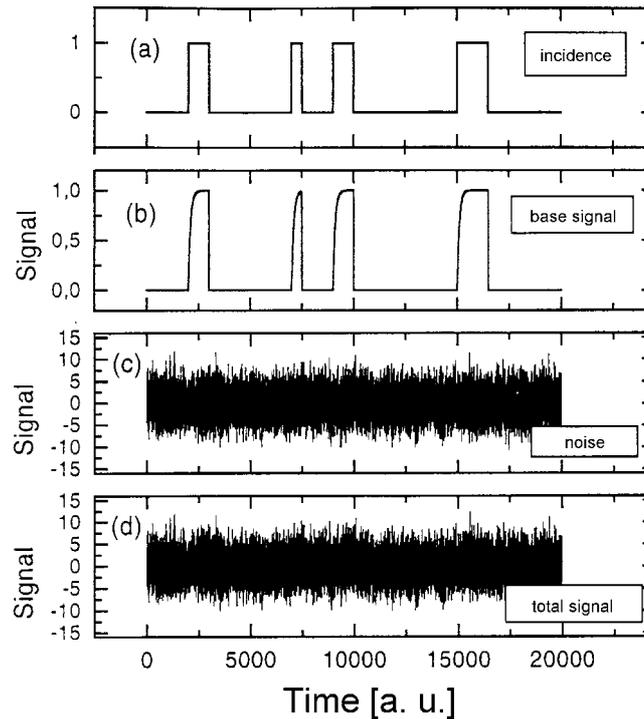
Im einfachsten Fall mißt das Sensorsystem eine Reihe von Beobachtungsgrößen, von denen angenommen wird, daß sie durch das Vorliegen des Ereignisfalls  $H_1$  in irgendeiner Weise beeinflußt werden. Diese Größen können zu einer „Alarmzahl“ aggregiert und mit einem Schwellwert verglichen werden. Je nach Lage des Schwellwertes erhält man im Extremfall ein System, daß entweder sehr empfindlich ist (keine Fehler 1. Art), oder eines, daß keine Fehlalarme (Fehler 2. Art) produziert.

Bei naiver Betrachtung könnte das Bestreben einzig darauf ausgerichtet sein, Fehler erster Art zu eliminieren und den Auslöseschwellwert entsprechend niedrig einzustellen. Betrachtet man jedoch ein Sensorsystem mit z.B. 50 Einheiten zur Überwachung einer industriellen Anlage und unterstellt, daß ein derart niedrig eingestellter Schwellwert dazu führt, daß jedes Gerät im Durchschnitt einmal im Jahr einen Fehler 2. Art produziert, so würde man fast jede Woche mit einem Fehler dieser Art für das Gesamtsystem rechnen müssen. Da Fehler 2. Art ebenfalls Kosten produzieren (Flugzeugnotlandungen, Feuerwehreinsätze u.a.), würde das System sehr schnell das Vertrauen des Anwenders verlieren und abgeschaltet werden.

Zur Vermeidung dieses Dilemmas müssen die Kosten für Fehler 2. Art ebenfalls in die Systemkonfiguration einbezogen werden.

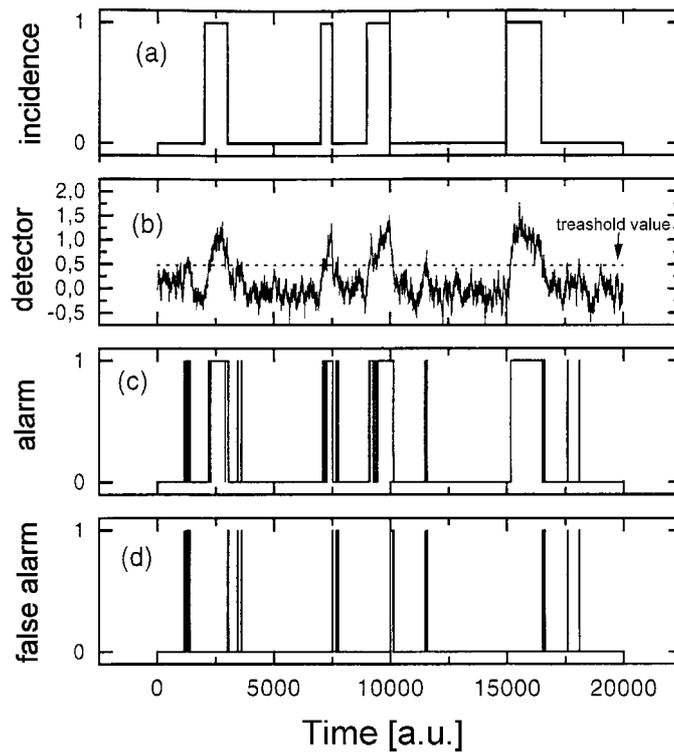
Ein weiterer kritischer Punkt, der das Auffinden geeigneter Schwellwertpositionierungen erschwert, ist die Unsicherheit, mit der Meßdaten behaftet sind. Wie im Kapitel über Sensorgrundlagen bereits dargelegt, verknüpfen Halbleitersensoren die Größen Gaskonzentration und Leitwertänderung nur innerhalb einer gewissen Bandbreite genau. Die folgende Abbildung soll diesen Sachverhalt demonstrieren:

Abb. 5.32 a. zeigt das Auftreten von Ereignissen, für deren Nachweis das Sensorsystem konzipiert wurde (z.B. Gasanwesenheit). Die in Abb. 5.32 b dargestellte „reine“ Reaktion des Sensors auf dieses Ereignis wird mit dem in c dargestellten weißen Rauschen überlagert, so daß die für die Signalverarbeitung zur Verfügung stehende Meßgröße das verrauschte Signal aus 5.32 d ist:



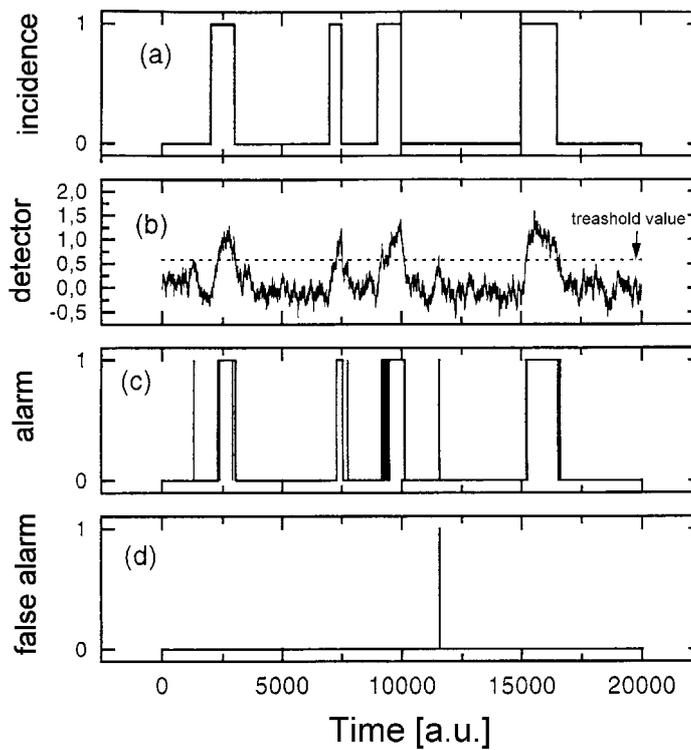
*Abb. 5.32: Verrauschte Sensordaten*

Die folgende Abbildung zeigt das Ergebnis einer Evolutionsstrategie zur Zielfunktionsoptimierung, wobei die korrekte Erkennung eines Alarm-/Nichtalarmfalles belohnt wurde. Freie Parameter im Optimierungsprozeß waren die Zeitkonstante des verwendeten Tiefpaßfilters, der über das Rohsignal gelegt wurde sowie der Schwellwert, ab dem ein Alarm ausgelöst werden soll. Abb. 5.33 a zeigt das Auftreten der Alarmfälle im Zeitverlauf, b das Ausgangssignal des Sensors und c den hypothetischen Detektoroutput. Abb. 5.33 d offenbart, daß insbesondere in den Übergangsbereichen zahlreiche Fehlalarme auftreten.



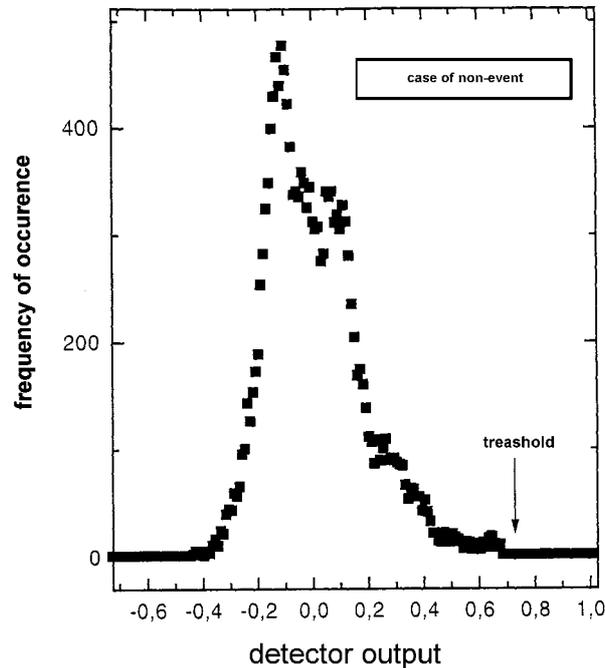
*Abb. 5.33: Optimierte Zielfunktion I*

Führt man zusätzlich in die Zielfunktion eine explizite Bestrafung von Fehlalarmen ein, so wird die Zahl der Falschklassifikationen deutlich herabgesetzt:



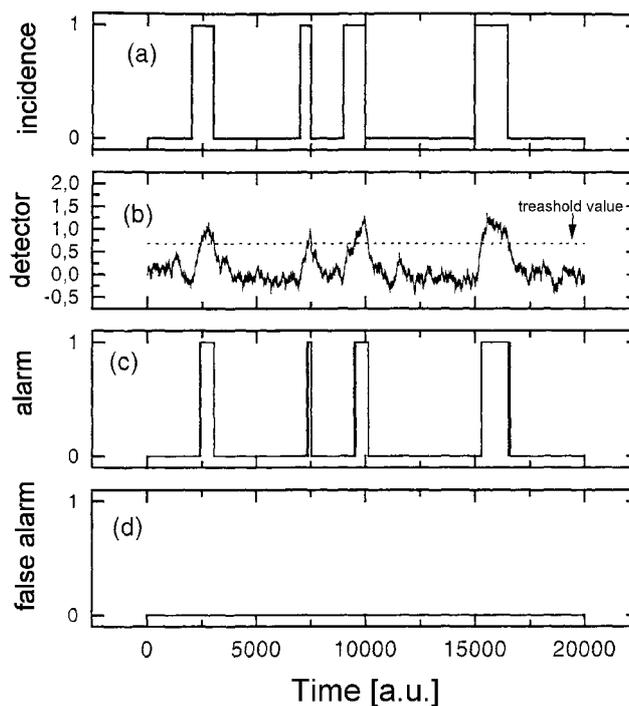
*Abb. 5.34: Optimierte Zielfunktion II*

Aus Abb. 5.34 d ist zu erkennen, daß nur noch ein einziger Fehlalarm verbleibt. Um auch diesen eliminieren zu können, muß die Signalstatistik für den Nicht-Ereignis-Fall ausgewertet werden:



*Abb. 5.35: Sensorsignalverteilung*

Abb. 5.35 zeigt die Verteilung der Sensorsignale für den nicht-Alarm Fall. Wenn das Vermeiden eines Fehlalarmes oberste Priorität genießt, so muß der einzustellende Schwellwert knapp oberhalb des höchsten im Normalbetrieb auftretenden Signales plaziert werden. Aus der folgenden Abbildung geht hervor, daß mit Hilfe dieser zusätzlichen Information ein Auftreten von Fehlalarmen vermieden werden kann, ohne daß „echte“ Alarmereignisse vom System übersehen werden:



*Abb. 5.36: Optimierte Zielfunktion III*

### 5.3.4.3 Bayes Strategie zur Mustererkennung

Das Grundproblem der Mustererkennung/Klassifikation besteht darin, einem Merkmalsvektor  $\bar{x}$  aus einem Merkmalsraum  $X$  einem Zustand  $\omega$  aus dem Zustandsraum  $\Omega$ , der  $q$  verschiedenen Klassen umfasse, zuzuordnen, d.h. es soll eine Entscheidung  $e_i$  aus dem Raum aller möglichen Entscheidungen  $E$  getroffen werden, wobei  $e_i$  die Entscheidung für die Klasse  $\omega_i$  aufgrund des Merkmalsvektors  $\bar{x}$  repräsentiert. Die Schadensfunktion  $s$  gebe die Kosten einer falschen Entscheidung an. Bei richtiger Klassifizierung entstehe kein Schaden, sonst ein Schaden der Größe

$$s_j(e_i) = s_{\omega_j}(e_i) = s(e_i | \omega_j) > 0, \text{ falls } i \neq j \quad (5.2)$$

Die statistische Entscheidungsfunktion  $d$  ordne jedem Merkmalsvektor  $\bar{x}$  aus  $X$  eine der möglichen Entscheidungen  $e_i$  aus  $E$  zu, d.h.  $d(\bar{x}) = e_i$ .

Die Auftrittswahrscheinlichkeit oder auch a priori Wahrscheinlichkeit

$$h_i = h_{\omega_i} = P(\omega_i) \quad (5.3)$$

gebe die Wahrscheinlichkeit für das Auftreten der Klasse  $\omega_i$  an, d.h. die Wahrscheinlichkeit dafür, dass ein beliebig ausgewähltes Muster  $\bar{x}$  der Klasse  $\omega_i$  angehört. Die bedingte Wahrscheinlichkeit

$$f_i(\bar{x}) = f_{\omega_i}(\bar{x}) = P(\bar{x} | \omega_i) \quad (5.4)$$

wird klassenspezifische Verteilung genannt und beschreibt die Verteilung der Merkmalsvektoren innerhalb eines bestimmten Klassengebietes. Demgegenüber ist  $P(\omega_i | \bar{x})$  die a posteriori Wahrscheinlichkeit oder Rückschlusswahrscheinlichkeit, die nach Beobachtung von  $\bar{x}$  zum Rückschluss auf dessen Klassenzugehörigkeit  $\omega_i$  dient.

Die Auftrittswahrscheinlichkeit  $P(\bar{x})$  gibt die Verteilung der Merkmalsvektoren  $\bar{x}$  im Merkmalsraum wieder:

$$P(\bar{x}) = \sum_{i=1}^q p(\bar{x} | \omega_i) P(\omega_i) = \sum_{i=1}^q f_i(\bar{x}) h_i \quad (5.5)$$

Sind die Dichten  $P(\omega_i)$  und  $P(\bar{x} | \omega_i)$  bekannt und liegt ein Merkmalsvektor  $\bar{x}$  vor, so erhält man nach dem Gesetz von Bayes die a posteriori Wahrscheinlichkeit von  $\omega_i$  aufgrund des Merkmalsvektors  $\bar{x}$  durch

$$P(\omega_i | \bar{x}) = \frac{P(\bar{x} | \omega_i) P(\omega_i)}{P(\bar{x})} = \frac{f_i(\bar{x}) h_i}{\sum_i f_i(\bar{x})} \quad (5.6)$$

Um zu einer Entscheidungsregel zu gelangen, betrachte man das bedingte Risiko, d.h. den Erwartungswert für den Schaden einer Entscheidung:

$$S(e_i | \bar{x}) = \sum_{j=1}^q s(e_i | \omega_j) P(\omega_j | \bar{x}) \quad (5.7)$$

Für die insgesamt zu erwartenden Kosten, d.h. für das Gesamtrisiko gilt dann:

$$S = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} S(d(\bar{x}) | \bar{x}) P(\bar{x}) d\bar{x} \quad (5.8)$$

S ist das sogenannte Bayes-Risiko. Die Bayes Strategie besteht darin, die Entscheidungsfunktion  $d$  so zu wählen, dass das Risiko  $S$  minimal wird. Da die Schadensfunktion  $s(e_i | \omega_j) \geq 0$  ist, ist nach (5.7) auch  $S(e_i | \bar{x}) \geq 0$ . Das Bayes Risiko ist daher genau dann minimal, wenn  $S(e_i | \bar{x}) \geq 0$  minimal ist.

Die Fehlklassifikationskosten seien gegeben durch

$$s_j(e_i) = s_{\omega_j}(e_i) = s(e_i | \omega_j) \begin{cases} 0 & \text{falls } i = j \\ s_j & \text{falls } i \neq j \end{cases} \quad (5.9)$$

Das bedingte Risiko ergibt sich somit zu

$$S(e_i | \bar{x}) = \sum_{\substack{j=1 \\ i \neq j}}^q s_j P(\omega_j | \bar{x}) = s_j (1 - P(\omega_i | \bar{x})) \quad (5.10)$$

Dieser Ausdruck wird minimal, wenn die a posteriori Wahrscheinlichkeit  $P(\omega_i | \bar{x})$  maximal wird. Damit ergibt sich als Entscheidungsfunktion für den Bayes-Klassifikator:

$$d(\bar{x}) = \omega_i \Leftrightarrow s_i \cdot P(\omega_i | \bar{x}) \geq s_j \cdot P(\omega_j | \bar{x}) \quad \forall j \text{ mit } j \neq i \quad (5.11)$$

Daraus ergibt sich mit (5.6):

$$d(\bar{x}) = \omega_i \Leftrightarrow s_i \cdot P(\bar{x} | \omega_i) \cdot P(\omega_i) \geq s_j \cdot P(\bar{x} | \omega_j) \cdot P(\omega_j) \quad \forall j \text{ mit } j \neq i \quad (5.12)$$

bzw.

$$d(\bar{x}) = \omega_i \Leftrightarrow s_i \cdot f_i(\bar{x}) \cdot h_i \geq s_j \cdot f_j(\bar{x}) \cdot h_j \quad \forall j \text{ mit } j \neq i \quad (5.13)$$

Diese Bayessche Entscheidungsregel stellt hinsichtlich des verbleibenden Klassifizierungsfehlers die im entscheidungstheoretischen Sinne optimale Lösung dar. Allerdings wird die Kenntnis der zugrundeliegenden Wahrscheinlichkeitsdichtefunktionen  $P(\bar{x} | \omega_i)$  vorausgesetzt. Da diese meist nicht explizit gegeben sind, müssen sie aus den verfügbaren Trainingsmustern geschätzt werden.

Hierfür gibt es zwei prinzipiell relevante Ansätze. Bei den parametrischen Ansätzen wird die tatsächliche klassenspezifische Verteilung unter Annahme einer Verteilungsfunktion, beispielsweise einer Gausfunktion, modelliert. Die Information über die Stichprobe geht somit auf die Parameter der gewählten Schätzfunktion über.

Beispiel für derartige Klassifikatoren sind der Bayes-Normalverteilungs-Klassifikator, der Mahalanobis-Klassifikator und der Euklidische-Abstandsklassifikator.

Der parametrische Ansatz setzt bei der Verwendung von Gaußfunktionen für die Dichteschätzung eine unimodale, quasi Gaußverteilung innerhalb der Klassengebiete einer Stichprobe voraus. Wird diese parametrische Annahme verletzt, beispielsweise durch bimodale Verteilung einer oder mehrerer Klassen in der Stichprobe, so ist keine akkurate Schätzung der Dichtefunktion möglich, so dass man auf nichtparametrische Verfahren ausweichen muß. Diese nehmen die Dichteschätzung lokal vor, indem die Information jedes Stichprobenelementes herangezogen wird. Beispiele hierfür sind der Parzen-Fenster-Klassifikator und der nächste-Nachbar-Klassifikator.

Eine Variante des Bayes-Klassifikators stellt der Maximum-Likelihood-Klassifikator dar, bei der die  $P(\omega_i)$  aus der Entscheidungsregel durch Definition einer geeigneten Kostenfunktion  $s_i$  herausfallen. Hierdurch können seltener auftretende Klassen, die sonst bei der Optimierung des Erkennungssystems unberücksichtigt bleiben würden, stärker gewichtet werden.

### 5.3.4.3.1 Parametrische Verfahren

#### 5.3.4.3.1.1 Bayes Normalverteilungsklassifikator und Maximum-Likelihood Normalverteilungsklassifikator

Für die klassenspezifischen Verteilungen werden Normalverteilungen zugrunde gelegt, deren Mittelwerte  $\bar{\mu}_i$  und Kovarianzmatrizen  $\mathbf{K}_i$  aus der Stichprobe für jede Klasse bestimmt werden. Die Entscheidungsfunktion ergibt sich nach einmaligem Logarithmieren unter Annahme gleicher Fehlklassifikationskosten zu

$$\begin{aligned} d(\bar{x}) = \omega_i &\Leftrightarrow \ln(P(\omega_i)) - \frac{1}{2} \ln(\det(\mathbf{K}_i)) - \frac{1}{2} ((\bar{x} - \bar{\mu}_i)^T \mathbf{K}_i^{-1} (\bar{x} - \bar{\mu}_i)) \\ &\geq \ln(P(\omega_j)) - \frac{1}{2} \ln(\det(\mathbf{K}_j)) - \frac{1}{2} ((\bar{x} - \bar{\mu}_j)^T \mathbf{K}_j^{-1} (\bar{x} - \bar{\mu}_j)) \quad \forall j \text{ mit } j \neq i \end{aligned} \quad (5.14)$$

Im Falle gleicher a priori Wahrscheinlichkeiten erhält man hieraus den Maximum-Likelihood Normalverteilungsklassifikator:

$$\begin{aligned} d(\bar{x}) = \omega_i &\Leftrightarrow -\frac{1}{2} \ln(\det(\mathbf{K}_i)) - \frac{1}{2} ((\bar{x} - \bar{\mu}_i)^T \mathbf{K}_i^{-1} (\bar{x} - \bar{\mu}_i)) \\ &\geq -\frac{1}{2} \ln(\det(\mathbf{K}_j)) - \frac{1}{2} ((\bar{x} - \bar{\mu}_j)^T \mathbf{K}_j^{-1} (\bar{x} - \bar{\mu}_j)) \quad \forall j \text{ mit } j \neq i \end{aligned} \quad (5.15)$$

#### 5.3.4.3.1.2 Mahalanobis Abstandsklassifikator

Geht man von näherungsweise gleichen klassenspezifischen Kovarianzmatrizen  $\mathbf{K}_i$  aus, so gelangt man zum Mahalanobis Abstandsklassifikator:

$$d(\bar{x}) = \omega_i \Leftrightarrow (\bar{x} - \bar{\mu}_i)^T \mathbf{K}^{-1} (\bar{x} - \bar{\mu}_i) \leq (\bar{x} - \bar{\mu}_j)^T \mathbf{K}^{-1} (\bar{x} - \bar{\mu}_j) \quad \forall j \text{ mit } j \neq i \quad (5.16)$$

#### 5.3.4.3.1.3 Euklidischer Abstandsklassifikator

Unter der Annahme unkorrelierte Merkmale, d.h.  $\mathbf{K} = \sigma^2 \cdot \mathbf{E}$  (mit  $\mathbf{E}$  = Einheitsmatrix), erhält man aus dem Mahalanobis-Abstandsklassifikator den euklidischen Abstandsklassifikator

$$d(\bar{x}) = \omega_i \Leftrightarrow (\bar{x} - \bar{\mu}_i)^2 \leq (\bar{x} - \bar{\mu}_j)^2 \quad \forall j \text{ mit } j \neq i \quad (5.17)$$

### 5.3.4.3.2 Nichtparametrische Verfahren

#### 5.3.4.3.2.1 Parzen-Fenster-Klassifikator

Falls keine parametrische Annahme über die Verteilungsfunktion getroffen werden kann, kann man auf eine lokale Schätzung der Dichtefunktion an jeweils einem Punkt im Merkmalsraum ausweichen. Dazu Betrachtet man in eine kleine Umgebung  $U(\bar{x})$  mit dem Volumen  $v$  um den Punkt  $\bar{x}$ . Die Wahrscheinlichkeit kann nun durch Auszählen der  $k$  Elemente aus der Stichprobe der Größe  $N$ , die in die Umgebung  $U(\bar{x})$  fallen, bestimmt werden:

$$p(\bar{x}) = \frac{k(\bar{x})}{N \cdot v} \quad (5.18)$$

Bei der Schätzung der Wahrscheinlichkeitsdichten kann nun einerseits das Volumen, andererseits die betrachtete Anzahl von Stichprobenelementen  $k$  innerhalb des Volumens  $v$  der Umgebung der Umgebung  $U(\bar{x})$  als fest betrachtet werden.

Eine Vorgehensweise mit festem Volumen  $v$ , die nichtparametrisch und ohne Verteilungsannahme zur Schätzung der Wahrscheinlichkeitsdichten arbeitet, bietet die Verwendung sogenannter *Kernschätzer*. Dieses ursprünglich von Parzen [Par62] zur Klassifikation eingeführte Verfahren wird nicht nur zur Schätzung von Dichtefunktionen verwendet, sondern kann auch für Regressionsaufgaben eingesetzt werden. Hier besteht ein wesentlicher Bezug zu *Radiale-Basisfunktionen-Netzwerken* und den weiter unten beschriebenen *Allgemeinen Regressionsnetzwerken*.

Die Schätzung der tatsächlichen Verteilungen erfolgt durch Glättung oder Verschmierung des Beitrags  $\frac{1}{N}$  eines jeden Punkts der Stichprobe der Größe  $N$  mittels einer Kernfunktion auf einen bestimmten Bereich, der durch diese bestimmt wird. Damit kann nun durch Summation aller Kernfunktionen der Stichprobenelemente eine Schätzung der Dichtefunktion im ganzen Raum, bzw. für einen zu klassifizierenden Merkmalsvektor in einem festen Volumen  $v$  vorgenommen werden:

$$p_{\text{Parzen}}(\bar{x}) = \frac{1}{N \cdot h_N^p} \sum_{i=1}^N \Phi\left(\frac{\bar{x} - \bar{x}_i}{h_N}\right) \quad (5.19)$$

Der Term  $h_N$  ist eine von der Stichprobengröße  $N$  abhängige Größe, die die Ausdehnung der Kernfunktion steuert. Mögliche Kernfunktionen sind die konstante Funktion, die Dreiecksfunktion und die Gaußfunktion. Die Kernfunktion ist im allgemeinen frei wählbar, muß aber selber eine Dichtefunktion sein und damit

$$\int \Phi(x) dx = 1 \quad (5.20)$$

genügen. Der auf der Kernschätzung beruhende Parzen-Fenster Klassifikator fällt seine Entscheidung gemäß der maximalen geschätzten Wahrscheinlichkeit, wobei die Schätzung wie oben beschrieben vorgenommen wird. Von Nachteil sind für dieses Verfahren neben der aufwendigen Berechnung der Werte der Kernfunktionen, die alle für einen betrachteten Datenpunkt ausgewertet werden müssen, die Schwierigkeiten bei der Wahl geeigneter Kernfunktionen und Parameter. Probleme bereiten vor allem geringe bzw. stark variierende Punktdichten, da dann global mit sehr ausgedehnten Kernfunktionen gearbeitet werden muß, was in Gebieten hoher Punktdichte nachteilhaft ist, weil dort die Lokalität der Schätzung verloren geht. Ein weiterer Nachteil ist, daß die gesamte Lernstichprobe zur Dichteschätzung und Klassifikation abgespeichert werden muß.

#### 5.3.4.3.2 Nächste-Nachbar Verfahren

Statt explizit eine Dichte zu konstruieren, um dann eine Entscheidung vornehmen zu können, kann über die Nächste-Nachbar Regel direkt eine Entscheidung herbeigeführt werden. Einem zu klassifizierenden Vektor wird bei dieser Entscheidungsregel die Klasse des nächsten Nachbarn aus der Trainingsstichprobe im Merkmalsraum zugeordnet. Die Verallgemeinerung der nächste-Nachbar Regel ist die  $k$ -nächste-Nachbar Regel (kNN), bei der eine feste Anzahl

k von Nachbarn zur Dichteschätzung herangezogen wird. Im Gegensatz zur Vorhergehenden Methode der Kernschätzer ist nun die Zahl k der Punkte in der Umgebung  $U(\bar{x})$  fest, das Volumen v hingegen variabel. Die Schätzung der Wahrscheinlichkeitsdichte erfolgt zu

$$p_{kNN}(\bar{x}) = \frac{k-1}{N \cdot v(\bar{x})} \quad (5.21)$$

Bei der vorab beschriebenen Methode der Kernschätzer handelt es sich um eine sogenannte *fixed-size* Methode, während es sich bei den kNN-Verfahren um sogenannte *fixed-mass* Verfahren handelt. Die kNN-Verfahren sind durch ihr variables Volumen v für Stichproben mit stark variierenden Punktdichten besser geeignet, als die bisher beschriebenen Verfahren der Kernschätzer.

Für die kNN-Verfahren gibt es zwei Bestimmungsverfahren, die *volumetrische Methode* und die *votierende Methode*. Bei der volumetrischen kNN-Methode werden für jede Klasse getrennt k Nachbarn bestimmt, und die Distanzen zwischen dem zu klassifizierenden Merkmalvektor  $\bar{x}$  und den gefundenen Nachbarn werden zur Entscheidungsfindung herangezogen. Die votierende kNN-Methode hingegen selektiert aus der gesamten gespeicherten Lernstichprobe k Nachbarn heraus und nimmt die Entscheidung anhand der mehrheitlich vertretenen Klasse in diesen k Nachbarn vor, d.h. die Zuordnung von  $\bar{x}$  zur Klasse  $\omega_i$  erfolgt nach der Entscheidungsregel:

$$e_i = \begin{cases} 1 & \text{wenn } k_i = \max_{j=1}^L k_j, \text{ mit } k_1 + \dots + k_L = k \\ 0 & \text{sonst} \end{cases} \quad (5.22)$$

Der Parameter k für die kNN-Entscheidung ist wesentlich unkritischer zu wählen, als die Wahl des Volumenelements für die Parzen-Methode. Somit ist der kNN-Ansatz als Klassifizierungsverfahren vorzuziehen.

### 5.3.4.3.2.3 Probabilistische Neuronale Netze (PNN) und General Regression Neural Network (GRNN)

Hierbei handelt es sich um außerordentlich leistungsfähige und zugleich leicht implementierbare statistische Algorithmen, die hier in einem nicht-neuronalen Rahmenwerk behandelt werden sollen. Die Formulierung als KNN findet sich ursprünglich bei [Spe 90], [Spe 91]. Eine kritische Würdigung im Vergleich mit anderen KNN-Architekturen findet sich in [Zel 94]. Beim GRNN handelt es sich um eine Erweiterung des PNN für nichtdiskrete Zuordnungen, d.h. GRNN können auch zur Funktionsapproximation benutzt werden.

#### 5.3.4.3.2.3.1 Probabilistische Neuronale Netze (PNN)

PNN bauen unmittelbar auf der Bayes-Strategie auf und verwenden folgende allgemeine Schätzer  $g_i(\bar{x})$  für die Dichtefunktion  $f_i(\bar{x})$ :

$$g_i(\bar{x}) = \frac{1}{\sigma \cdot N_i} \sum_{j=1}^{N_i} W \left( \frac{d(\bar{x}, \bar{x}_{\omega_i, j})}{\sigma} \right) \quad (5.23)$$

mit

$g_i(\bar{x})$ :	Geschätzte Dichte von $\bar{x}$ für die Klasse $i$ (interpretierbar als Maß der „Zugehörigkeit“ von $\bar{x}$ zur Klasse $i$ )
$\bar{x}$ :	Zu klassifizierender Vektor $\bar{x}$
$N_i$ :	Anzahl der (bekannten) Beobachtungsvektoren $\bar{x}_{\omega_i,j}$ mit Zugehörigkeit zur Klasse $\omega_i$
$\sigma$	Skalierungsparameter (keine Standardabweichung!)
$W(\cdot)$	Gewichtungsfunktion
$d(\cdot)$	Distanzfunktion für zwei Vektoren mit vorzugebender Metrik (euklidisch, City-Block)

Um den Grad der Zugehörigkeit eines unbekanntem Vektors  $\bar{x}$  zu Klasse  $\omega_i$  zu ermitteln, wird er mit allen bekannten Referenzvektoren  $\bar{x}_{\omega_i,j}$  einer Klasse  $\omega_i$  verglichen. Dabei wird die Distanz zwischen  $\bar{x}$  und  $\bar{x}_{\omega_i,j}$  bestimmt, mit einem Parameter  $\sigma$  skaliert und anschließend mittels der Funktion  $W$  gewichtet. Die Gewichtungsfunktion  $W$  muß allgemein hohe Werte bei kleinen Abständen und niedrige Werte bei großen Abständen zwischen  $\bar{x}$  und  $\bar{x}_{\omega_i,j}$  annehmen. Die geschätzte Dichte wird umso größer, je mehr bekannte Beobachtungsvektoren  $\bar{x}_{\omega_i,j}$  mit Klassenzugehörigkeit  $\omega_i$  sich in „geringem Abstand“ von  $\bar{x}$  befinden. Die Zuordnung selbst erfolgt nach der Bayesschen Entscheidungsregel. Bei gleichen a priori Wahrscheinlichkeiten und identischen Fehlklassifikationskosten findet die Zuordnung zu derjenigen Klasse  $\omega_i$ , für die sich die größte geschätzte Dichte  $g_i(\bar{x})$  ergibt.

Häufig wird für  $W(\cdot)$  eine Gaußsche Funktion verwendet, womit sich folgende konkrete Form ergibt:

$$g_i(\bar{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} \sigma^n} \cdot \frac{1}{N_i} \sum_{j=1}^{N_i} \exp\left(-\frac{d^2(\bar{x}, \bar{x}_{\omega_i,j})}{2\sigma^2}\right) \quad (5.24)$$

Hierbei ist  $n$  die Dimension des Merkmalsvektors,  $N_i$  die Anzahl der Trainingsmuster der Klasse  $\omega_i$ ,  $\bar{x}_{\omega_i,j}$  das  $j$ -te Trainingsmuster der Klasse  $\omega_i$  und  $\sigma$  ein Glättungsparameter. Diese Funktion stellt die Summe kleiner mehrdimensionaler Gaußverteilungen dar, die in jedem Trainingsmuster zentriert sind.

Aus Gründen der Vereinfachung wird oftmals auf die Normierung in (5.24) verzichtet und reduziert sich somit zu

$$g_i(\bar{x}) = \sum_{j=1}^{N_i} \exp\left(-\frac{d^2(\bar{x}, \bar{x}_{\omega_i,j})}{2\sigma^2}\right) \quad (5.25)$$

Die Verwendung einer Gaußschen Gewichtungsfunktion erfüllt die Anforderungen aus Parzen's Methode und hat nichts mit einer Normalverteilungsannahme zu tun.

Neben der Auswahl der Gewichtungsfunktion  $W$  bestimmt bei diesem Verfahren der Skalierungsparameter  $\sigma$  entscheidend die geschätzten Dichten  $g_i(\bar{x})$ .

Der Glättungsparameter  $\sigma$  bestimmt die Gestalt der Diskriminanzgrenze und hängt von der Dimensionalität des Merkmalsraumes  $n$  und der Anzahl der Trainingsmuster  $N$  ab, d.h.  $\sigma = \sigma(n, N)$ . Nach [Spe 90] kann immer ein geeignetes  $\sigma$  gefunden werden, wobei nach den Bedingungen von Parzen [Par 62] gelten muß:

$$\lim_{N \rightarrow \infty} \sigma(n, N) = 0 \quad \text{und} \quad \lim_{N \rightarrow \infty} N \cdot \sigma(n, N) = \infty \quad (5.26)$$

Im Grenzfall  $\sigma \rightarrow 0$  erhält man eine hochgradig nichtlineare Grenze, die dem Nächste-Nachbar-Klassifikator entspricht. Im Grenzfall  $\sigma \rightarrow \infty$  erhält man eine Hyperebene.

Der konkrete Wert, den eine geschätzte Dichtefunktion  $g_i$  an der Stelle  $\bar{x}$  annimmt, stellt eine nicht normierte *Maßzahl* für die Zugehörigkeit von  $\bar{x}$  zur Klasse  $\omega_i$  dar. Für die reine Klassifikation ist dies ausreichend. Möchte man darüber hinaus konkrete Wahrscheinlichkeitsaussagen treffen, muß anschließend noch eine Normierung vorgenommen werden, indem durch die Summe der geschätzten Dichten aller Klassen dividiert wird:

$$P(\omega_i | \bar{x}) = \frac{g_i(\bar{x})}{\sum_{k=1}^K g_k(\bar{x})} \quad (5.27)$$

Zur Bestimmung des „optimalen“ Skalierungsparameters  $\sigma$  sollen hier zwei Verfahren kurz vorgestellt werden. Das erste Verfahren, welches in der Literatur quasi als Standardverfahren vorgeschlagen wird [Spe91], beruht auf der *one-hold-out*-Methode und wird weiter unten behandelt. Das zweite, hier zunächst vorzustellende Verfahren ist eine Variante des ersten, jedoch einfacher darzustellen und leichter zu implementieren.

Zur Schätzung des „optimalen“ Skalierungsparameters  $\sigma$  ist, wie bei jeder Art der Parameterschätzung, eine Lernstichprobe oder Trainingsmenge notwendig. Die Trainingsmenge besteht dabei aus insgesamt  $T$  beobachteten Vektoren  $\bar{x}_l$ ,  $l = 1, \dots, T$ , deren jeweilige Klassenzugehörigkeit bekannt ist. Die gesamte Trainingsmenge wird nun in zwei Teilmengen  $U$  und  $V$  Vektoren zerlegt ( $U+V=T$ ), wobei die Anzahl der Vektoren  $U$  in der ersten Teilmenge deutlich größer als in der zweiten ist ( $U \gg V$ ). Dann werden die  $V$  Vektoren der zweiten Teilmenge auf Basis der  $U$  Vektoren der ersten Teilmenge klassifiziert. Der Skalierungsparameter  $\sigma$  ist nun so zu wählen, daß die Summe der Fehlklassifikationen auf den  $V$  Vektoren der zweiten Teilmenge minimal (bzw. die Summe der richtigen Klassifikationen maximal) wird. Die Fehlerfunktion ist also hier die Summe der Fehlklassifikationen (auf den  $V$  Vektoren der zweiten Teilmenge), die durch geeignete Wahl eines Skalierungsparameters  $\sigma$  zu minimieren ist. Dazu können Verfahren der nichtlinearen Optimierung eingesetzt werden.

Dieses Vorgehen besitzt allerdings den Nachteil, eine relativ große Trainingsmenge zu erfordern. Liegen z.B. 2000 beobachtete Vektoren mit zugehöriger Klassifikation vor, so ist eine Aufteilung mit  $U=1800$  und  $V=200$  unproblematisch. Sind allerdings z.B. nur 50 Beobachtungen verfügbar, wären mit einer Aufteilung von  $U=45$  und  $V=5$  schwerwiegende Probleme verbunden: Eine Anzahl von nur  $V=5$  ist zu klein, um eine zuverlässige Schätzung des „optimalen“ Skalierungsparameters zu gewährleisten. Die Anzahl  $V$  müßte deutlich erhöht werden, z.B. auf  $V=25$ . Damit beträgt jetzt aber  $U$  nur noch 25, d.h. die Anzahl der „Stützstellen“, auf deren Basis die Klassifikation der  $V$  Vektoren der zweiten Teilmenge erfolgt, wurde deutlich verringert.

Bei kleinen Datenmengen muß daher die *one-hold-out*-Methode angewendet werden. Dazu wird ein beobachteter Vektor aus der gesamten Trainingsmenge herausgenommen und in die zweite Teilmenge eingestellt, d.h.  $U=T-1$  und  $V=1$ . Dieser wird dann auf Basis der  $U$  Vektoren der ersten Teilmenge klassifiziert. Danach wird er gegen einen anderen Vektor der ersten Teilmenge ausgetauscht, d.h. nach dem Austausch gilt immer noch  $U=T-1$  und  $V=1$ , nur be-

findet sich jetzt ein anderer, bisher nicht benutzter Beobachtungsvektor in der zweiten Teilmenge. Dann wird jener auf Basis der  $U$  Vektoren der ersten Teilmenge klassifiziert und der Austausch (gegen einen bisher „unbenutzten“ Beobachtungsvektor) wiederholt. Das Verfahren wird fortgesetzt und erst beendet, nachdem alle  $T$  Beobachtungsvektoren genau *einmal* auf Basis aller anderen Beobachtungsvektoren klassifiziert wurden. Während einer derartigen, kompletten Iteration wird die Anzahl der Fehlklassifikationen bestimmt. Das gesamte Verfahren wird nun unter Variation des Skalierungsparameters  $\sigma$  solange wiederholt, bis der Wert für  $\sigma$  gefunden wurde, für den sich eine minimale Summe an Fehlklassifikationen ergibt.

Ein letzter Aspekt verdient beim praktischen Einsatz eines PNN (oder des aus ihm abgeleiteten GRNN) besondere Aufmerksamkeit: Die beim PNN (oder GRNN) benutzten Distanzmaße wie die Euklidische Distanz oder die City-Block-Distanz sind *nicht skaleninvariant*. Ein Abstandsmaß ist dann skaleninvariant, wenn der gemessene Abstand zwischen zwei Vektoren  $\bar{x}$  und  $\bar{y}$  nicht von den Maßeinheiten abhängt, in denen die einzelnen Komponenten der Vektoren ausgedrückt sind. Dies ist bei diesen Abstandsmaßen jedoch nicht der Fall, so daß eine geeignete Skalierung der einzelnen Komponenten (Z-Transformation, zumindest gleiche Spannweiten) zwingend erforderlich ist.

### 5.3.4.3.2.3.2 Das General Regression Neural Network (GRNN)

Bei dem in [Spe91] präsentierten Ansatz mit der Bezeichnung *General Regression Neural Network* (GRNN) handelt es sich um eine Weiterentwicklung des oben dargestellten PNN. Während mit dem PNN lediglich Klassifikationen vorgenommen werden können, ermöglicht das GRNN eine Funktionsapproximation. Die Art des funktionalen Zusammenhangs ist dabei beliebig, d.h. es können prinzipiell auch hochgradig nichtlineare Funktionen approximiert werden.

Auch das GRNN beruht auf einem alten, aus der Statistik bekannten Verfahren und soll hier in einem *nicht-neuronalen* Rahmen dargestellt werden. Wie auch bei der klassischen Regressionsanalyse geht es bei diesem Verfahren darum, auf Grundlage einer Lernstichprobe (bisher auch Trainingsmenge genannt) von Tupeln  $(\bar{x}, y)$  zu einer Schätzfunktion  $\hat{y} = f(\bar{x}) := \hat{y}(\bar{x})$  zu gelangen, mit der dann bei Auftreten eines Vektors  $\bar{x}$  eine Schätzung für  $y$  abgegeben werden kann. Während beim PNN die Zuordnung zu einer diskreten Kategorie vorgenommen werden soll, erfolgt beim GRNN eine Zuordnung zu einem Wertekontinuum, so daß man es als eine Verallgemeinerung des PNN auf unendlich viele Klassen interpretieren kann.

Analog zum PNN sollen die bekannten, zugehörigen Funktionswerte  $y_l$  von Vektoren  $\bar{x}_l$  um so stärker in die Schätzung von  $y$  für  $\bar{x}$  eingehen, je geringer die Distanz zwischen  $\bar{x}_l$  und  $\bar{x}$  ist. Als Gewicht bietet sich hier wie beim PNN die Gewichtungsfunktion  $W$  an. Die Schätzung von  $\hat{y}(\bar{x})$  könnte also nach (5.28) vorgenommen werden:

$$\hat{y}(\bar{x}) = \sum_{l=1}^T y_l W\left(\frac{d(\bar{x}, \bar{x}_l)}{\sigma}\right) \quad (5.28)$$

mit

$\hat{y}(\bar{x})$ : Schätzwert von  $y$  in Abhängigkeit vom aktuellen Vektor  $\bar{x}$  der unabhängigen Variablen

$y_l$ :	beobachteter Wert für $y$ für das $l$ -te Tupel der Trainingsmenge
T:	Anzahl der bekannten Tupeln $(\bar{x}_l, y_l)$ (Trainingsmenge)
$\sigma$	Skalierungsparameter
W(.)	Gewichtungsfunktion
d(.)	Distanzfunktion für zwei Vektoren mit vorzugebender Metrik (euklidisch, City-Block)

Gleichung (5.28) beschreibt den grundlegenden Ansatz beim GRNN, ist aber in dieser Form nicht sinnvoll anwendbar. Das Problem der Formulierung nach (5.28) liegt darin, daß die Gewichtungsfunktion  $W$  unnormierte Werte liefert. Die absolute Höhe der so resultierenden Gewichtungsfaktoren ist somit – alle übrigen Bedingungen konstant – u.a. abhängig vom Wert des Skalierungsparameters  $\sigma$  (oder anderen Faktoren, die bei der konkreten Definition von  $W$  verwendet werden), so daß noch eine geeignete Normierung vorgenommen werden muß:

$$\hat{y}(\bar{x}) = \frac{\sum_{l=1}^T y_l W\left(\frac{d(\bar{x}, \bar{x}_l)}{\sigma}\right)}{\sum_{j=1}^T W\left(\frac{d(\bar{x}, \bar{x}_j)}{\sigma}\right)} \quad (5.29)$$

Als Gewichtungsfunktion kann die einfache Exponentialfunktion ( $\exp(-x) = e^{-x}$ ) verwendet werden, als Distanzfunktionen kommen wieder die Euklidische Distanz oder die City-Block-Distanz in Frage. Die konkrete Schätzung eines beliebigen  $\hat{y}(\bar{x})$  erfolgt unter Verwendung der Exponentialfunktion und der Euklidischen Distanz nach (5.30)

$$\hat{y}(\bar{x}) = \frac{\sum_{l=1}^T y_l \exp\left(-\frac{d^2(\bar{x}, \bar{x}_l)}{2\sigma^2}\right)}{\sum_{j=1}^T \exp\left(-\frac{d^2(\bar{x}, \bar{x}_j)}{2\sigma^2}\right)} \quad (5.30)$$

Das Zustandekommen der Schätzung von  $Y$  wird hier (im Prinzip genauso wie beim PNN) nachhaltig vom Skalierungsparameter  $\sigma$  beeinflusst. Die Darstellungen zum PNN gelten hier sinngemäß, sollen aber nochmals im Kontext des GRNN wiederholt werden. Dazu sei beispielhaft auf die konkrete Formulierung nach (5.30) Bezug genommen und diese zur Veranschaulichung wie folgt formuliert:

$$\hat{y}(\bar{x}) = \sum_{l=1}^T y_l \frac{\exp\left(-\frac{d^2(\bar{x}, \bar{x}_l)}{2\sigma^2}\right)}{\sum_{j=1}^T \exp\left(-\frac{d^2(\bar{x}, \bar{x}_j)}{2\sigma^2}\right)} \quad (5.31)$$

Die Exponentialfunktion fällt mit zunehmender Entfernung von einem bekannten Beobachtungsvektor  $\bar{x}_l$  gegen Null ab. Wie schnell (in Abhängigkeit von der Distanz von  $\bar{x}$  zu  $\bar{x}_l$ ) sie gegen Null abfällt, wird vom Skalierungsparameter  $\sigma$  kontrolliert.

Es seien nun zwei extreme Fälle angenommen. Im ersten Grenzfall sei der Skalierungsparameter  $\sigma$  sehr groß ( $\sigma \rightarrow \infty$ ). Dann wird unabhängig von der Euklidischen Distanz von  $\bar{x}$  zu  $\bar{x}_l$  das Argument der Exponentialfunktion immer nahezu Null und die Argumente werden über alle T bekannten Beobachtungsvektoren  $\bar{x}_l$  in etwa gleich groß sein. Damit werden in der

Folge aber auch alle *relativen* Gewichtungsfaktoren über alle  $T$  bekannten Beobachtungsvektoren  $\bar{x}_i$  weitgehend gleich sein (nämlich ungefähr  $1/T$ ). Die Schätzung von  $y$  ergibt sich in diesem Grenzfall einfach als arithmetisches Mittel über alle bereits bekannten Funktionswerte  $y_i$ . Die geschätzte Dichte verläuft für große  $\sigma$  „flach und breit“ und wird im betrachteten Grenzfall sogar zu einer Ebene „platt gedrückt“ (woraus die nahezu gleichen relativen Gewichtungsfaktoren resultieren).

Der zweite Grenzfall besteht in der Betrachtung eines sehr kleinen Skalierungsparameters  $\sigma$  ( $\sigma \rightarrow 0$ ). Ein sehr kleiner Skalierungsparameter bewirkt in (5.31) zunächst, daß die Argumente der Exponentialfunktion dem Absolutbetrag nach sehr groß und damit die Funktionswerte sehr klein werden. Je weiter nun der Abstand von  $\bar{x}$  zu  $\bar{x}_i$  ist, um so schneller nehmen die Werte der Exponentialfunktion gegen Null ab. Dieser Effekt besteht zwar allgemein, wird hier aber infolge des sehr kleinen Skalierungsparameters  $\sigma$  extrem verstärkt. In diesem Grenzfall wird nur der Gewichtungsfaktor der bekannten Beobachtung  $\bar{x}_i$ , die zu  $\bar{x}$  am nächsten liegt, *relativ* von Bedeutung sein (der absolute Gewichtungsfaktor ist zwar auch hier klein, relativ zu allen anderen aber sehr groß). Der *relative* Gewichtungsfaktor für  $\bar{x}_i$  wird damit nahezu Eins, während die relativen Gewichtungsfaktoren für alle anderen  $\bar{x}_i$  nahezu Null werden. In die Schätzung von  $y$  fließt also nur der Funktionswert  $y_i$  der  $\bar{x}$  am nächsten gelegenen Beobachtung  $\bar{x}_i$  ein. In diesem Grenzfall wird das GRNN zu einem *nearest neighbor estimator*. Die geschätzte Dichte verläuft hier „sprunghaft“ mit einzelnen „Peaks“ an den Stützstellen (d.h. bei den bekannten Beobachtungsvektoren  $\bar{x}_i$ ).

Die Art der Schätzung von  $y$  bewegt sich hier zwischen den beiden Grenzfällen einer arithmetischen Mittelwertbildung und einem *nearest neighbor estimator*, je nach Wert des Skalierungsparameters  $\sigma$ . Mit der Variation nur eines Parameters ermöglicht also das GRNN ein breites Spektrum bezüglich der Art der Schätzung von  $y$ , was es – zumindest theoretisch – als ein faszinierendes Instrument der Funktionsapproximation erscheinen läßt.

Mit der konkreten Umsetzung des Verfahrens verbinden sich vielfältige Detailprobleme, die im Zusammenhang mit dem PNN bereits diskutiert wurden. Zunächst ist auf die Distanzfunktion hinzuweisen, die auf verschiedene Weise formuliert werden kann. Üblicherweise wird hier das Euklidische Distanzmaß oder alternativ die City-Block-Distanz verwendet. Im Prinzip ist jedoch die Verwendung jeglicher Art von Abstandsmaßen denkbar, soweit diese in geeigneter Weise (für das betrachtete Problem) den Abstand von Vektoren im mehrdimensionalen Inputraum beschreiben. Sofern die verwendeten Distanzmaße nicht skaleninvariant sind, ist eine Normierung der Daten (z.B. Z-Transformation) notwendig.

Ein besonderes Problem stellt hier wiederum (wie beim PNN) die geeignete Wahl des Skalierungsparameters  $\sigma$  dar, welcher allerdings der einzige Optimierungsbedürftige Parameter bei diesem Verfahren ist. Ein dazu mögliches Vorgehen wurde bereits beim PNN beschrieben: Die  $T$  bekannten Beobachtungsvektoren  $\bar{x}_i$  werden in zwei Teilmengen zu  $U$  und  $V$  Vektoren aufgeteilt. Dann werden die Funktionswerte der  $V$  Vektoren der zweiten Teilmenge auf Basis der  $U$  Vektoren der ersten Teilmenge geschätzt. Der Skalierungsparameter  $\sigma$  ist dabei so zu wählen, daß eine geeignet definierte Fehlerfunktion minimiert wird, z.B.:

$$E(\sigma) = \sum_{i=1}^V (\hat{y}(\bar{x}_i) - y(\bar{x}_i))^2 \rightarrow \min! \quad (5.32)$$

mit

$y(\bar{x}_i)$ : zugehöriger, bekannter Funktionswert  $y$  des  $i$ -ten Vektors  $\bar{x}_i$  der zweiten Teilmenge

$\hat{y}(\bar{x}_i)$ : geschätzter Funktionswert in Abhängigkeit vom Vektor  $\bar{x}_i$

Das Optimierungsproblem ist dabei „recht leicht“, da nur ein Parameter optimiert werden muß. Allerdings ist es hier im Regelfall nicht möglich, ein Verfahren zur Berechnung der ersten oder zweiten Ableitung der Fehlerfunktion nach dem Skalierungsparameter  $\sigma$  angeben zu können. Deshalb wird man mit Suchverfahren oder Approximationen des Gradienten arbeiten müssen.

Bei geringen Datenmengen ist ebenfalls das weiter oben beschriebene *one-hold-out* Verfahren bei der Bestimmung des Optimums der Fehlerfunktion anzuwenden. Dies kann dann zwar ein relativ aufwendiges Unterfangen werden, allerdings reagiert das GRNN recht „gutmütig“ auf nicht ganz optimale Werte für den Skalierungsparameter  $\sigma$ .

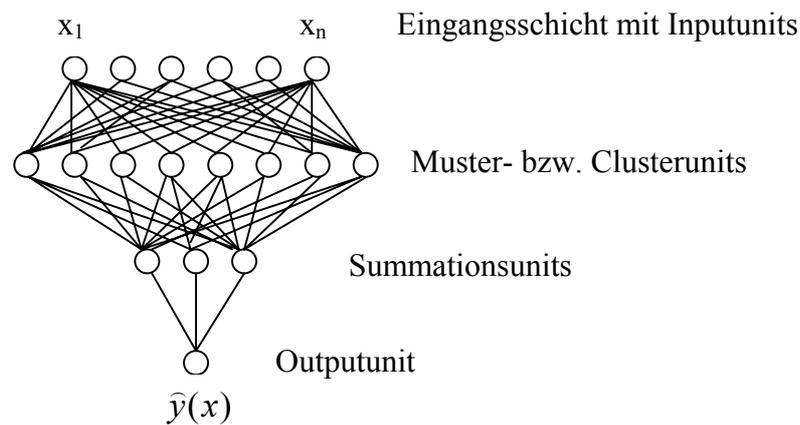
Abgesehen von diesen nicht unbedeutenden Detailproblemen besticht das GRNN durch eine Reihe von Vorzügen gegenüber der klassischen Regressionsanalyse. Es eignet sich hervorragend für adaptive Schätzungen, da jeder neue auftretende Inputvektor  $\bar{x}$  mit seinem assoziierten Wert  $y$  (sobald dessen Realisierung bekannt ist) problemlos in die bestehende Datenbasis integriert werden und damit sofort in die Schätzung des nächsten Wertes für  $y$  einfließen kann. Bei der klassischen Regressionsanalyse wäre – wollte man ein derartiges adaptives Verfahren verwenden – eine ständige Neuschätzung der Regressionskoeffizienten oder – noch aufwendiger – eine komplette Neuentwicklung der Regressionsfunktion pro neu hinzukommendem Tupel  $(\bar{x}, y)$  erforderlich.

Noch wichtiger als der vorangegangene Aspekt ist die Fähigkeit zur nichtlinearen Datenanalyse zu nennen, die sich auch auf die Approximation von Schwellenwertfunktionen ausdehnt.. Da der Wert des Glättungsparameters  $\sigma$  die Art der Dichteschätzung eines GRNN bestimmt, kann umgekehrt die Bestimmung dieses Wertes im Zuge einer Optimierung Aufschlüsse darüber liefern, in welchem Ausmaß vermutlich „Nichtlinearitäten“ in den Trainingsdaten vorhanden sind.

Nachteilig bei diesem Verfahren ist gerade bei umfangreichen Datenmengen der hohe Speicherplatzbedarf, da jedes beobachtete Tupel  $(\bar{x}, y)$  gespeichert werden muß. Da weiterhin jedes Tupel  $(\bar{x}, y)$  bei der Schätzung eines neuen  $y$  stets wieder herangezogen werden muß, entsteht ein proportional zur Größe der Datenbank anwachsender Rechenbedarf, so daß dieses Verfahren schnell unpraktisch werden kann. In derartigen Situationen bietet es sich an, Techniken der Gruppenbildung („Clustering“) einzusetzen, um nicht jedes einzelne Tupel  $(\bar{x}, y)$  in der Datenbasis zu halten, sondern sie durch Cluster zu repräsentieren.

Die Arbeitsweisen von PNN und GRNN beruhen, wie bereits erwähnt, auf einem alten, sehr leistungsfähigen statistischen Algorithmus. Die beiden vorgestellten Verfahren haben also nichts mit KNN zu tun und lassen sich auch völlig außerhalb dieser Klasse von Verfahren realisieren.

Allerdings läßt sich das vorgestellte Verfahren auch innerhalb eines KNN-Ansatzes realisieren. Die folgende Abbildung zeigt die Umsetzung des hinter dem GRNN stehenden Verfahrens in ein „neuronales Netz“:



Die Inputschicht dient – wie üblich – lediglich als Eintrittspunkt für externe Informationen.

Hier liegen die Elemente des Inputvektors  $\bar{x}$  an. In der folgenden Schicht befinden sich die Muster bzw. Clusterunits. Sie repräsentieren in ihren Gewichtsvektoren die schon bekannten Tupel  $(\bar{x}, y)$  (bzw. die Clustermittelpunkte von Clustern) von schon bekannten Beobachtungen  $(\bar{x}, y)$ . Die Summationsunits der folgenden Schicht ermitteln Zähler und Nenner der Gleichung (5.30). Die Outputunit nimmt die Division vor und ermittelt den Schätzwert  $\hat{y}(\bar{x})$ .

Diese Realisierung erlaubt einen direkten Vergleich mit den bekannteren KNN-Typen: Die Realisierung der Muster- bzw. Clusterschicht ähnelt, sofern hier keine Gruppenbildung vorgenommen wird, einem *Restricted Coulomb Energy Netzwerk* (RCE). Bei Anwendung einer Gruppenbildung („Clustering“) können die Clustering-Techniken des RCE oder auch die der *Lernenden Vektorquantisierung* (LVQ) Verwendung finden. Das so entstehende Netzwerk ist dann als Erweiterung oder Variante eines RCE oder LVQ interpretierbar, mit dem jetzt nicht mehr reine Klassifikation (wie beim RCE oder LVQ), sondern zusätzlich die Ausgabe reellwertiger Outputs möglich ist. In diesem Sinne kann das GRNN als eine ausgesprochen interessante Erweiterung oder Variante eines RCE oder LVQ aufgefaßt werden.

### 5.3.4.3.2.3 Beurteilung

Bei dem PNN und seiner Weiterentwicklung zum GRNN handelt es sich um zwei Verfahren, die vom Prinzip her sehr verschieden zu den bislang „klassischen“ Methoden (z.B. Regressionsanalyse, Perceptrons) sind. Bei jenen Methoden zur Funktionsapproximation wird der geschätzte Funktionswert  $y$  aus den Werten der *unabhängigen Variablen* des Vektors  $\bar{x}$  gewonnen, indem die Werte seiner Komponenten mit Koeffizienten multipliziert und anschließend summiert (bei der Regressionsanalyse) und gegebenenfalls nach demselben Prinzip auf nachfolgenden Stufen weiter verarbeitet werden (beim Perceptron). Eine Folge dieses Bearbeitungsprinzips ist, daß im multivariaten Fall (d.h. bei mehreren unabhängigen Variablen) viele (bei der Regressionsanalyse) oder sogar sehr viele (beim Perceptron) Koeffizienten (Regressionskoeffizienten bzw. Gewichte eines Perceptrons) geschätzt werden müssen.

Das GRNN nimmt dagegen die Schätzung des Funktionswertes  $Y$  durch eine gewichtete, arithmetische Mittelwertbildung über den (bereits bekannten) Werten der *abhängigen Variablen* vor. Die Werte der unabhängigen Variablen des Vektors  $\bar{x}$  werden hier „nur“ zur Ermittlung der einzelnen Gewichte bei der Mittelwertbildung benötigt, jedoch wird der Schätzwert für  $y$  nicht direkt aus ihnen abgeleitet (wie bei den anderen Verfahren). Die Bestimmung der Gewichte basiert auf der Schätzung der Dichte der Vektoren  $\bar{x}$  im Inputraum. Die Form der geschätzten Dichte wird dadurch gewonnen, indem über die einzelnen Stützstellen (d.h. den bekannten Beobachtungen) „Gaußsche Hüte“ gelegt werden. Der Skalierungsparameter kon-

trolliert dabei die Weite der „Gaußschen Hüte“. Für hohe Werte von  $\sigma$  sind diese „flach und breit“, für kleine Werte dagegen „eng und spitz“. Die Form der geschätzten Dichte ist schließlich die gemeinsame Überlagerung aller „Gaußschen Hüte“. Die Folge dieses Verarbeitungsprinzips ist schließlich, daß selbst im multivariaten Fall immer nur ein freier Parameter (nämlich der Skalierungsparameter  $\sigma$ ) geschätzt werden muß.

Das GRNN ist geradezu „genial einfach“, leicht zu implementieren und erfordert nur einfache Algorithmen zur Schätzung des „optimalen“ Skalierungsparameters  $\sigma$ . Es trifft keine Annahmen hinsichtlich der Art des zu approximierenden funktionalen Zusammenhangs (dieser kann auch hochgradig nichtlinear sein) und arbeitet – nach praktischen Erfahrungen – ausgesprochen „robust“.

Die Leistungsfähigkeit von GRNN und PNN beruht maßgeblich auf der Güte der geschätzten Dichte. Um die „wahre“ Dichte möglichst gut zu schätzen, sind möglichst viele „Stützstellen“ (Beobachtungen) notwendig. Mit wenigen Stützstellen wird gerade dann eine gute Schätzung der Dichte unmöglich sein, wenn die „wahre“ Dichte multimodal ist (d.h. viele „Höcker“ besitzt) und keine „Gaußsche Form“ aufweist. Ob man aber über viele „Stützstellen“ verfügt, hängt nicht nur von der reinen Anzahl an Beobachtungsvektoren ab, sondern auch von deren Dimension. Besitzen die Vektoren  $\bar{x}$  nur eine Komponente (unabhängige Variable), so sind 2000 Beobachtungen „viel“ und erlauben auch sehr komplexe Verläufe der „wahren“ Dichte hinreichend genau zu schätzen. Bestehen diese aber aus 10 Komponenten, so sind selbst bei 2000 Beobachtungen die Stützstellen im 10-dimensionalen Inputraum nur sehr spärlich verteilt. Damit lassen sich keine komplexen Verläufe der „wahren“ Dichte (falls sie komplex sein sollte) schätzen. Je komplexer der vermutete Verlauf der wahren Dichte ist, um so mehr Beobachtungen müssen vorhanden sein und/oder um so kleiner muß die Dimension des Vektors der unabhängigen Variablen gehalten werden. Für die praktische Arbeit leitet sich daraus letztlich die Forderung ab, so wenig unabhängige Variablen wie möglich zu verwenden. Wie sollen diese aber bestimmt werden?

In der Grundform des GRNN (PNN) besteht in der Tat das bedeutsamste Problem in der fehlenden Möglichkeit zur Selektion der relevanten unabhängigen Variablen. So kann etwa im Rahmen der Regressionsanalyse mit Hilfe der t-Statistik getestet werden, ob ein vermuteter Einflußfaktor auch wirklich statistisch signifikant ist. Ist er es nicht, kann man ihn aus der Regressionsgleichung entfernen. Ähnliche Techniken stehen für die Entwicklung von Perceptrons zur Verfügung. Beim GRNN in seiner Grundform ist man hingegen weitgehend darauf angewiesen, die wirklich relevanten Größen zu kennen. Spezielle Techniken stehen hier nicht bereit, vielmehr ist man auf ein „Probieren“ angewiesen. So kann beispielsweise temporär eine unabhängige Variable aus den Vektoren  $\bar{x}$  entfernt und anschließend geprüft werden, ob sich die Prognoseleistung verschlechtert. Bleibt sie in etwa gleich, kann dies als Indiz für eine irrelevante Einflußgröße gewertet werden. Die Kombination von effizienten Merkmalsextraktionsalgorithmen mit dem GRNN zu einem integrierten Verfahren beseitigt diese inhärente Schwäche und wird im Anschluß näher beschrieben.

Schließlich existieren beim GRNN (PNN) noch weitere „Freiheitsgrade“. So sind verschiedene Varianten bei der Festlegung der Gewichtungsfunktion und des Abstandsmaßes denkbar. In der konkreten Anwendung bleibt diesbezüglich auch hier nur als Lösungsansatz ein systematisches „Ausprobieren“.

Diese kurze Skizze einiger praktischer Probleme mag den anfänglichen „Charme“ des GRNN und PNN deutlich relativieren. Deren Vorteile gegenüber Regressionsanalyse und Perceptrons werden tatsächlich mit Problemen an anderer Stelle erkaufte. Trotzdem darf die Anwendung des GRNN vor dem Hintergrund der geschilderten Probleme nicht als hoffnungslos abgetan werden, denn aufgrund praktischer Erfahrungen kann man (natürlich nur im Sinne einer Heuristik) einige bewährte Spezifikationen und Vorgehensweisen angeben:

- Im Regelfall sollte als Distanzmaß die City-Block Distanz gewählt werden. Die Euklidische Distanz ist sensibler gegen Ausreißer und führt meistens zu einem weniger robusten Verhalten des GRNN.
- Als Gewichtungsfunktion hat sich die einfache Exponentialfunktion bewährt.
- Die Anzahl der unabhängigen Variablen sollte so gering wie möglich gehalten werden. Hier bietet sich die Kombination mit einem leistungsfähigen Merkmalsextraktionsverfahren zu einem integrierten Algorithmus (MSMEV) an.
- Das integrierte Verfahren sollte über einen weiten Bereich des Nachbarschaftsparameters  $\sigma$  iteriert werden.

#### Gegenüberstellung mit andern KNN-Typen

KNN-Typ (Perceptron Familie)	Eigenschaften
Single-Layer Perceptron	<ul style="list-style-type: none"> <li>• einfaches, zweilagiges Netzwerk</li> <li>• rechentechnisch einfach und schnell</li> <li>• nur lineare (oder sehr einfache nichtlineare) Probleme bearbeitbar</li> </ul>
Multi-Layer-Perceptron	<ul style="list-style-type: none"> <li>• mehrere verdeckte Schichten</li> <li>• bei Einsatz nichtlinearer Outputfunktionen nichtlineare Datenanalysen möglich</li> <li>• deutlich höhere Rechenzeiten</li> </ul>
Recurrent Perceptron	<ul style="list-style-type: none"> <li>• beliebige Verknüpfungsstrukturen</li> <li>• Berücksichtigung von Interdependenzen auch auf der Outputschicht</li> <li>• durch Fixpunktiteration Rechentechnisch sehr langsam</li> <li>• bei Fehlen eines stabilen Fixpunktattraktors für alle Inputvektoren und Zustände der Gewichtsmatrix undefinierter Outputzustand</li> </ul>
KNN-Typ (RBF Familie)	Eigenschaften
LVQ	<ul style="list-style-type: none"> <li>• einfaches, zweilagiges Netzwerk</li> <li>• rechentechnisch einfach und schnell</li> <li>• reiner Klassifikator</li> <li>• auch nichtlinear trennbare Probleme bearbeitbar</li> </ul>
CPN	<ul style="list-style-type: none"> <li>• fünfziges Netzwerk (in reduzierter Form nur dreilagig)</li> <li>• doppelt-assoziativer Speicher (in reduzierter Form nur einfach-assoziativ)</li> </ul>

	<ul style="list-style-type: none"> <li>• im Kern aber reiner Klassifikator</li> <li>• rechentechnisch einfach und schnell</li> <li>• auch nichtlineare Probleme bearbeitbar</li> </ul>
RCE	<ul style="list-style-type: none"> <li>• dreilagiges Netzwerk</li> <li>• algorithmisch etwas aufwendiger, aber sehr schnell</li> <li>• reiner Klassifikator</li> <li>• auch nichtlinear trennbare Probleme bearbeitbar</li> </ul>
GRNN	<ul style="list-style-type: none"> <li>• nicht unbedingt ein KNN, aber als vierlagiges Netzwerk realisierbar</li> <li>• als Erweiterung des LVQ oder RCE interpretierbar</li> <li>• algorithmisch sehr einfach und rechentechnisch (bei nicht zu umfangreichen Datenmengen) sehr schnell</li> <li>• auch nichtlineare Funktionen approximierbar</li> </ul>

#### 5.3.4.4 Merkmalsextrahierendes Mustererkennungsverfahren (MSMEV)

Wie in den vorangegangenen Kapiteln bereits dargelegt, gibt es eine Reihe von Sekundäreffekten, die eine analytisch geschlossene Darstellung von Sensorsystemen mittlerer bis hoher Komplexität vereiteln. Zwischen den folgenden beiden Positionen der Informationsverarbeitungskette klafft daher eine Lücke, die nicht durch ein einfaches mathematisches Modell geschlossen werden kann: Auf der einen Seite liegen Meßdaten vor, über deren Zusammenhang mit den mannigfaltigen physikalischen Ursachen zwar Theorien vorliegen, die aber (zumindest zur Zeit) nicht durch eine Umkehrabbildung in diese zurücküberführt werden können (wie bei einer einfachen Sensorkennlinie). Auf der anderen Seite besteht die Anforderung nach einer möglichst zuverlässigen Modellierung des Ursache-Wirkungs-Zusammenhangs, d.h. aus den vorliegenden Daten soll eine möglichst präzise Aussage über den verursachenden Zustand abgeleitet werden.

Das im folgenden näher beschriebene Verfahren soll zur Schließung dieser Lücke dienen und hat sich bereits in zahlreichen Anwendungen bewährt. In der ersten Stufe werden die sensorischen Daten unter Einbeziehung des vorhandenen Vorwissens zu einer Merkmalsdatenbank aufbereitet. Typischerweise handelt es sich hierbei um einen mehrdimensionalen Zeitreihenvektor, der durch einfache Transformationen aus den Meßdaten generiert wird und möglichst viele Charakteristika derselben widerspiegelt (z.B. charakteristische Punkte im Kurvenverlauf eines temperaturmoduliert betriebenen Halbleitersensors).

Die unterstellte Tatsache, daß ähnliche physikalische Ursachen zu ähnlichen Reaktionen des Sensors führen wird anschließend dazu genutzt, die Umkehrabbildung Meßvektor->Ursache auf implizite Weise, durch einen Ähnlichkeitsgewichteten Mustervergleich zu konstruieren.

Allerdings sind in der Regel nicht alle im Rohvektor enthaltenen Informationen jedoch tatsächlich mit allen nachzuweisenden physikalischen Ursachen korreliert; einige nur mit Teilen davon, andere überhaupt nicht. Das Vorhandensein irrelevanter Information führt bei gleicher Gewichtung zu einer Verfälschung der Aussage, so daß aus der Gesamtheit der aus der Messung hervorgegangenen Informationen für die weitere Verarbeitung diejenigen herausgefiltert

werden müssen, aus denen sich eine Aussage über das Vorhandensein eines Gases, eines Gemisches, eines Brandes usw. ableiten läßt.

Als Maß für die Relevanz der einzelnen Parameter kommen verschiedene Gütefunktionen in Betracht. Entscheidend hierbei ist, daß ein einzelner Parameter für sich alleine eine nur geringe, in Kombination mit einem weiteren jedoch eine sehr hohe Aussagekraft haben kann. Daher werden mehrere Parameter zu einem Merkmalsvektor zusammengefaßt und dieser mit Hilfe einer Gütefunktion bewertet. Die Merkmalsgüte kann sodann auf verschiedene Arten ermittelt werden, die zweckmäßigste ist jedoch, gleich die Abbildungsgüte eines zuvor gewählten Approximators als Bewertungsmaß heranzuziehen. Diese Gütefunktion dient als Maß bei der Auswahl der relevanten Merkmale aus dem Rohdatensatz. Als universell geeignet hat sich dabei das weiter oben vorgestellte GRNN erwiesen. Der Wert der Gütefunktion entspricht dann dem reziproken Approximations-/Klassifikationsfehler bei gegebener Merkmalsanzahl. Ein gewünschter Nebeneffekt des Verfahrens besteht darin, daß das Training des Approximators simultan zur Merkmalselektion erfolgt.

Die Matrix der Merkmalsvektoren wird dazu in zwei Teile zerlegt: die Referenz- bzw. Codebook-Vektoren und die Testvektoren. Bei kleineren Datensätzen empfiehlt sich ein Verfahren, bei dem man jeweils einen Vektor aus der Gesamtmenge entfernt und diesen mit Hilfe der restlichen probeweise klassifiziert (Jack-Knife bzw Kreuzvalidierungsverfahren). Zusätzlich wird ein Teil der Daten für späteres Out-of-Sample-Testing reserviert, um die Generalisierungsfähigkeit des erzeugten Modells zu überprüfen (Vermeidung des „Auswendiglernens“ der Trainingsdaten).

Die Funktionsapproximation erfolgt durch eine gewichtete Superposition der Funktionswerte der nächsten Nachbarn in Abhängigkeit von deren Abstand zum aktuellen Merkmalsvektor.

$$f(\bar{x}) = \frac{\sum_{i=1}^N f(\bar{x}_i) \cdot w(\sigma, \|\bar{x} - \bar{x}_i\|)}{\sum_{i=1}^N w(\sigma, \|\bar{x} - \bar{x}_i\|)} \quad (5.33)$$

Hierbei sind  $\bar{x}$  der zu klassifizierende Merkmalsvektor und  $\bar{x}_i$  die in der Referenzmusterbibliothek enthaltenen Codebook-Vektoren. Mit Hilfe des Skalierungsparameters  $\sigma$  läßt sich der Radius der für die Interpolation maßgeblichen Nachbarschaft einstellen.

Als Gewichtungsfunktion  $w$  kann zum Beispiel eine quadratisch abklingende Exponentialfunktion verwendet werden. Bei der Wahl der Metrik ist die City-Block Distanz aufgrund ihrer größeren Robustheit gegenüber Ausreißern der euklidischen vorzuziehen.

Die zur Ermittlung der Gütefunktion notwendige probeweise Klassifizierung erfolgt, indem man eine bestimmte Anzahl von Komponenten für die Zusammensetzung der Referenzvektoren auswählt und den Testvektor auf die beschriebene Weise klassifiziert. Die Differenz zwischen der erzielten Ausgabe und dem gewünschten Wert wird als Maß für die Güte des Klassifikators verwendet. Dieses kann symmetrisch sein (z.B. mittlere quadratische Abweichung zwischen Soll- und Ist-Werten) oder aber toleranter gegenüber der einen oder anderen Art von Fehlern sein (z.B. stärkere Bestrafung eines nicht erkannten Fehlers als eines fälschlicherweise als Fehler deklarierten Nichtfehlers).

Für das probeweise Zusammensetzen der Merkmalsvektoren aus einer Untermenge der Merkmalsgesamtheit ist meist eine Suchstrategie erforderlich, da die systematische Suche über alle Merkmalskombinationen exponentiell mit der Merkmalszahl anwächst. Im Rahmen der vorliegenden Untersuchungen wurde ein bottom-up Ansatz gewählt, bei dem probeweise zunächst alle Zweierkombinationen getestet werden. Die mit dem höchsten Wert der Fitneßfunktion wird beibehalten. In den darauffolgenden Schritten wird jeweils dasjenige Merkmal hinzugenommen, das bei der jeweiligen Merkmalsanzahl den höchsten Gütewert erbringt.

Abschließend wird diejenige Merkmalskombination beibehalten (Anzahl und Art), bei der die Gütefunktion ihr globales Maximum erreicht. Die im Rahmen dieses Prozesses selektierten Komponenten werden anschließend zum Aufbau der Referenzvektorbibliothek verwendet.

Der MSMEV-Algorithmus protokolliert die Fehlerfunktion in Abhängigkeit von Merkmalsanzahl und Art der ausgewählten Merkmale auf jeder Stufe. Für die global beste Merkmalskombination wird automatisch die passende Referenzmusterbibliothek für den GRNN-Klassifikator generiert.

#### 5.3.4.4.1 Einfluß der freien Parameter

Der Charme des Verfahrens liegt in der geringen Anzahl der freien (und somit vom Anwender vorzugebenden) Parameter. Da das Verfahren eine Bottom-up-Suchstrategie verwendet, eignet es sich hervorragend zur Beschränkung der maximal erwünschten Merkmalsanzahl. Wird es z.B. zur Sensorauswahl eingesetzt, so kann man den Suchvorgang auf die gewünschte maximale Sensoranzahl einstellen und alle anderen Lösungen bereits im Vorfeld verwerfen. Der Suchvorgang wird hierdurch bei hoher Dimensionalität des Rohdatenvektors erheblich beschleunigt.

Wie in 5.3.4.3.2.3 bereits dargelegt, handelt es sich bei dem Nachbarschaftsparameter  $\sigma$  um den entscheidenden (weil einzigen „echten“) freien Parameter des GRNN/PNN. Zur Vereinfachung der folgenden Betrachtungen werden folgende Substitutionen vorgenommen:

$$R_0 = \frac{1}{2\sigma^2} \quad (5.34)$$

$d(k)$  = Abstand zwischen Eingabevektor (unbekannt) und  $k$ -tem Referenzvektor

$f(k)$  = Faktor der Gewichtung von Referenzvektor  $k$ , unnormiert

$m(k)$  = Merkmalsausprägung des  $k$ -ten Referenzvektors

$v$  = zu klassifizierender Eingabevektor

Nahe benachbarte Referenzvektoren sollen bei der Bestimmung des zu klassifizierenden Eingabevektors ein sehr hohes „Mitspracherecht“ besitzen:  $d(k) \rightarrow 0 \Rightarrow f(k) = 1$

Die abstandsabhängige Gewichtung  $f(k) = e^{-R_0 \cdot d^2}$  sieht für  $R_0=1$  folgendermaßen aus:

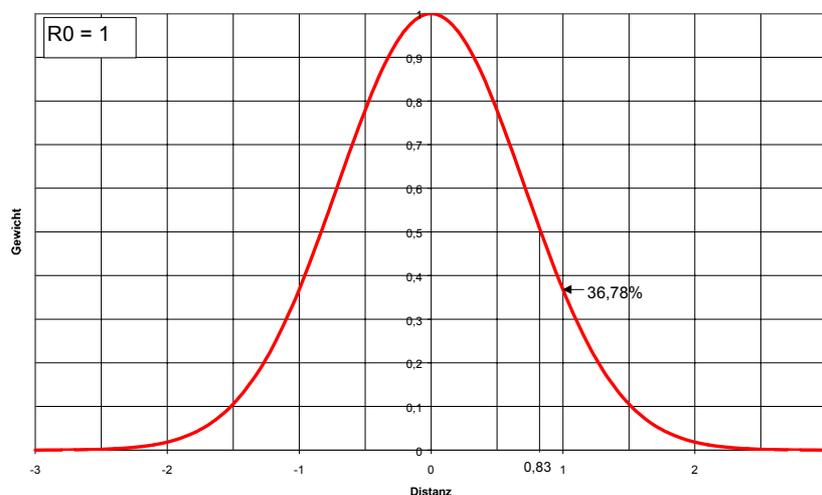
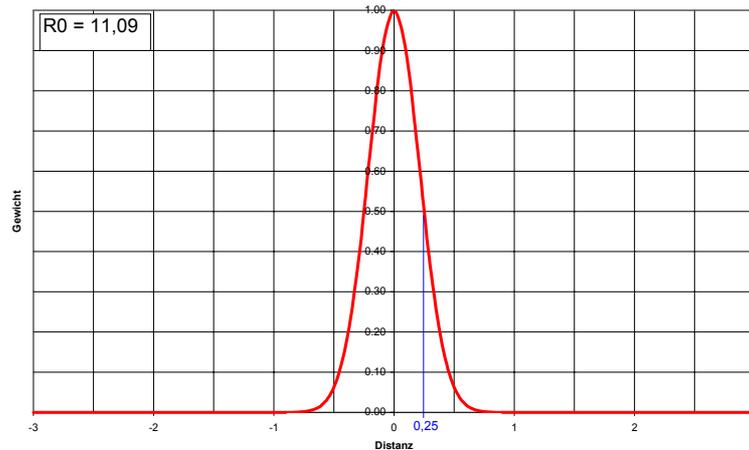


Abb. 5.37: Gewichtungsfunktion für  $R_0=1$

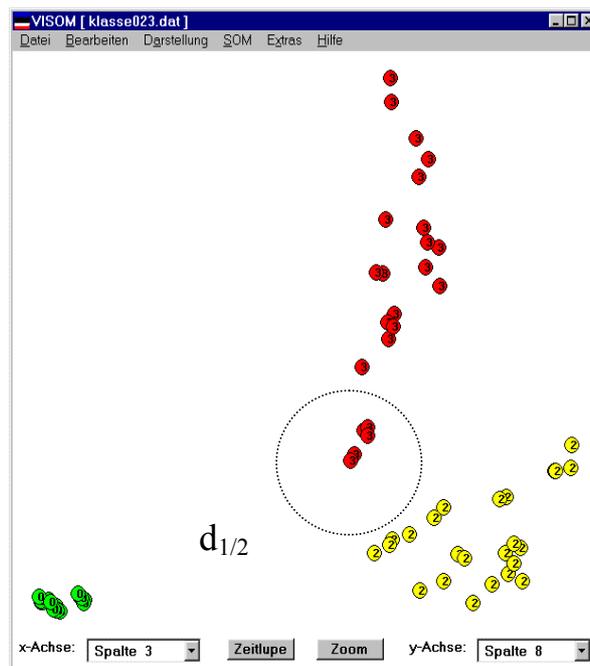
Der Halbwertsabstand befindet sich auf einer Hypersphäre mit Radius  $d_{1/2} = \sqrt{\frac{\ln(2)}{R_0}}$  um die Codebookvektoren herum. Die Summe der Gewichtungsfaktoren wird auf 1 normiert. Im Falle von  $N$  Codebookvektoren im gleichen Abstand von  $v$  würde also jeder ein "Mitspracherecht" von  $1/N$  erhalten.

Aus Gründen der numerischen Stabilität ist eine Skalierung der Merkmale auf das Intervall  $[0...1]$  empfehlenswert. Die Manhattan-Metrik-Distanzfunktion ist dabei so implementiert, daß durch die Merkmalszahl dividiert wird, d.h.  $R_0$  kann näherungsweise unabhängig von der Merkmalsanzahl festgehalten werden, so daß  $d(k)$  immer im Intervall  $[0...1]$  liegt, sofern die Einzelkomponenten entsprechend skaliert wurden. Je größer  $R_0$  gewählt wird, desto mehr wird das Verfahren zum nächste-Nachbar-Klassifikator.



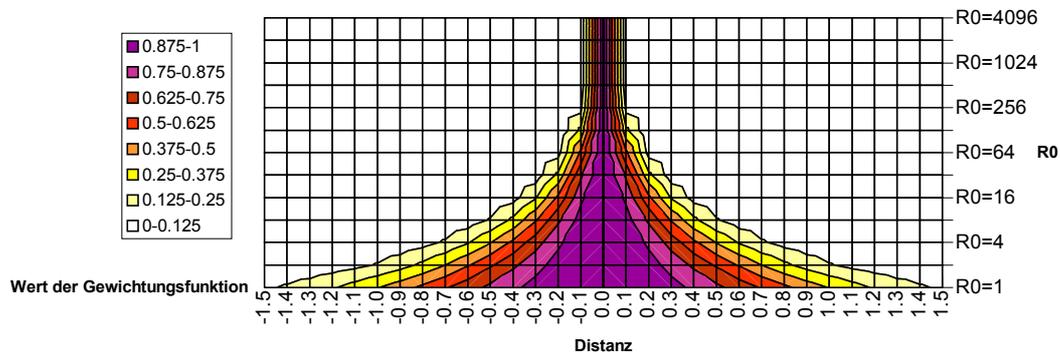
**Abb. 5.38:** Gewichtungsfunktion mit kleinerem Hypersphärenradius

Die folgende Abbildung zeigt den Halbwertsradius für die Gewichtungsfunktion aus Abb. 5.38 ( $R_0=11,09 \Rightarrow r_{1/2} = 0,25$ ) in einem Meßdatensatz mit 3 Klassen und Merkmalsausprägungen im Intervall  $[0...1]$ :



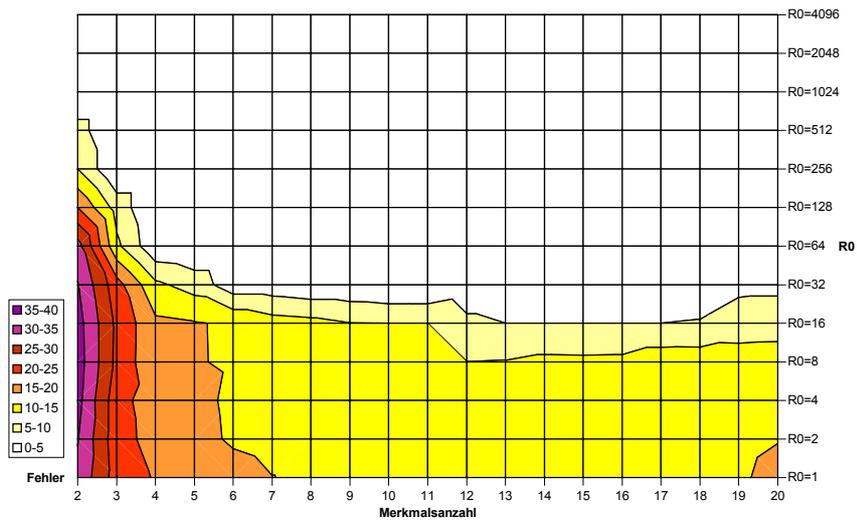
**Abb. 5.39:** 50%-Einflußsphäre für Referenzvektor der Klasse 3

Der folgende Kontur-Plot gibt einen Überblick über den Zusammenhang von gewähltem  $R_0$  und Wert der Gewichtungsfunktion versus  $R_0$ :



**Abb. 5.40:** Wert der Gewichtungsfunktion in Abhängigkeit von  $d$  und  $R_0$

Für die Merkmalsdaten aus Abb. 5.39 ergibt sich daraus folgende Fehlerfläche:



**Abb. 5.41:** Fehler versus Merkmalsanzahl und  $R_0$

### 5.3.4.4.2 Zerlegung von Entscheidungsprozessen

#### Beispielapplikation: Sensorsystem zur Früherkennung von Lösungsmitteldämpfen

In Fortführung der Untersuchungen aus 3.2 wurden in Zusammenarbeit mit dem Lehrstuhl für Meßtechnik der Universität des Saarlandes Versuchsreihen mit temperaturmodulierten Halbleitersensoren des selben Typs gefahren. Angestrebt wurde die Identifikation von insgesamt sechs Lösungsmitteln deutlich unterhalb UEG, und zwar im Konzentrationsbereich

$$\frac{UEG}{10000} \leq Konz. \leq \frac{UEG}{10} \quad (5.35)$$

Der Identifizierungs- und Klassifikationsprozeß wurde zur Entlastung des Klassifikationssystems in Teilentscheidungsprozesse zerlegt: Die Konzentrationsbestimmung erfolgte mit Hilfe eines Breitbandsensors, der wie in 3.2. mit konstanter Temperatur der gassensitiven Schicht betrieben wurde, wobei diese so gewählt wurde, daß ein möglichst gutes Ansprechen auf alle Zielgase gewährleistet ist.

Zur Unterscheidung von insgesamt 6 Lösungsmitteln sollten lediglich 1 bis 2 Identifizierungssensoren eingesetzt werden. Um dieses Ziel erreichen zu können wurden diese Sensoren temperaturmoduliert betrieben, um zusätzliche Informationen über die Art des vorliegenden Lösungsmittels zu gewinnen.

Der Auswertealgorithmus wurde dreistufig aufgebaut:

- Auswertung von Frühindikatoren
- Identifikation der Lösungsmittel
- Ermittlung des Konzentrationsbereichs

Basis für die Auswertung bildete ein 30-sekündiger Zyklus, der von einem Mikrokontroller aufgezeichnet und zur Auswertung an einen PC übertragen wurde. Folgende Signale erweisen sich als zuverlässige Frühindikatoren für die Anwesenheit eines Lösungsmittels, noch bevor dieses einwandfrei identifiziert werden kann:

- Signal des Breitbandsensors oberhalb eines festgelegten Schwellwertes
- mittleres Signal des Breitbandsensors noch unter Schwellwert, aber hoher Signalanstieg im Zyklus
- Veränderungen im Signalverlauf der temperaturmodulierten Sensoren

Die Identifizierung des Lösungsmittels erfolgte im ersten vollständig auswertbaren (=quasistationären) Zyklus anhand charakteristischer Merkmale im Signalverlauf der temperaturmodulierten Sensoren. Dieser Prozeß wurde wiederum in Teilprozesse zerlegt (s.u.). Nachdem das Lösungsmittel identifiziert wurde, kann sodann die korrespondierende Kennlinie des Breitbandsensors zur Konzentrationsbestimmung herangezogen werden.

Die folgenden beiden Abbildungen zeigen die Signalverläufe innerhalb eines Temperaturzyklus für verschiedene Lösungsmittel:

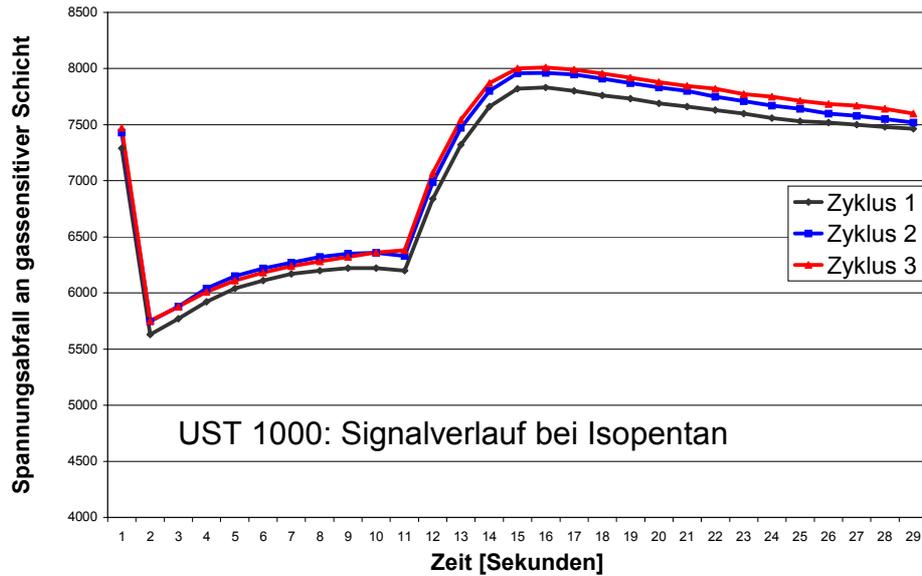


Abb. 5.42: 3 Zyklen des UST1000 bei Anwesenheit von Isopentan

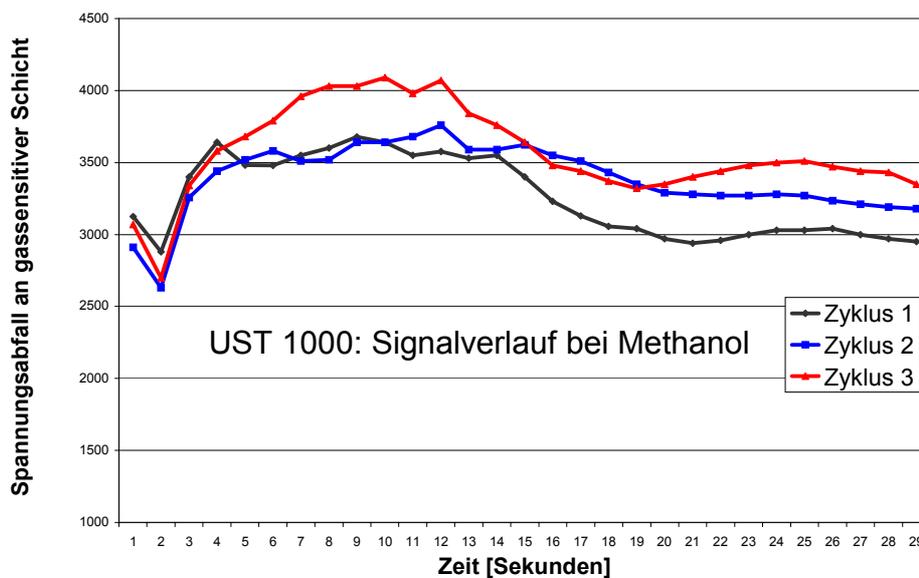


Abb. 5.43: 3 Zyklen des UST1000 bei Anwesenheit von Methanol

Die vorliegende Aufgabenstellung ist prototypisch für eine ganze Reihe verwandter Probleme: Zwar sind einzelne Merkmale im Signalverlauf sehr gut zur Unterscheidung einzelner Gruppen/Lösungsmittel geeignet, nicht jedoch zur Separierung aller Lösungsmittel auf einmal (zu hoher Noise-Anteil). Daher wurde der Klassifikationsprozeß in Teilschritte aufgespalten.

Die Hierarchie des Klassifikators wurde anhand der Separierbarkeit verschiedener Lösungsmittel(-klassen) ausgewählt; Chemische Ähnlichkeiten (CH-Verbindungen vs. CHO-Verbindungen) spiegeln sich in wohlseparierbaren Clustern auf der Merkmalsebene wieder.

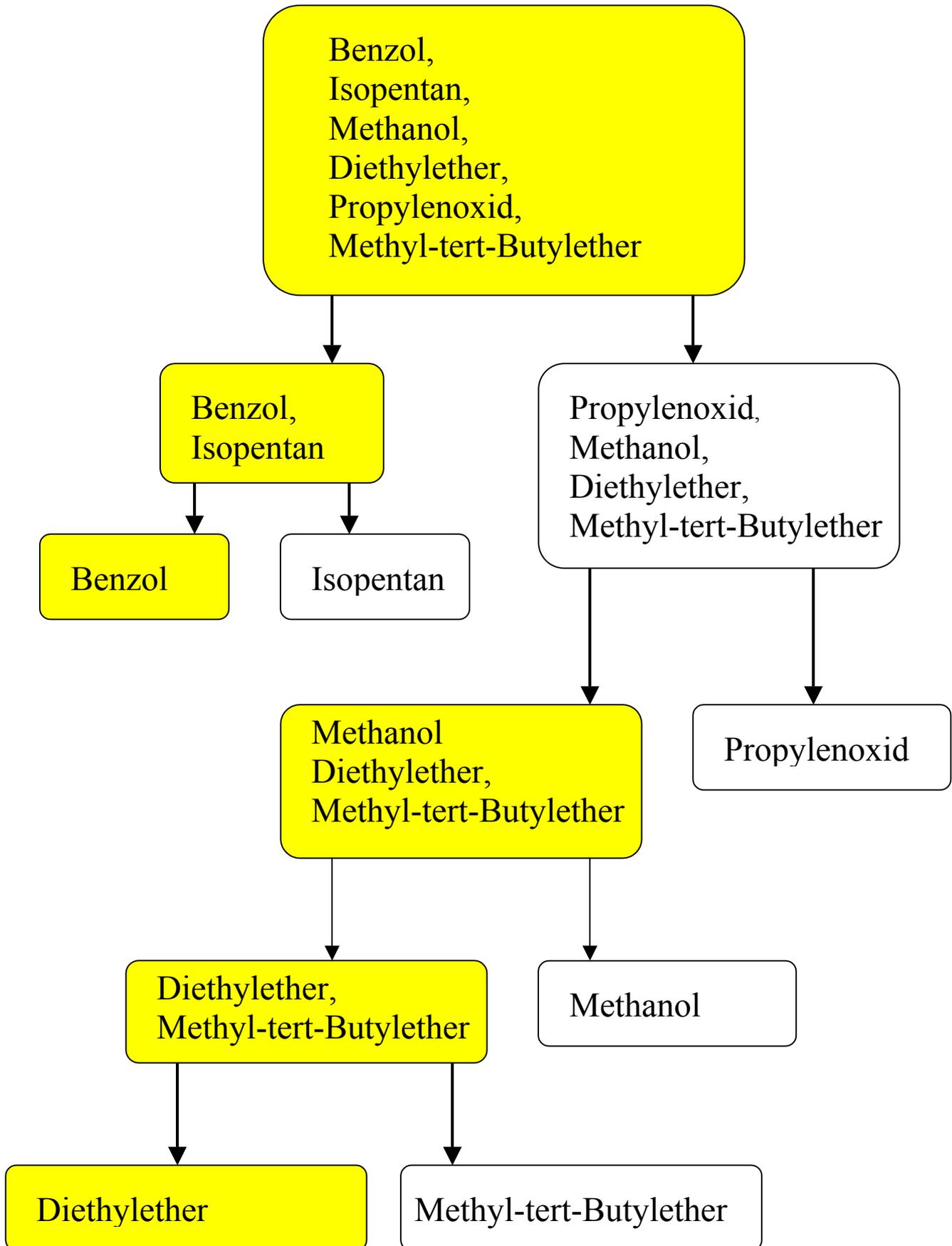


Abb. 5.44: Hierarchische Klassifikation der Lösungsmittel(-gruppen)

Durch die Zerlegung der Gesamtentscheidung in eine Klassifikationsbaum mit Binärentscheidungen ist es möglich, Merkmale zu extrahieren, die eine fehlerfreie Klassenzuordnung gewährleisten:

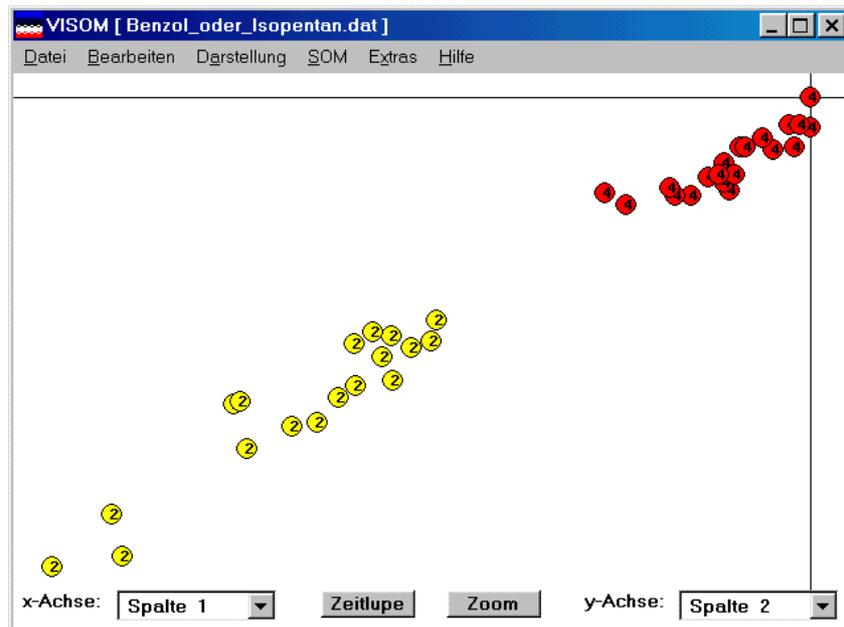


Abb 5.45: Entscheidung innerhalb der Untergruppe ( **Isopentan**, **Benzol** )

Die Signale des Sensors UST 3000 genügten zur Gewinnung der für diese Teilentscheidung notwendigen Merkmale:

$$\text{Merkmal1} = \frac{UST3000(t = 3...11)}{UST3000(t = 19...29)}, \quad \text{Merkmal2} = \frac{UST3000(t = 9...11)}{UST3000(t = 27...29)}$$

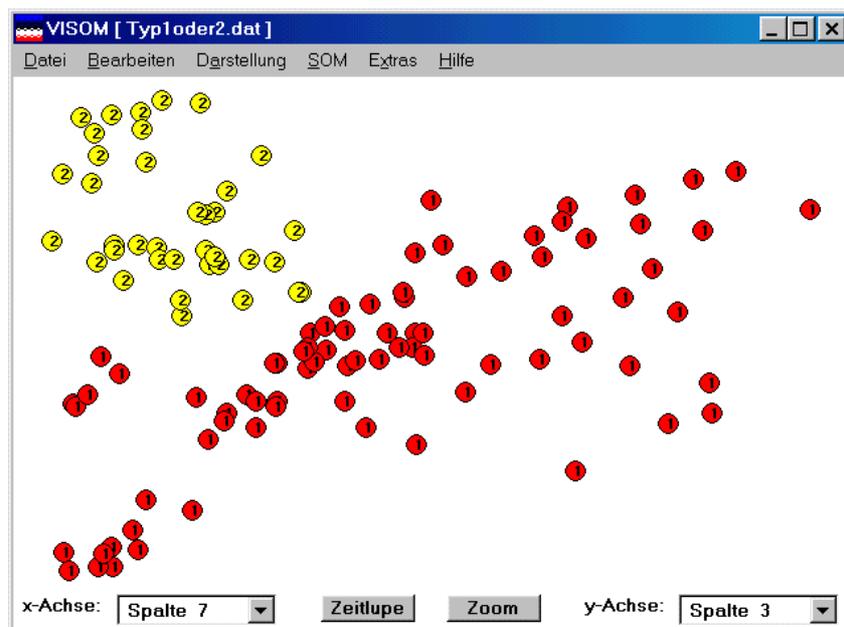


Abb. 5.46: Entscheidung innerhalb der Untergruppe ( **Gruppe1**, **Gruppe2** )

wobei **Gruppe 1** = ( Methyl-tert-Buthylether, Methanol, Diethylether, Propylenoxid )  
 und **Gruppe 2** = ( Benzol, Isopentan )

Zur Zuverlässigen Separierung dieser beiden Gruppen sind die Beiträge zweier Sensoren zu verschiedenen Zeitpunkten im Zyklus erforderlich:

$$\text{Merkmal3} = \frac{UST1000(t=9\dots11)}{UST1000(t=27\dots29)} \quad \text{und} \quad \text{Merkmal7} = \frac{\text{Steigung}(UST1000(t=3\dots11))}{TGS2620(t=1\dots29)}$$

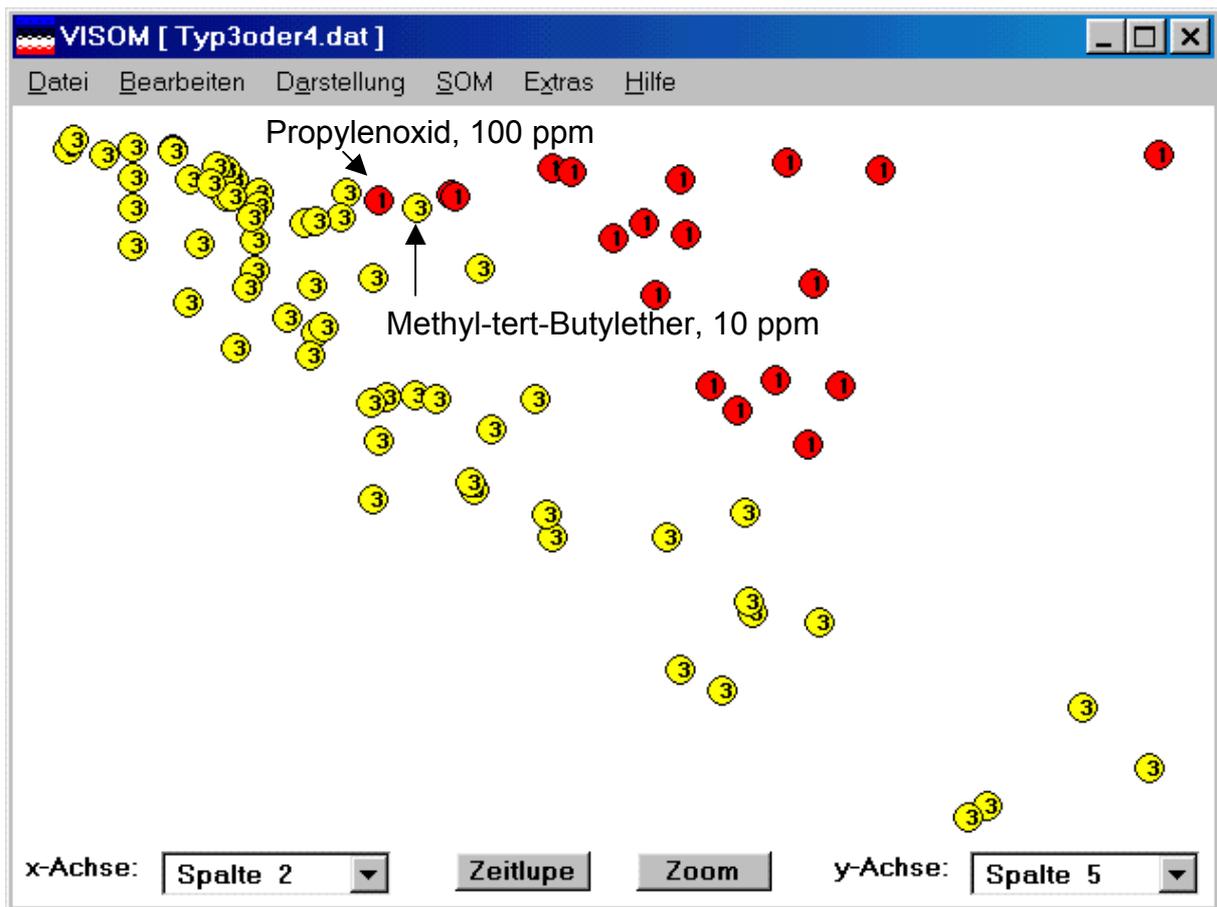
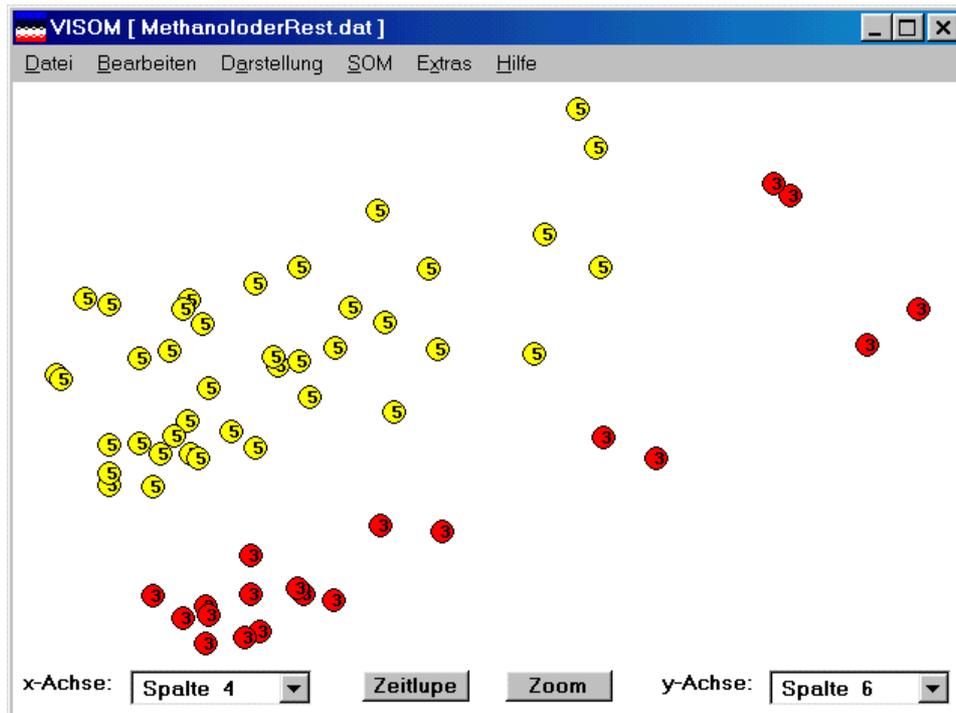


Abb. 5.47: Überlappende Merkmale

Abb. 5.47 zeigt die Merkmale zur Unterscheidung zwischen **Propylenoxid** und der Untergruppe (**Methanol, Diethylether, Methyl-tert-Buthylether**). Im Vergleich zu den beiden vorherigen Beispielen wird erkennbar, daß diese Entscheidung mit höherer Unsicherheit behaftet ist (überlappende Merkmale).

$$\text{Merkmal2} = \frac{\text{Anstieg}(UST1000(t=19\dots29))}{TGS2620(t=1\dots29)} \quad \text{und} \quad \text{Merkmal5} = \frac{UST3000(t=3\dots11)}{UST3000(t=19\dots29)} / TGS2620(t=1\dots29)$$



*Abb. 5.48: Gute Separierbarkeit*

In Abb. 5.48 sind die Merkmale zur Unterscheidung der Gruppen **Methanol** und **(Diethylether, Methyl-tert-Buthylether)** dargestellt:

$$\text{Merkmal}_4 = \frac{\text{Anstieg}(UST1000(t=19\dots29))}{TGS2620(t=1\dots29)}, \quad \text{Merkmal}_6 = \frac{UST1000(t=9\dots11) - UST1000(t=13\dots15)}{TGS2620(t=1\dots29)}$$

Die Merkmalsmengen sind Überlappungsfrei, eine fehlerfreie Zuordnung ist bereits mit nur zwei Merkmalen erreichbar.

## 5.4 Erkennung kritischer Zustände in Dynamischen Systemen

### 5.4.1 Log-periodische Oszillationen als Frühindikatoren für Phasenübergänge

Jüngere Untersuchungen weisen darauf hin, daß scheinbar unvorhersehbare und auf den ersten Blick mit wenig Gemeinsamkeiten behaftete Ereignisse wie z.B. Erdbeben und sog. Börsencrashes (d.h. einschneidende Marktkorrekturen) sich möglicherweise innerhalb eines gemeinsamen theoretischen Rahmenwerks als Phasenübergänge mit log-periodischem Verhalten im Vorfeld beschreiben lassen [Hua 96], [Joh 98].

Ging man beispielsweise im Fall des Aktienmarktes im Rahmen der Efficient Market Theorie davon aus, daß sich Preisveränderungen als Random-Walk vollziehen und allenfalls einer „Fat-Tail“-Ergänzung der Normalverteilung zu ihrer Beschreibung bedürfen, legen die Ergebnisse dieser Untersuchungen den Schluß nahe, daß es sich statistisch gesehen bei den sog. Crashes um Ausreißer handelt, die sich darüber hinaus durch sog. Precursor ankündigen [Van 98]. So läßt sich beispielsweise die Anzahl  $N$  der Marktrückgänge ( $DD = Draw\ Downs$ ) des Dow Jones in Abhängigkeit von ihrer Stärke mit Hilfe folgender Gleichung sehr gut beschreiben [Joh 98]:

$$N(DD) = N_0 \cdot e^{-DD/DD_c}, \quad \text{mit } DD_c \approx 1,8\% \quad (5.36)$$

Marktrückgänge wie in 1929 bzw. 1987 dürften demnach nur alle 160-3000 Jahrhunderte auftreten. Ihr häufigeres Auftreten gibt Anlaß zur gesonderten Modellierung dieser Phänomene.

Zur Beschreibung der wachsenden Korrektur-Wahrscheinlichkeit in einem zunehmend überbewerteten Markt betrachte man das Crashrisiko  $h(t)$ . Das Auftreten eines Crashes wird nie mit 100%iger Genauigkeit vorhersagbar sein, da die Marktteilnehmer noch eine hinreichend große Wahrscheinlichkeit für ein Ausbleiben desselben annehmen können müssen, um zumindest investiert zu bleiben.

Ein Crash/abrupter Marktrückgang findet dann statt, wenn eine große Gruppe von Marktakteuren gleichzeitig Verkaufsaufträge platziert, d.h. es tritt plötzlich ein emergentes Phänomen auf: der Zustand niedriger Ordnung ausgeglichener Käufe und Verkäufe schlägt um in einen Zustand koordinierter Verkäufe. In diesem Sinne stellt ein Crash einen Phasenübergang von einem ungeordneten Zustand in einen hochgradig geordneten Zustand dar, hervorgerufen durch die zunehmende Korreliertheit im Verhalten der Akteure.

Ein relativ einfacher Ansatz zur Beschreibung der zeitlichen Entwicklung des Crashrisikos stellt Gleichung (5.37) dar:

$$\frac{dh}{dt} = C \cdot h^\delta, \quad \text{mit } \delta > 1 \quad (5.37)$$

wobei  $C$  eine positive Konstante ist und  $\delta$  die Anzahl der beteiligten Agenten repräsentiert. Die Integration von (5.37) liefert

$$h(t) = \frac{B}{(t_c - t)^\alpha}, \quad \text{mit } \alpha \equiv \frac{1}{\delta - 1} \quad (5.38)$$

Aus ökonomischen Gründen muß  $\alpha$  zwischen 0 und 1 liegen, so daß für  $\delta$  die Beziehung  $2 < \delta < +\infty$  gilt. Der kritische Zeitpunkt  $t_c$  markiert das Auftreten des gesuchten kritischen Zustandes, d.h. das „Platzen“ der Blase. Allerdings existiert eine endliche Wahrscheinlichkeit für ein Ausbleiben des kritischen Zustandes, d.h. einer weichen Landung, da die Marktteilnehmer anderenfalls den Crash antizipieren und vorzeitig aus dem Markt ausscheiden würden, was zu einer Paradoxie in der Modellierung führen würde, d.h. es muß gelten

$$1 - \int_{t_0}^{t_c} h(t) dt > 0 \quad (5.39)$$

Unter der vereinfachenden Annahme, daß der Preis  $p$  des untersuchten Index beim Auftreten des Crashes um einen festen Prozentsatz  $\gamma \in (0,1)$  fällt, z.B. 30% des Zuwachses gegenüber einem niedrigeren Niveau  $p_1$  aus der Vergangenheit, läßt sich die Preisentwicklung vor dem Crash wie folgt beschreiben:

$$dp = \mu(t)p(t)dt - \gamma[p(t) - p_1]dj \quad (5.40)$$

wobei  $j$  eine Sprungvariable darstellt, die vor dem Sprung den Wert 0 und danach den Wert 1 annimmt und  $E_t[dj] = h(t)dt$  gilt.

Unter der Annahme, daß alle zukünftigen Erwartungen sofort „eingepreist“ werden (Martingale-Bedingung, d.h. Unmöglichkeit eines „free lunch“), also

$$E_t[p(t')] = p(t) \quad \forall t' > t \quad (5.41)$$

erhält man aus (5.40)

$$\mu(t)p(t) = \gamma[p(t) - p_1]h(t) \quad (5.42)$$

d.h. der Return  $\mu$  wächst mit der Crashwahrscheinlichkeit  $h(t)$ , um die Akteure für das erhöhte Risiko zu kompensieren. Mit anderen Worten: die Marktteilnehmer werden durch zunehmend höhere Renditen gegen Ende der Überbewertungsphase belohnt und zum Investiertbleiben bzw. zum Neuengagement bewegt.

Setzt man (5.42) in (5.40) ein, so erhält man für  $p(t) - p(t_0) < p(t_0) - p_1$  als Lösung der Differentialgleichung für  $t < t_c$ :

$$p(t) \approx p(t_0) + \gamma[p(t_0) - p_1] \int_{t_0}^t h(t') dt' \quad (5.43)$$

Durch Einsetzen von (5.38) in (5.43), erhält man aus der Martingale-Bedingung folgende Preisentwicklung vor dem Crash:

$$p(t) \approx p_c - \gamma \frac{B}{\beta} (t_c - t)^\beta \quad (5.44)$$

mit  $\beta = 1 - \alpha \in (0,1)$ .  $p_c$  ist der Preis zum kritischen Zeitpunkt  $t_c$  unter der Annahme, daß der Crash noch nicht stattgefunden hat.

Ausgefeiltere Modelle zur Beschreibung von Imitationsverhalten in hierarchischen Systemen führen ebenfalls zu dem bereits in (5.44) sichtbaren Potenzgesetzcharakter [Joh 98b], [Sor 98]. Darüber hinaus prognostizieren diese Modelle, daß es sich bei dem kritischen Exponenten  $\alpha$  um eine komplexe Zahl handelt. Die Terme erster Ordnung der allgemeinen Lösung für  $h(t)$  lauten dann

$$h(t) \approx B_0(t_c - t)^{-\alpha} + B_1(t_c - t)^{-\alpha} \cos[\omega \log(t_c - t) + \Psi] \quad (5.45)$$

Die Preisentwicklung vor dem Crash ergibt sich aus

$$p(t) \approx p_c - \frac{\gamma}{\beta} \{B_0(t_c - t)^\beta + B_1(t_c - t)^\beta \cos[\omega \log(t_c - t) + \phi]\} \quad (5.46)$$

Das bereits in (5.44) sichtbare Power-Law-Verhalten wird also von log-periodischen Oszillationen überlagert.

### 5.4.2 Beispiel: Nasdaq-Korrektur 2000

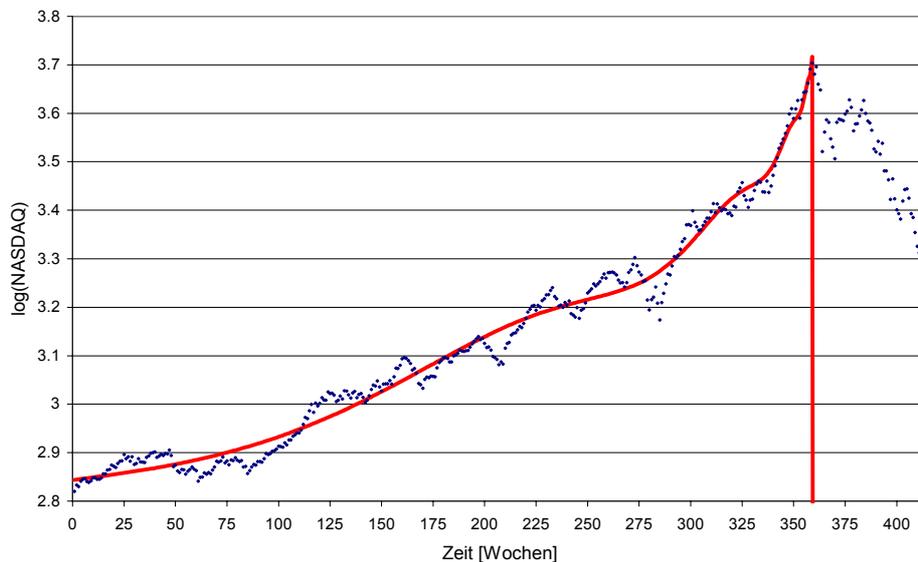
Als Datenbasis für die folgenden Untersuchungen diente die logarithmierte Wochendaten-Zeitreihe des NASDAQ Composite Index im Zeitraum vom 02.05.1993- 12.08.2001. Mit Hilfe der in 2.3.7 vorgestellten Verfahren für multivariate Optimierungsprobleme wurde die folgende Gleichung an die historischen Daten angefitet:

$$p(t) = A + B(t_c - t)^z + C(t_c - t)^z \cos[\omega \ln(t_c - t) + \phi] \quad (5.47)$$

Das Markthoch wurde in Woche 359 erreicht. Verwendet man die Daten bis zu diesem Zeitpunkt für den Fit, erhält man folgende Werte für die freien Parameter

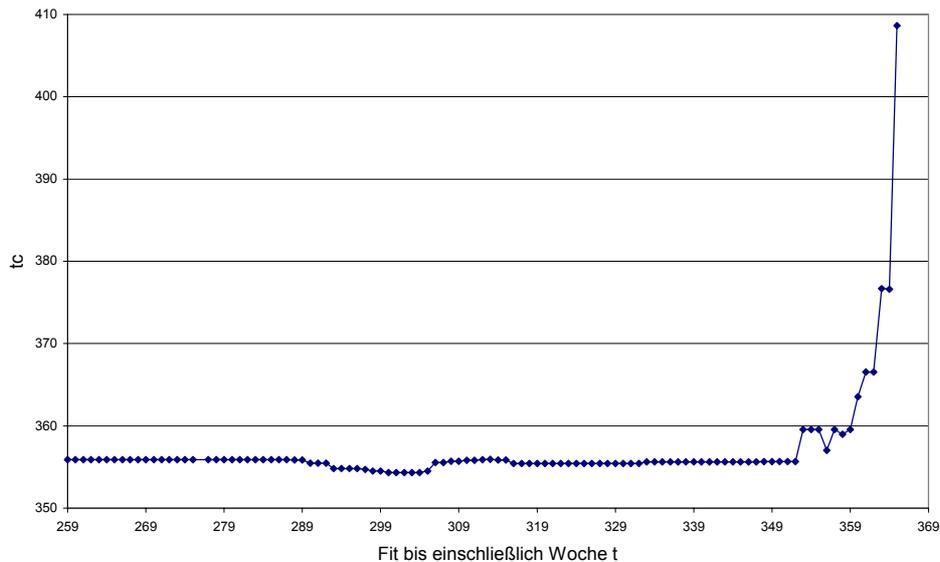
Fitparameter	
A	3.72245
B	-0.05325
C	0.00098
$t_c$	358.998
z	0.48048
$\omega$	7.17461
$\phi$	0.83145

, d.h. der Wert für den kritischen Zeitpunkt  $t_c$  „rastet“ auf dem historischen Hoch in Woche 359 ein. Abb. 5.49 gibt den Verlauf der Fit-Funktion vor dem Hintergrund der historischen Daten wieder.



**Abb. 5.49:** NASDAQ Composite mit Log-Periodischem Fit (02.05.1993- 12.08.2001)

Ausgehend von dieser für die Zeitreihe bis  $t = 359$  erhaltenen Startlösung wurde sukzessive jeweils das letzte Wertepaar  $(t, F(t))$  aus der Zeitreihe entfernt und ein erneuter Fit vorgenommen. Abb. 5.50 zeigt die Werte für den kritischen Zeitpunkt  $t_c$  der jeweils gefundenen Lösungen:



**Abb. 5.50:** Vorhersage des Crashzeitpunktes  $t_c$  versus maximal in den Fit einbezogene Woche  $t$

Aus Abb. 5.48 geht hervor, daß der Wert  $t_c$  bereits lange vor dem kritischen Ereignis auf einen sehr guten und vor allen Dingen zeitstabilen Näherungswert für das Ende der Aufwärtsbewegung zu liegen kommt. Bei Annäherung an den kritischen Punkt springt  $t_c$  auf den exakten Wert von 359. Verlängert man die für den Fit verwendete Zeitreihe schließlich über den kritischen Zeitpunkt hinaus, divergiert die vormals stabile Lösung für  $t_c$ .

Für einen Einsatz als Frühwarnsystem muß die Vorgehensweise lediglich invertiert werden, d.h. man beginnt zu einem Zeitpunkt  $t_k$  mit einem Fit, der bis zu einem Punkt  $t_{k-m}$  in die Vergangenheit zurückreicht und erweitert im Verlauf der Zeit sukzessive das Ende des Fits auf die Werte  $t_{k+1}$ ,  $t_{k+2}$  usw.

Für den praktischen Einsatz sind allerdings noch zwei wesensverwandte Fragen zu klären:

1. Unter welchen Umständen konvergiert bereits eine „frühe“ Lösung gegen den später tatsächlich auftretenden Endwert?
2. Wie lassen sich Fehlalarme zuverlässig ausgrenzen?

Zur Klärung von 1. und möglicherweise auch von 2. könnten sich sog. *Basin-of-Attraction* Untersuchungen als sehr aufschlußreich erweisen, wie sie z.B. zur Visualisierung von Julia- und anderer fraktaler Mengen verwendet werden.

Die hier skizzierte Vorgehensweise zur Erkennung kritischer Zustände ist aufgrund ihrer potentiellen Universalität möglicherweise auch auf andere Anwendungsgebiete übertragbar. Neben den Bereits erwähnten Erdbeben treten log-periodische Oszillationen auch bei gerichteten Diffusionsprozessen in dreidimensionalen Perkulationsclustern auf [Sta 97], womit der Kreis zu den in 5.3.1 beschriebenen Untersuchungen zur Sensormodellierung geschlossen wäre.

## 6. Literatur

- [Alb 95] T. Albrecht, *Gassensoren-Array mit Mustererkennung zur Bestimmung von Gefahrstoffen in Luft*, Berichte aus der Elektrotechnik, Verlag Shaker, 1995
- [Blä 95] G. Bläser, *Studien zur schnellen und selektiven Erkennung toxischer Gasgemische in Luft mit Multisensoren*, Diplomarbeit, Institut für Angewandte Physik der Universität Gießen, 1995
- [Blä 00] G. Bläser, *Zeitabhängige Signale und Kalibration bei Halbleitersensoren*, Inauguraldissertation, Institut für Angewandte Physik der Universität Gießen, 2000
- [Bot 95] H.-H. Bothe: *Fuzzy-Logic*, Springer Verlag, 2. Auflage, Berlin 1995
- [Bun 94] A. Bunde, S. Havlin, eds: *Fractals in Science* (Springer, Heidelberg, 1994)
- [Bun 96] A. Bunde, S. Havlin, eds: *Fractals and Disordered Systems* (Springer, Heidelberg, 2<sup>nd</sup> ed., 1996)
- [Cer 85] V. Cerny, *Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm*, Journal of Optimization Theory and Applications 45 (1985), S. 41-51
- [Dat 95] *Datenanalyse, Anwendung von DataEngine mit Fuzzy Technologien und Neuronalen Netzen*, Hrsg.: H.-J. Zimmermann, VDI-Verlag GmbH, Düsseldorf 1995
- [Due 90] G. Dueck, T. Scheuer, *Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing*, Journal of Computational Physics 90 (1990), S. 161-175
- [Due 93] G. Dueck, *New Optimization Heuristics. The Great Deluge Algorithm and the Record-to-Record Travel*, Journal of Computational Physics 104 (1993), S. 86-92
- [Fle 92] M. Fleischer, *Präparation und Charakterisierung von Ga<sub>2</sub>O<sub>3</sub> –Dünnschichten, Untersuchungen der gassensitiven elektrischen Eigenschaften*, Dissertation, Physik-Departement E-23 der TU München, 1992
- [Fuz 93] fuzzyTECH 3.1, Manual and Reference Book, Revision 310, Inform Software Corporation, December 16, 1993

- [Hei 88] G. Heiland und D. Kohl, *Physical and Chemical Aspects of Oxidic Semiconductor Gas Sensors*, in T. Seiyama (ed.), Chemical Sensor Technology, Vol. I, Kodansha Ltd. Tokyo, Japan in cooperation with Elsevier Science Publishers, Amsterdam, Netherlands, 1988, 15-38
- [Hua 96] Y. Huang et. al., *Precursors, aftershocks, criticality and self-organized criticality*, cond-mat/9612065, Dez. 1996
- [Joh 98] A. Johansen, D. Sornette, *Stock market crashes are outliers*, The European Physical Journal B, Springer Verlag, 1998
- [Joh 98b] A. Johansen, O. Ledoit, D. Sornette, *Crashes as critical points, preprint 1998* (<http://xxx.lanl.gov/abs/cond-mat/9810071>)
- [Juk 94] T. Jukarainen, E. Kärpänoja, P. Vuorimaa, *Gas Recognition Using Fuzzy Self-Organizing Map*, 4. Dortmunder Fuzzy-Tage, 6.-8. Juni 1994, Tagungsband
- [Kel 96] J. Kelleter, *Künstliche Nase für gasförmige Emissionen aus unvollständigen Verbrennungen, Aufbau und Erprobung eines Multi-Gassensor-Systems*, Dissertation, Institut für Angewandte Physik der Universität Gießen, 1996
- [Kie 99] J. Kiesewetter, *Langzeituntersuchungen an Wärmetönungssensoren*, Vortragsunterlagen zum Nosebox-Statusseminar am 14.12.1999 in Oelde
- [Kir 83] S. Kirkpatrick et. al., *Optimization by Simulated Annealing: Theory and Applications*, Dordrecht: Reidel 1987
- [Koh 89] D. Kohl, *Surface Processes in the Detection of Reducing Gases with  $\text{SnO}_2$ -based Devices*, Sensors and Actuators, 18 (1989), 71-113, 1989
- [Koh 89] D. Kohl, *Surface Processes in the Detection of Reducing Gases with  $\text{SnO}_2$ -based Devices*, Sensors and Actuators, 18 (1989), 71-113
- [Koh 90] D. Kohl, *The role of noble metals in the chemistry of solid-state gas-sensors*, Sensors and Actuators B, 1 (1990), 158-165, 1990
- [Koh 93] D. Kohl, *Elements and Arrays for Gas Mixtures*, Physica Scripta. Vol T49, 544-548, 1993.

- [Koh 95] T. Kohonen, *Self-Organizing Maps*, Springer Series in Information Science, Springer-Verlag, 1995
- [Koh+95] T.Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, K. Torkkola, *LVQ-PAK, The Learning Vector Quantization Program Package*, Version 3.0, LVQ Programming Team of the Helsinki University of Technology, Lab. of Comp. and Inform. Science, Rakentajanaukio 2 C, SF-02150 Espoo, Finland
- [Koh+95b] T.Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, *SOM-PAK, The Self-Organizing Map Program Package*, Version 3.0, SOM Programming Team of the Helsinki University of Technology, Lab. of Comp. and Inform. Science, Rakentajanaukio 2 C, SF-02150 Espoo, Finland
- [Koh+96] D. Kohl, C. Hummel, V. Uwira, J. Häusler, T. Rühl, *Stofferkennung mit Klassifikatoren und Gassensorarray*, SENSOR report 4-1996, S. 30f.
- [Kön 95] A. König, *Neuronale Strukturen zur sichtgestützten Oberflächeninspektion von Objekten in industrieller Umgebung*, Darmstadt, Diss., 1995
- [Kos 92] B. Kosko, *Neural Networks and Fuzzy Systems, A Dynamical Systems Approach to Machine Intelligence*, Prentice Hall, Eaglewood Cliffs, NJ, 1992
- [Kru 98] C. Krummel, *Charakterisierung der Oberflächenreaktionen von  $CH_4$  und  $H_2$  auf  $Ga_2O_3$* , Dissertation, Institut für Angewandte Physik, Uni Gießen, 1998
- [Lüt 93] H. Lüth, *Surfaces and Interfaces of Solids*, Springer, Heidelberg, 2<sup>nd</sup> ed, 1993
- [Mad 89] M.J. Madou, S.R. Morrison, *Chemical Sensing with Solid State Devices*, Academic Press, London, 1989
- [Mar 95] G. Marinell, *Multivariate Verfahren*, 4. Erw. Auflage, Oldenbourg Verlag, 1995
- [Met 53] N. Metropolis et. al., *Equation of State Calculations by Fast Computing Machines*, Journal of Chemical Physics 21 (1953), S. 1087-1092
- [Neu 96] K.U. Neumann, *Fernmessung der Temperatur mit Hilfe eines Paares von Gassensoren*, Diplomarbeit, Institut für Angewandte Physik der Universität Gießen, 1996
- [Nis 97] V. Nissen, *Einführung in Evolutionäre Algorithmen: Optimierung nach dem Vorbild der Evolution*, Viewg, Braunschweig, Wiesbaden, 1997

- [Nrc 92] W. H. Press et al., *Numerical Recipes in C, The Art of Scientific Computing*, Second Edition, Cambridge University Press, Cambridge, 1992
- [Par 62] E. Parzen, *On estimation of a probability density function and mode*, *Annals of Mathematical Statistics*, Vol. 33, S. 1065-1076
- [Roj 93] R. Rojas, *Theorie der neuronalen Netze, Eine systematische Einführung*, Springer Verlag, Berlin Heidelberg New York, 1993
- [Sam 69] J. W. Sammon, *A Nonlinear Mapping for Data Structure Analysis*, *IEEE Transactions on Computers*, Vol. C-18, No. 5, May 1969
- [Sat 94] Satake et al., *Titania NO<sub>x</sub> Sensors for Exhaust Monitoring*, *Sensors and Actuators B*, June 1994
- [Sch 95] A. Schwarz, *Entwurf und Aufbau eines Gassensorsystems mit neuronaler Auswertung zur Messung reduzierender Gase*, Diplomarbeit, Institut für Angewandte Physik der Universität Gießen, 1995
- [Sor 98] D. Sornette, A. Johansen, *A Hierarchical Model of Financial Crashes*, *Physica A* 261, 581-598, 1998
- [Spe 90] D.F. Specht, *Probabilistic Neural Networks*, *Neural Networks*, Vol. 3, 109-118, 1990
- [Spe 91] D.F. Specht, *A General Regression Neural Network*, *IEEE Transactions on Neural Networks*, Vol. 2, No. 6, S. 568 - 576, 1991
- [Sta 85] D. Stauffer: *Introduction to Percolation Theory* (Taylor and Francis, London 1985)
- [Sta 97] D. Stauffer, D. Sornette, *Log-periodic oscillations for biased diffusion on random lattice*, *Physica A* 252 (1998) 271-277
- [Sze 81] S.M. Sze, *Physics of semiconductor Devices*, Wiley, New York, 1981
- [Ulr 99] M. Ulrich, *Leitfähigkeitsmodell gassensitiver polykristalliner Metalloxidschichten unter Berücksichtigung von Perkolationseffekten*, Diplomarbeit, Institut für Angewandte Physik, Uni gießen, 1999
- [Ult 92] Ultsch, A., *Self-organizing Neural Networks for Visualisation and Classification*, *Proc. Conf. Soc. for Information and Classification*, Dortmund, April 1992

- [Ult 95] Ultsch A., Korus D., *Automatic Acquisition of Symbolic Knowledge from Sub-symbolic Neural Networks*, Proc. EUFIT '95, Aachen, August 1995
- [Ust 95] Umweltsensortechnik GmbH, *Geraberger Gassensor GGS*, Datenblatt GGS, 1995
- [Uwi 96] V. Uwira, persönliche Mitteilung
- [Vai 92] S. Vaihinger, *Mehrkomponentenanalyse durch zeitabhängige Signale chemischer Sensoren*, Tübingen, Diss., 1992
- [Van 98] N. Vandevallé et. al., *The crash of October 1987 seen as a phase transition: amplitude and universality*, Physica A 255 (1998) 201-210
- [Vog 95] H. Vogel, *Gerthsen Physik*, 18. Auflage, Springer Verlag, Berlin 1995
- [Wei 92] U. Weimar, *Oxidgassensoren und Multikomponentenanalyse*, Tübingen, Diss., 1992
- [Wfd 90] *Wie funktioniert das? Die Ernährung*, hrsg. u. bearb. von E. Menden in Zusammenarbeit mit Meyers Lexikonred., Meyers Lexikonverlag, Mannheim, Wien, Zürich
- [Zel 94] A. Zell, *Simulation Neuronaler Netze*, Addison-Wesley GmbH, 1994
- [Zem 88] J. Zemel, *Theoretical Description of Gas-Film Interaction on SnO<sub>x</sub>*, Thin Solid Films, 163 (1988), 189-202
- [Zup 93] J. Zupan, J. Gasteiger, *Neural Networks for Chemists*, VCH Verlagsgesellschaft mbH, 1993

## 7. Anhang

### 7.1 VISOM (Visualisierungs-Tool (incl. SOM) für Windows)

#### Inhalt:

<b>1 Funktionsumfang</b> .....	<b>I</b>
<b>2 Installation</b> .....	<b>II</b>
<b>3 Dateiformate</b> .....	<b>III</b>
<b>4 Erläuterung der einzelnen Menüpunkte</b> .....	<b>V</b>
<b>4.I. Menü Datei</b> .....	<b>V</b>
<b>4.II Menü Bearbeiten</b> .....	<b>VII</b>
<b>4.III Menü Darstellung</b> .....	<b>VII</b>
<b>4.IV Menü SOM</b> .....	<b>VIII</b>
<b>4.V Menü Extras</b> .....	<b>XIII</b>

#### 1 Funktionsumfang

##### a) Datensichtung

Das Programm bietet die Möglichkeit, charakteristische Merkmale in hochdimensionalen Datensätzen bereits ohne aufwendige Datenpräparation zu erkennen. Das unterstützte Datenformat ist kompatibel zu dem von *SOM\_PAK* [1]. Es können  $n(n-1)/2$  Projektionen des von den  $n$ -dimensionalen Merkmalsvektoren aufgespannten Vektorraums in die Ebene dargestellt werden. Ein Wechsel der gerade gegeneinander aufgetragenen Achsen erfolgt mit Hilfe der Maus. Beim Neuladen einer Datei werden automatisch die Daten in der ersten Spalte als x-Achsenwerte verwendet und die in der zweiten als y-Achsenwerte. Sollte es sich um eindimensionale Daten handeln, so werden diese als Ordinate aufgetragen und der (obligatorische) Index in der letzten Spalte als Abszisse verwendet. Dieser Index kann eine beliebige Zahl sein. Er geht nicht in das Netztraining ein. Bei der Einteilung der einzelnen Datenpunkte in verschiedene Kategorien hat der Anwender zwei Möglichkeiten:

a) Verwendung eines fortlaufenden Index in der letzten Spalte und Unterscheidung nach Intervallzugehörigkeit.

b) Verwendung eines festen Zahlwertes für jede Klasse.

Speziell für den Fall von sich überlappenden Punktwolken unterschiedlicher Klassenzugehörigkeit hat der Benutzer die Möglichkeit, per Maus Ausschnitte zu vergrößern. Das Programm speichert dabei bis zu 50 Zoom-Stufen, erlaubt also im Fall eines versehentlichen Hineinzoomens in einen weniger interessanten Bereich, stufenweise heraus und erneut hineinzuzoomen. Bei einem Wechsel der Achsen wird der Fensterausschnitt automatisch wieder angepaßt. Ferner hat man die Option, die zeitliche Entwicklung der Datenpunkte in „Zeitlupe“ darzustellen.

Darüber hinaus können sowohl SOM als auch Codebook-Vektoren (LVQ, PNN, GRNN) im Eingaberaum visualisiert werden.

## b) SOM-Training und -Visualisierung

### Training

Die Eingabe der Trainingsparameter erfolgt über eine Eingabemaske (s. 4.IV.2). Das Programm verwendet zum Netztraining wahlweise *VFIND.EXE* aus *SOM\_PAK* oder eine an Windows angepaßte Version namens *WINSOM.EXE*. Beide sind in der Konfigurationsdatei *TRAIN.INF* eingetragen. Der aktuelle Eintrag in der ersten Zeile entscheidet darüber, welche Version verwendet wird. Beide Versionen werden als Icon ausgeführt. Durch einen Doppelklick auf das Icon kann das Fenster auf normale Größe gebracht werden. In diesem Fall kann man den zeitlichen Fortschritt des Trainings mitverfolgen. Ein Weiterarbeiten im Hauptfenster ist in beiden Fällen möglich.

### Visualisierung und Kalibrierung von Netzen

#### Vektor-Lagekarten:

Ein Datensatz mit passender Vektordimension wird auf ein trainiertes Netz abgebildet. Jeder Datenpunkt aktiviert ein Neuron, welches anhand der Klassenzugehörigkeit des aktivierenden Datenpunktes eingefärbt wird. Für den Fall, daß Merkmalsvektoren unterschiedlicher Klassenzugehörigkeit dasselbe Neuron aktivieren, gibt es zwei Möglichkeiten:

Das Neuron wird mit der Farbe des zuletzt aktivierenden Merkmalsvektors neu gezeichnet oder als Konflikt markiert. Der Benutzer kann entscheiden, welche Variante zur Anwendung kommt. In der Standardeinstellung werden Konflikte als solche gekennzeichnet.

Gleichzeitig mit der grafischen Darstellung auf dem Bildschirm erfolgt eine Kalibrierung des Netzes. Dabei wird jedem Neuron die Klasse des aktivierenden Datenpunktes zugeordnet.

#### Komponenten-Lagekarten:

Die Codebook-Vektoren (Neuronen) eines Netzes können komponentenweise inspiziert werden. Zur optischen Orientierung wird ein Farbverlauf verwendet (von schwarz für die kleinste bis hin zu rot für die größte Komponente einer Ebene).

#### U-Matrix Darstellung:

Die Abstände zwischen zwei Codebook-Vektoren werden als Graustufen dargestellt. (schwarz = maximaler Abstand).

#### Darstellung des Netzes im Eingabraum

Zur Beurteilung der Entfaltung der Karte (topologische Defekte!) können die Netze in die jeweilige Vektor-Komponentendarstellung im Eingaberaum projiziert werden

#### Klassifizierung unbekannter Datensätze

Neue Meßdaten können mit einem trainierten und bereits kalibrierten Netz klassifiziert werden. Ein aktiviertes Neuron wird dabei mit der Farbe seiner Klassenzugehörigkeit und dem (möglicherweise fortlaufenden) Index des aktivierenden Datenpunktes dargestellt. Aktiviert ein zu klassifizierender Vektor ein bei der Kalibrierung nicht getroffenes, sg. leeres Neuron, so wird dieser Sachverhalt durch ein Fragezeichen grafisch dargestellt.

## 2 Installation

Installationsdiskette in Laufwerk A: oder B: einlegen. Im *Programm-Manager* den Menüpunkt Datei | Ausführen... wählen und unter Befehlszeile: das Kommando *a:\setup* (bzw. *b:\setup*) eingeben. Das Installationsprogramm wird gestartet und fragt nach dem Verzeichnis, in dem das

Programm installiert werden soll (default: C:\VISOM). Diese Vorgabe kann akzeptiert oder geändert werden. Der Rest der Installation verläuft automatisch.

### 3 Dateiformate

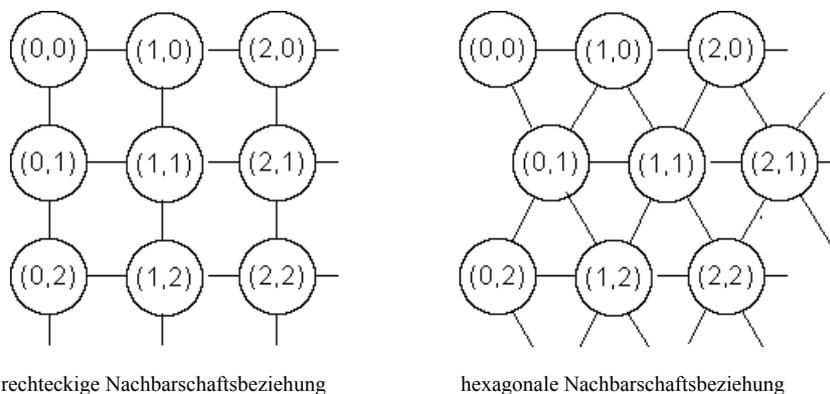
Das verwendete Dateiformat ist so gewählt, daß es zu dem von Kohonens SOM\_PAK kompatibel ist, d.h. die erste Zeile eines Datensatzes muß folgende Informationen enthalten ([1], S.11):

**Vektordimension [Topologie] [x-Dimension] [y-Dimension] [Nachbarschaftsfunktion]**

Erläuterung der einzelnen Einträge:

**Vektordimension** (Natürliche Zahl) - entspricht der um eins reduzierten (näheres dazu s.u.) Gesamtspaltenzahl der zugehörigen Datei

**Topologie** (String: *hexa* oder *rect*) - Zwei Netztopologien werden unterstützt: rechteckige und hexagonale Gitterstruktur, wobei die folgenden Nachbarschaftsbeziehungen bestehen ([1], S.13):



rechteckige Nachbarschaftsbeziehung

hexagonale Nachbarschaftsbeziehung

**x-Dimension** (Natürliche Zahl) - Ausdehnung des Netzes in horizontaler Richtung

**y-Dimension** (Natürliche Zahl) - Ausdehnung des Netzes in vertikaler Richtung

**Nachbarschaftsfunktion** (String: *bubble* oder *gaussian*) - Auswahl zwischen zylindrischer und Gaußscher Nachbarschaftsfunktion.

Bei den Rohdatensätzen ist nur eine Angabe der Vektordimension obligatorisch, alle anderen sind optional. Da die *Netz*-Dateien vom Programm erzeugt werden, muß der Anwender lediglich bei neuen *Trainings*- und *Test*-Dateien die Vektordimension in der ersten Zeile eines neuen Datensatzes eintragen.

Die letzte Spalte eines Rohdatensatzes ist für den sg. Klassifizierungsindex [Typ Integer] reserviert. Diese zusätzliche Spalte muß für das einwandfreie Funktionieren aller Programmkomponenten vorhanden sein. Der Eintrag **Vektordimension** entspricht also der um eins verminderten Gesamtspaltenzahl eines Datensatzes.

Der Klassifizierungsindex dient der späteren Unterscheidung verschiedener Klassen. Es kann sich dabei um einen fortlaufenden Index handeln (Zeitzähler) oder aber um eine einheitliche Zahl für alle Angehörigen einer Klasse. Daß die letzte Spalte nicht zur Vektordimension addiert wird, hat die erwünschte Konsequenz, daß die Programme aus den Packages LVQ\_PAK und SOM\_PAK die letzte Spalte als Label betrachten, so daß diese Zahlenwerte nicht ungewollt zum Training mitverwendet werden.

Das Programm VISOM stellt Datenpunkte auf dem Bildschirm entweder als Kreise mit dem Index modulo 10 dar oder

als kleine Rechtecke mit dem vollständigen Index. Der Index kann zur optischen Unterscheidung verschiedener Klassenzugehörigkeiten verwendet werden: Bei fortlaufendem Index lassen sich entsprechende Intervalle für eine Klassenzugehörigkeit angeben.

### Beispiel:

Es liegt ein Datensatz 3-dimensionaler Vektoren vor, bei dem die Datenvektoren im Laufe der Zeit von einem Zustand A in einen Zustand B übergegangen sind. Die einzelnen Vektoren sind mit einem fortlaufenden Index gekennzeichnet. Der entsprechende Rohdatensatz sähe dann wie folgt aus:

Datei <DATEINAME>.DAT:

```
3
1.3   2.4   1.7   180
1.9   2.1   2.0   181
...   ...   ...   ...
2.9   2.8   1.5   200
3.0   2.7   1.4   201
...   ...   ...   ...
etc.
```

Hier würde eine Klassifizierungsdatei <DATEINAME>.CLS mit folgenden Einträgen angelegt werden, wobei der hier mit <DATEINAME> bezeichnete Dateiname vor .CLS mit dem obigen (vor .DAT) identisch sein muß. (Erläuterungen in Klammern bitte nicht mit eingeben!)

DATEI <DATEINAME>.CLS:

```
2      (Anzahl der Intervalle)
180    (Beginn Intervall 1)
200    (Ende Intervall 1)
1      (beliebiger, eindeutiger Index für Intervall 1)
201    (Beginn Intervall 2)
300    (Ende Intervall 2)
2      (beliebiger, eindeutiger Index für Intervall 2)
...
etc.
```

Die Legende benötigt folgende Information:

<DATEINAME>.LBL (auch hier muß der Dateiname vor .LBL mit dem von oben übereinstimmen)

```
1      (Index des ersten Intervalls)
Zustand A  (Name des Zustandes)
2      (Index des zweiten Intervalls)
Zustand B  (Name des zweiten Intervalls)
```

Zur individuellen Gestaltung der Achsenbezeichnung kann eine Datei <DATEINAME>.AXS angelegt werden, die die Bezeichnungen für die einzelnen Spalten enthält:

Datei <DATEINAME>.AXS

```
x-Achse
y-Achse
z-Achse
```

Folgende Dateien werden vom Programm automatisch generiert:

```
*.SOM      Netz-Datei
```

\*.MAP wird immer dann erzeugt, wenn ein Test-Datensatz auf ein Netz abgebildet wird (entspricht der Ausgabe von *visual* aus *SOM\_PAK*)

\*.UDM enthält die Unified-Distance-Matrix eines Netzes. Der größte vorkommende Abstand zweier Codebook-Vektoren wird auf 100 skaliert. Die Datei kann z.B. zur 3-dimensionalen grafischen Aufbereitung in EXCEL eingelesen werden.

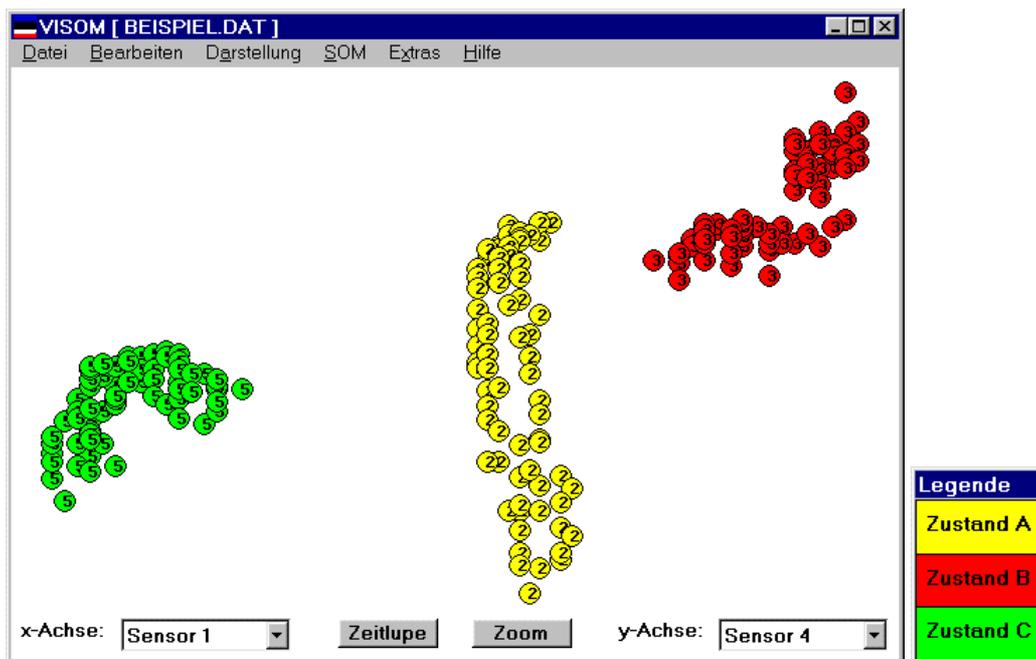
\*.NKI speichert die Klasseninformation der einzelnen Neuronen aus \*.SOM

## 4 Erläuterung der einzelnen Menüpunkte

### 4.I. Menü Datei

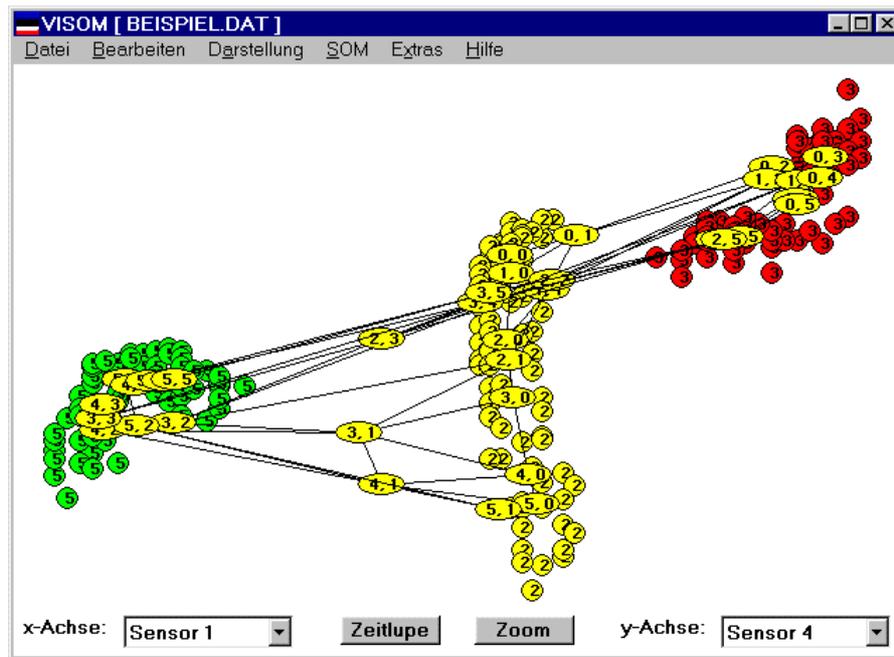
#### 4.I.1. Datensatz laden...

Liest einen Datensatz vom oben beschriebenen Format ein und stellt ihn auf dem Bildschirm dar. Bei eindimensionalen Daten wird diese Komponente nach oben und der Klassifizierungsindex nach rechts aufgetragen, ansonsten die erste Komponente auf der x-Achse und die zweite auf der y-Achse. Beliebiges Umschalten durch Wahl einer anderen Spalte (im unteren Fensterbereich) mit der Maus:



#### 4.I.2. SOM-Datei laden...

Öffnet eine bereits vorhandene SOM-Datei, die ebenfalls im Eingaberaum angezeigt wird:

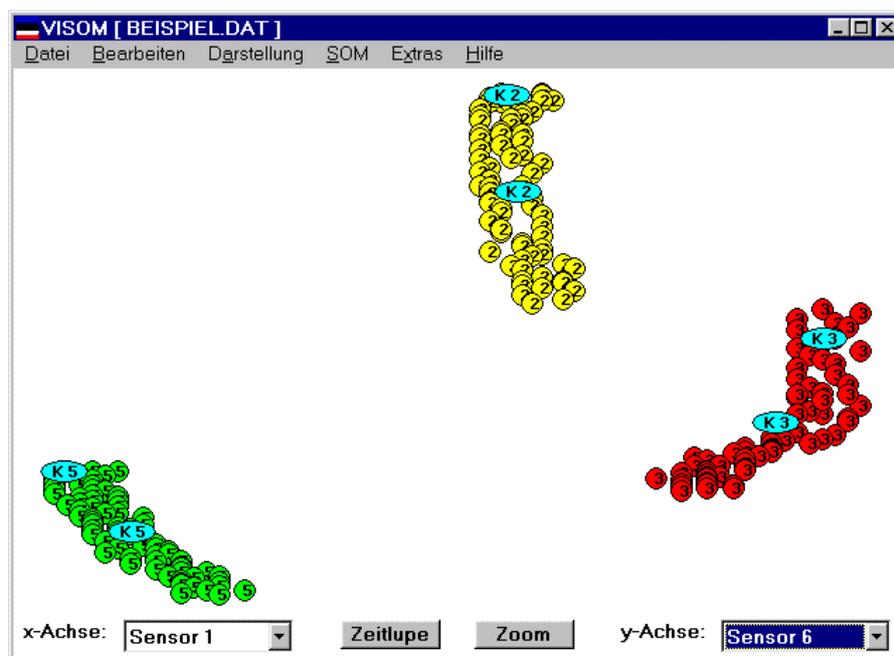


#### 4.I.3. SOM-Datei schließen...

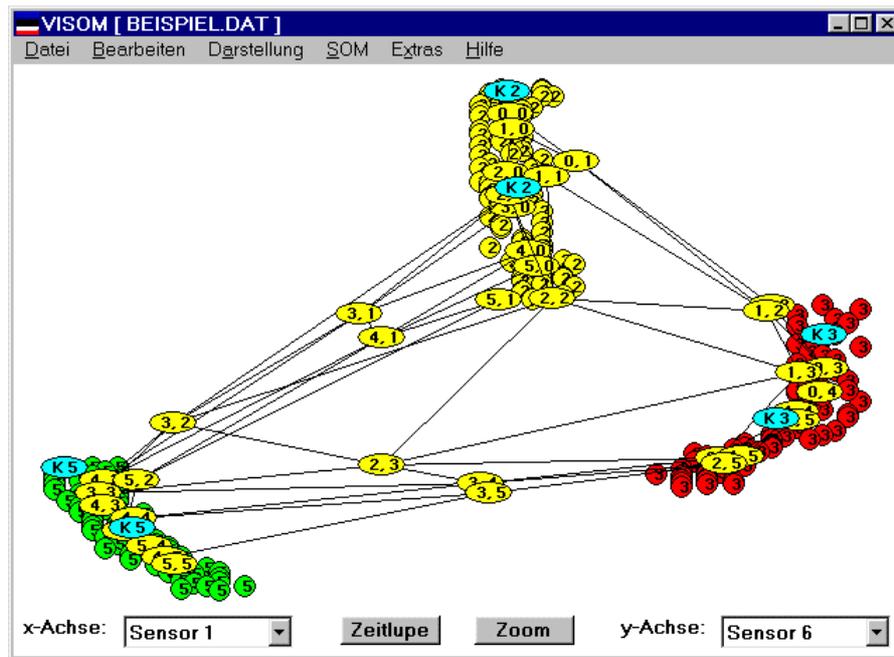
Dieser Menüpunkt ist nur bei geöffneter SOM-Datei aktiv und entfernt die Kohonen Karte wieder aus der Darstellung

#### 4.I.4. LVQ-Datei laden...

Zeigt die Codebook-Vektoren eines LVQ-Klassifikators (hellblaue Ovale mit Klassenzugehörigkeitsanzeige)



Die beiden letztgenannten Menüpunkte können selbstverständlich auch kombiniert werden:



#### 4.I.4. LVQ-Datei schließen...

Blendet die Darstellung der Codebook-Vektoren wieder aus

#### 4.I.5. Benden

Beendet das Programm.

### 4.II. Menü Bearbeiten

#### 4.II.1 Kopieren

Kopiert die aktuelle Grafik in die Zwischenablage. Dort steht sie zur Weiterverarbeitung zur Verfügung.

### 4.III. Menü Darstellung

#### 4.III.1 Ausschnitt

##### 4.III.1.1 Anzeigen

Zeigt die Koordinaten des aktuellen Fensterausschnitts an.

##### 4.III.1.2 Ändern

Öffnet eine Dialogbox zum Ändern des aktuellen Ausschnitts. Die gleiche Dialogbox erscheint, wenn man nach Anklicken des Zoom-Buttons im unteren Teil des Hauptfensters einen Fensterbereich mit der Maus markiert hat. In diesem Fall entspricht die Vorgabe dem markierten Bereich.

Ausschnitt	
xMin:	0.9328
xMax:	0.9665
yMin:	0.8878
yMax:	1.0492
<input type="button" value="Abbruch"/> <input type="button" value="OK"/>	

### 4.III.2 Form

#### 4.III.2.1 Kreis

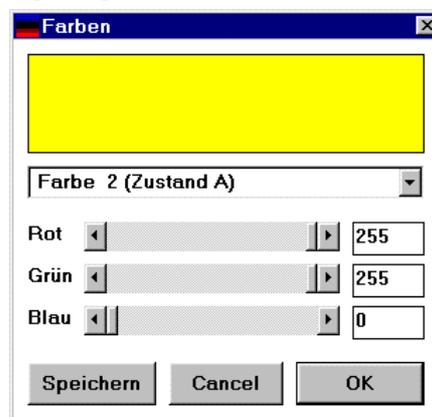
Darstellung der Datenpunkte als kleine Kreise (Voreinstellung)

#### 4.III.2.2 Rechteck

Darstellung der Datenpunkte als kleine Rechtecke (ermöglicht direktes Ablesen auch größerer Werte aus der Index-Spalte)

### 4.III.3 Farben

Öffnet einen Farbeditierdialog für bis zu 100 verwendete Farben zur Klassenunterscheidung. Wenn eine bestimmte Farbe zur Kennzeichnung einer bestimmten Klasse verwendet wurde, so wird der Klassenname im Dialogfeld mit angezeigt



### 4.III.4 Zeitlupe

Drei Zeitlupengeschwindigkeiten sind einstellbar; der Datensatz wird neu gezeichnet. Die hier eingestellte Geschwindigkeit ist auch dann wirksam, wenn im Hauptfenster der Button „Zeitlupe“ angeklickt wird.

4.III.4.1 Langsam (langsamste Zeitlupengeschwindigkeit.)

4.III.4.2 Mittel (mittlere Zeitlupengeschwindigkeit)

4.III.4.3 Schnell (schnellste Zeitlupengeschwindigkeit, default)

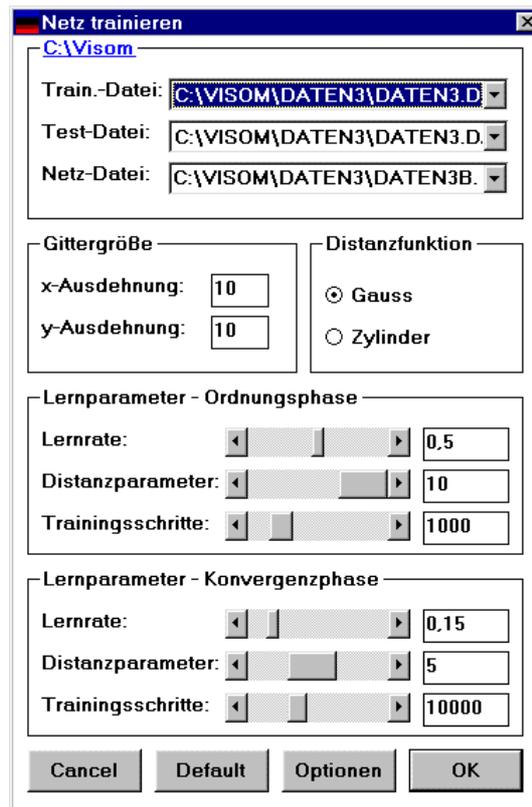
## 4.IV. Menü SOM

### 4.IV.1 Demo

Eingabevektoren dieses Demoprogramms sind die Punkte einer Dreiecksfläche. Eine eindimensionale Kohonen-Kette mit Dreiecks-Nachbarschaftsfunktion wird mit diesen Punkten trainiert; der Lernfortschritt kann auf dem Bildschirm verfolgt werden.

### 4.IV.2 Netz trainieren...

Die Eingabemaske für das Netztraining erscheint:



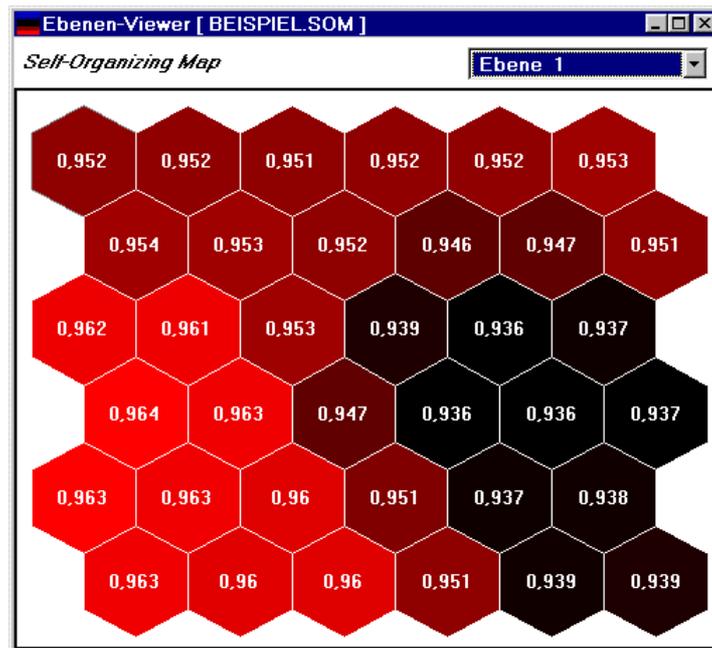
Mit Hilfe des Optionen-Buttons können zwei weitere Parameter eingestellt werden:



#### 4.IV.3 Netz anschauen (Verzweigung zu den Unterpunkten):

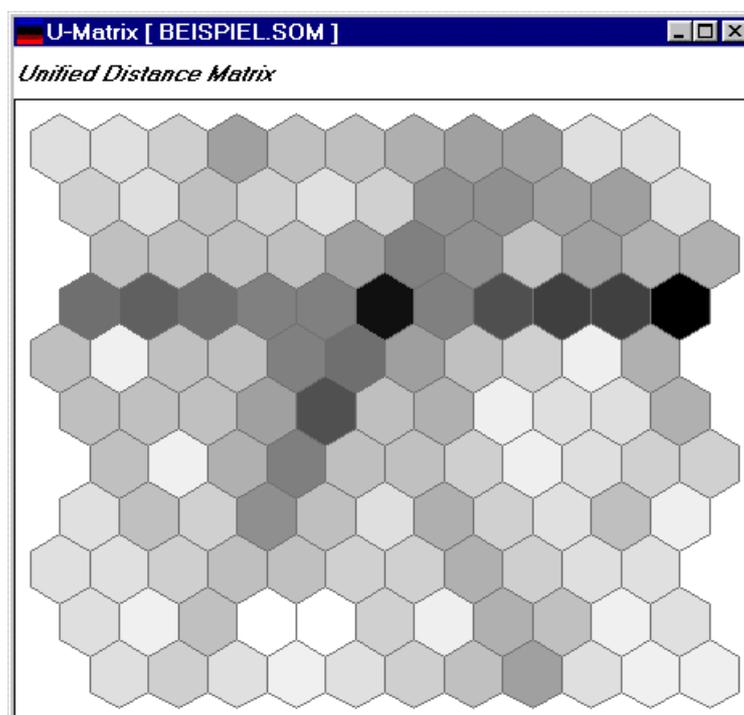
##### 4.IV.3.1 Ebenen (Darstellung der Komponentenkarten eines Netzes)

Eine Dialogbox zur Auswahl eines Netzes (<DATEINAME>.SOM) erscheint. Nachdem das gewünschte Netz ausgewählt wurde, werden die Komponenten der einzelnen Ebenen angezeigt (Start mit Ebene 1, weitere Ebenen via Combo-Box auswählbar). Hinweis: Falls das eingestellte Zahlenformat für eine Ebene nicht angemessen sein sollte, so kann es durch Doppelklick auf die Zeichenfläche korrigiert werden (eine Dialogbox zur Auswahl eines anderen Zahlenformats erscheint):



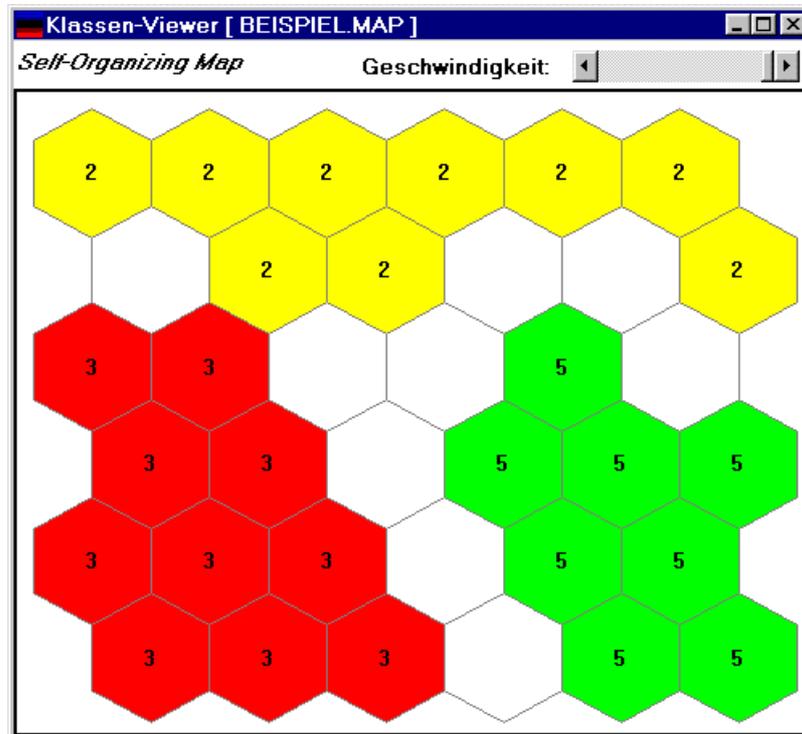
#### 4.IV.3.2 U-Matrix

Eine Dialogbox zur Auswahl eines Netzes erscheint. Die U-Matrix des Netzes wird berechnet und als Graustufigen grafik auf dem Bildschirm ausgegeben. (Falls eine dreidimensionale Darstellung mit „Mauern“ gewünscht werden sollte: Die entsprechenden Daten sind unter <DATEINAME>.UDM abgespeichert (maximaler Abstand := 100). Diese Datei kann problemlos z.B. in EXCEL eingelesen und die Matrix als 3D-Grafik dargestellt werden:



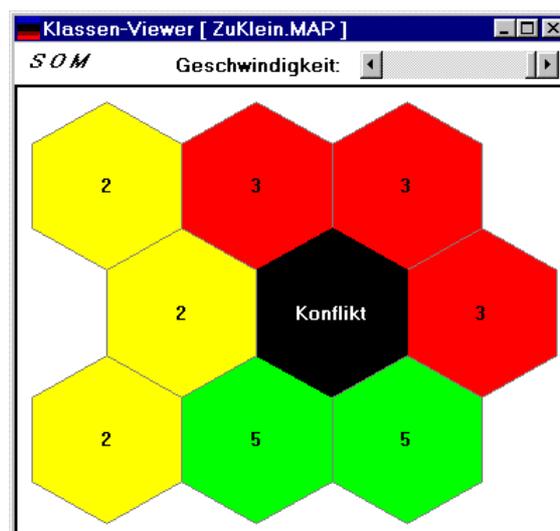
#### 4.IV.3.3 Klassen/Kalibrieren (Visualisierung der Klassenverteilung und Kalibrierung des Netzes)

Zwei Dateiauswahlboxen erscheinen hintereinander (die erste für den Datensatz, die zweite für das Netz). Das Netz wird anhand der Daten kalibriert und dargestellt:



#### 4.IV.3.4 Konflikte anzeigen

Ist dieser Menüpunkt aktiviert (Voreinstellung), so wird jedesmal dann, wenn ein Neuron von Vektoren unterschiedlicher Klassenzugehörigkeit erregt wird, ein Konflikt angezeigt. Anderenfalls wird das Neuron mit der jeweils aktuellen Klassen-Farbe neu gezeichnet:



### 4.IV.3.5 Einzeldarstellung

Ist dieser Menüpunkt aktiviert (default), so wird die Aktivierung der einzelnen Neuronen unter **4.IV.3.3** datenpunktweise dargestellt. Dieses Feature ist dann interessant, wenn die Datenpunkte einer kontinuierlichen Zeitreihe entstammen und man den zeitlichen Verlauf der angeregten Gebiete in der Karte mitverfolgen will. Wegen des hohen Zeitaufwandes bei langen Test-Dateien kann diese Option ausgeschaltet werden. Die Darstellung erfolgt dann neuroneweise.

### 4.IV.4. Daten klassifizieren

#### 4.IV.4.1. Auf SOM abbilden...

Bisher unklassifizierte Rohdaten (letzte Spalte beliebig) werden von einem mit **4.IV.3.3** kalibrierten Netz klassifiziert. Voraussetzung ist, daß sowohl eine Labeldatei (\*.LBL) als auch eine Klassendatei (\*.CLS) für das Netz (\*.SOM) existieren. Beim Ausführen von **4.IV.3.3** wurde automatisch eine Datei mit dem Namen <NETZNAME>.NKI angelegt, die jedem Neuron eine Klasse zuweist und die Namen der relevanten \*.LBL / \*.CLS-Dateien enthält. Aktiviert ein Vektor des zu untersuchenden Datensatzes ein als "Konflikt" gekennzeichnetes Neuron, so wird dies angezeigt. Bei Aktivierung eines "leeren" Neurons wird ein Fragezeichen an der entsprechenden Stelle sichtbar.

#### 4.IV.4.2. SOM (unbekannte Daten)

Zeigt an, wieviel Prozent der Daten welcher Klasse zugeordnet wurden, den Anteil der Daten, der auf nicht gelabelte Neuronen („leere Treffer“) entfiel und den Anteil, der auf als Konflikt markierte Neuronen entfiel (s. auch **4.IV.4.3.**)

#### 4.IV.4.3. SOM (bekannte Daten)

Zeigt an, wieviel Prozent der Daten den welcher Klasse zugeordnet wurden, den Anteil der Daten, der auf nicht gelabelte Neuronen („leere Treffer“) entfiel, den Anteil, der auf als Konflikt markierte Neuronen entfiel, sowie den prozentualen Anteil der Fehlklassifikationen:

Kohonen-Klassifikator	
<b>Ergebnis der Klassifikation:</b>	
<b>Zustand A</b>	<b>33.2 %</b>
<b>Zustand B</b>	<b>33.2 %</b>
<b>Zustand C</b>	<b>33.6 %</b>
<b>Leere Treffer</b>	<b>0.0 %</b>
<b>Falsch klassifiziert</b>	<b>0.0 %</b>
<b>Konflikte</b>	<b>0.0 %</b>
Schließen	

**4.IV.5 Quantisierungsfehler**

Anzeige der Datei ERR.INF, die den Quantisierungsfehler des zuletzt trainierten Netzes enthält

**4.V Extras****4.V.1 A<sub>ch</sub>senbeschriftung**

Lädt \*.AXS (falls vorhanden) zum Editieren von Hand.

**4.V.2 Klassen-Einteilung**

Lädt \*.CLS (falls vorhanden) zum Editieren von Hand.

**4.V.3 Klassen-L<sub>a</sub>bel**

Lädt \*.LBL (falls vorhanden) zum Editieren von Hand

**4.V.4 Änderungen ausführen!**

Zeichnet den aktuellen Ausschnitt mit den geänderten Parametern aus **4.V.1 - 4.V.3** neu.

**4.VI F<sub>e</sub>nster** (wird erst sichtbar wenn mindestens eines der Fenster aus **4.IV.3** bzw. **4.IV.4** vorhanden ist)

Zeigt die neben dem Hauptfenster noch vorhandenen weiteren Fenster an und holt diese gegebenenfalls in der Vordergrund.

## Danksagung

Mein besonderer Dank gilt Herrn Prof. Dr. Claus Dieter Kohl für die Betreuung dieser Arbeit und für die zahlreichen Diskussionen und Anregungen, die wesentlich zum Gelingen derselben beigetragen haben. Darüberhinaus möchte ich mich für seine Förderung und Unterstützung bei den verschiedensten Projekten bedanken, die meinen Erfahrungsschatz weit über den akademischen Rahmen hinaus bereichert haben.

Ganz besonders bedanken möchte ich mich auch bei Dr. Günter Bläser für die hervorragende Zusammenarbeit bei zahllosen Projekten, Vorträgen und Veröffentlichungen.

Ingolf Teetz möchte ich an dieser Stelle für zahlreiche Diskussionen und die Teilhabe an seinem ans legendäre grenzenden IT-Kow-How danken.

Dr. Carsten Diehl danke ich für die unzähligen Gespräche und Anregungen, ebenso Dr. Axel Schwarz, Dr. Volker Uwira, Dr. Jörg Kelleter und all den anderen ehemaligen *IAP-lern*.

Stellvertretend für die aktuelle Belegschaft des IAP möchte ich Andreas Eberheim, Sven Baumann, Markus Lämmer und Dr. Frank Becker für die zahllosen bereichernden Diskussionen und die gute Zusammenarbeit danken.

Alexandra Gabriel danke ich für die immer überaus freundliche Unterstützung bei allen Verwaltungsangelegenheiten.

Herrn Dr. Marc v. Kreutzbruck danke ich für die zahlreichen Gespräche, die wir seit Beginn unseres Studiums geführt haben.

Herrn Dr. Holger Wysk und Herrn Dr. Torsten Clajus von Merck sowie Prof. Dr. Andreas Schütze und Andreas Gramm vom Lehrstuhl für Meßtechnik der Universität des Saarlandes in Saarbrücken danke ich für die konstruktive Zusammenarbeit beim Lösungsmittelprojekt.

Herrn Dr. André Freiling und Herrn Konstantin Kallergis von DaimlerChrysler Aerospace Airbus / EADS sowie Herrn Ulrich Wollenweber und Herrn Dieter Wittenbrink von AOA Gauting danke ich für die Zusammenarbeit beim FIREDETEX-Projekt und anderen Luftfahrtprojekten.

Für die gute Zusammenarbeit beim Nosebox-Projekt möchte ich mich bei Herrn Dr. U. Sillmen von Miele, Herrn Dr. Jörg Kiesewetter von der Fa. DMT, Herrn Dr. Wenker von Winter, Herrn Prof. Dr. W. Mokwa und Stefan Bender von der RWTH Aachen, Herrn Peter Oechtering von der SICAN bedanken sowie Herrn Rump von ETR, Dr. O. Kiesewetter von UST und Herrn Klaus Frers, Vorstandsvorsitzender der Paragon AG, für die Vermittlung unternehmerischen Denkens.

Meinen Eltern möchte ich für die stete Unterstützung in all den Jahren danken.

Meiner Lebensgefährtin Julia möchte ich für die unendliche Geduld und die Unterstützung bei meiner Arbeit danken.