

**Prognose der Eigenschaften stochastischer Prozesse  
mittels Neuronaler Netze mit spezifischen Anwendungen  
in der Kommunikationstechnik**

Von der Fakultät Elektrotechnik der Universität Stuttgart  
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)  
genehmigte Abhandlung

vorgelegt von  
**Markus Eberspächer**  
geb. in Aarau

Hauptberichter: Prof. Dr.-Ing. Dr. h. c. mult. P. J. Kühn  
Mitberichter: Prof. Dr.-Ing. Dr. h. c. R. Lauber

Tag der mündlichen Prüfung: 11.2.1999

Institut für Nachrichtenvermittlung und Datenverarbeitung  
der Universität Stuttgart  
1999

## Summary

For complex systems often the question arises how the system states or measurable values evolve in time, especially in the future. This applies not only to technical systems but also to economic or social systems. The knowledge of future values or even tendencies can be used in several ways, e.g., for planning, design or control.

Traditionally, future system values or states are determined by prediction methods. Different approaches exist depending on the characteristics of the system that can be, e.g., deterministic or stochastic and the observable values which may be continuous, discrete, etc. Prediction is done by taking into account some knowledge about the past behavior of the system and knowledge or assumptions concerning the internal structure and relations of the system.

Most real complex systems don't behave completely deterministic but have significant random or stochastic parts which may be arbitrarily correlated. Therefore, they have to be described by stochastic models that take into account this behavior. In case of stochastic system models stochastic prediction methods are used.

The currently known prediction methods most often predict mean values. This means that they predict a value that corresponds to the expectation of all possible values that theoretically could occur at a particular instant. Certainly, various applications would benefit from even more information about the future behavior, e.g., the variance of the future values as an indicator of confidence for the mean value.

In this report a new prediction method is introduced that extends the prediction of future values towards distributions of them. This allows the calculation of various other statistical parameters like variance or quantiles. These parameters can be directly used, e.g., in control (Chapter 6), or they can help to estimate the achievable accuracy of the prediction when only the mean value is used for further calculations.

The prediction of future system states and values can also be used to simulate the system behavior in order to achieve a deeper insight into the system dynamics. A closely related application is the generation of a stochastic process with the same statistics as a specific (sub)system. Hence, in a simple manner source models can be built for the corresponding stochastic process (Chapter 5).

The prediction algorithm is based on two types of neural networks. Due to their inherent learning ability, the algorithm can adapt automatically to a wide spectrum of stochastic processes, including highly correlated ones. The first neural network learns the mapping of a number of past values of the stochastic process to a set of distributions. All of these distributions are represented by instances of the second type of neural networks that was newly developed for this purpose.

There are many applications of such an algorithm. One area where the application seems to be very promising are communications systems. Therefore, two examples from this field demonstrate the applicability of the new algorithm and the way how it is used. The first example uses the distribution prediction for source modelling (Chapter 5), and the second applies it to efficient bandwidth reservations for the available bit rate (ABR) service in asynchronous transfer mode (ATM) networks (Chapter 6).

This report is divided into two main parts. The first part, consisting of Chapters 1 to 3, introduces the new prediction method and relevant theoretical background. In the second part (Chapters 4 to 6), several ways to apply the new algorithm to different scenarios are shown. The report closes with a summary and some future prospects of the method (Chapter 7).

Chapter 1 motivates the new prediction method. Chapter 2 gives a detailed introduction in neural networks and introduces some relevant types of stochastic processes, vectors of random variables, and time series.

In Chapter 3, the theory and algorithm of the new distribution prediction method are derived. The basic approach as well as specific extensions are described. Some areas of deployment are indicated by simple examples.

Starting the second part of the report, Chapter 4 gives an introduction to principles and methods of modelling and performance evaluation and some methods for analysis and simulation which are relevant for this work. The examples in Chapters 5 and 6 are based on this framework.

Chapter 5 presents a first extensive example that uses the distribution prediction for the modelling of traffic sources. After an introduction to known techniques for source modelling, the distribution prediction is deployed to this application and a basic as well as some extended models are introduced. The evaluation of the approach is done by simulation, the results are assessed by statistical tests.

The second extensive example is given in Chapter 6 by using the distribution prediction for efficient bandwidth reservation for the ABR service in ATM networks. The first part of this chapter introduces ATM, the ABR traffic class, overload control, and local area network (LAN) interconnection using ABR. Secondly, alternative ABR source algorithms, including one which is based on distribution prediction, are compared. This performance evaluation is done by simulations. The chapter finishes with a discussion of the results.

Chapter 7 closes the report with a summary of the most important results and an outlook of possible directions for future enhancements of the newly presented method.

# Inhalt

|   |            |
|---|------------|
| <b>Abkürzungen</b> .....  | <b>vii</b> |
| <b>Formelzeichen</b> .....  | <b>ix</b>  |
| <b>1 Einleitung</b> .....   | <b>1</b>   |
| 1.1 Prognoseverfahren für stochastische Prozesse .....                        | 1          |
| 1.2 Übersicht über die Arbeit .....   | 2          |
| <b>2 Grundlagen</b> .....   | <b>3</b>   |
| 2.1 Neuronale Netze .....   | 3          |
| 2.1.1 Biologisches Vorbild .....  | 3          |
| 2.1.2 Künstliche neuronale Netze .....  | 5          |
| 2.1.3 Klassifizierung und Anwendungen .....                                   | 9          |
| 2.1.4 Bewertung .....   | 10         |
| 2.2 Stochastische Prozesse .....  | 13         |
| 2.2.1 Einführung .....  | 13         |
| 2.2.2 Verfahren zur Bestimmung statistischer Kenngrößen .....                 | 14         |
| 2.2.2.1 Empirischer Mittelwert, Varianz und Korrelation .....                 | 14         |
| 2.2.2.2 Test für unkorrelierte Prozesse .....                                 | 14         |
| 2.2.2.3 Test für Dauer der Autokorrelation .....                              | 15         |
| 2.2.2.4 Anpassungstest für Verteilungen .....                                 | 15         |
| 2.2.3 Prozesse mit Langzeitkorrelation .....                                  | 16         |
| 2.3 Zufallsvektoren .....   | 17         |
| 2.4 Zeitreihen .....  | 18         |
| 2.4.1 Einführung .....  | 18         |
| 2.4.2 Zeitreihenmodelle .....   | 19         |
| 2.4.2.1 AR-Modelle .....  | 19         |
| 2.4.2.2 MA-Modelle .....  | 20         |
| 2.4.2.3 ARMA-Modelle .....  | 20         |
| 2.4.3 Chaotische Zeitreihen .....   | 20         |
| <b>3 Ein Verfahren zur Verteilungsprognose</b> .....                          | <b>22</b>  |
| 3.1 Motivation .....  | 22         |
| 3.2 Herleitung .....  | 23         |
| 3.2.1 Prinzip .....   | 23         |
| 3.2.2 Beziehung zwischen Verteilung und Autokorrelation einer Zeitreihe ..... | 24         |
| 3.2.3 Vorhersage für einen Schritt .....                                      | 26         |
| 3.2.4 Approximation .....   | 26         |
| 3.2.5 Realisierung .....  | 27         |
| 3.3 Erweiterung für die Vorhersage mehrerer Schritte .....                    | 30         |
| 3.4 Erweiterung für mehrere Eingänge .....                                    | 31         |
| 3.5 Beispiele .....   | 32         |
| 3.5.1 Stochastische Prozesse .....  | 32         |
| 3.5.2 Deterministische Prozesse .....   | 36         |

|  |           |
|--|-----------|
| 3.6 Einsatzgebiete . . . . .   | 37        |
| 3.6.1 Einsatz zur Zeitreihenmodellierung. . . . .  | 37        |
| 3.6.2 Einsatz in Regelsystemen . . . . .   | 38        |
| 3.7 Bewertung des Verfahrens . . . . .   | 40        |
| <b>4 Methodische Grundlagen von Modellierung und Leistungsuntersuchung. . . . .</b>                            | <b>41</b> |
| 4.1 Analysemethoden . . . . .  | 41        |
| 4.1.1 Warteschlangentheorie . . . . .  | 41        |
| 4.1.2 Zeitreihenanalyse. . . . .   | 43        |
| 4.1.3 Methoden der Regelungstechnik. . . . .   | 43        |
| 4.1.4 Perturbationstheorie. . . . .  | 44        |
| 4.1.5 Bewertung . . . . .  | 45        |
| 4.2 Simulation . . . . .   | 45        |
| <b>5 Einsatz der Verteilungsprognose zur Modellierung von Verkehrsquellen. . . . .</b>                         | <b>51</b> |
| 5.1 Einführung . . . . .   | 51        |
| 5.2 Bekannte Verfahren . . . . .   | 51        |
| 5.3 Anwendung der Verteilungsprognose. . . . .   | 53        |
| 5.4 Erweiterte Modelle . . . . .   | 55        |
| 5.4.1 Mehrwertige Quellmodelle . . . . .   | 55        |
| 5.4.2 Quellmodelle mit aggregierter Rückkopplung . . . . .   | 55        |
| 5.4.3 Hierarchische Modelle. . . . .   | 56        |
| 5.5 Statistische Tests. . . . .  | 57        |
| 5.5.1 Bewertung der Verteilung . . . . .   | 57        |
| 5.5.2 Bewertung des Korrelogramms. . . . .   | 57        |
| 5.6 Simulationsmodell . . . . .  | 59        |
| 5.7 Beispiele und Leistungsbewertung . . . . .   | 60        |
| 5.7.1 Notation . . . . .   | 60        |
| 5.7.2 ARMA-Prozesse . . . . .  | 61        |
| 5.7.3 MPEG-kodierte Videodaten . . . . .   | 63        |
| 5.7.3.1 Korrelation und Verteilung . . . . .   | 63        |
| 5.7.3.2 LRD-Verhalten. . . . .   | 66        |
| 5.7.4 Deterministische Zahlenfolge . . . . .   | 67        |
| 5.7.5 Ergebnisse . . . . .   | 69        |
| 5.8 Zusammenfassung. . . . .   | 70        |
| <b>6 Einsatz der Verteilungsprognose zur Bandbreitereservierung für den ABR-Dienst in ATM-Netzen . . . . .</b> | <b>72</b> |
| 6.1 Überblick. . . . .   | 72        |
| 6.2 Der Asynchrone Transfermode als Transportmechanismus für breitbandige Kommunikationsnetze . . . . .        | 72        |
| 6.2.1 Das ATM-Prinzip . . . . .  | 73        |
| 6.2.2 ATM-Referenzmodell . . . . .   | 74        |
| 6.2.2.1 Bitübertragungsschicht. . . . .  | 76        |
| 6.2.2.2 ATM-Schicht . . . . .  | 76        |

|   |            |
|---|------------|
| 6.2.2.3 ATM-Anpassungsschicht . . . . .                                   | 76         |
| 6.2.2.4 Höhere Schichten . . . . .  | 78         |
| 6.3 Überlastabwehr und -vermeidung . . . . .                              | 79         |
| 6.4 Die ABR-Verkehrsklasse . . . . .                                      | 80         |
| 6.4.1 Allgemeines . . . . .   | 80         |
| 6.4.2 Das ABR-Protokoll . . . . .   | 81         |
| 6.4.2.1 Quelle . . . . .  | 83         |
| 6.4.2.2 Senke . . . . .   | 85         |
| 6.4.2.3 Vermittlungsknoten . . . . .                                      | 85         |
| 6.4.3 Regelungstechnische Betrachtung . . . . .                           | 86         |
| 6.5 Realisierungsalternativen für ABR-Quellen . . . . .                   | 87         |
| 6.5.1 Reservierung der ausgehandelten Maximalrate . . . . .               | 88         |
| 6.5.2 Reservierung der zuletzt beobachteten Rate . . . . .                | 88         |
| 6.5.3 Anwendung der Verteilungsprognose . . . . .                         | 88         |
| 6.6 LAN-Kopplung über ABR . . . . .                                       | 89         |
| 6.7 Modellierung . . . . .  | 93         |
| 6.7.1 Modell für das ATM-Netz . . . . .                                   | 93         |
| 6.7.1.1 Übersicht . . . . .   | 93         |
| 6.7.1.2 Feste und variable Verzögerung im ATM-Netz . . . . .              | 94         |
| 6.7.1.3 Modifikation von $ER_b$ -Werten im Netz . . . . .                 | 95         |
| 6.7.1.4 Zustandsprozeß zur Nachbildung realer Netzlast . . . . .          | 97         |
| 6.7.2 Modell für einen Netzübergangsknoten . . . . .                      | 99         |
| 6.7.2.1 Übersicht . . . . .   | 99         |
| 6.7.2.2 Zeitverhalten . . . . .   | 101        |
| 6.7.2.3 Reservierungsverfahren . . . . .                                  | 102        |
| 6.7.2.4 Modellparameter . . . . .   | 106        |
| 6.8 Leistungsuntersuchung . . . . .                                       | 107        |
| 6.8.1 Diskussion einiger Modellparameter . . . . .                        | 107        |
| 6.8.2 Untersuchung der unterschiedlichen Reservierungsverfahren . . . . . | 109        |
| 6.9 Bewertung . . . . .   | 111        |
| <b>7 Zusammenfassung und Ausblick . . . . .</b>                           | <b>112</b> |
| <b>8 Literaturverzeichnis . . . . .</b>                                   | <b>113</b> |
| <b>Anhang: Algorithmen zur Verteilungsprognose . . . . .</b>              | <b>121</b> |
| A1 Vektor-Quantisierer . . . . .  | 121        |
| A1.1 Einführung . . . . .   | 121        |
| A1.2 Lernalgorithmus . . . . .  | 122        |
| A1.3 Beispiel . . . . .   | 123        |
| A1.4 Parameterwahl . . . . .  | 123        |
| A1.5 Bewertung . . . . .  | 123        |
| A2 Verteilungs-Approximation . . . . .                                    | 125        |
| A2.1 Lernalgorithmus . . . . .  | 126        |
| A2.2 Beispiel . . . . .   | 130        |

|  |     |
|--|-----|
| A2.3 Parameterwahl .....                   | 131 |
| A2.4 Bewertung.....                        | 132 |
| A3 Exponentiell gewichtete Mittelung ..... | 134 |

## Abkürzungen

|        |  |
|--------|--|
| AAL    | ATM Adaptation Layer   |
| ABR    | Available Bit Rate   |
| ABT    | ATM Block Transfer   |
| ACR    | Allowed Cell Rate  |
| AR     | Autoregressive   |
| ARMA   | Autoregressiv Moving Average                                   |
| ATM    | Asynchronous Transfer Mode                                     |
| B-ISDN | Broadband Integrated Services Digital Network                  |
| CAC    | Connection Admission Control                                   |
| CBR    | Constant Bit Rate  |
| CCITT  | Comité Consultatif International Téléphonique et Télégraphique |
| CCR    | Current Cell Rate  |
| CLP    | Cell Loss Priority   |
| CPU    | Central Processing Unit  |
| CRC    | Cyclic Redundancy Check  |
| CS     | Convergence Sublayer   |
| DBR    | Deterministic Bit Rate   |
| DQDB   | Distributed Queue Dual Bus                                     |
| EFCI   | Explicit Forward Congestion Indication                         |
| EPRCA  | Enhanced Proportional Rate Control Algorithm                   |
| ER     | Explicit Cell Rate   |
| GFC    | Generic Flow Control   |
| GMDP   | Generally Modulated Deterministic Process                      |
| HEC    | Header Error Control   |
| ICR    | Initial Cell Rate  |
| IETF   | Internet Engineering Task Force                                |
| IP     | Internet Protocol  |
| ISDN   | Integrated Services Digital Network                            |
| ISO    | International Standards Organisation                           |
| ITU    | International Telecommunication Union                          |
| ITU-T  | Telecommunication Standardization Sector of ITU                |
| KNN    | Künstliches Neuronales Netz                                    |
| LAN    | Local Area Network   |



|         |   |
|---------|---|
| LANE    | LAN Emulation                             |
| LRD     | Long Range Dependence                     |
| MA      | Moving Average                            |
| MCR     | Minimum Cell Rate                         |
| MMPP    | Markov Modulated Poisson Process          |
| MPEG    | Moving Picture Experts Group              |
| NNI     | Network Node Interface                    |
| OAM     | Operation, Administration and Maintenance |
| OSI     | Open Systems Interconnection              |
| PCR     | Peak Cell Rate                            |
| PHY     | Physical Layer                            |
| PM      | Physical Medium                           |
| PT      | Payload Type                              |
| PVC     | Permanent Virtual Channel                 |
| RM      | Resource Management                       |
| RM-Cell | Resource Management Cell                  |
| SAR     | Segmentation and Reassembly               |
| SBR     | Statistical Bit Rate                      |
| SVC     | Switched Virtual Channel                  |
| TC      | Transmission Convergence                  |
| TCP     | Transmission Control Protocol             |
| UBR     | Unspecified Bit Rate                      |
| UNI     | User Network Interface                    |
| UPC     | Usage Parameter Control                   |
| VA      | Verteilungs-Approximation                 |
| VBR     | Variable Bit Rate                         |
| VC      | Virtual Channel                           |
| VCI     | Virtual Cannel Identifier                 |
| VP      | Virtual Path                              |
| VPI     | Virtual Path Identifier                   |
| VPM     | Verteilungsprognose-Modul                 |
| VQ      | Vektor-Quantisierer                       |
| WAN     | Wide Area Network                         |
| ZG      | Zufallszahlen-Generator                   |

## Formelzeichen

|                |   |
|----------------|---|
| $\mathbb{N}$   | Menge der natürlichen Zahlen  |
| $\mathbb{R}$   | Menge der reellen Zahlen  |
| $\mathbb{Z}$   | Menge der ganzen Zahlen   |
| $f(x)$         | Verteilungsdichtefunktion einer Zufallsvariablen $X$  |
| $F(x)$         | Verteilungsfunktion einer Zufallsvariablen $X$  |
| $w_{ij}$       | Gewichtsfaktor bei Neuronalen Netzen  |
| $\{x_k\}$      | Zahlenfolge   |
| $X_k, x_k$     | Zufallsvariable mit einem Wert für den Zeitpunkt $k$  |
| $\alpha$       | Erstes Moment einer Verteilung  |
| $\hat{\alpha}$ | Empirischer Mittelwert einer Zahlenfolge  |
| $\gamma$       | Autokovarianz   |
| $\hat{\gamma}$ | Empirische Kovarianz einer Zahlenfolge  |
| $\mu$          | Zweites Moment einer Verteilung   |
| $\hat{\mu}$    | Empirische Varianz einer Zahlenfolge  |
| $\rho_i$       | Autokorrelationskoeffizienten   |
| $\hat{\rho}$   | Empirische Autokorrelation einer Zahlenfolge  |
| $N$            | Anzahl Eingänge eines Verteilungsprognosemoduls bzw.<br>Anzahl der Eingänge eines Vektor-Quantisierers              |
| $M$            | Anzahl Verteilungsapproximationen für die Verteilungsprognose bzw.<br>Anzahl der Klassen eines Vektor-Quantisierers |
| $L$            | Anzahl Klassen zur Approximation einer Verteilungsdichtefunktion  |
| $P$            | Vorhersageweite der Verteilungsprognose   |



# Kapitel 1

## Einleitung

### 1.1 Prognoseverfahren für stochastische Prozesse

Bei der Betrachtung komplexer Systeme unterschiedlichster Art stellt sich oft die Frage nach der zeitlichen Weiterentwicklung von Systemgrößen. Die Behandlung dieses Problembereichs erfolgt üblicherweise durch den Einsatz von Prognose- oder Schätzverfahren. Prinzipiell können derartige Vorhersagen nur dann getroffen werden, wenn die betroffene Systemgröße ein serielles Korrelationsverhalten aufweist.

Zur Vorhersage des Verhaltens stochastischer Prozesse werden häufig Verfahren eingesetzt, die aufgrund bekannter vergangener Werte Prognosen über die Weiterentwicklung des Prozesses abgeben. Dabei wird bei den meisten Methoden lediglich der Erwartungswert ermittelt, den die Systemgröße aufgrund der beobachteten vergangenen Werte und aufgrund ihres stochastischen Verhaltens im Mittel annehmen wird.

Für viele Anwendungen ist es sinnvoll, weitere statistische Informationen über den zu erwartenden Wert zu kennen, wie z. B. mittlere Größe und Wahrscheinlichkeit der Abweichung des tatsächlich auftretenden Werts von dem Prognosewert. Diese Möglichkeit bietet ein neuentwickeltes Prognoseverfahren, das die Verteilung zukünftiger Werte einer Systemgröße vorher sagt. Aufgrund dieser Verteilung ist die Bestimmung unterschiedlicher statistischer Kenngrößen möglich, zu denen auch der Mittelwert gehört. Andere Kenngrößen sind beispielsweise Varianz oder Quantile, die unter anderem Aussagen über die Sicherheit einer Vorhersage erlauben.

Ergänzend zu diesen Auswertemöglichkeiten können die prognostizierten Verteilungen auch zur Bestimmung von Zufallswerten herangezogen werden. Dies ermöglicht die einfache Erzeugung von Quellmodellen für den jeweiligen Prozeß.

Die Verteilungsprognose ist mit Hilfe künstlicher neuronaler Netze realisiert. Aufgrund der Lernfähigkeit dieser Netze kann sich die Verteilungsprognose während eines automatisch ablaufenden Lernvorgangs an das Verhalten eines stochastischen Prozesses anpassen. Die stochastischen Prozesse, für die eine Anpassung möglich ist, unterliegen nur wenigen Einschränkungen.

Ein Gebiet, auf dem heute sehr komplexe, weit verteilte Systeme entworfen, untersucht und verwaltet werden müssen, ist die Kommunikationstechnik. Die Modellierung von Systemgrößen durch stochastische Prozesse ist bei den dort betrachteten Systemen eine übliche Vorgehensweise. Für viele Anwendungen sind auch hier Vorhersagen über Systemgrößen erforderlich, die z. B. im Systementwurf oder in der Systemsteuerung weiter verwertet werden. Neben

dem Einsatz auf diesen Anwendungsfeldern erlaubt die Verteilungsprognose auch die Modellierung von Verkehrsquellen.

## 1.2 Übersicht über die Arbeit

Im folgenden **Kapitel 2** werden die Methoden eingeführt, die der Verteilungsprognose zugrundeliegen. Den ersten Teil bildet eine Übersicht über die Geschichte künstlicher neuronaler Netze und ihre wesentlichen Eigenschaften sowie über Anwendungen, speziell aus dem Bereich der Kommunikationstechnik. Im zweiten Teil erfolgt eine Einführung für stochastische Prozesse und deren statistische Kenndaten sowie für Zeitreihen und Zeitreihenmodelle.

**Kapitel 3** bildet den Kern der Arbeit. Hier wird mit Hilfe der Methoden aus Kapitel 2 das neue Verfahren der Verteilungsprognose hergeleitet. Nach der theoretischen Beschreibung des grundlegenden Prinzips folgen die Beschreibungen einiger Erweiterungen. Im Anschluß daran wird an unterschiedlichen Beispielen die Funktionsweise des Verfahrens demonstriert. Das Kapitel endet mit einer Übersicht über potentielle Anwendungsgebiete.

**Kapitel 4** behandelt methodische Grundlagen der Modellierung und Leistungsuntersuchung. In einem ersten Teil werden unterschiedliche Analysemethoden im Hinblick auf ihre Eignung zur Behandlung von Problemen diskutiert, die die Verteilungsprognose mit einbeziehen. Der zweite Teil behandelt die simulative Untersuchung von Systemen. Weiterhin wird eine Simulationskomponente eingeführt, die die Verteilungsprognose realisiert.

Die folgenden beiden Kapitel dienen dem Nachweis der Funktion und Leistungsfähigkeit der Verteilungsprognose anhand exemplarischer Anwendungen aus dem Gebiet der Kommunikationstechnik.

In **Kapitel 5** wird als erstes Anwendungsbeispiel die Modellierung und Nachbildung von Verkehrsquellen mit Hilfe der Verteilungsprognose gezeigt. Nach einer Diskussion der Vor- und Nachteile dieser Art von Quellmodellierung werden mehrere prinzipielle Formen des Quellmodells eingeführt, die für unterschiedliche Anwendungsfälle geeignet sind. Diese Modelle werden dann anhand mehrerer stochastischer und deterministischer Prozesse auf ihre Leistungsfähigkeit untersucht.

**Kapitel 6** behandelt eine zweite Anwendung, die Nutzung der Verteilungsprognose zur Vorhersage geeigneter Werte für die Bandbreitereservierung in ATM-Netzen. Die Anwendung besteht aus der Kopplung zweier lokaler Netze über ein ATM-Netz. Für diese Kopplung wird, basierend auf vergangenen Übertragungsraten, die benötigte Bandbreite vorhergesagt und reserviert. Das Kapitel beginnt mit einer Einführung in ATM und die Verkehrsklasse ABR. Anschließend wird ein Systemmodell der Anwendung erstellt, das im weiteren zur simulativen Untersuchung der Leistungsfähigkeit unterschiedlicher Reservierungsstrategien verwendet wird.

Die Ergebnisse der Arbeit werden in **Kapitel 7** zusammengefaßt und bewertet. Für weitere Anwendungsmöglichkeiten der Verteilungsprognose werden abschließend einige Anregungen gegeben.

# Kapitel 2

## Grundlagen

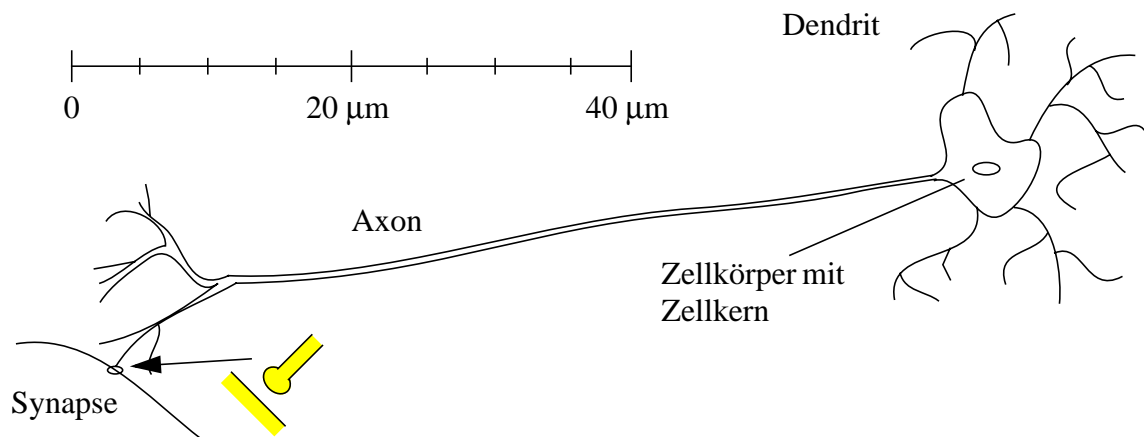
### 2.1 Neuronale Netze

#### 2.1.1 Biologisches Vorbild

Die Großhirnrinde ist der Bereich des (menschlichen) Gehirns, der für die Verarbeitung von Sensorinformationen (Augen, Gehör, Geruchs-, Geschmacks- und Tastsinn) sowie für das Gedächtnis zuständig ist. Da ein Teil dieser Verarbeitungsfähigkeiten auch in der Technik angestrebt wird (autonome Systeme, Bildverarbeitung, Spracherkennung, etc.), beinhaltet primär dieser Teil des Gehirns eine Art von „Funktionalität“, die eine Nachbildung für technische Anwendungen naheliegender erscheinen lässt. Im weiteren Verlauf dieses Abschnitts wird eine Einführung in die Funktionsweise und das Zusammenwirken einiger Bestandteile der Großhirnrinde gegeben.

Die Großhirnrinde besteht hauptsächlich aus untereinander vernetzten Nervenzellen, den Neuronen, die die Informationsverarbeitung im Gehirn vornehmen. Bild 2.1 zeigt eine für das Gehirn repräsentative schematische Darstellung eines Neurons. Je nach Aufgabe werden drei generelle Neuronentypen unterschieden: Interneuronen (Verbindungen nur zu anderen Neuronen), Motorneuronen (Ansteuerung von Muskeln) und Rezeptorneuronen (Verbindungen von Rezeptorzellen des Auges, des Ohrs, etc.).

Ein Neuron besteht aus einem Zellkörper (Soma), davon ausgehenden Auswüchsen (Dendriten) sowie einem langen Fortsatz mit schmalen Verästelungen (Axon). Zwischen den Enden des Axons und Dendriten anderer Neuronen liegen Verbindungsstellen (Synapsen), über die Signale zwischen Neuronen ausgetauscht werden. Über die Synapsen und Dendriten erhält das



**Bild 2.1:** Neuron - biologisches Vorbild

Neuron seine Eingangswerte von anderen Neuronen oder Rezeptorzellen, über das Axon und weitere Synapsen werden Ausgangswerte an andere Neuronen oder Muskeln weitergegeben (transportiert). Die Übertragung von Nervensignalen erfolgt innerhalb des Neurons durch elektrische Signale, an den Synapsen zwischen Neuronen chemisch durch Ionen-transport.

Der Zellkörper eines Neurons arbeitet als Summierer für die Effekte der verschiedenen Eingänge über Synapsen und Dendriten. Die Signale werden je nach Abstand der Synapse vom Zellkörper (Länge des Dendrits) abgeschwächt. Wenn mehrere Signale innerhalb eines bestimmten Zeitfensters an einem Neuron ankommen, summieren sich die Effekte auf. Dies wird als Erregung des Neurons bezeichnet. Übersteigt die Erregung einen bestimmten Schwellwert, erfolgt in dem Neuron ein Potentialsprung, der anschließend über das Axon wandert und weitere Neuronen erregen kann. Dies wird als Feuern bezeichnet. Nach dem Feuern braucht das Neuron eine gewisse Erholungsphase, bevor es wieder erregt werden und feuern kann. Der aktive Zustand hat immer dasselbe Potential und nimmt entlang des Axons nicht ab. Trotzdem hat man kein binäres Verhalten (Aktivität ja/nein), da die kontinuierliche Information durch die Pulsfrequenz sowie durch die Phasenlage unterschiedlicher Signale codiert ist. Die Häufigkeit, mit der Neuronen feuern, kann zwischen einem und rund hundert Ereignissen pro Sekunde schwanken. Durch Kombination dieses binären und kontinuierlichen Verhaltens wird eine optimale Qualität und Sicherheit der Informationsübertragung erzielt.

Die Synapsen weisen einen kleinen nichtleitenden Spalt mit ca. 200 nm Breite auf, über den die Signale durch Ionen-transport übertragen werden. Es gibt verstärkende (exitatorische) und abschwächende (inhibitorische) Synapsen. Inhibitorische Synapsen greifen teilweise an anderen Axonen an und unterbinden dort die Ausbreitung der Signale. Die Stärke der Synapsen schwankt abhängig von ihren chemischen Eigenschaften.

Die synaptische Stärke liegt nicht von vornherein fest. Synapsen, die häufig aktiv werden, werden verstärkt, Synapsen, die weniger häufig aktiv werden, schwächen sich mit der Zeit ab. Diese Eigenschaft wird durch die sogenannte Hebb-Regel beschrieben und spielt eine wichtige Rolle im Lernprozeß. Synapsen werden als elementare Einheiten des Gedächtnisses angesehen [40].

Drei Eigenschaften einer Synapse bestimmen im wesentlichen ihre Auswirkung auf ein Neuron: erstens die Stärke, zweitens der Abstand vom Neuron und drittens die Wiederholrate der ankommenden Signale.

Das gesamte Gehirn besteht aus sehr vielen unterschiedlichen Neuronentypen. Sie unterscheiden sich in der Größe, im Grad der Verzweigung der Dendritenbäume, der Länge der Axonen und anderen Strukturdetails. Grundsätzlich haben alle dasselbe beschriebene Basisprinzip. Es wird angenommen, daß in der menschlichen Großhirnrinde jedes Neuron seine Eingangswerte von im Mittel 10.000 Synapsen bezieht. Andererseits ist jede Zelle ausgangsseitig mit vielen hundert anderen Neuronen verbunden, oft durch eine große Anzahl von Synapsen, die eine einzige Nervenzelle berühren. Schätzungen geben eine Gesamtzahl von mindestens  $3 \times 10^{10}$  Neuronen in der Großhirnrinde an. Verbunden mit der mittleren Anzahl Synapsen pro Neuron führt dies zu einer Anzahl von rund  $10^{15}$  synaptischen Verbindungen im menschlichen Gehirn, von denen die meisten während der ersten Lebensmonate entwickelt werden. Die Anzahl der synaptischen Verbindungen überschreitet bei weitem die Anzahl der möglichen genetischen Informationen der DNA, so daß die detaillierte Struktur des neuronalen Netzes im Gehirn aus einer

Kombination genereller Prinzipien und zusätzlicher Informationen gewonnen werden muß. Diese zusätzliche Information kann aus der Umwelt stammen oder durch das Gehirn selbst generiert werden.

Trotz der Fortschritte und Erkenntnisse der letzten Jahrzehnte sind noch viele Vorgänge im Gehirn unerforscht. Die bereits verstandenen Mechanismen sind zu komplex und die Dichte des Gehirns an „Schaltelementen“ ist zu groß, um erwarten zu können, daß es der Forschung in naher Zukunft gelingt, das Gehirn in seiner Gesamtheit zu verstehen.

Hier stellt sich generell die Frage, ob das intellektuelle Potential des Menschen ihm jemals erlauben wird, das Gehirn, das ja die für dieses Verständnis notwendige Intelligenz hervorbringt, in seiner Gesamtheit zu verstehen.

### **2.1.2 Künstliche neuronale Netze**

Mitte der 40er Jahre begannen Wissenschaftler unterschiedlichster Fachrichtungen wie Psychologie, Medizin, Biologie und Philosophie (McCulloch und Pitts [78]) die Idee zu verfolgen, biologische Neuronen und ihre Verknüpfungen durch künstliche neuronale Netze nachzubilden. Zu diesem Zeitpunkt lagen erste Ergebnisse und Hypothesen zum Aufbau und der Funktion des Gehirns der Primaten und insbesondere des Menschen vor. Das Ziel dieser Forschungsaktivitäten waren Modelle für bestimmte Vorgänge im Gehirn zum Zweck der Verifikation von Theorien und dem besseren Verständnis der Abläufe auf Zellebene.

Erst später kamen Versuche hinzu, einzelne Fähigkeiten des Gehirns für technische Anwendungen nachzuahmen. Erste Erfolge in dieser Richtung waren Ende der 50er Jahre zu verzeichnen (Rosenblatt [90]). Ab Mitte der 60er Jahre war ein Rückgang der Forschung auf diesem Gebiet zu beobachten, hervorgerufen durch einige kritische Veröffentlichungen, die die Anwendbarkeit künstlicher neuronaler Netze in Zweifel zogen (Minsky und Papert [79]). Erst Anfang der 80er Jahre erfolgte durch die Entdeckung neuer leistungsstarker Algorithmen ein Durchbruch, der zu intensiven weltweiten Forschungen führte. Diese Phase brachte viele verschiedene Typen künstlicher neuronaler Netze hervor und dauerte bis Ende der 80er Jahre (Rumelhart, McClelland [91]). Seit Anfang dieses Jahrzehnts steht die Anwendungsforschung im Vordergrund.

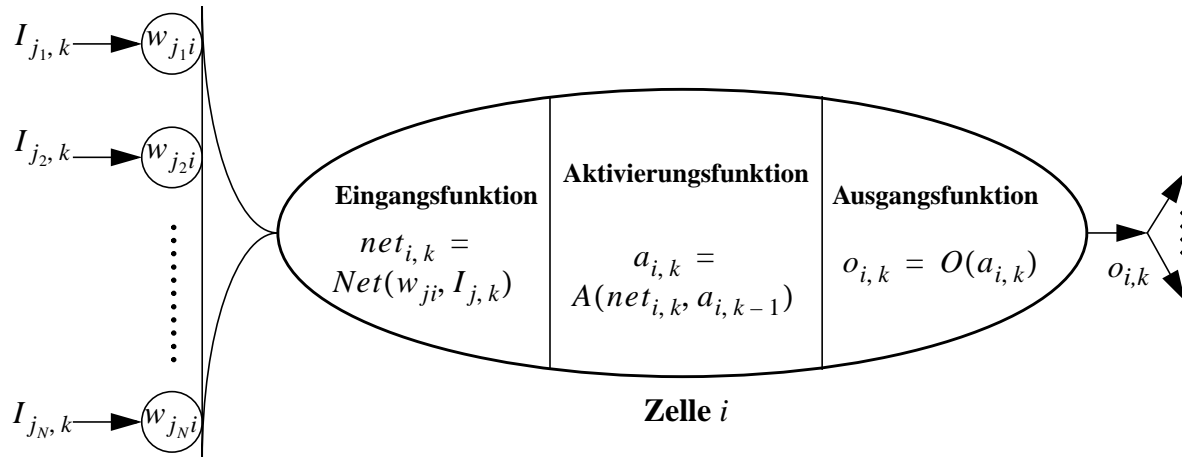
Ein künstliches neuronales Netz (KNN) ist definiert als die Verschaltung gleichartiger, relativ einfacher Bausteine zu einem Netzwerk. Die Informationsweiterleitung erfolgt in diesem Netz nach ähnlichen Prinzipien wie bei biologischen Nervenzellen (Neuronen). Wie das biologische Vorbild setzt sich ein KNN aus Verarbeitungselementen (Zellkörper der Neuronen) und Koppelungselementen (Axon, Synapsen und Dendriten) zusammen. Zur eindeutigen Unterscheidung von biologischen neuronalen Netzen wird im weiteren der Begriff „Zelle“ für künstliche Neuronen und der Begriff „Verbindungsgewicht“ für künstliche Synapsen verwendet.

Für die meisten künstlichen neuronalen Netze wird die Aktivität eines Neurons stark vereinfacht als ein reeller Wert dargestellt, anstatt die Erregungszustände und die Impulsfrequenz nachzubilden. Dabei geht die Phaseninformation der Impulse verloren. Es gibt allerdings auch Ansätze, diese Information nachzubilden [106].

Bild 2.2 zeigt ein allgemeines Modell für ein künstliches Neuron  $i$  zu einem Zeitpunkt  $k$ . Auf der linken Seite sind die Eingänge  $I_j$  zu erkennen, die entweder Ausgänge anderer Zellen oder



Sensorsignale darstellen können. Die Eingangswerte werden durch eine Eingangsfunktion  $Net$ , die im einfachsten Fall eine Summation ist, zu einem gemeinsamen Eingangswert  $net_{i,k}$  zusammengefaßt. Dabei erfolgt auch die Gewichtung der einzelnen Eingangswerte durch Verbindungsgewichte  $w_{ji}$ . Die zentrale Funktionalität der Zelle wird durch die Aktivierungsfunktion  $A$  realisiert. Sie bewertet den kumulierten Eingang  $net_{i,k}$  und beinhaltet in einigen Fällen auch ein Gedächtnis. Am Ausgang der Zelle kann zusätzlich eine Ausgangsfunktion  $O$  den Aktivierungswert  $a_{i,k}$  zum Ausgangswert der Zelle  $o_{i,k}$  modifizieren, der dann an weitere Zellen weitergeführt wird oder einen Ausgangswert des KNN bildet. Häufig begrenzt die Ausgangsfunktion das Ausgangssignal auf einen bestimmten Wertebereich.



**Bild 2.2:** Allgemeines Modell eines künstlichen Neurons

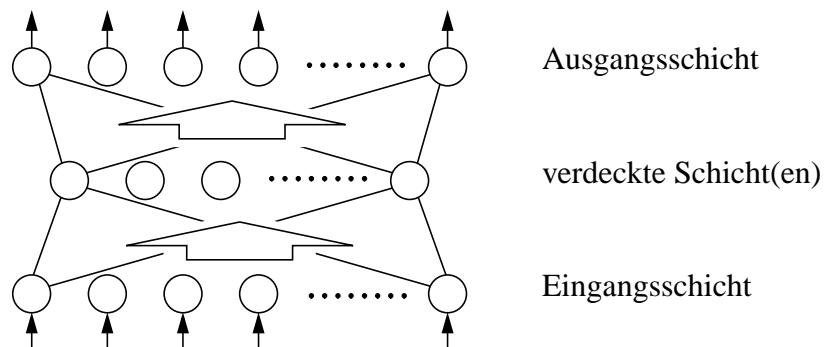
Ein KNN besteht aus vielen dieser Zellen und kann als eine Prozeßstruktur mit verteilter Information in Form eines gerichteten Graphen angesehen werden. Die eigentliche Information eines KNN ist in den Verbindungsgewichten codiert („verteilte Repräsentation“). Diese können entweder durch Vorgabe von Werten „fest verdrahtet“ werden oder durch einen Lernvorgang aus gegebenen Daten gewonnen werden. Die Festlegung eines KNN erfolgt durch den Zellentyp, die Netztopologie und die Lernregeln, durch die die Verbindungsgewichte verändert werden.

Bezüglich des Lernalgorithmus und der Bestimmung der Gewichte gibt es zwei grundlegende Typen von KNNs: Netze, deren Lernvorgang überwacht erfolgt, indem eine Bewertung des Lernfortschritts durch eine übergeordnete Instanz erfolgt, sowie Netze, deren Lernvorgang nicht überwacht abläuft.

Die wichtigsten Vertreter der Netze mit überwachtem Lernvorgang sind Perceptrons (s. Rosenblatt [14]). Sie werden auch Abbildungs-Netze genannt, da sie eine Abbildung  $f: A \subset \mathbb{R}^N \rightarrow \mathbb{R}^M$  realisieren (z. B. Funktionsapproximation). Die Frage, wie komplex die Abbildung sein kann, wurde lange kontrovers diskutiert [79, 36, 101]. Als Lernalgorithmus wird heute in der Regel Backpropagation [41] verwendet, für einlagige Perceptrons z. B. auch die Grossberg-Lernregel [41]. Die Messung und Beurteilung des Lernresultats erfolgt meist über die Summe der quadratischen Abweichungen der Ausgangswerte von vorgegebenen Zielwerten in  $\mathbb{R}^M$ . Im Bereich der Funktionsapproximation sind im Vergleich zur klassischen statistischen Regressionsanalyse durch Verwendung solcher KNNs allgemeinere Funktionen möglich.

Ein Problem der Abbildungsnetze ist die Überanpassung an einen Lerndatensatz (Overtraining), wodurch die Verallgemeinerungsfähigkeit des Netzes leidet. Gelöst wird dieses Problem durch geeignete Abbruchkriterien beim Lernen [41]. Für die Backpropagation-Lernregel gibt es zahlreiche Verbesserungsmöglichkeiten [31, 32, 92, 96].

Perceptrons sind in der Regel als mehrschichtige, ausschließlich vorwärtsgekoppelte Netze ausgeführt (s. Bild 2.3). Vorwärtsgekoppelt bedeutet, daß nur Verbindungen zwischen Zellen niedriger Schichten (näher am Eingang) mit Zellen höherer Schichten (näher am Ausgang) existieren. Durch den Verzicht auf Rückkopplungen wird eine höhere Stabilität der Lernalgorithmen erreicht, deren populärste die Backpropagation-Lernregel ist. Die Eingangsschicht dient in der Regel nur zur Aufspaltung des Signals.



**Bild 2.3:** Allgemeine Struktur eines vorwärtsgekoppelten neuronalen Netzes

Eine Erweiterung der Abbildungsnetze für zeitabhängige Probleme (z. B. Erkennung bestimmter zeitabhängiger Trajektorien im  $\mathbb{R}^N$ ) sind KNNs, die neben räumlichen auch zeitliche Abhängigkeiten berücksichtigen (spaciotemporal). Beispiele sind die Kosko/Klopf-Lernregel [41] und Recurrent Backpropagation für rückgekoppelte Perceptrons [4, 108]. Die Realisierung erfolgt durch Rückkopplungen und das Gedächtnis der Verarbeitungselemente. Anwendungen sind u. a. Filter. Prinzipiell sind rückgekoppelte neuronale Netze anfälliger gegen Instabilitäten [95].

Bei Netzen mit nicht überwachtem Lernvorgang findet keine Beurteilung der Ausgangswerte durch eine übergeordnete Instanz statt, sondern die Anpassung der Verbindungsgewichte erfolgt aufgrund einer allgemeinen und für den jeweiligen Netztyp spezifischen Gütefunktion.

Eine Klasse der Lernregeln für KNNs mit nicht überwachtem Lernen sind die auf Wettbewerb basierenden Lernverfahren. Aufgrund einer allgemeinen Gütefunktion wird ein „Gewinner“ aus einer Gruppe von Zellen bestimmt. Die Gewichte dieses Gewinners werden in geeigneter Weise verstärkt, die Gewichte der anderen abgeschwächt. Da dadurch die Anpassung mehrerer Gewichte voneinander abhängt, spricht man von „nicht lokalen“ Lernverfahren. Ein Beispiel für diesen Netztyp ist das Kohonen-Netz mit der Kohonen-Lernregel [41, 68], das eine Abbildung zweidimensionaler Bereiche aufeinander realisiert, bei der Nachbarschaftsbeziehungen erhalten bleiben.

Eine weitere Klasse von KNNs mit nicht überwachtem Lernen sind stochastische Netze, also Netze mit Zellen, deren Aktivität zufälligen Gesetzmäßigkeiten folgt. Das durch die zufallsbehaftete Natur der Zustandsübergänge entstehende Rauschen dient dem Überwinden lokaler Minima einer zu minimierenden Zielfunktion. Der bekannteste Vertreter dieser Klasse ist das

Hopfield-Netz [41, 45, 82], das aus einer Zellschicht mit Rückkopplungen besteht. Die bevorzugte Anwendungen dieser Netze sind Assoziativspeicher zur Bilderkennung oder -vervollständigung (Boltzmann-Maschine: Hopfield-Netz mit zusätzlichen verdeckten Schichten [82]) sowie Optimierungsprobleme (Suche des Minimums einer Funktion unter Randbedingungen). Beispiele sind kombinatorische Optimierungsprobleme wie das bekannte Problem des Handlungsreisenden [46].

Neben den genannten KNN-Typen, die sich relativ einfach einer grundlegenden Klasse zuordnen lassen, existieren eine Fülle weiterer Typen, die entweder Kombinationen bereits bekannter Typen sind oder die Lösungen für spezielle Probleme darstellen. Beispielsweise ist das Counterpropagation-Network [41] eine Kombination aus einem Kohonen-Netz und einem einschichtigen Perceptron mit Grossberg-Lernregel; Anwendungen sind selbstlernende Assoziativspeicher (Lookup-Table).

Bei einer weiteren Klasse von KNNs wird die Netztopologie nicht fest vorgegeben, sondern sie wird während des Lernprozesses abhängig vom Problem automatisch gefunden. Beispiele hierfür sind GMDH-Netze (Group Method of Data Handling, geeignet zur Approximation von kontinuierlichen Funktionen  $f : A \subset \mathbb{R}^n \rightarrow \mathbb{R}$  [41]) und Cascade-Correlation-Netze [32]. In beiden Fällen werden verdeckte Schichten hinzugefügt, bis das Lernziel erreicht ist.

Mit hierarchischen Netzen wird häufig versucht, einzelne wohlverstandene Teilfunktionen zu einem Ganzen zu kombinieren (z. B. Neocognitron zur Erkennung handgeschriebener Zeichen [35]).

Neben KNNs, die der durch die Biologie getriebenen Forschungsrichtung entsprungen sind, gibt es weitere, wie z. B. neuronale Netze, die ähnlichen statistischen Gesetzen gehorchen wie der Elektronenspin in magnetischen Materialien [82]. Hier werden die Verbindungsgewichte als Beziehungen zwischen den Spins mehrerer Elektronen interpretiert. Diese Typen gehören in die Klasse der stochastischen Netze.

Wichtige Eigenschaften vieler künstlicher neuronaler Netze sind Verallgemeinerungsfähigkeit und Robustheit. Verallgemeinerungsfähigkeit bedeutet, daß Eingangskombinationen, die nicht Teil des Lernvorgangs waren, trotzdem zu sinnvollen Ausgangswerten führen. KNNs sind in mehreren Beziehungen robust: erstens liefern sie meist auch für verrauschte Eingangswerte korrekte Ausgangswerte und zweitens beeinträchtigt das Herausnehmen einzelner Zellen oder Verbindungen das Gesamtverhalten eines KNN häufig nur unwesentlich.

Alle bisher genannten und beschriebenen KNN-Typen dienen der Zuordnung einer Zahl, eines Wahrheitswertes oder einer Aussage zu einem Eingangsmuster, realisieren also deterministische Abbildungen. Solche neuronalen Netze sind nicht für den Einsatz in Gebieten, die auf stochastischen Prinzipien beruhen, geeignet. Um KNNs dort einsetzen zu können, müßten sie statistische Größen wie Wahrscheinlichkeiten, Verteilungen, etc. in geeigneter Form verarbeiten können. Ein erster Ansatz für ein KNN zur Verwendung in stochastischen Systemen, das Verteilungen lernen kann, wird in dieser Arbeit vorgestellt.

Die Architektur künstlicher neuronaler Netze (viele sehr einfache Verarbeitungselemente) legt eine parallele Implementierung in Soft- oder Hardware nahe. Es existieren viele Ansätze in dieser Richtung, u. a. komplette Chipsätze zur Realisierung in Hardware. Die Realisierung künstlicher neuronaler Netze erfolgt heute meist trotzdem durch serielle Rechner, da hier die

Prototypentwicklung sehr viel einfacher ist. Ein Problem speziell der Hardware-Implementierungen ist die schwache Lokalität der Vernetzung bei den meisten KNN-Typen.

Wenn der Lernvorgang eines KNNs abgeschlossen ist, gibt es weitere Möglichkeiten der Optimierung. Ein Ziel hierbei ist die Reduktion der Netzgröße bei gleichbleibender Qualität der Netzfunktionalität. Dies ist deshalb möglich, weil während des Lernvorgangs unterschiedliche Zellen und ihre Gewichte dieselben Eigenschaften lernen. Dadurch entsteht eine redundante interne Darstellung, die durch geeignete Tests und Maßnahmen (Sensibilitätstests, Pruning: Löschen und Zusammenfassen von Zellen und Gewichten [62, 63]) reduziert werden kann. Durch diese Maßnahmen wird auch der oben beschriebene Effekt durch Überanpassung gemindert. Eine weitere Optimierungsmaßnahme – speziell für Assoziativspeicher – besteht im gezielten Schwächen von Gewichten mit dem Ziel, lokale Minima aufzulösen. Dadurch wird vermieden, daß während der Erkennungsphase nicht gespeicherte Muster auftreten.

In den letzten Jahren wurden gegenüber den 80er Jahren deutlich weniger neue Algorithmen für KNNs, wie sie bisher verstanden wurden, publiziert. Dagegen wurde zunehmend versucht, Erkenntnisse aus anderen Gebieten, die sich mit der Anwendung von aus der Biologie bekannten Mechanismen befassen, zur Verbesserung von KNN-Algorithmen zu verwenden. Dazu gehören die Gebiete Fuzzy Logic und Genetische Algorithmen. Kombinationen aus Fuzzy Logic und KNNs wurden unter den Begriffen „Fuzzy-Neuro“ und „Neuro-Fuzzy“ publiziert, je nachdem, welches Gebiet im Vordergrund stand. Genetische Algorithmen werden für neue Lernalgorithmen eingesetzt [87].

### **2.1.3 Klassifizierung und Anwendungen**

Für den Einsatz künstlicher neuronaler Netze haben sich im Lauf der Zeit einige spezielle Einsatzgebiete herauskristallisiert (s. Tabelle 2.1). Bei diesen Anwendungsfeldern handelt es sich in der Regel um Gebiete, auf denen das menschliche Gehirn der „Rechenmaschine“ Computer überlegen ist. Nur dort haben sich KNNs gegenüber konventionellen Ansätzen durchsetzen können.

Eine weitere Aufzählung von Anwendungsgebieten aus der Kommunikationstechnik erfolgt in Tabelle 2.2 (einen Überblick geben auch [24, 34]).

Im folgenden wird eine Reihe von Eigenschaften von Zell- und Netztypen sowie von Lernregeln angegeben, die als Klassifizierungskriterien für KNNs dienen können.

#### **Klassifizierungskriterien für Zell- und Netztypen:**

- Informationsspeicherung vs. Abbildung
- zufallsabhängige (z. B. Hopfield-Netz) vs. deterministischen Netzen (feed-forward-Netze, Perceptrons)
- Art des Eingangs- und Wertebereiches (kontinuierlich bzw. diskret (in der Regel binär))
- Anzahl der Lagen der Netztopologie
- Interner Zustand (Gedächtnis) von Zellen

**Tabelle 2.1:** Einsatzgebiete künstlicher neuronaler Netze

| Gebiet                  | Beispiele  |
|-------------------------|--|
| Mustererkennung         | - Sprach- und Bildererkennung, z. B. aus verrauschten Daten  |
| Mustervervollständigung | - Assoziativspeicher [41, 81]  |
| Optimierungsverfahren   | - Traveling Salesman   |
| Regelungstechnik        | - nichtlineare Systeme, z. B. Robotik (etwa Bahnregelung eines Roboterarms, da hier extrem nichtlineare Bedingungen herrschen)<br>- Systeme mit unbekannter Systemfunktion |
| Zeitreihen              | - Vorhersage von Zeitreihen [25, 104]<br>- Rauschreduktion von Zeitreihen (z. B. Rauschreduktion von EKG durch Backpropagation Filter [24])                                |
| Datenanalyse            | - Bestimmung der Kreditwürdigkeit von Bankkunden   |

- Art der Nachbildung der Erregungszustände und Impulsfrequenz biologischer Neuronen (bei der üblichen Nachbildung durch Analogwerte geht die Phaseninformation der Impulse verloren)
- Art der Aktivierungsfunktionen der Zellen (linear oder nichtlinear)
- Analysierbarkeit (z. B. ist bei Hopfield-Netzen eine exakte Analyse anwendbar)

#### **Klassifizierungskriterien für Lernregeln:**

- Berücksichtigung von Zeitaspekten des Eingangssignals
- Art der Bewertung des Lernerfolgs (überwachtes oder nicht überwachtes Lernen)
- Ermittlung der Verbindungsgewichte (iterativer Lernalgorithmus (z. B. Backpropagation) oder direkte Gewichtsbestimmung aus den Lernmustern)
- Lokalität des Lernverfahrens („nicht lokal“ bedeutet in diesem Zusammenhang, daß eine Gewichtsänderung weitere Gewichte beeinflusst, z. B. Kohonen-Lernverfahren)
- Abhängigkeit der Netztopologie vom Lernverfahren (z. B. Änderung der Netztopologie während des Lernens [32, 41])

#### **2.1.4 Bewertung**

Im Laufe der Zeit haben sich einige Gebiete herauskristallisiert, auf denen künstliche neuronale Netze selbst nach kritischem Vergleich mit anderen Verfahren in technisch-wissenschaftlichen Anwendungen eindeutige Vorteile bringen ([110] bietet einen Überblick). Speziell auf den Gebieten Mustererkennung, Orientierungsfähigkeit in komplexen Umgebungen und Lernfähigkeit (Adaptionsfähigkeit), also für Aufgabenstellungen, die sich häufig einem analytischen Lösungsansatz widersetzen, sind diese neuen Ansätze bisherigen Lösungen und teilweise sogar dem menschlichen Gehirn überlegen.

**Tabelle 2.2:** Anwendungsbeispiele aus der Kommunikationstechnik

| KNN-Typ und Lernregel                    | Anwendung   | Literatur                      |
|--|---|--------------------------------|
| Hopfield-Netz o. ä.                      | Koppelnetzsteuerung                                       | [2, 3, 18, 34, 77, 103]        |
|  | VP-Verkehrslenkung in ATM-Netzen                          | [22]                           |
|  | Verkehrslenkung in verbindungsorientierten Netzen         | [34]                           |
|  | Verkehrslenkung in paketvermittelnden Netzen              | [64, 65]                       |
|  | Kanalzuteilung für Mobilkommunikationssysteme             | [34]                           |
| Kohonen-Netz o. ä.                       | Verkehrslenkung in verbindungsorientierten Netzen         | [5, 86]                        |
|  | Codierung und Kompression digitaler Daten (z. B. Sprache) | [15, 58, 70]                   |
|  | Zuteilung von Satellitenkanälen                           | [34]                           |
| Perceptron mit Backpropagation-Lernregel | Bildkompression   | [13, 27, 33]                   |
|  | Überlastabwehr in ATM-Netzen                              | [20]                           |
|  | Verbindungsannahme in ATM-Netzen                          | [42, 43, 80, 88, 97, 102, 111] |
|  | Codierung   | [84]                           |
|  | Verkehrslenkung in verbindungsorientierten Netzen         | [34]                           |
| Perceptrons mit Rückkopplungen           | Bildkompression   | [27]                           |
|  | Verbindungsannahme in ATM-Netzen                          | [83]                           |
| Lineare Programmierung durch KNNs        | Verkehrslenkung in verbindungsorientierten Netzen         | [1]                            |

Der traditionelle, nicht auf KNNs basierende Lösungsansatz für technische Problemstellungen beruht auf einem problemspezifischen Algorithmus, der direkt aus der Analyse des Problems hervorgeht. Besonders für Problemstellungen, für die eine vollständige Analyse zu aufwendig (zu komplex, zu teuer, etc.) ist, bietet sich der Einsatz von KNNs an. Auch hier ist eine Problemanalyse notwendig, um Netztyp, Lernregel, Topologie des Netzes, Parameter, Initialwerte, usw. festzulegen. Diese Problemanalyse ist aber wesentlich weniger aufwendig als eine vollständige Analyse.

Für die Erstellung und Erprobung künstlicher neuronaler Netze existiert eine Reihe von Programmierwerkzeugen, mit deren Hilfe Prototypen in sehr kurzer Zeit und mit minimalem Programmieraufwand erstellt werden können.

Ein häufig erwähnter Vorteil künstlicher neuronaler Netze ist die Fehlertoleranz, d. h. wenn einzelne Neuronen oder Verbindungen ausfallen, bleibt die Gesamtfunktionalität trotzdem – evtl. etwas schlechter – erhalten. Das gilt allerdings häufig nicht für stark spezialisierte Netze. Dies ist in Analogie zum Gehirn zu sehen: Das Entfernen von Gehirnteilen führt häufig nicht zum Ausfall kompletter Funktionen, sondern zur Verschlechterung mehrerer Funktionen. Wenn allerdings Teile stark spezialisierter Gehirnteile entfernt werden, fällt die damit verbundene Funktion komplett aus. Ein Nachteil dieser an sich positiven Robustheit der meisten KNNs ist, daß häufig nicht nachvollziehbar ist, wie ein KNN eine Aufgabe löst. Die Nachvollziehbarkeit der internen Vorgänge ist häufig nur in kleinen oder spezialisierten Netzen möglich. Dadurch wird die Anwendung in technischen, womöglich sicherheitsrelevanten Bereichen oft nicht toleriert.

Der Einsatzschwerpunkt von KNNs liegt folglich bei Problemstellungen, die sich einer analytischen Lösung widersetzen oder deren vollständige analytische Betrachtung einen zu großen Aufwand bedeuten würde. Der Vorteil der KNNs liegt darin, daß man sich die Lernfähigkeit und Dynamik dieser Systeme zunutze macht und somit eine herkömmliche Problemanalyse überflüssig wird. Hinzu kommt, daß sich die Implementierung von KNNs wesentlich effektiver und einfacher gestaltet als die konventioneller Problemlösungen.

## 2.2 Stochastische Prozesse

### 2.2.1 Einführung

Ein stochastischer Prozeß wird beschrieben durch eine indizierte Familie von Zufallsvariablen  $X_t, t \in T$ , wobei der Index  $t$  häufig für die Zeit steht. Falls  $T \subset \mathbb{Z}$ , dann wird  $\{X_t; t \in T\}$  ein zeitdiskreter stochastischer Prozeß genannt, während im Fall  $T \subset \mathbb{R}$   $\{X_t; t \in T\}$  ein zeitkontinuierlicher stochastischer Prozeß genannt wird.

Einzelne Realisierungen  $x_t$  der Zufallsvariablen  $X_t$  bilden eine Realisierung des stochastischen Prozesses. Die Gesamtheit aller möglichen Realisierungen wird als Ensemble bezeichnet.

Die statistischen Kenngrößen des stochastischen Prozesses werden über das gesamte Ensemble bestimmt:

Erwartungswert oder erstes Moment:

$$\alpha_t = E[X_t] \quad t = 1, \dots, T_{max} \quad (2.1a)$$

Varianz oder zweites zentrales Moment:

$$\mu_t = \text{var}(X_t) = E[(X_t - \alpha_t)^2] \quad t = 1, \dots, T_{max} \quad (2.1b)$$

Kovarianz:

$$\mu_{t, t-\tau} = \text{cov}(X_t, X_{t-\tau}) = E[(X_t - \alpha_t)(X_{t-\tau} - \alpha_{t-\tau})] \quad t = \tau + 1, \dots, T_{max} \quad (2.1c)$$

In der Regel steht nur eine Realisierung des stochastischen Prozesses als Stichprobe zur Verfügung. In diesem Fall können statistische Kenngrößen nicht mehr für einzelne Zeitpunkte bestimmt werden, sondern nur noch *über* die Zeit aus der einzelnen Stichprobe. Dafür müssen aber die statistischen Kenngrößen des zugrundeliegenden Prozesses zeitunabhängig sein. Wenn ein stochastischer Prozeß diese Eigenschaft hat, wird er als stationär bezeichnet und es müssen folgende Bedingungen für alle  $t$  erfüllt sein:

$$E[X_t] = \alpha_t = \alpha \quad (2.2a)$$

$$E[(X_t - \alpha)^2] = \mu_t = \mu = \gamma(0) \quad (2.2b)$$

$$E[(X_t - \alpha)(X_{t-\tau} - \alpha)] = \gamma(\tau) \quad \tau = 1, 2, \dots \quad (2.2c)$$

Dabei ist  $\gamma$  die Autokovarianz des Prozesses für die Verschiebung  $\tau$ .

Die Bedingung in (2.2a-c) stellt eine Bedingung für schwache Stationarität dar. Im Gegensatz dazu steht die strikte Stationarität, die eine Übereinstimmung der  $r$ -dimensionalen Verbundwahrscheinlichkeit für die Stichprobenwerte zu den Zeitpunkten  $t_1, \dots, t_r$  mit der  $r$ -dimensionalen Verbundwahrscheinlichkeit der Stichprobenwerte zu den um ein beliebiges  $\tau$  verschobenen Zeitpunkten  $t_{1+\tau}, t_{2+\tau}, \dots, t_{r+\tau}$  fordert. Strikte Stationarität impliziert schwache Stationarität, vorausgesetzt, die ersten zwei Momente der Verbundwahrscheinlichkeit existieren.



Falls alle statistischen Kenngrößen des Ensembles und der Stichprobe übereinstimmen, spricht man von einem ergodischen Prozeß [85].

Für einen stationären stochastischen Prozeß können die Eigenschaften im Zeitbereich durch die Autokovarianzfunktion  $\gamma(\tau)$  zusammengefaßt werden (s. Gl. (2.2c)). Die Autokovarianzen lassen sich durch Division durch die Varianz normieren, was zur Autokorrelation des Prozesses führt:

$$\rho(\tau) = \gamma(\tau)/\gamma(0) \quad \tau = 0, 1, 2, \dots \quad (2.3)$$

Durch die Normierung ist  $\rho(0) = 1$ .  $\rho(\tau)$  ist symmetrisch zu  $\tau=0$ , daher genügt es, die Werte nach Gl. (2.3) über nicht-negativen Werten von  $\tau$  aufzutragen, um die Autokorrelationsfunktion zu erhalten.

## 2.2.2 Verfahren zur Bestimmung statistischer Kenngrößen

### 2.2.2.1 Empirischer Mittelwert, Varianz und Korrelation

Für die Bestimmung der statistischen Kenngrößen eines stochastischen Prozesses steht nur eine begrenzte Anzahl von  $N$  Meßwerten zur Verfügung. Man erhält die folgenden Schätzwerte für den empirischen Mittelwert  $\hat{\alpha}$ , die empirische Varianz  $\hat{\mu}$ , die empirische Kovarianz  $\hat{\gamma}$  und die empirische Autokorrelation  $\hat{\rho}$ :

$$\hat{\alpha} = \frac{1}{N} \sum_{t=1}^N x_t \quad (2.4a)$$

$$\hat{\mu} = \frac{1}{N} \sum_{t=1}^N (x_t - \hat{\alpha})^2 \quad (2.4b)$$

$$\hat{\gamma}(\tau) = \frac{1}{N} \sum_{t=1}^N (x_t - \hat{\alpha})(x_{t-\tau} - \hat{\alpha}) \quad (2.4c)$$

$$\hat{\rho}(\tau) = \hat{\gamma}(\tau)/\hat{\gamma}(0) \quad \tau = 0, 1, 2, \dots \quad (2.4d)$$

Wenn die Werte nach Gl. (2.4d) über nicht-negative Werte von  $\tau$  aufgetragen werden, erhält man die empirische Autokorrelationsfunktion oder das Korrelogramm.

### 2.2.2.2 Test für unkorrelierte Prozesse

Häufig stellt sich die Frage, ob ein beobachteter stochastischer Prozeß unkorreliert ist. Eine Testmöglichkeit ist die Beurteilung der empirischen Autokorrelation durch einen statistischen Test [39]. Ausgehend von der Annahme, daß die einzelnen Stichprobenwerte unabhängig voneinander sind, gilt, daß die Werte der empirischen Autokorrelation asymptotisch (für  $N \rightarrow \infty$ ) normal verteilt sind mit Mittelwert Null und Varianz  $1/N$ . Die Absolutwerte der empirischen Autokorrelation werden mit dem Faktor  $\sqrt{N}$  normiert und mit einer Schwelle  $s$  verglichen. Tabelle 2.3 zeigt Schwellwerte für verschiedene Signifikanzniveaus (Irrtumswahrscheinlichkeiten). Falls einzelne Werte  $\sqrt{N} \cdot \rho(\tau)$  die Schwelle überschreiten, wird die Annahme der Unkorreliertheit zurückgewiesen. Obwohl der Test eigentlich nur für unabhängige Beobach-

tungen durchgeführt werden kann, liefert er auch erste Anhaltspunkte für stochastische Prozesse, deren Werte voneinander abhängig sind.

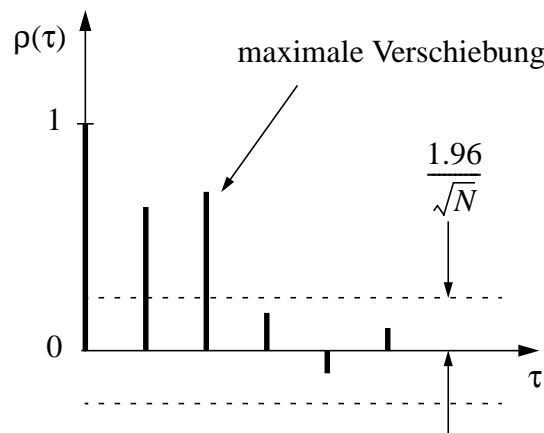
**Tabelle 2.3:** Schwellwerte zu verschiedenen Signifikanzniveaus

| Signifikanzniveau | Schwellwert $s$ |
|-------------------|-----------------|
| 0,01              | 2,58            |
| 0,05              | 1,96            |
| 0,1               | 1,65            |

### 2.2.2.3 Test für Dauer der Autokorrelation

Der obige Test kann modifiziert werden, um eine Abschätzung für die „Dauer“ der Autokorrelation eines stochastischen Prozesses zu liefern; es ist die Verschiebung  $\tau$  gesucht, ab der die Autokorrelation verschwindet.

Wie oben wird das Korrelogramm mit einer Schwelle verglichen; Bild 2.4 zeigt ein Beispiel für ein Signifikanzniveau von 0,05. Fällt der Absolutwert der Korrelation unter die Schwelle, wird angenommen, daß ab dieser Verschiebung keine Korrelation mehr vorhanden ist und die von Null abweichenden Werte durch die endliche Stichprobengröße verursacht werden.



**Bild 2.4:** Test für Dauer der Autokorrelation

### 2.2.2.4 Anpassungstest für Verteilungen

Der Vergleich einer empirischen Verteilungsfunktion mit einer hypothetischen Soll-Verteilungsfunktion  $F(x)$  kann durch den Anpassungstest nach Kolmogorow-Smirnow erfolgen [17]. Die empirische Verteilungsfunktion einer Stichprobe mit  $N$  Werten einer Zufallsvariable  $X$  ist definiert durch

$$W_N(x) = \frac{M_N(x)}{N}, \tag{2.5}$$

wobei  $M_N(x)$  die Anzahl der Stichprobenwerte ist, die kleiner oder gleich  $x$  sind. Bei diskreten Stichprobenwerten ergibt sich eine Treppenfunktion. Der Anpassungstest nach Kolmogorow-Smirnow bewertet nun den maximalen Abstand  $D_N$  zwischen empirischer und hypothetischer Soll-Verteilungsfunktion:

$$D_N = \sup_{x \in \mathbb{R}} |W_N(x) - F(x)|. \quad (2.6)$$

Falls für ein gegebenes Signifikanzniveau  $\alpha$  der Wert  $\sqrt{N} \cdot D_N$  kleiner als ein zu  $\alpha$  gehörendes  $\lambda$  ist, wird die Hypothese, daß  $X$  nach  $F$  verteilt ist, akzeptiert. Die Werte von  $\lambda$  sind in [17] tabelliert, Tabelle 2.4 zeigt Beispiele für einige Signifikanzniveaus (gültig ab einer Stichprobengröße von  $N=36$ ).

**Tabelle 2.4:** Schwellwerte für Kolmogorow-Smirnow-Test

| Signifikanzniveau $\alpha$ | Schwellwert $\lambda$ |
|----------------------------|-----------------------|
| 0,01                       | 1,63                  |
| 0,05                       | 1,36                  |
| 0,1                        | 1,22                  |

### 2.2.3 Prozesse mit Langzeitkorrelation

Einige in Natur und Technik beobachtbare stochastische Prozesse haben die Eigenschaft, daß zeitlich weit auseinanderliegende Zufallsvariablen eine nicht vernachlässigbare Korrelation aufweisen. Diese Eigenschaft basiert auf der sogenannten Selbstähnlichkeit dieser Prozesse und wird durch den Begriff der Langzeitkorrelation charakterisiert (Long Range Dependence, LRD). Selbstähnliche oder fraktale Prozesse zeichnen sich dadurch aus, daß auch der über verschiedene Zeitmaßstäbe aggregierte Prozeß  $\{X_t^{(m)}\}$  (Ersetzen von jeweils  $m$  aufeinanderfolgenden Werten durch ihren Mittelwert) mit

$$X_t^{(m)} = \frac{(X_{tm-m+1} + \dots + X_{tm})}{m} \quad m = 1, 2, \dots \quad (2.7)$$

dieselbe Autokorrelationsfunktion besitzt wie der ursprüngliche Prozeß. Diese Eigenschaft kann z. B. bei der Aktivität in Ethernet-LANs [73] und der Größe von MPEG-codierten Video-Rahmen [28, 29] beobachtet werden.

Je nach Art des Systems, in dem ein stochastischer Prozeß mit LRD-Eigenschaft auftritt, kann diese Eigenschaft starken Einfluß auf das Systemverhalten haben. Daher ist für die Auslegung und Beurteilung solcher Systeme eine quantitative Bewertung der beteiligten selbstähnlichen Prozesse notwendig.

Die quantitative Charakterisierung selbstähnlicher Prozesse erfolgt durch den sogenannten Hurst-Parameter [48]. Er stellt ein Maß für den Grad der Selbstähnlichkeit dar und kann auf verschiedene Arten bestimmt werden [28], von denen an dieser Stelle die R/S-Analyse (Rescaled Adjusted Range Analysis [75, 76]) kurz vorgestellt wird. Die R/S-Analyse kann dazu verwendet werden, den Hurst-Parameter einer Meßreihe auf graphische Weise zu bestimmen.

Ausgehend vom empirischen Summenprozeß für die  $N$ -elementige Stichprobe  $\{x_i\}$

$$x_S(t) = \sum_{i=1}^t x_i \quad 1 \leq t \leq N \quad (2.8)$$

wird die „sample sequential Range“  $R(t,s)$  der Stichprobe für ein Intervall der Länge  $s$  ( $s \geq 2$ ) gebildet:

$$R(t,s) = \max_{0 \leq u \leq s} \left[ x_S(t+u) - x_S(t) - \frac{u}{s}(x_S(t+s) - x_S(t)) \right] - \min_{0 \leq u \leq s} \left[ x_S(t+u) - x_S(t) - \frac{u}{s}(x_S(t+s) - x_S(t)) \right] \quad (2.9)$$

Innerhalb des betrachteten Intervalls ist die empirische Varianz  $S^2(t,s)$  der Stichprobe:

$$S^2(t,s) = \frac{1}{s} \sum_{u=1}^s \left[ x(t+u) - \frac{1}{s}(x_S(t+s) - x_S(t)) \right]^2. \quad (2.10)$$

Daraus ergibt sich die „rescaled adjusted range“  $R(t,s)/S(t,s)$ . Der Erwartungswert von  $R(t,s)/S(t,s)$  verhält sich für stochastische Prozesse mit LRD-Eigenschaft nach dem Potenzgesetz

$$E[R(t,s)/s(t,s)] \sim s^H \quad \text{für} \quad s \rightarrow \infty \quad \text{und} \quad 0,5 < H < 1. \quad (2.11)$$

Die R/S-Analyse läßt sich praktisch als empirischen Test für die LRD-Eigenschaft und zur Bestimmung des Hurst-Parameters  $H$  anwenden. Man unterteilt die Meßreihe in Teilsequenzen und berechnet für jede Teilsequenz die „rescaled adjusted range“ für verschiedene Werte von  $s$ , die logarithmisch äquidistant gewählt werden. In einem Diagramm werden dann alle Punkte mit den Koordinaten  $(\log[s], \log[(R(t,s))/(S(t,s))])$  aufgetragen. Durch die resultierende Punktmenge wird eine Regressionsgerade für  $s_0 \leq s \leq N$  gelegt, wobei man  $s_0$  so wählt, daß nur der relativ lineare Teil der Punktmenge für größere Werte von  $s$  für die Regression berücksichtigt wird. Die Steigung der Regressionsgeraden liefert einen Schätzwert für den Hurst-Parameter  $H$ .

## 2.3 Zufallsvektoren

Mehrere in Zusammenhang stehende Zufallsgrößen können zu einem  $N$ -dimensionalen Zufallsvektor  $(X_1, X_2, \dots, X_N)$  zusammengefaßt werden. Die einzelnen Zufallsvariablen können auch durch je einen stochastischen Prozeß ersetzt werden, wodurch der Zufallsvektor zu einem mehrdimensionalen stochastischen Prozeß wird.

Ein Zufallsvektor kann durch seine  $N$ -dimensionale Verteilungsfunktion

$$F(x_1, x_2, \dots, x_N) = P\{X_1 \leq x_1, \dots, X_N \leq x_N\} \quad (2.12)$$

charakterisiert werden. Wenn  $k$  der Veränderlichen von  $F(\bullet)$  zu  $\infty$  gesetzt werden, erhält man die  $(N-k)$ -dimensionale Randverteilungsfunktion von  $F(\bullet)$ . Für  $k = N-1$  sind das die Verteilungsfunktionen der einzelnen Zufallsvariablen  $X_1, X_2, \dots, X_N$ .

Analog zu den Definitionen (2.1a-c) und (2.3) werden für die Komponenten eines stetigen Zufallsvektors die statistischen Kenngrößen angegeben:

Erwartungswert von  $X_j$  (erstes Moment):

$$\alpha_j = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} x_j dF(x_1, \dots, x_N) \quad (2.13a)$$

Kovarianz von  $X_i$  und  $X_j$ :

$$\mu_{ij} = \text{cov}(X_i, X_j) = \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} (x_i - \alpha_i) \cdot (x_j - \alpha_j) dF(x_1, \dots, x_N) \quad (2.13b)$$

Varianz von  $X_j$  (zweites zentrales Moment):

$$\text{var}(X_j) = \mu_{jj} \quad (2.13c)$$

Korrelationskoeffizient von  $X_i$  und  $X_j$ :

$$\rho_{ij} = \frac{\text{cov}(X_i, X_j)}{\sqrt{\mu_{ii} \cdot \mu_{jj}}} \quad (2.13d)$$

## 2.4 Zeitreihen

### 2.4.1 Einführung

Zeitreihen bestehen aus einer Menge von Beobachtungen einer Größe  $x$ , die üblicherweise zu äquidistanten Zeitpunkten aufgenommen werden [16, 38, 39, 61, 107]. Die Werte der Stichprobe werden mit  $x_1, \dots, x_T$  bezeichnet. Beim Umgang mit Zeitreihen sind zwei Aspekte zu berücksichtigen: die Analyse und die Modellierung. Der Zweck der Analyse ist die Zusammenfassung und Charakterisierung der Eigenschaften einer Zeitreihe. Dies kann entweder im Zeitbereich oder im Frequenzbereich geschehen. Die beiden Formen der Analyse sind komplementär und nicht im Widerstreit zueinander zu sehen. Es wird dieselbe Information auf verschiedene Weise verarbeitet und liefert unterschiedliche Einsichten in die Natur einer Zeitreihe.

Die Motivation zur Modellierung einer Zeitreihe ist in der Regel die Vorhersage zukünftiger Werte. Ein Beispiel dafür ist die Vorhersage von Börsen- oder Devisenkursen. Je nach Art der Problemstellung können in Zeitreihenmodelle mehrere Größen eingehen und deren gegenseitige Beziehungen berücksichtigt werden. Dies ist bei komplexen Ökonomiemodellen der Fall.

Prinzipiell kann eine Zeitreihe stochastisches oder deterministisches Verhalten aufweisen. Da diese Arbeit sich primär mit dem Verhalten stochastischer Prozesse beschäftigt, werden diese im weiteren vorrangig betrachtet.

Vorhersagen werden aufgrund der Vergangenheit der Zeitreihe und/oder in Abhängigkeit von der Zeit durchgeführt. Bei bekannter serieller Korrelation bzw. Autokorrelation ist die Vorher-

sage zukünftiger Werte einfacher als wenn diese nicht bekannt ist. Falls keine serielle Korrelation vorliegt, kann als Vorhersagewert nur der Mittelwert der Zeitreihe verwendet werden, was trivial ist. Der statistische Ansatz der Vorhersage basiert auf der Konstruktion eines Modells. Das Modell definiert einen Mechanismus, von dem angenommen wird, daß er in der Lage ist, Werte mit denselben Eigenschaften wie die Beobachtungen  $x_t$  zu generieren. Ein solches Modell ist immer stochastischer Natur. Das bedeutet, daß bei der Generierung mehrerer Zeitreihen mit Hilfe desselben Modells jedesmal eine andere Zeitreihe entsteht, die aber alle denselben statistischen Gesetzmäßigkeiten gehorchen. In der Terminologie stochastischer Prozesse bilden die durch ein Zeitreihenmodell generierten Wertefolgen aufgrund ihrer gemeinsamen statistischen Eigenschaften ein Ensemble.

Bei der Analyse und Modellierung von Zeitreihen muß zwischen stationären und instationären Zeitreihen unterschieden werden, da dadurch die Komplexität von Analyse bzw. Modellierung wesentlich beeinflusst wird. Stationäre Prozesse lassen sich relativ einfach behandeln; für ihre Modellierung sind beispielsweise ARMA-Modelle (s. Abschnitt 2.4.2.3) gut geeignet. Ein Beispiel für nichtstationäre Zeitreihen sind Zeitreihen mit lokalen oder globalen Trends, wie z. B. Passagierzahlen im Flugverkehr, die von Jahr zu Jahr wachsen. Ein anderer einfacher nichtstationärer Prozeß ist der Random-Walk, bei dem von Schritt zu Schritt eine neue, unabhängige Entscheidung über die Richtung getroffen wird. Dadurch können die Werte dieses Prozesses unbeschränkt wachsen und die Definition eines Mittelwerts ist nicht möglich. Letztlich führen auch feste, z. B. jährliche Zyklen zu einem instationären Verhalten von Zeitreihen. Ein Beispiel sind Arbeitslosenzahlen, die regelmäßig im Winter stark zunehmen.

Nichtstationäre Prozesse haben häufig die Eigenschaft, daß ihre ersten oder höheren Differenzen (Differenzbildung zwischen aufeinanderfolgenden Werten, z. B. zur Eliminierung von Trends) wieder stationär sind und daher wieder durch ARMA-Prozesse modelliert werden können. Dafür geeignete Modelle sind die in der Literatur häufig beschriebenen ARIMA-Modelle [39].

Andere Effekte neben Trends, die eine direkte Verwendung von ARMA-Modellen nicht zulassen, sind Saisoneffekte und Zyklen. In diesem Fall kann der Prozeß häufig als Überlagerung eines ARMA-Prozesses mit einem weiteren, z. B. deterministischen Anteil modelliert werden.

Die Betrachtung von Vektoren  $\mathbf{x}_t$  anstatt eindimensionaler Werte  $x_t$  führt zu mehrwertigen Zeitreihen. In diesem Fall wird die zeitliche Entwicklung mehrerer zueinander in Beziehung stehender Variablen analysiert bzw. modelliert.

## 2.4.2 Zeitreihenmodelle

### 2.4.2.1 AR-Modelle

Autoregressive Modelle (AR) sind seit langer Zeit ein populärer Ansatz zur Modellierung von Zeitreihen. Zur Bestimmung des momentanen Werts der Reihe werden  $p$  vergangene Werte sowie ein normalverteilter Zufallswert verwendet:

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + \varepsilon_t \quad t = 1, 2, \dots \quad (2.14)$$

Hierbei sind  $\phi_1, \dots, \phi_p$  Koeffizienten und  $\varepsilon_t$  der Wert einer normalverteilten Zufallsvariable mit Erwartungswert Null und Varianz  $\sigma^2$ .

Dieser Prozeß der Ordnung  $p$  wird mit  $AR(p)$  abgekürzt. Für ein stationäres Verhalten gelten Einschränkungen für die Koeffizienten  $\phi_i$  [39]. AR-Modelle sind für die Modellierung stark korrelierter Prozesse ein guter Ansatz, da die Autokorrelationsfunktion erst für  $\tau$  gegen unendlich verschwindet.

### 2.4.2.2 MA-Modelle

Moving average-Modelle (MA) bestehen aus einer Sequenz von  $q$  unabhängigen Zufallsvariablen  $\varepsilon_t$  mit Erwartungswert 0 und konstanter Varianz  $\sigma^2$ :

$$x_t = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} \quad t = 1, 2, \dots \quad (2.15)$$

Dieser Prozeß der Ordnung  $q$  mit den Parametern  $\theta_1, \dots, \theta_q$  wird mit  $MA(q)$  abgekürzt und ist bei endlichem  $q$  immer stationär. Die Autokorrelationsfunktion kann direkt aus den Parametern des Prozesses bestimmt werden:

$$\rho(\tau) = \begin{cases} (\theta_\tau + \theta_1 \theta_{\tau+1} + \dots + \theta_{q-\tau} \theta_q) \sigma^2 & \tau = 1, \dots, q \\ 0 & \tau > q \end{cases} \quad (2.16)$$

und hat ab  $\tau = q$  den Wert Null. Dadurch ist im Gegensatz zu AR-Modellen auch die Modellierung von Prozessen mit beschränkter Autokorrelation möglich.

### 2.4.2.3 ARMA-Modelle

Allgemeine ARMA-Modelle (autoregressive-moving average) für Zeitreihen setzen sich aus einer AR-Komponente und einer MA-Komponente zusammen. Da es sich um eine lineare Überlagerung der beiden Teilprozesse handelt, hat der Gesamtprozeß die Summe der Eigenschaften der Teilprozesse.

Ein solcher Prozeß der Ordnung  $(p, q)$  wird üblicherweise mit  $ARMA(p, q)$  abgekürzt und folgendermaßen geschrieben [39]:

$$x_t = \phi_1 x_{t-1} + \dots + \phi_p x_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q}. \quad (2.17)$$

## 2.4.3 Chaotische Zeitreihen

Bei manchen Systemen läßt sich ein Verhalten beobachten, das als „chaotisch“ bezeichnet wird. Damit wird beschrieben, daß bereits für kleinste Unterschiede im Anfangszustand oder in den Randbedingungen die jeweiligen Entwicklungen des betrachteten Systems exponentiell divergieren. Aus dem Bereich der Klimaforschung stammt der Begriff „Schmetterlingseffekt“, der dieses Verhalten treffend beschreibt: kleinste Einflüsse – hier das Flügelschlagen eines Schmetterlings – können überspitzt formuliert das globale Systemverhalten (das Wetter) nachhaltig beeinflussen.

Eine chaotische Zeitreihe  $\{x_t\}$  wird durch einen Definitionsbereich  $D$  und eine Abbildungsvorschrift  $A$ , die den Definitionsbereich auf sich selbst abbildet, beschrieben:

$$\begin{aligned} x_t &\in D \subset \mathbb{R} \\ x_{t+1} &= A(x_t) \quad t = 0, 1, \dots \end{aligned} \tag{2.18}$$

Zeitreihen mit chaotischem Verhalten lassen sich häufig nur schwer von stochastischen Zeitreihen unterscheiden, obwohl sie tatsächlich bis auf den Startwert ein deterministisches Verhalten aufweisen [93]. Aussagen zur Stationarität sind hier schwieriger, da die Verteilung des Ausgangswerts in jedem Schritt durch die Abbildungsvorschrift transformiert wird. Stationarität fordert aber für jeden Zeitpunkt dieselbe Verteilung. Daher können nur chaotische Prozesse stationär sein, deren Verteilung des Startwerts bezüglich der Abbildungsvorschrift invariant ist, was nur für wenige, sehr spezielle Verteilungen erfüllt ist.

Es ist also sehr wichtig, chaotisches Verhalten als solches zu erkennen und das Bildungsgesetz zu finden, da statistische Standardverfahren (z. B. Bestimmung der Korrelation) für chaotische Prozesse im allgemeinen keine sinnvollen Aussagen liefern. In [93] werden Verfahren genannt, wie chaotisches Verhalten aufgespürt werden kann.

Für die Modellierung solcher Systeme bedeutet dies, daß Zeitreihenmodelle nach Abschnitt 2.4.2 ungeeignet sind.



# Kapitel 3

## Ein Verfahren zur Verteilungsprognose

### 3.1 Motivation

Zeitreihen und stochastische Prozesse spielen in vielen Bereichen der Wirtschaft, der Technik und des täglichen Lebens eine wichtige Rolle. Dabei stellt sich häufig das Problem der Prognose bzw. Vorhersage zukünftiger Werte oder der Schätzung fehlender, nicht meßbarer oder verrauschter Werte zum momentanen Zeitpunkt. Üblicherweise erfolgt sowohl die Schätzung als auch die Prognose auf der Basis einer Reihe aktueller und vergangener Meßwerte.

Beispiele für Informationen, zu deren Vorhersage Prognoseverfahren eingesetzt werden, sind Devisen- und Aktienkurse, Wetterdaten sowie Pegelstände von Gewässern. Die Behandlung derartiger Problemstellungen erfolgt meist durch Ansätze und Methoden der Zeitreihenanalyse.

Im Gegensatz zu einer Prognose oder Vorhersage liegen bei einer Schätzung Meßwerte für den Schätzzeitpunkt vor. Ein Einsatzgebiet ist die Schätzung von Größen aus verrauschten Meßwerten oder von zum Teil nicht meßbaren Daten in der Regelungstechnik. Es werden Systemzustandsgrößen geschätzt, die z. B. als Eingangswerte für Regler benötigt werden. Man spricht hier auch von Filterung.

Schätz- und Prognoseverfahren spielen auch in der immer komplexer werdenden Kommunikationstechnik eine wichtige Rolle: Auch hier sind Aussagen über die zukünftige Entwicklung einzelner Größen notwendig und es werden regelungstechnische Verfahren eingesetzt, für die häufig die Werte einzelner Größen geschätzt werden müssen. Ein Beispiel ist die Prognose von Lastsituationen, um Maßnahmen zur Überlastvermeidung ergreifen zu können.

Bei der Schätzung von Systemzuständen werden bestimmte Annahmen für die zu berücksichtigenden Störgrößen bezüglich ihrer Verteilung gemacht. Dies führt zur Schätzung eines Werts, der den Mittelwert der tatsächlich auftretenden Werte approximiert. Bei der Prognose zukünftiger Werte werden Kenntnisse über den zugrundeliegenden Prozeß genutzt, um einen zukünftigen Wert vorherzusagen, der wiederum den Mittelwert der tatsächlich auftretenden Werte approximiert. Je nach Problemstellung kann dieser mittlere Wert noch durch eine Zufallsgröße modifiziert werden, z. B. bei der Modellierung von Zeitreihen.

Generell erfolgt die Schätzung oder Prognose von Größen aufgrund von Vorwissen über das stochastische Verhalten der bisher beobachteten Folge, beschrieben durch deren Verteilung und Autokorrelation.

In dieser Arbeit wird ein neuer Ansatz vorgestellt, der die Güte und Flexibilität von Schätzung und Prognose erhöht. Die Neuerung besteht darin, daß nicht wie bisher üblich der Mittelwert

der zukünftigen Werte ermittelt wird, sondern deren Verteilung. Aus der Verteilung können neben dem Mittelwert wesentlich mehr statistische Kenndaten gewonnen werden, wie z. B. Varianz und Quantile. Die Art der Auswertung muß abhängig von der Problemstellung geeignet gewählt werden.

Dieses Kapitel beinhaltet die Herleitung, die Realisierung sowie einfache Beispiele zur Anwendung dieses Verfahrens. Die Realisierung kann gleichermaßen für Schätz- und Prognoseaufgaben eingesetzt werden, es ändert sich lediglich die Interpretation einiger Werte.

## 3.2 Herleitung

### 3.2.1 Prinzip

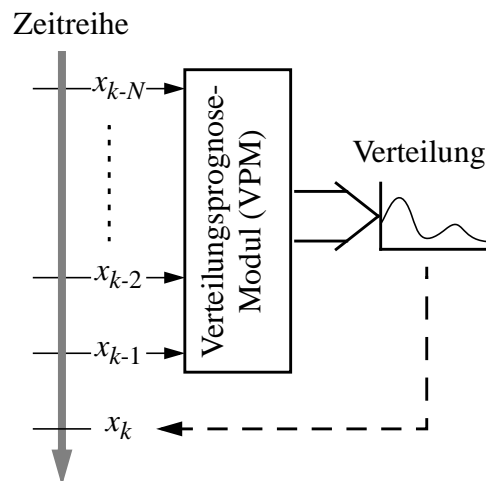
Das in diesem Abschnitt beschriebene Verfahren dient zur Prognose der Verteilung des nächsten Werts einer Zahlenfolge  $\{x_k\}$ ,  $k \in \mathbb{Z}$ . Diese Zahlenfolge kann beliebigen Ursprungs sein; es kann sich z. B. um eine Zeitreihe gemäß der Definition in Abschnitt 2.4.1 handeln. Bei Zeitreihen ist der betrachtete aktuelle Zeitpunkt  $k$ ; der Wert der Folge zu diesem Zeitpunkt sei noch nicht bekannt.

Wie bei allen Prognoseverfahren beruhen die Vorhersagen für  $x_k$  auch in diesem Fall auf bekannten Werten von vergangenen Zeitpunkten. Für die einfachste Version des Verfahrens sind dies alle Werte eines einzigen stochastischen Prozesses. Eine Erweiterung für mehrere Eingangsgrößen erfolgt in Abschnitt 3.4.

Bild 3.1 zeigt das Prinzip der Verteilungsprognose: Die letzten  $N$  Werte der Folge, die dem Wert  $x_k$  vorangegangen sind, werden zu einem Vektor

$$\mathbf{i}_k = (i_1, i_2, \dots, i_N) = (x_{k-1}, x_{k-2}, \dots, x_{k-N}) \quad (3.1)$$

zusammengefaßt, wobei angenommen wird, daß alle Folgenwerte für  $k < 0$  den Wert Null haben. Der Vektor  $\mathbf{i}_k$  wird in den Block „Verteilungsprognose-Modul“ (VPM) eingespeist. Innerhalb dieses Moduls wird die Prognose für die Verteilung des Werts für den Zeitpunkt  $k$  ermittelt. Für jeden Vektor  $\mathbf{i}_k$  am Eingang existiert eine eindeutige Verteilung am Ausgang.



**Bild 3.1:** Prinzip der Verteilungsprognose

Die Verteilung am Ausgang des VPM ist die bedingte Verteilung für das „Gedächtnis“  $\mathbf{i}_k$ . Sie ist für einen gegebenen stochastischen Prozeß  $\{x_k\}$  mit bekannter Korrelation und Verteilung eindeutig angebar (s. Abschnitt 3.2.2).

In der Regel sind die stochastischen Prozesse, die in realen Systemen vorkommen, nicht im Detail bekannt. Daher ist die Realisierung des hier vorgestellten Verfahrens dafür ausgelegt, sich automatisch an einen stochastischen Prozeß zu adaptieren.

### 3.2.2 Beziehung zwischen Verteilung und Autokorrelation einer Zeitreihe

Der beschriebene Mechanismus prognostiziert Verteilungen. Der Nachweis, daß die entsprechenden Vorhersagen auch die Korrelation des Prozesses berücksichtigen, wird in diesem Abschnitt erbracht. Um diesen Nachweis besser führen zu können, wird einschränkend gefordert, daß der betrachtete stochastische Prozeß  $\{x_k\}$  schwach stationär ist (s. Abschnitt 2.2.1). Dies schließt auch zyklische Prozesse mit ein, deren Startzeitpunkt zufällig gewählt wird.

Der Wert  $x_k$  und der Eingangsvektor  $\mathbf{i}_k$  werden im weiteren als ein  $(N+1)$ -dimensionaler Vektor  $\mathbf{x}_k = (x_k, \mathbf{i}_k)$  behandelt. Die  $N+1$  Komponenten des Zufallsvektors  $\mathbf{x}_k$  (s. Abschnitt 2.3) können aufgrund der Stationaritätsannahme als Realisierungen identisch verteilter Zufallsvariablen  $X_k, X_{k-1}, \dots, X_{k-N}$  aufgefaßt werden.

Andererseits gehorcht der Zufallsvektor  $\mathbf{x}_k = (x_k, \mathbf{i}_k)$  einer  $(N+1)$ -dimensionalen Verteilungsfunktion

$$\begin{aligned} F(\mathbf{x}_k) &= P\{\mathbf{X}_k \leq \mathbf{x}_k\} \\ F(x_k, x_{k-1}, \dots, x_{k-N}) &= P\{X_k \leq x_k, \dots, X_{k-N} \leq x_{k-N}\} \end{aligned} \quad (3.2)$$

und Dichtefunktion

$$\begin{aligned} f(\mathbf{x}_k) &= \frac{d}{d\mathbf{x}_k} F(\mathbf{x}_k) \\ f(x_k, x_{k-1}, \dots, x_{k-N}) &= \frac{\partial^N}{\partial x_k \dots \partial x_{k-N}} F(x_k, \dots, x_{k-N}) \end{aligned} \quad (3.3)$$

$F(\mathbf{x}_k)$  ist also die  $(N+1)$ -dimensionale Verbundverteilungsfunktion des Zufallsvektors  $\mathbf{x}_k$ , deren 1-dimensionalen Randverteilungen die Verteilungen der einzelnen Zufallsvariablen darstellen. Damit ist auch die vorherzusagende Verteilung von  $X_k$  vollständig durch  $F(\mathbf{x}_k)$  beschrieben.

Zur Berechnung der Autokorrelationskoeffizienten der Zeitreihe  $\{x_k\}$  sind das erste Moment  $\alpha_j$  und das zweite zentrale Moment  $\mu_{ij}$  der Zufallsvariablen  $X_k, X_{k-1}, \dots, X_{k-N}$  notwendig. Für das erste Moment gilt:

$$\begin{aligned}\alpha_j &= \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} x_j \cdot f(x_k, x_{k-1}, \dots, x_{k-N}) dx_k \dots dx_{k-N} \\ &= \int_{\mathbb{R}^{N+1}} x_j \cdot f(\mathbf{x}_k) d\mathbf{x}_k \quad j = k-N, \dots, k\end{aligned}\tag{3.4}$$

Da aufgrund der Stationaritätsannahme alle  $N+1$  Zufallsvariablen derselben Verteilung gehorchen, haben sie dasselbe erste Moment:  $\alpha_j = \alpha$ . Damit gilt für die zweiten zentralen Momente:

$$\begin{aligned}\mu_{ij} &= \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} (x_i - \alpha) \cdot (x_j - \alpha) \cdot f(x_k, x_{k-1}, \dots, x_{k-N}) dx_k \dots dx_{k-N} \\ &= \left( \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} x_i \cdot x_j \cdot f(x_k, x_{k-1}, \dots, x_{k-N}) dx_k \dots dx_{k-N} \right) - \alpha^2 \\ &= \int_{\mathbb{R}^{N+1}} x_i \cdot x_j \cdot f(\mathbf{x}_k) d\mathbf{x}_k - \alpha^2 \quad i, j = k-N, \dots, k\end{aligned}\tag{3.5}$$

Für  $i=j$  ergeben sich aus (3.5) die Varianzen  $\mu_{ii}$  der einzelnen Zufallsvariablen. Diese sind wie die ersten Momente alle gleich:  $\mu_{ii} = \mu$ .

Die Korrelationskoeffizienten sind dann definiert als

$$\begin{aligned}\rho_{ki} &= \frac{\mu_{ki}}{\sqrt{\mu_{kk}\mu_{ii}}} = \frac{\mu_{ki}}{\mu} \\ &= \frac{1}{\mu} \left\{ \left( \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} x_k \cdot x_i \cdot f(x_k, x_{k-1}, \dots, x_{k-N}) dx_k \dots dx_{k-N} \right) - \alpha^2 \right\} \\ &= \frac{1}{\mu} \left\{ \int_{\mathbb{R}^{N+1}} x_k \cdot x_i \cdot f(\mathbf{x}_k) d\mathbf{x}_k - \alpha^2 \right\} \quad i = k-N, \dots, k\end{aligned}\tag{3.6}$$

Falls der Wert  $N$  so groß gewählt wird, daß er den Bereich der Autokorrelation des Prozesses  $\{x_k\}$  abdeckt, kann man aus den Korrelationskoeffizienten nach (3.6) die Autokorrelationskoeffizienten  $\rho_\tau$  von  $\{x_k\}$  durch die Substitution  $\tau = k-i$  bestimmen:

$$\begin{aligned}\rho_\tau &= \frac{\mu_{k, k-\tau}}{\mu} \\ &= \frac{1}{\mu} \left\{ \int_{\mathbb{R}^{N+1}} x_k \cdot x_{k-\tau} \cdot f(\mathbf{x}_k) d\mathbf{x}_k - \alpha^2 \right\} \quad \tau = 0, \dots, N\end{aligned}\tag{3.7}$$

Die Bestimmung von  $N$  kann z. B. durch einen statistischen Test wie in Abschnitt 2.2.2.3 erfolgen.

Die Herleitung von Gleichung (3.7) zeigt, daß die Autokorrelationskoeffizienten des stochastischen Prozesses vollständig durch die Verbundverteilungsfunktion  $F(\mathbf{x}_k)$  beschrieben sind.

Damit wurde gezeigt, daß die Verteilungsprognose bei einer hinreichend großen Anzahl von  $N$  vergangenen Werten zu einer Fortschreibung des stochastischen Verhaltens – insbesondere auch der Autokorrelation – des stochastischen Prozesses führt.

### 3.2.3 Vorhersage für einen Schritt

Das Ziel der Verteilungsprognose ist die Vorhersage der Verteilung von  $x_k$  unter der Voraussetzung, daß die Werte  $x_{k-1}, \dots, x_{k-N}$  bekannt sind. Dies entspricht der Kenntnis der bedingten Verteilungsdichtefunktion

$$\tilde{f}(x_k | X_{k-1} = x_{k-1}, \dots, X_{k-N} = x_{k-N}). \quad (3.8)$$

Theoretisch ließe sich diese bedingte Verteilungsdichte als Quotient aus der Dichte nach Gl. (3.3) und der  $N$ -dimensionalen Dichte der Randverteilung des Zufallsvektors  $(X_{k-1}, \dots, X_{k-N})$  berechnen. Da diese Funktionen normalerweise nicht bekannt sind, müßten sie aus einer Stichprobe bestimmt werden. Dies ist nur schwer möglich, da es sich um Funktionen im  $\mathbb{R}^{N+1}$  handelt und die Stichprobengröße dafür in der Regel nicht groß genug ist. Anstelle der genauen Bestimmung von  $\tilde{f}(x_k | \bullet)$  wird daher eine Approximation verwendet.

### 3.2.4 Approximation

In Abschnitt 3.2.2 wurde gezeigt, daß die Kenntnis der Verbundverteilungsfunktion  $F(\mathbf{x}_k)$  nach Gl. (3.2) ausreicht, um Vorhersagen für die Verteilung zukünftiger Werte machen zu können. Für die Verteilungsprognose wird die bedingte Verteilung von  $X_k$  benötigt, die zwar theoretisch bestimmbar, aus einer Stichprobe aber kaum zu ermitteln ist.

Aus diesem Grund wird hier eine Approximation der Verteilungsdichte von  $X_k$  durchgeführt. Jeder mögliche Eingangsvektor wird auf eine bestimmte Dichtefunktion  $\tilde{f}$  abgebildet. Falls die Komponenten des Eingangsvektors, die den sogenannten Eingangsraum aufspannen, kontinuierliche Werte annehmen, sind dies unendlich viele unterschiedliche Funktionen. Um diese Vielfalt zu reduzieren, wird unter der Annahme der Stetigkeit von  $\tilde{f}$  über den Koordinaten des Eingangsraums der Eingangsraum in eine endliche Anzahl von  $M$  Bereichen unterteilt und die Dichte  $\tilde{f}$  in jedem dieser Bereiche durch die Dichte  $\hat{f}$  approximiert. Dadurch wird die theoretisch unendliche Zahl von Dichtefunktionen auf  $M$  approximierte Funktionen reduziert. Die Annahme der Stetigkeit ist notwendig, um bei der Aufteilung des Eingangsraums nicht Unstetigkeitsstellen der Dichte berücksichtigen zu müssen.

Die gesamte Approximation erfolgt also durch lokale Approximationen in  $M$  Teilbereichen  $I_i$  des  $N$ -dimensionalen kontinuierlichen Eingangsraums. In jedem dieser Bereiche wird die bedingte Dichte von  $X_k$ , Gl. (3.8), durch die Funktion  $\hat{f}_i(x_k)$  lokal approximiert. Durch Vereinigung aller Bereiche  $I_i$  entsteht der  $N$ -dimensionale Raum  $\mathbb{R}^N$ :

$$\bigcup_{i=1}^M I_i = \mathbb{R}^N. \quad (3.9)$$

Die Approximation der Dichte von  $X_k$  in Bereich  $I_i$  wird als Mittelwert der  $(N+1)$ -dimensionalen Dichte nach (3.3) in diesem  $N$ -dimensionalen Bereich definiert. Als freie Variable bleibt  $x_k$ :

$$\hat{f}_i(x_k) = \frac{1}{norm_i} \cdot \int \dots \int_{I_i} f(x_k, t_1, \dots, t_N) dt_1 \dots dt_N \quad (3.10)$$

$$norm_i = \int \dots \int_{I_i} f^*(t_1, \dots, t_N) dt_1 \dots dt_N \quad (3.11)$$

Dabei ist  $f^*$  die  $N$ -dimensionale Randverteilung von  $f$  ohne  $x_k$ :

$$f^*(t_1, \dots, t_N) = \int_{-\infty}^{\infty} f(x_k, t_1, \dots, t_N) dx_k \quad (3.12)$$

Durch den Normierungsfaktor  $norm_i$  hat das Integral  $\int_{-\infty}^{\infty} \hat{f}_i(x_k) dx_k$  den Wert 1.

Durch die Approximation in den einzelnen Bereichen  $I_i$  ist die Dichte von den Koordinaten  $x_{k-1}, \dots, x_{k-N}$  unabhängig. Durch eine Erhöhung der Bereichszahl  $M$  kann eine beliebig gute Näherung der tatsächlichen Dichte erreicht werden, d. h.

$$\tilde{f}(x_k | \bullet) = \lim_{M \rightarrow \infty} \hat{f}_i(x_k). \quad (3.13)$$

### 3.2.5 Realisierung

Die Realisierung der Approximation der Verteilungsprognose erfolgt so, daß sie für weitgehend beliebige stochastische Prozesse automatisch abläuft und sich auch an den Wertebereich des jeweiligen Prozesses anpaßt. Dadurch werden die vorangestellte Analyse des stochastischen Prozesses und die Anzahl der einzustellenden Parameter minimiert.

Der Eingangsvektor  $i_k$  wird durch ein reellwertiges Schieberegister realisiert, in das die Werte der Folge  $\{x_k\}$  geschoben werden. Dadurch speichert das Schieberegister die vergangenen Werte und bildet ein „Gedächtnis“ für  $\{x_k\}$ .

Die Aufteilung des  $\mathbb{R}^N$  in die Bereiche  $I_i$  erfolgt mit Hilfe eines Vektor-Quantisierungs-Algorithmus (Vector Quantizer, VQ), dessen Eingangswerte durch die Ausgangswerte des Schieberegisters gebildet werden. Durch den VQ wird der Raumbereich bestimmt, in den ein Eingangsvektor fällt. Am Ausgang erscheint dann die Nummer  $i$  dieses Bereichs, was einer Abbildung  $\mathbb{R}^N \rightarrow \mathcal{I}$  entspricht. Auf diese Weise wird eine starke Reduktion der Komplexität erzielt. Dieser Algorithmus basiert auf einem speziellen künstlichen neuronalen Netz. Er wird an dieser Stelle anschaulich beschrieben, eine genauere Beschreibung erfolgt in Anhang A1.

Jeder der Raumbereiche ist durch einen Mittelpunktvektor  $\hat{p}_i = (\hat{x}_1^{(i)}, \dots, \hat{x}_N^{(i)})$  gekennzeichnet, wobei der Wert  $\hat{x}_j^{(i)}$  für den Bereich  $i$  ( $i=1, \dots, M$ ) gilt und mit der Komponente  $i_j$  ( $j=1, \dots, N$ ) des Eingangsvektors korrespondiert. Jeder Vektor  $\hat{p}_i$  ist der Ortsvektor des Zentrums eines Bereichs  $I_i$  des  $\mathbb{R}^N$ . Die Zugehörigkeit eines Eingangsvektors zu einem Bereich  $I_i$  bzw. zu einem Bereichsvektor  $\hat{p}_i$  wird über den Abstand der Vektorendpunkte bestimmt.

Die Grenze eines Bereichs  $I_i$  ist definiert als die Sphäre des  $\mathbb{R}^N$ , für deren Punkte  $\mathbf{q}$  gilt:

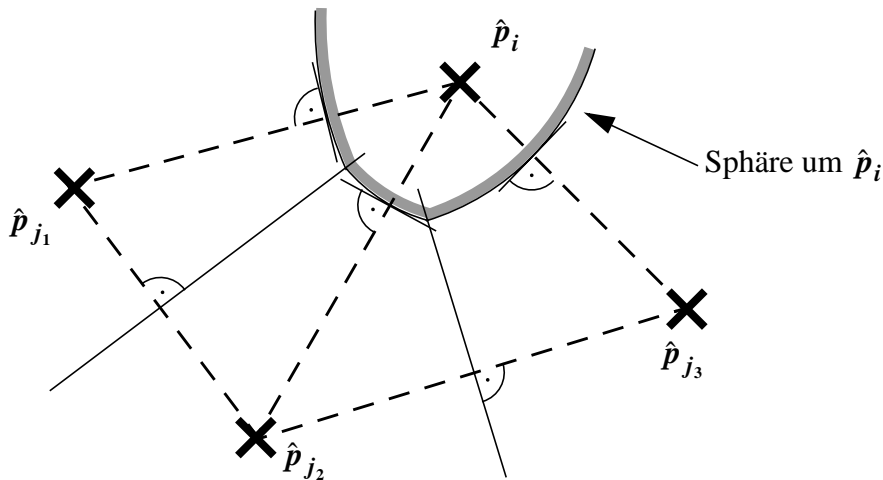
$$b_i \cdot \|\mathbf{q} - \hat{\mathbf{p}}_i\| = \min_{j=1, \dots, M; j \neq i} (b_j \cdot \|\mathbf{q} - \hat{\mathbf{p}}_j\|). \quad (3.14)$$

Die Raumaufteilung durch den Algorithmus erfolgt durch Anpassung der Bereichsmittelpunkte  $\hat{\mathbf{p}}_i$  und der Gewichtungsfaktoren  $b_i$  so, daß alle Bereiche  $I_i$  mit derselben Häufigkeit ausgewählt werden. Dies wird durch geeignete Werte der  $b_i$  erreicht, was einer Verschiebung der Bereichsgrenzen entspricht. Falls die Gewichtungsfaktoren  $b_i$  zweier benachbarter Bereiche denselben Wert haben, liegt die Bereichsgrenze auf halbem Weg zwischen den Bereichsmittelpunkten.

$\|\dots\|$  symbolisiert das Euklidische Abstandsmaß, wonach der Abstand  $D$  zweier Vektoren  $\mathbf{a}$  und  $\mathbf{b}$  folgendermaßen definiert ist:

$$D = \|\mathbf{a} - \mathbf{b}\| = \sqrt{(a_1 - b_1)^2 + \dots + (a_N - b_N)^2}. \quad (3.15)$$

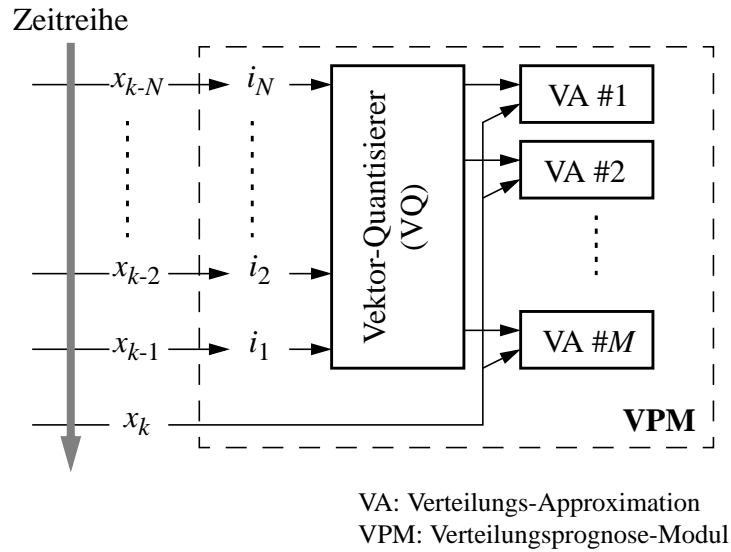
Bild 3.2 veranschaulicht die Raumaufteilung durch einen VQ für den zweidimensionalen Fall ( $N=2$ ). Im dargestellten Beispiel gibt der Endpunkt von  $\hat{\mathbf{p}}_i$  den Bereichsmittelpunkt des hervorgehobenen Bereichs an. Das  $b_i$  des hervorgehobenen Bereichs hat den Wert 2, alle anderen haben den Wert 1. Dadurch unterteilt die trennende Sphäre zwischen  $\hat{\mathbf{p}}_i$  und den anderen Punkten  $\hat{\mathbf{p}}_j$  die jeweiligen Verbindungslinien im Verhältnis 2 zu 1. Die Sphäre um  $\hat{\mathbf{p}}_i$  setzt sich aus mehreren gekrümmten Flächen zusammen (im 2-dimensionalen Fall aus mehreren Kurven), da nur bei gleichem  $b_i$  zweier Punkte die sie trennende Sphäre eine Ebene ist (im 2-dimensionalen Fall eine Gerade).



**Bild 3.2:** Raumaufteilung durch Vektor-Quantisierer

Durch die Approximation der Verteilungsdichte über eine innerhalb eines Bereichs  $I_i$  nur von  $x_k$  abhängige Dichte entsteht ein Fehler. Die Fehler in den einzelnen Bereichen werden aufgrund der Stetigkeit der Dichte durch Verkleinerung der Bereiche ebenfalls kleiner. Daher wird die Summe aller Fehler mit steigender Bereichszahl  $M$  immer kleiner und nähert sich für  $M \rightarrow \infty$  dem Wert Null.

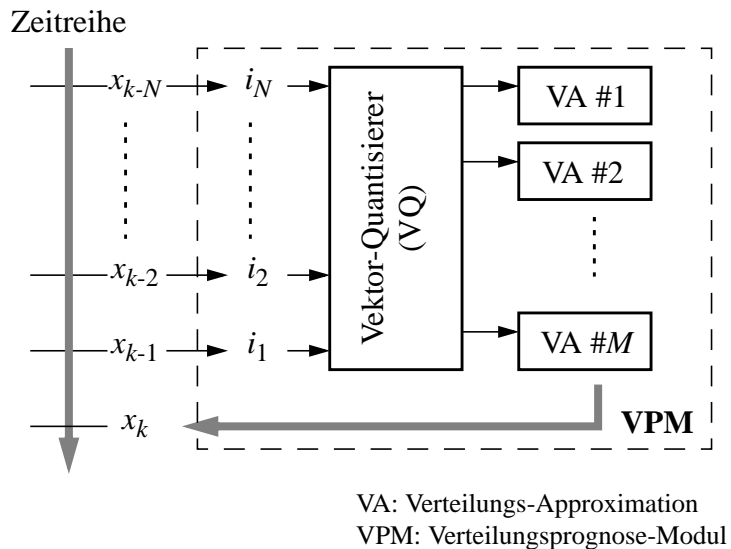
Die approximative Dichte  $\hat{f}_i(x_k)$  des Bereichs  $I_i$  wird wie die Bereichsaufteilung des VQs automatisch aus der gegebenen Zeitreihe gewonnen. Ein zu diesem Zweck neuentwickeltes



**Bild 3.3:** Lernvorgang

künstliches neuronales Netz paßt sich automatisch der Wertefolge  $\{x_k\}$  an. Dieser Algorithmus wird in Anhang A2 detailliert beschrieben. Er basiert darauf, daß sich ein KNN während einer Lernphase so an einen stochastischen Prozeß adaptiert, daß sich an seinem Ausgang die Verteilungsdichte des stochastischen Prozesses einstellt. Da die Regionen  $I_i$  gleich häufig durch Eingangsvektoren getroffen werden, werden auch alle Verteilungsapproximationen gleich häufig zur Adaption ausgewählt, was für einen gleichmäßigen Lernvorgang unerlässlich ist. Die Approximation der Dichte erfolgt durch eine stückweise konstante Funktion aus  $L$  Abschnitten. Auch hier wird durch die Approximation ein Fehler gemacht, der durch eine steigende Zahl von Abschnitten gegen Null geht.

Die Bilder 3.3 und 3.4 zeigen ein detailliertes Modell der Verteilungsprognose, das Verteilungsprognose-Modul (VPM). Der Vektor-Quantisierer bildet jeden Eingangsvektor auf eine zugehörige Verteilungs-Approximation ab.



**Bild 3.4:** Wiedergabevorgang



In Bild 3.3 ist der Lernvorgang dargestellt. Der Eingangsvektor  $i_k$  dient dem VQ zur Bestimmung einer Region  $I_i$  und der zu dieser Region gehörenden Verteilungs-Approximation (VA). Der nächste Wert  $x_k$  wird hier als Eingangswert für die durch den VQ ausgewählte VA verwendet. Der gesamte Lernvorgang läuft in zwei Phasen ab: in der ersten Phase wird der VQ an die auftretenden Eingangsvektoren  $i_k$  adaptiert, in der zweiten Phase erfolgt die Anpassung der Verteilungs-Approximationen. Für beide Phasen werden die Werte der Folge  $\{x_k\}$  als Lerndaten verwendet.

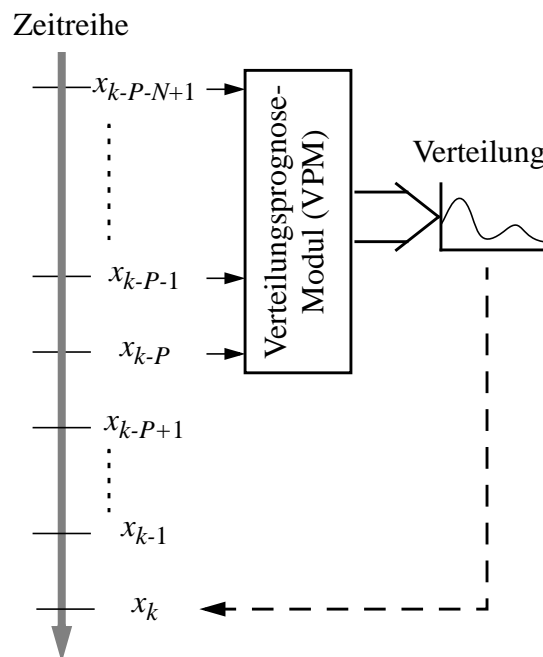
Bild 3.4 zeigt den Wiedergabevorgang des vorher trainierten Modells. In diesem Fall wird die Verteilung der VA, die durch den VQ aufgrund des momentanen Eingangsvektors ausgewählt wird, als Ausgangswert verwendet.

Wegen des Aufbaus und der Abläufe innerhalb des Modells kann die Zuständigkeit der Komponenten für die Adaption an unterschiedliche statistische Kenngrößen eindeutig angegeben werden: Der VQ ist für die Detektion von Korrelationen in der Zeitreihe zuständig, die VAs für die Verteilung.

### 3.3 Erweiterung für die Vorhersage mehrerer Schritte

Für manche Anwendungen ist eine Erweiterung des Modells für die Vorhersage von weiter in der Zukunft liegenden Verteilungen notwendig.

Für die Vorhersage der Verteilung eines um  $P$  Schritte in der Zukunft liegenden Wertes der Zeitreihe ( $P=1$  ist der bisher behandelte Normalfall) müssen die dann nicht in den Prognose-Algorithmus eingehenden Werte  $r_k = (x_{k-1}, \dots, x_{k-P+1})$  berücksichtigt werden. Der Eingangsvektor für das Prognosemodul wird dann  $i_k = (x_{k-P}, x_{k-P-1}, \dots, x_{k-P-N+1})$ .



**Bild 3.5:** Prognose für mehrere Schritte

Der Folgewert  $x_k$ , der Vektor  $\mathbf{r}_k$  und der Eingangsvektor  $\mathbf{i}_k$  werden zu einem  $(N+P)$ -dimensionalen Vektor  $\mathbf{x}_k = (x_k, \mathbf{r}_k, \mathbf{i}_k)$  mit einer  $(N+P)$ -dimensionalen Verteilungsfunktion zusammengefaßt.

Falls aufgrund der im VPM zugänglichen Information die ersten und zweiten Momente berechnet werden würden, müßte dies auf der Basis des Vektors  $\mathbf{x}_k' = (x_k, 0, \dots, 0, \mathbf{i}_k)$  erfolgen, da die tatsächlichen Werte für  $\mathbf{r}_k$  nicht bekannt sind. Dann sind Erwartungswert und Varianz der Zufallsvariablen  $X_{k-1}, \dots, X_{k-P+1}$  Null. Eine Grenzwertbetrachtung zeigt, daß für diesen Fall die Autokorrelationskoeffizienten nach Gl. (3.7) für  $i = 1, \dots, P-1$  ebenfalls den Wert Null haben.

Da der Prognosealgorithmus diese Werte implizit approximiert, haben die aufgrund der Prognose auftretenden Autokorrelationskoeffizienten  $\rho_1, \dots, \rho_R$  den Wert Null.

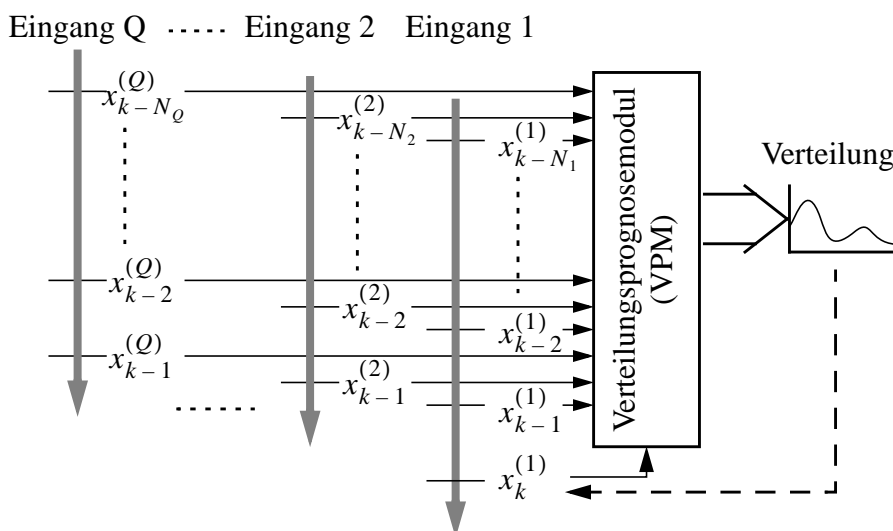
Für diese Erweiterung muß das Modell nur minimal geändert werden, s. Bild 3.5. Es ist eine Vergrößerung des Schieberegisters am Eingang von  $N$  auf  $N+P-1$  Plätze erforderlich, wobei nur die  $N$  letzten Plätze mit dem Vektor-Quantisierer verbunden sind.

### 3.4 Erweiterung für mehrere Eingänge

Die vorhergehenden Abschnitte behandeln den Fall *einer* Zeitreihe, für deren zukünftige Werte die Verteilung vorhergesagt werden soll. Häufig ist allerdings die Situation gegeben, daß ein stochastischer Prozeß nicht unabhängig von anderen Prozessen betrachtet werden kann. Im Fall von Zeitreihen spricht man hier von „multivariaten“ oder mehrwertigen Zeitreihen. Bei der Schätzung von Größen in der Regelungstechnik ist dieser Fall sogar der Normalfall, da hier eigentlich immer Schätzungen aufgrund von mehreren Meßgrößen erfolgen.

In diesem Abschnitt wird das bisher vorgestellte Prinzip erweitert, um die Adaption des Prognosemoduls an mehrere Eingangsgrößen zu ermöglichen.

Bild 3.6 zeigt die dafür geänderte Struktur des Modells für  $Q$  Eingangsgrößen  $\{x_k^{(1)}\}, \dots, \{x_k^{(Q)}\}$ . Anstatt eines einzigen Eingangsschieberegisters existiert nun für jeden Eingang ein getrenntes Schieberegister. Die Ausgänge aller Schieberegister sind mit dem Vek-



**Bild 3.6:** Prognose für mehrere Eingänge

tor-Quantisierer verbunden. Die Reihenfolge spielt dabei wegen des VQ-Algorithmus keine Rolle. Die Dimension des Eingangsraums des VQ ist dann  $N = N_1 + \dots + N_Q$ .

In der Regel wird der Wert  $x_k$  einer der Folgen als Eingangswert für den Lernprozeß der Verteilungs-Approximation verwendet – in Bild 3.6 von Folge  $\{x_k^{(1)}\}$ . Es ist aber auch möglich, hierfür die Werte einer Folge zu verwenden, die nicht zu den Eingangswerten des VQ beiträgt.

Die Erweiterungen für mehrere Eingänge und für die Vorhersage mehrerer Schritte sind beliebig kombinierbar.

## 3.5 Beispiele

Im folgenden wird die Funktionsweise des Verfahrens zur Verteilungsprognose anhand einiger Beispiele demonstriert. Die Beispiele in Abschnitt 3.5.1 basieren auf stochastischen Prozessen, diejenigen in Abschnitt 3.5.2 auf deterministischen Prozessen.

### 3.5.1 Stochastische Prozesse

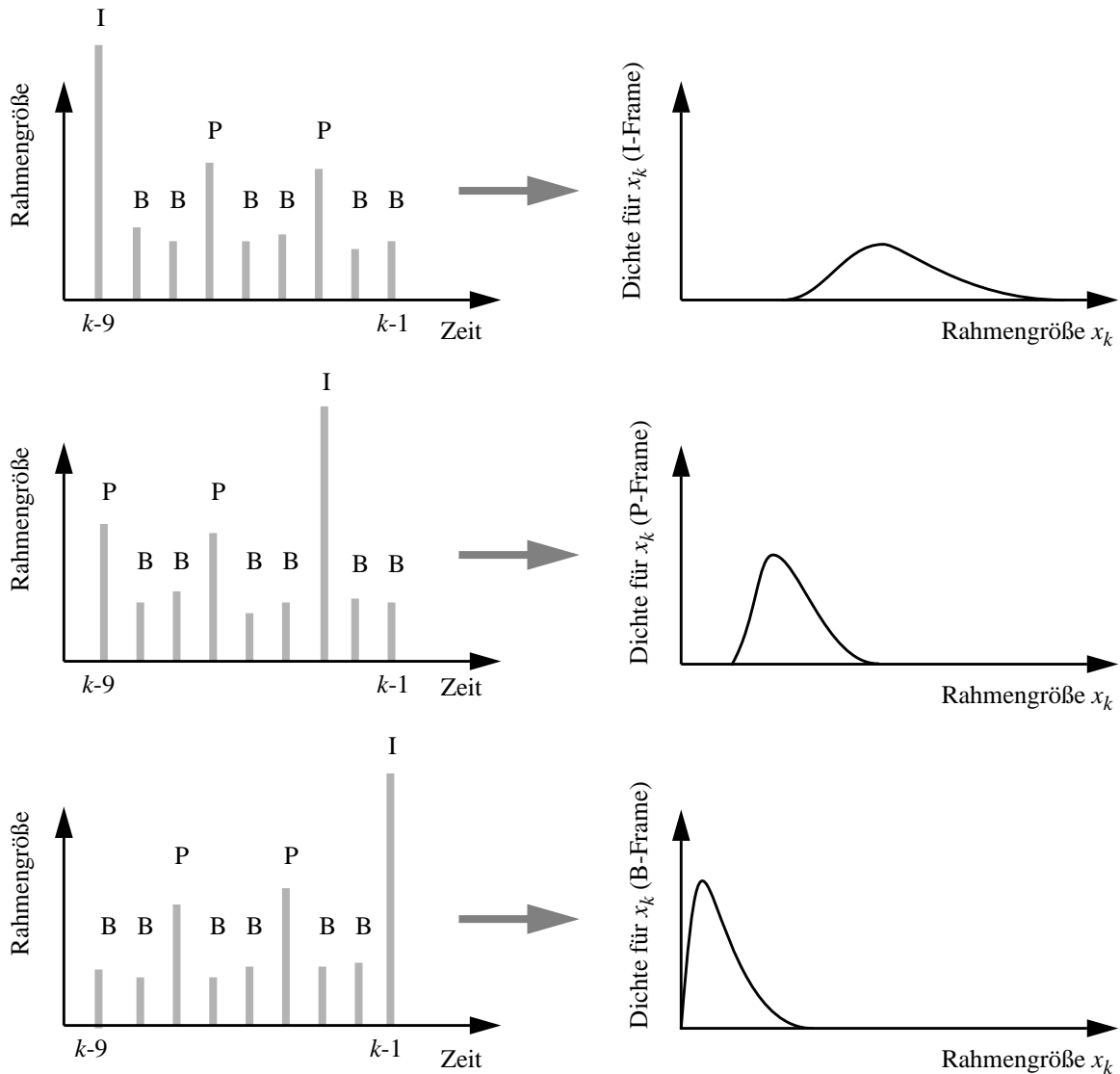
Für die Beispiele dieses Abschnitts werden MPEG-kodierte Videodaten des Kinofilms „Starwars“ als Eingangsdaten der Verteilungsprognose verwendet. MPEG ist ein standardisiertes Verfahren zur komprimierten Übertragung von Video- und Audiodaten [72]. Je nach MPEG-Parametern wird durch den Komprimierungsvorgang die Redundanz oder auch die Irrelevanz der Daten reduziert.

Die Werte der Folge  $\{x_k\}$  sind die Größen der in äquidistanten zeitlichen Abständen (40 ms) übertragenen MPEG-Rahmen. Jeder Wert  $x_k$  gibt die Größe eines Rahmens in Oktetts an. Für die MPEG-Parameter, die den „Starwars“- Daten zugrundeliegen, gibt es drei unterschiedliche Rahmentypen, die sich in ihrer Bedeutung und Größe unterscheiden: die größten Rahmen – sogenannte I-Rahmen – enthalten die komprimierte Information kompletter Bilder, die P-Rahmen mittlerer Größe und die kleinsten B-Rahmen enthalten Differenz- und vorausschauende Information. Die unterschiedlichen Rahmentypen treten je nach MPEG-Parametern in einer typischen und festen Sequenz auf, z. B. -I-B-B-P-B-B-P-B-B-, die sich zyklisch wiederholt. Diese Sequenz ist auf der linken Seite von Bild 3.7 gut erkennbar.

Die Stationarität einer MPEG-Zeitreihe hängt von einer Annahme über den Anfangszustand der Folge ab: Falls immer vom selben Start-Rahmentyp ausgegangen wird, ist die Folge nicht stationär, da die Verteilungen der Zufallsvariablen  $X_k$  der Zeitreihe dann nicht übereinstimmen. Falls dagegen von einem zufälligen, entsprechend der Größenverteilung aller MPEG-Rahmen verteilten Startwert ausgegangen wird, gilt die Annahme der Stationarität. Im folgenden wird von dieser Annahme ausgegangen, um die Voraussetzung in 3.2.2 zu erfüllen.

Das Verfahren funktioniert allerdings auch für zyklische Prozesse mit festen Startwerten, da diese eine Überlagerung eines stochastischen Prozesses mit einem deterministischen Prozeß darstellen und das Verfahren neben stochastischen Prozessen auch für deterministische Prozesse eingesetzt werden kann (s. Abschnitt 3.5.2).

Die Vorhersage von Verteilungen zukünftiger Werte hängt von der aktuellen Position im MPEG-Zyklus ab. Bild 3.7 zeigt verschiedene Möglichkeiten, wann die unterschiedlichen Rahmentypen als nächstes auftreten. Auf der linken Seite ist jeweils eine Momentaufnahme



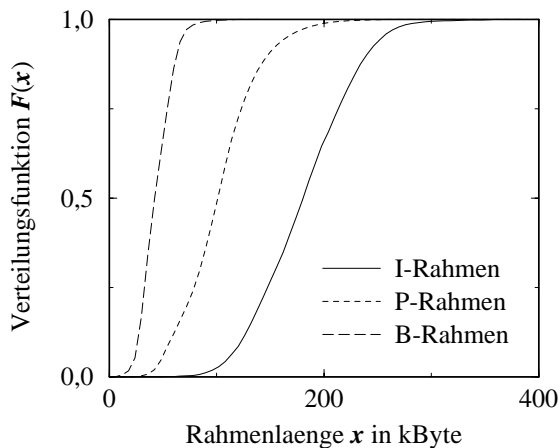
**Bild 3.7:** Beispiele für die Vorhersage von Verteilungsdichten von MPEG-Rahmen

der MPEG-Sequenz abgebildet und auf der rechten Seite die korrespondierende Verteilungsdichtefunktion der nächsten Rahmengröße (qualitativ). Die Verteilungsfunktionen für I-, P- und B-Rahmen, die aus den zum Lernen des Modells verwendeten Meßwerten ermittelt wurden, zeigt Bild 3.8. <sup>(1)</sup>

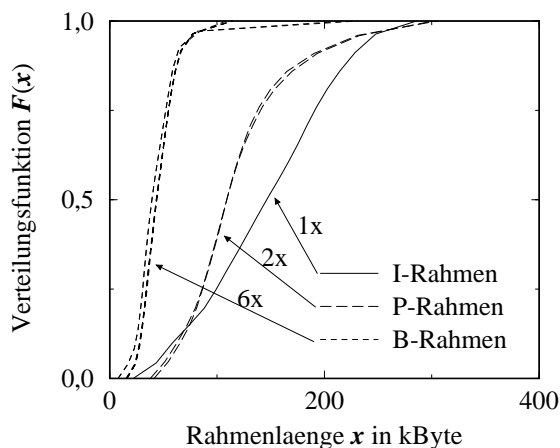
Im weiteren wird als Kurzschreibweise für ein Modell mit  $N$  VQ-Eingängen,  $M$  Verteilungsapproximationen,  $L$  Approximationsbereichen pro Verteilung und einer Vorhersage für  $P$  Schritte die Abkürzung  $N/M/L/P$  verwendet.

Wenn die beschriebene MPEG-Sequenz mit dem Verteilungsprognosemodul gelernt wird, ergeben sich die in Bild 3.9 dargestellten Verteilungsfunktionen. Für das Beispiel wurden folgende Parameter für die Verteilungsprognose verwendet:  $N=6$  VQ-Eingänge,  $M=9$  Verteilungsapproximationen,  $L=20$  Bereiche pro Verteilungsapproximation und eine Vorhersage-

(1) In Bild 3.8 ff. werden Verteilungsfunktionen dargestellt, da dort die Gruppenbildung für die Rahmentypen deutlicher sichtbar ist als bei Dichtefunktionen.



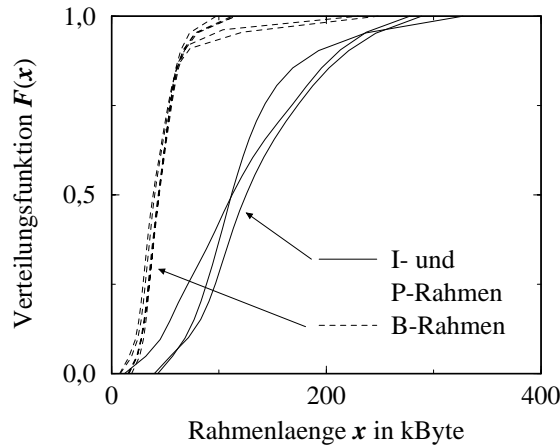
**Bild 3.8:** Tatsächliche Verteilungsfunktionen der MPEG-Rahmengrößen



**Bild 3.9:** Gelernte Verteilungsfunktionen der MPEG-Rahmengrößen für ein 6/9/20/1-Modell

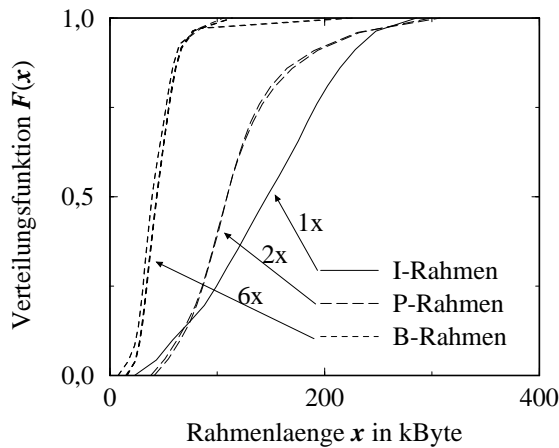
weite von  $P=1$  (6/9/20/1-Modell). Deutlich sichtbar sind unterschiedliche Gruppen von Verteilungsfunktionen, die analog zu Bild 3.7 je nach Vergangenheit der MPEG-Sequenz Gültigkeit haben. Man erkennt weiterhin, daß für die B-Rahmen deutlich mehr Verteilungen (6 Stück) gelernt wurden, als für die P-Rahmen (2 Stück), während nur eine Verteilung für I-Rahmen existiert. Dies ist das Resultat der VQ-Eigenschaft, den Eingangsraum in Gebiete gleicher Wahrscheinlichkeit zu unterteilen. Dadurch wird hier für jede der möglichen und gleich häufigen Kombinationen aus vergangenen Rahmentypen genau eine Verteilung gelernt: 6 Verteilungen für B-Rahmen, 2 Verteilungen für P-Rahmen und 1 Verteilung für I-Rahmen. Die Differenzen zwischen den Bildern 3.8 und 3.9 sind auf die Modellparameter zurückzuführen und werden im folgenden diskutiert.

Die gewählten Parameter stellen die minimale Konfiguration der Verteilungsprognose für das MPEG-Beispiel dar. Für weniger Eingänge des VQ (kleineres  $N$ ) kann die Sequenz nicht mehr reproduziert werden. Für  $N=4$  können z. B. die I- und P-Rahmen nicht mehr unterschieden werden. Bild 3.10 zeigt die entsprechenden Verteilungsfunktionen, die nicht mehr die klare Gruppenstruktur zeigen wie oben.

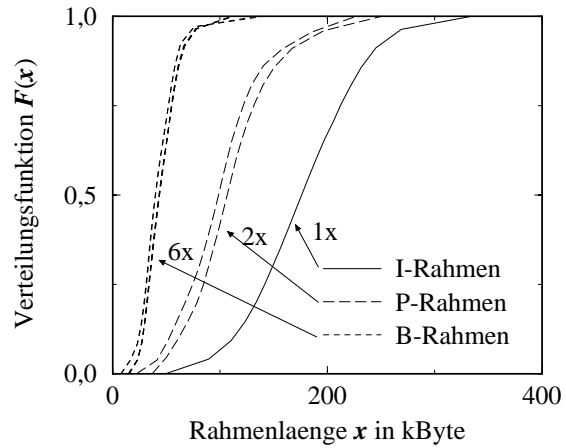


**Bild 3.10:** Gelernte Verteilungsfunktionen für zu kleines Gedächtnis (4/9/20/1-Modell)

An der MPEG-Sequenz kann auch die Vorhersage für mehrere Schritte in die Zukunft gut demonstriert werden. Falls beispielsweise die Rahmengröße für 2 oder 5 Schritte in der Zukunft vorhergesagt werden soll, ergeben sich die Verteilungsfunktionen in den Bildern 3.11 und 3.12. Wie in Bild 3.9 ergeben sich drei Gruppen von Verteilungsfunktionen für I-, P- und B-Rahmen. Anhand der Unterschiede zwischen den Bildern 3.11 und 3.12 werden die Eigenschaften der Verteilungsprognose im folgenden weiter erläutert. Diese Unterschiede – insbesondere eine klarere Trennung der Verteilungsfunktionen für P- und I-Rahmen in Bild 3.12 – können durch die Art und Anzahl der vergangenen Werte am VQ-Eingang erklärt werden: In den Fällen, in denen einer der 1. bis 3. zukünftigen Werte ein I-Rahmen ist, kommt in der Sequenz am VQ-Eingang kein I-Rahmen vor. Da I- und P-Rahmen nicht immer eindeutig aufgrund ihrer Größe unterscheidbar sind, kann in diesen Fällen durch den VQ nicht immer die korrekte Sequenz detektiert werden. Im Fall einer falschen Detektion führt dies zur Auswahl der falschen Verteilungsapproximation für den Lernvorgang. Als Resultat wird die Verteilung der I-Rahmen durch fehlerhaft zugeordnete P-Rahmen verfälscht, was zu den entsprechenden Kurven in den Bildern 3.9 und 3.11 führt. Im Gegensatz dazu ist in allen Fällen, in denen einer



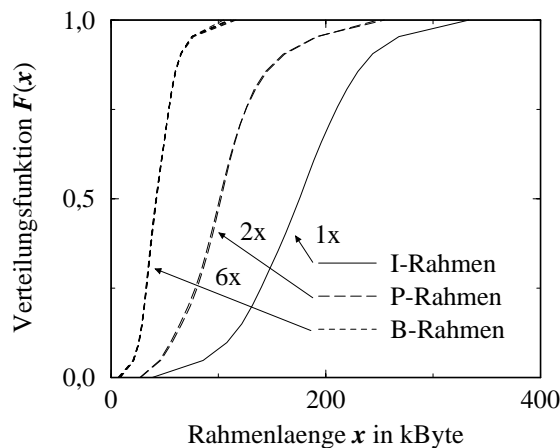
**Bild 3.11:** Gelernte VF für Vorhersage von 2 Schritten (6/9/20/2-Modell)



**Bild 3.12:** Gelernte VF für Vorhersage von 5 Schritten (6/9/20/5-Modell)

der 4. bis 9. zukünftigen Werte ein I-Rahmen ist, auch ein I-Rahmen in der Sequenz am Eingang des VQ enthalten, was zu der besseren Trennung der Kurven in Bild 3.12 führt.

Der beschriebene Effekt bei der Anpassung der Verteilungsprognose an einen MPEG-Datenstrom für unterschiedliche Werte von  $P$  tritt nicht auf, wenn die Anzahl  $N$  der vergangenen Werte, die zur Bestimmung einer Verteilungsapproximation herangezogen werden, auf 9 erhöht wird <sup>(1)</sup>. Dann kann der VQ praktisch immer die korrekte Verteilung auswählen. Die in diesem Fall für die verschiedenen Rahmentypen klar getrennten und mit den Verteilungsfunktionen des Originals in Bild 3.8 sehr gut übereinstimmenden Verteilungsfunktionen zeigt Bild 3.13.



**Bild 3.13:** Gelernte Verteilungsfunktionen bei 9 Eingängen (9/9/20/1-Modell)

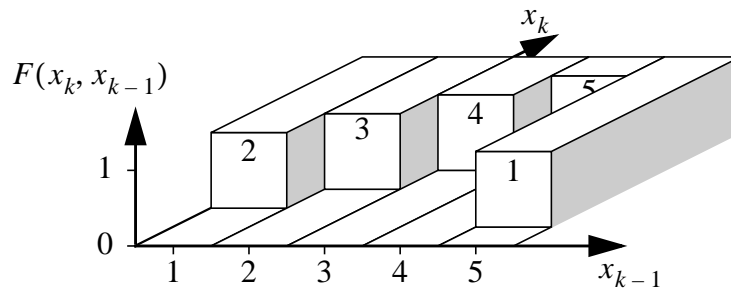
### 3.5.2 Deterministische Prozesse

In diesem Abschnitt wird gezeigt, daß die Verteilungsprognose auch für deterministische Prozesse, die im Sinne der Definition nach Abschnitt 2.2.1 nicht stationär sind, verwendet werden kann.

Der deterministische Prozeß, der hier zur Demonstration verwendet wird, besteht aus der zyklisch wiederholten Zahlenfolge 1-2-3-4-5. Für eine bestimmte Vergangenheit liegt also genau fest, welcher Wert als nächstes kommen wird, daher genügt hier ein Gedächtnis der Größe  $N=1$ . Der Determinismus des nächsten Werts drückt sich in der bedingten Verteilungsdichte  $\tilde{f}$  als Dirac-Distribution aus. Die Verbundverteilungsfunktion nach Gl. (3.2) läßt sich in diesem Fall grafisch veranschaulichen (s. Bild 3.14). Für die Werte 1-5 für  $x_{k-1}$  ergibt sich jeweils eine Schrittfunktion über  $x_k$  von 0 auf 1, deren Schritt an folgenden Stellen erfolgt:

$$x_k = \begin{cases} x_{k-1} + 1 & \text{für } k = 1, 2, 3, 4 \\ 1 & \text{für } k = 5 \end{cases} \quad (3.16)$$

(1) Dieser Wert entspricht der Zykluslänge 9 der MPEG-Daten und stellt üblicherweise die erste Wahl für den Parameter  $N$  dar.



**Bild 3.14:** Gelernte Verbundverteilungsfunktion für eine deterministische 1-2-3-4-5-Folge

Das verwendete Modell hat die Parameter  $N=1$ ,  $M=5$ ,  $L=3$ ,  $P=1$ . Die erste Stufe der Verteilungsprognose, der Vektor-Quantisierer, lernt, zwischen den Werten 1-5 zu unterscheiden, die zweite Stufe, die Verteilungsapproximation, lernt die fünf unterschiedlichen bedingten Dichtefunktionen  $\hat{f}$ . Die Verteilungsprognose lernt das Verhalten des deterministischen Prozesses so gut, daß eine graphische Gegenüberstellung der gelernten Verteilungsfunktion mit der theoretischen Verteilungsfunktion keine Unterschiede zeigt.

### 3.6 Einsatzgebiete

Für das Verfahren der Verteilungsprognose gibt es vielfältige Einsatzgebiete, von denen ein Teil in diesem Abschnitt dargestellt wird. Prinzipiell stellt sich die Frage, auf welche Weise die vorhergesagten Verteilungen weiterverarbeitet werden sollen. Hierfür gibt es z. B. die folgenden Möglichkeiten:

- Ausgehend von der Verteilung kann ein Zufallswert bestimmt werden. Diese Möglichkeit kann in der Zeitreihenmodellierung oder zur Quellmodellierung (s. Kapitel 5) eingesetzt werden.
- Die Verteilung kann zur Berechnung von statistischen Kenngrößen wie Quantilen, Momenten, etc. herangezogen werden. Die Weiterverarbeitung dieser Kenngrößen kann auf sehr unterschiedliche Weise erfolgen. Ein Beispiel für die Verbesserung der Effizienz eines Regelalgorithmus wird in Kapitel 6 gezeigt.

#### 3.6.1 Einsatz zur Zeitreihenmodellierung

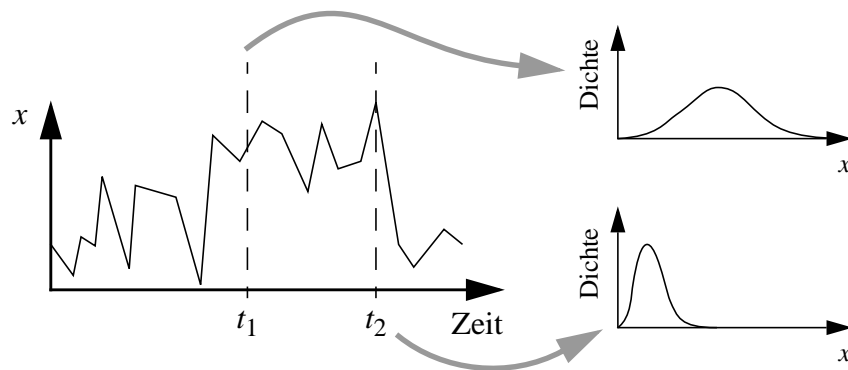
Eine naheliegende Weise, die vorhergesagten Verteilungen weiterzuverarbeiten, ist die Bestimmung eines Zufallswerts zur Nachbildung einer Zeitreihe. Damit ist der Aufbau eines Modells für eine beobachtete Zeitreihe möglich, das Werte erzeugt, deren stochastisches Verhalten dem der Originalreihe entspricht. Eine spezielle Anwendung der Zeitreihenmodellierung ist die Modellierung von Verkehrsquellen. Die zu modellierenden stochastischen Prozesse sind hier Signal- oder Datenquellen der Datenverarbeitung und der Kommunikationstechnik. In Kapitel 5 werden ausführliche Beispiele zu diesem Themengebiet gezeigt und die Leistungsfähigkeit des Verfahrens untersucht.

Eine weitere Verarbeitungsmöglichkeit der vorhergesagten Verteilungen ist die Auswertung der Verteilung im Hinblick auf ihre statistischen Kenngrößen wie Quantile, Erwartungswert



oder Varianz. Dies kann z. B. bei der Vorhersage von Börsen- oder Devisenkursen gegenüber traditionellen Zeitreihenmodellen zusätzliche Informationen über die Streuung der zu erwartenden Werte liefern. Über die Varianz der prognostizierten Verteilung ist dann eine Aussage über die Sicherheit des Mittelwerts möglich, was wiederum Vorhersagen über das Risiko von Spekulationsgeschäften erlaubt. Derartige Entscheidungen können z. B. mit Hilfe unscharfer Logik (Fuzzy Logic) getroffen werden.

Bild 3.15 enthält ein Beispiel für eine Zeitreihe, bei der die Verteilungsprognose zu unterschiedlichen Zeitpunkten aufgrund der unterschiedlichen Varianzen verschieden sichere Aussagen über die weitere Entwicklung der Zeitreihe zuläßt. Die Vorhersage zum Zeitpunkt  $t_1$  ist aufgrund der großen Varianz der Verteilung relativ unsicher; die Vorhersage zum Zeitpunkt  $t_2$  erlaubt dagegen zuverlässigere Aussagen über die Entwicklung von  $x$ .



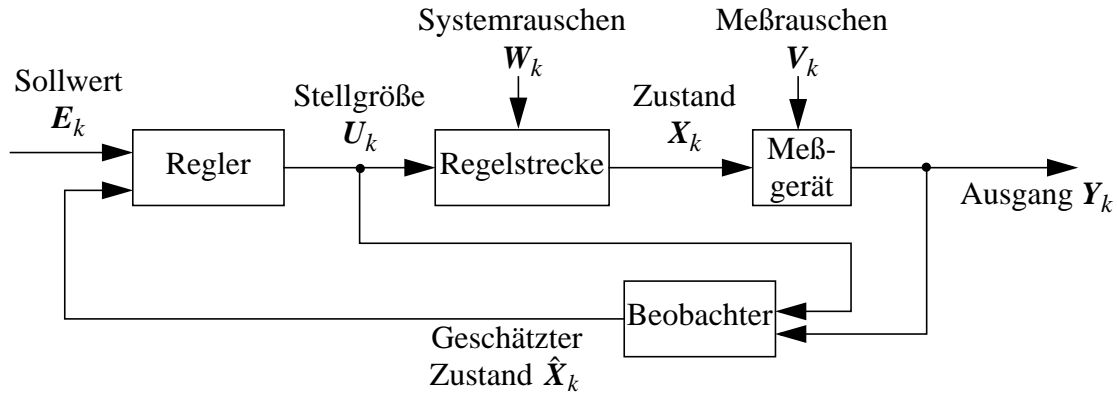
**Bild 3.15:** Verteilungen unterschiedlicher Varianz

### 3.6.2 Einsatz in Regelsystemen

Ein weiteres Gebiet, auf dem die Verteilungsprognose gewinnbringend eingesetzt werden kann, ist die Regelungstechnik.

Die Regelungstechnik befaßt sich mit Analyse und Modellierung von Regelstrecken und der Synthese von Reglern. Bild 3.16 zeigt ein Beispiel für einen Regelkreis mit zeitdiskretem Regler, wobei alle Größen für den allgemeinen Fall als Vektoren angegeben werden. Auf die Darstellung von Abtast- und Haltegliedern wurde aus Gründen der Übersichtlichkeit verzichtet. Die Regelstrecke umfaßt den Teil eines dynamischen Systems, der durch geeignete Werte der Stellgröße  $U_k$  in seinem Verhalten so gesteuert werden soll, daß ein definiertes Regelziel erreicht wird. Ziel des Reglers ist es, den Ausgangsvektor  $Y_k$  über die Stellgröße so einzustellen, daß seine Abweichung vom Sollwertvektor  $E_k$  minimal wird. Als Nebenbedingungen kommen häufig Anforderungen an die Einregelzeit und eine Begrenzung der Stellenergie hinzu. Die Regelziele lassen sich durch eine Kostenfunktion  $\gamma(k, X_k, U_k, W_k)$  ausdrücken, die durch einen konkreten Regelalgorithmus durch Erzeugung einer Folge von Stellgrößen minimiert werden soll. Je nach Art der Regelstrecke und der Störeinflüsse auf Aktoren und Sensoren wird ein deterministischer oder ein stochastischer Regleransatz gewählt.

Sowohl für deterministische als auch für stochastische Regelungen wird, falls nicht alle Zustandsgrößen  $X_k$  meßbar sind, üblicherweise ein Beobachteransatz gewählt. Der Beobachter schätzt, ausgehend von Vorwissen über die Systemparameter, die fehlenden Zustandsgrößen.



**Bild 3.16:** Blockschaltbild für Regelkreis mit Beobachter

Zustandsgrößen können beispielsweise nicht meßbar sein, weil es sich um systeminterne Größen handelt, die nicht in Form einer physikalischen Größe meßbar sind oder sich die Erfassung einer Größe aufgrund der Kosten für ein Meßgerät oder einen Sensor verbietet. Ein weiterer Grund für den Einsatz eines Beobachters können nur fehlerbehaftet meßbare Systemgrößen sein. Dies wird in Bild 3.16 durch den Störgrößenvektor  $V_k$  bei der Messung der zugänglichen Zustandsgrößen  $X_k$  berücksichtigt.

Im Kontext dieser Arbeit ist die Regelung stochastischer Systeme interessant. Die stochastische Natur solcher Regelsysteme tritt auf durch nicht beschreibbare oder im Modell nicht berücksichtigte „zufällige“ Vorgänge, verrauschte Meßwerte, etc. Die Modellierung von Systemgrößen erfolgt dann durch stochastische Prozesse.

Im Beispiel nach Bild 3.16 wird der Zustand  $X_k$  wegen des Vorhandenseins der Rauschsequenz  $\{W_k\}$  zu einem Zufallsvektor.  $X_{k+1}$  hängt wie im deterministischen Fall von  $k$ ,  $X_k$  und  $U_k$  ab. Der Beobachter wird für stochastische Systeme häufig als Kalman-Filter realisiert [39].

Die Einsatzmöglichkeiten der Verteilungsprognose in stochastischen Regelsystemen zur Vorhersage zukünftiger oder zur Schätzung aktueller Werte und ihrer statistischen Kenngrößen sind vielfältig. So können beispielsweise in einem Beobachter nicht zugängliche oder verrauschte Zustandswerte geschätzt werden. Dabei ist durch Auswertung der Verteilung zusätzlich eine Bewertung der Streuung des Werts möglich. In einem Regler könnte durch Vorhersage der Verteilung zukünftiger Werte eine effizientere Regelung bei besseren Kosten erreicht werden, indem nicht nur der momentane Zustand und die Vergangenheit berücksichtigt werden, sondern Informationen über die Zukunft in Form von Wahrscheinlichkeiten, Momenten, etc. in den Algorithmus einbezogen werden. Derartige zukünftige Größen können mit konventionellen Beobachtern oder Kalman-Filtern nicht geschätzt werden.

Methoden der Regelungstechnik werden neben ihren klassischen Einsatzgebieten in der Automatisierungs- und Fertigungstechnik auch in anderen Gebieten wie der Kommunikationstechnik eingesetzt. Die Einsatzmöglichkeiten in der Kommunikationstechnik liegen häufig in Anwendungen der Überlastvermeidung und -abwehr, bei denen ein Teil eines Kommunikationssystems, die Regelstrecke, durch einen Regler in einem optimalen Betriebszustand gehalten werden soll. Anwendungsbeispiele hierfür sind Verbindungsannahme-Algorithmen (optimale Ressourcenauslastung bei gleichzeitiger Einhaltung von Dienstgütegarantien),

Feldstärkeregelung in Funknetzen oder optimale Wahl von Protokollparametern wie Fenstergrößen in Transportprotokollen zur Durchsatzoptimierung.

### **3.7 Bewertung des Verfahrens**

Mit dem vorgestellten Verfahren existiert die Möglichkeit, die Verteilung unterschiedlichster Zeitreihen aufgrund ihrer Vergangenheit mit beliebiger Genauigkeit vorherzusagen. Die Stärke des Verfahrens liegt auf der Vorhersage der Eigenschaften von stochastischen Prozessen, allerdings ist auch die Vorhersage des Verhaltens von periodischen deterministischen Prozessen möglich. Für die Anwendung des Verfahrens existieren nur wenige Einschränkungen für folgende Typen von Zeitreihen:

- Instationäre Prozesse im allgemeinen mit folgenden Ausnahmen: Die Adaption an Prozesse mit festen Zyklen ist möglich, da diese durch das Modell erlernbar sind. Weiterhin ist die Adaption an bestimmte trendbehaftete Prozesse möglich, indem die Trends durch Differenzenbildung analog zum Vorgehen bei ARIMA-Modellen für Zeitreihen (s. Abschnitt 2.4.1) eliminiert werden.
- Chaotische Prozesse, da diese zwar ein stochastisches Verhalten zu haben scheinen, tatsächlich aber einer deterministischen Abbildungsvorschrift gehorchen (s. Abschnitt 2.4.3). Außerdem bestehen für chaotische Prozesse Einschränkungen bezüglich der Stationarität.

Das Verfahren basiert auf zwei unterschiedlichen, für diesen Zweck entworfenen künstlichen neuronalen Netzen. Aufgrund der Lernfähigkeit von KNNs ermöglicht dies die automatische Anpassung der Verteilungsprognose an das stochastische Verhalten und den Wertebereich eines stochastischen Prozesses. Diese Anpassung erfolgt automatisiert in einem einzigen Durchlauf und erfordert keine vorangehende Vorverarbeitung der Werte. Dadurch ist auch die Anpassung an Prozesse in Echtzeit möglich. Aufgrund dieser Eigenschaften und nur weniger Parameter, die vorab gewählt werden müssen, gestaltet sich der Einsatz dieses Verfahrens sehr einfach.

## **Kapitel 4**

# **Methodische Grundlagen von Modellierung und Leistungsuntersuchung**

Die Untersuchung dynamischer Systeme zur Gewinnung von Einsichten in das Systemverhalten oder zur Bestimmung von Leistungsdaten kann sowohl analytisch als auch simulativ erfolgen. Vor der Analyse oder Simulation muß durch geeignete Abstraktion des realen Systems ein Modell gebildet werden, das anschließend mit den Mitteln von Analyse bzw. Simulation untersucht wird.

Ist für eine bestimmte Abstraktion eine Analyse möglich, wird sie in der Regel der Simulation vorgezogen, da sie umfassendere Aussagen und Ergebnisse liefern kann. Häufig lassen sich allerdings nur relativ kleine oder vereinfachte Systeme vollständig durch analytische Methoden untersuchen, wogegen mit Simulationen auch die detaillierte Untersuchung großer und komplexer Systeme möglich ist. Ein häufiger Anwendungsfall von Simulationen ist die stichprobenhafte Überprüfung von Analyseergebnissen; ihre Beschränkung liegt in der benötigten Rechenzeit. Durch die zunehmende Leistungsfähigkeit von Digitalrechnern ist die Grenze zwischen Analyse und Simulation nicht mehr so eindeutig wie früher. Die Folge ist, daß auch in Analysen komplexe numerische Lösungsverfahren Anwendung finden und Teile von Simulationen zur Reduzierung von Rechenzeit durch analytische Modelle ersetzt werden (hybride Ansätze).

In Kapitel 3 dieser Arbeit wird ein Verfahren eingeführt, das die Vorhersage von Verteilungen für stochastische Prozesse erlaubt. Dieses Verfahren läßt sich auf verschiedenen Gebieten auf unterschiedliche Weise einsetzen. Da sich in den vielen, zum Teil stark unterschiedlichen technischen Disziplinen sehr unterschiedliche Analyse- und Simulationsverfahren durchgesetzt haben, die häufig nur auf dem jeweiligen Spezialgebiet sinnvoll einsetzbar sind, ist eine Aussage über die Eignung dieser Verfahren zur Berücksichtigung der Verteilungsprognose notwendig.

In den folgenden Abschnitten werden daher einige Verfahren kurz vorgestellt, ihre Einsatzgebiete charakterisiert und eine Bewertung bezüglich ihrer Eignung zur Berücksichtigung des Prognoseverfahrens durchgeführt.

### **4.1 Analysemethoden**

#### **4.1.1 Warteschlangentheorie**

Die Warteschlangentheorie entstand vor dem Hintergrund immer komplexer werdender technischer Systeme wie Kommunikations- oder Fertigungssysteme, die zu dimensionieren, auszula-

sten und in ihren Abhängigkeiten zu überblicken ohne formale Hilfsmittel nicht mehr möglich war. Zu diesem Zweck wurden Modellierungsmittel und Analysemethoden entwickelt, die es erlauben, komplexe Vorgänge zu modellieren, Ausfallzeiten und Alternativen zu berücksichtigen und auf formale Weise Ergebnisse zu liefern.

Die Modellierung von Kommunikationsnetzen und Datenverarbeitungsanlagen erfolgt durch die in Tabelle 4.1 angegebenen sowie durch weitere, ggf. spezialisierte Modellierungskomponenten [69]. Die betrachteten Prozesse sind in der Regel stochastischer Natur, weswegen auch Analyseergebnisse meist in Form von Wahrscheinlichkeiten, Verteilungen oder deren Momenten vorliegen. Mit Hilfe der Komponenten ist die Modellierung komplexer Systeme unterschiedlichsten Abstraktionsgrads möglich, die zeitdiskret oder zeitkontinuierlich, wertdiskret oder wertkontinuierlich sein können. Die Modellbildung erfolgte bis vor einigen Jahren meist zeitkontinuierlich und wertdiskret; für diesen Fall existieren die meisten Ergebnisse. In jüngerer Zeit werden für die Betrachtung getakteter Systeme zunehmend auch zeit- und wertdiskrete Modelle wichtig.

**Tabelle 4.1:** Wesentliche Modellierungskomponenten der Warteschlangentheorie <sup>(a)</sup>

| Komponente                                 | Funktion  | Parameter  |
|--|---|--|
| Verkehrsquelle                             | Erzeugung von Anforderungen   | Ankunftsabstand der Anforderungen (Beschreibung durch Mittelwert, Verteilung, Zustandsmodelle, etc.)   |
| Bedieneinheit                              | Bearbeitung von Anforderungen   | Bedienzeit (Beschreibung z. B. durch eine Verteilung),<br>Anzahl parallel verarbeitbarer Anforderungen |
| Wartespeicher endlicher Größe              | Zwischenspeicherung von Anforderungen, Abweisung von Anforderungen bei vollem Wartespeicher | Anzahl der Warteplätze,<br>Prioritätsmechanismus für unterschiedliche Anforderungstypen                |
| Wartespeicher mit unendlich vielen Plätzen | Zwischenspeicherung von Anforderungen   | Prioritätsmechanismus für unterschiedliche Anforderungstypen   |
| Aufspaltung                                | Vervielfältigung einer Anforderung zur Weiterleitung auf mehrere Wege                       | Anzahl der Wege  |
| Verzweigung                                | Verzweigung von Anforderungen auf mehrere Wege  | Verzweigungswahrscheinlichkeit für jeden Weg   |
| Zusammenführung                            | Zusammenführung von Anforderungen unterschiedlichen Ursprungs                               | Reihenfolge bei gleichzeitiger Ankunft mehrerer Anforderungen  |

(a) In Tabelle 4.2 auf Seite 49 werden Sinnbilder für die Komponenten angegeben.

Wesentliche Merkmale, die die Modellierung mit Warteschlangennetzen von anderen Modellierungsarten abheben, sind die inhärente Nichtlinearität sowie die einfachen Möglichkeiten zur Berücksichtigung von Verzögerungen. Dadurch können Systeme modelliert werden, die anderen Methoden unzugänglich sind.

Die Modellierung liefert entweder einstufige Warteschlangen-Bediensysteme oder netzartig verkoppelte Bediensysteme. Zur Analyse werden stochastische Zustandsvariable (z. B. Warteschlangenlänge) eingeführt, Beziehungen zwischen stochastischen Variablen hergestellt und in entsprechende Beziehungen zwischen ihren statistischen Kenngrößen (Verteilungsfunktion, Verteilungsdichtefunktion, Momente, usw.) transformiert. Mathematisch führen diese Beziehungen abhängig von der Art des betrachteten Systems auf Zustandsgleichungen in Form von Differentialgleichungen, Integralgleichungen oder lineare Gleichungssysteme [66, 69].

Die Analysemethoden der Warteschlangentheorie sind sehr gut geeignet, Aussagen über das Verhalten von kleineren Systemen oder Teilsystemen zu machen. Für größere Systeme sind ebenfalls geschlossene Lösungen möglich, hier gelten jedoch Einschränkungen, z. B. bezüglich der für die Modellierung erlaubten Komponenten und Verkehrsquellen.

### **4.1.2 Zeitreihenanalyse**

Das Ziel der Zeitreihenanalyse ist, wie der Name schon sagt, die Untersuchung und Modellierung von Zeitreihen (s. Abschnitt 2.4). Dabei beschränkt man sich in der Regel auf die Untersuchung von Stichproben und verzichtet auf die Herleitung von Modellen für die Systeme, die eine konkrete Zeitreihe generieren. Das liegt daran, daß auf den klassischen Einsatzgebieten der Zeitreihenanalyse (Vorhersage von Ökonomiedaten, Wetterdaten, etc.) über das System, das diese Daten generiert, sehr wenig bekannt ist. Das Ziel der Zeitreihenanalyse ist daher im Gegensatz zu anderen Analysemethoden nicht die Herleitung eines detaillierten Systemmodells, sondern es werden allein aufgrund von beobachteten Werten und deren statistischen Kenndaten Aussagen über den Prozeß gemacht, der diese Werte generiert und ggf. einfache Modelle hergeleitet. Daher überlappen sich die Einsatzbereiche der Zeitreihenanalyse nur schwach mit denen der anderen hier behandelten Analysemethoden.

### **4.1.3 Methoden der Regelungstechnik**

Die Regelungstechnik beschäftigt sich mit Methoden zur Analyse und Modellierung von Regelsystemen sowie mit der Auslegung von Reglern. Im Zusammenhang mit Systemen, die aufgrund ihres überwiegend stochastischen Charakters den Einsatz der klassischen Regelungstechnik für deterministische Systeme nicht zulassen, spricht man von „stochastischer Regelungstechnik“.

Das Ziel einer regelungstechnischen Aufgabenstellung besteht normalerweise in der Ermittlung eines für eine bestimmte Regelstrecke und unter Kostenaspekten optimalen Reglers. Die hier eingehenden Kosten können Investitionen für den realen Regler sein, aber auch Stellenergie, Regelzeit, usw.

Die Modellierung führt häufig auf ein Modell wie in Bild 3.16. Dieses Modell ist für viele regelungstechnische Aufgabenstellungen gültig. Zur Analyse werden Zustandsgrößen der Regelstrecke identifiziert (z. B. Lage, Geschwindigkeit, usw.) und Beziehungen zwischen die-

sen Größen hergestellt. Je nach Art des betrachteten Systems werden Lösungen für die Zustandsgleichungen sowie optimale Regelalgorithmen im Zeit- oder Frequenzbereich gesucht. Im Fall der klassischen Regelungstechnik führt dies häufig auf Differential- oder Differenzgleichungen, die unter den gegebenen Randbedingungen (Kosten, Regelzeit, etc.) gelöst werden.

Die meisten Methoden der Regelungstechnik zur Systemanalyse und -modellierung sind zur Untersuchung linearer oder schwach nichtlinearer Systeme geeignet. Allgemeine nichtlineare Systeme sind nur selten geschlossen analysierbar. Ein häufig verfolgter Ansatz zur Umgehung dieser Problematik ist die Linearisierung um einen Arbeitspunkt, wodurch wieder der Einsatz klassischer Techniken möglich wird. Da dies eine starke Abstraktion des realen Systemverhaltens darstellt, sind für starke Anregungen des Systems ggf. genauere Betrachtungen bezüglich der Stabilität erforderlich.

Im Kontext dieser Arbeit ist primär die Regelung stochastischer Systeme interessant. Die stochastische Natur solcher Regelsysteme zeigt sich durch nicht beschreibbare oder im Modell nicht berücksichtigte „zufällige“ Vorgänge, verrauschte Meßwerte, etc. Die Modellierung von Systemgrößen erfolgt hier durch stochastische Prozesse.

#### **4.1.4 Perturbationstheorie**

Die Perturbationstheorie [44] dient zur Bestimmung des Verhaltens dynamischer Systeme, wobei keine analytische Beschreibung des Systems vorausgesetzt wird. Sie basiert auf der Beobachtung von Reaktionen eines Systems oder eines Systemmodells auf kleinste Änderungen von Eingangswerten oder Parametern. Durch Interpretation der Systemreaktion wird auf das grundsätzliche Systemverhalten geschlossen. Diese Methode kann beispielsweise für Parameterstudien oder für die Bestimmung von Übertragungsfunktionen eingesetzt werden.

Die Perturbationstheorie ist eine Kombination aus Simulation und Analyse. Dabei werden Simulationen für unterschiedliche, sich nur sehr wenig unterscheidende Eingangswerte oder Parameter durchgeführt. Die Simulationsergebnisse werden anschließend durch die Analyse geeignet aufbereitet und extrapoliert. Das Resultat dieser Analyse dient wiederum zur Wahl neuer Simulationsparameter. Damit ist beispielsweise bei der Suche nach Optima für bestimmte Systemparameter eine Verminderung der Anzahl von Simulationsläufen zur Bestimmung der Parameter möglich, da aus der Analyse Hinweise auf die richtige „Richtung“ der Parametermodifikation gewonnen werden können.

Durch Methoden der Perturbationstheorie können sowohl der eingeschwungene Zustand als auch transiente Vorgänge komplexer dynamischer Systeme untersucht werden.

Der einfachste Ansatz setzt voraus, daß die genannten Parameteränderungen keine Änderungen der Ereignis- und Zustandsfolge des Systems zur Folge haben. Dies bedeutet eine Einschränkung, denn es wird eine gewisse Robustheit des Systems vorausgesetzt. Durch die genannte Einschränkung ist diese Art der Perturbationsanalyse unter anderem nicht für chaotische Systeme geeignet, da die Voraussetzung der Robustheit gegenüber kleinen Parameterschwankungen dort nicht gilt. Der einfache, häufig ungenügende Ansatz kann dahingehend erweitert werden, daß auch Systeme untersucht werden können, bei denen kleinste Parameteränderungen zu Änderungen in der Ereignis- und Zustandsfolge führen. Diese Erweiterung

führt allerdings zu einem wesentlich höheren Aufwand, da aufwendige Vergleiche der entstehenden Ereignis- und Zustandsfolgen notwendig werden.

### **4.1.5 Bewertung**

In Abschnitt 3.6 wurden einige potentielle Einsatzgebiete für die Verteilungsprognose angegeben und diskutiert. Auf diesen Gebieten haben sich individuelle Analyse- und Modellierungstechniken etabliert, die in den vorstehenden Abschnitten kurz charakterisiert wurden.

Die beschriebenen Analyse- und Modellierungstechniken sind mehr oder weniger geeignet, Systeme zu analysieren und zu modellieren, in denen die Verteilungsprognose eingesetzt wird. Daher erfolgt in diesem Abschnitt eine Abgrenzung der Analysemethoden gegeneinander und eine Bewertung ihrer Eignung zur Einbeziehung der Verteilungsprognose. Prinzipiell stellt sich die Frage, ob die jeweilige Methode eine Funktion in ihren Formalismus einbeziehen kann, die vergangene Folgewerte auf die Verteilung eines zukünftigen Folgewerts abbildet.

In der Zeitreihenanalyse und in der stochastischen Regelungstechnik werden üblicherweise normalverteilte Zufallsvariablen, die durch Erwartungswert und Varianz ausreichend beschrieben sind, zur Modellierung von Systemgrößen verwendet; die genauere Charakterisierung durch Angabe einer Verteilung erfolgt nicht. Beide Gebiete zeichnen sich durch einen ausgeprägten Formalismus aus, der zur Berücksichtigung der erwähnten Abbildungsvorschrift geeignet erweitert werden muß. Diese Erweiterung sollte vor allem dann einfach durchführbar sein, wenn die Auswertung der vorhergesagten Verteilungen durch Bestimmung von Erwartungswert und Varianz des zukünftigen Werts erfolgt, da dies dem prinzipiellen Modellierungskonzept dieser Methoden entspricht.

Die Analysemethodik der Perturbationstheorie bezieht sich nicht so sehr auf die direkte Analyse eines Systemmodells, als vielmehr auf die analytische Auswertung der zugehörigen Systemsimulationen. Dadurch kann auch ein Modell, das die Verteilungsprognose beinhaltet, problemlos behandelt werden.

Die Warteschlangentheorie zeichnet sich im Gegensatz zu den Methoden der Regelungstechnik und der Zeitreihenanalyse durch einen weniger stark ausgeprägten Formalismus bei der Behandlung von Problemstellungen aus. Es existieren eine Reihe von ganz unterschiedlichen Ansätzen, spezielle oder relativ einfache Probleme zu behandeln. Diese Ansätze müssen zur Lösung komplexerer Probleme individuell kombiniert und ausgebaut werden. Die Einbeziehung der Verteilungsprognose durch Berücksichtigung einer Abbildungsfunktion sollte daher als weitere spezielle Lösung ohne Probleme möglich sein.

## **4.2 Simulation**

Die Simulation dynamischer Systeme besteht aus der modellbasierten Nachbildung der realen Vorgänge in einem System, z. B. durch Datenverarbeitungsanlagen. Dabei werden die Zustände eines realen Systems auf Datenstrukturen eines Simulationsprogramms abgebildet und diese Datenstrukturen in einer dem realen System nachempfundenen Weise bearbeitet. Der Vorteil gegenüber analytischen Ansätzen besteht darin, daß auch sehr große und komplexe Systeme durch Simulationen untersucht werden können, Fehlfunktionen zu keinen Schäden führen und die Kosten für den Aufbau einer realen Testumgebung eingespart werden.



Für den Zweck der Modellbildung für eine Simulation müssen mehrere Klassen von Systemen unterschieden werden: zeitdiskrete und zeitkontinuierliche Systeme sowie deterministische und stochastische Systeme. Bei zeitdiskreten Systemen können Änderungen der Zustandsgrößen nur zu bestimmten, vorher bekannten Zeitpunkten auftreten; bei zeitkontinuierlichen Systemen treten derartige Änderungen dagegen zu beliebigen Zeitpunkten auf.

Je nach Art des Systemmodells kommen unterschiedliche Simulationsarten zum Einsatz. Dabei ist einschränkend zu sagen, daß grundsätzlich jede Computersimulation aufgrund der beschränkten Auflösung der internen Zahlendarstellung wertdiskret und aufgrund der schrittweisen Befehlsverarbeitung zeitdiskret ist. Die Genauigkeit ist jedoch in den meisten Fällen ausreichend für die Simulation wertkontinuierlicher Modelle. Auch für die Simulation zeitkontinuierlicher Modelle, wie z. B. analoger elektronischer Schaltungen, existieren entsprechende Simulationswerkzeuge, die die näherungsweise Nachbildung der Zeitkontinuität an die Dynamik des simulierten Modells anpassen (quasikontinuierliche Modelle). Folgende Simulationsarten wurden aufgrund unterschiedlicher Anforderungen entwickelt:

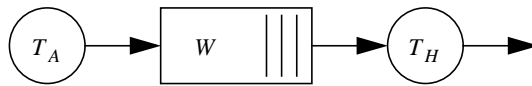
- **Zeitgesteuerte Simulation**  
Hier steht die Simulationszeit zur Realzeit in einem festen Verhältnis. Es werden alle Zeitintervalle – auch solche, in denen keine Aktionen stattfinden – simuliert. Diese Art der Simulation wird bei der prozeßorientierten und der transaktionsorientierten Simulation [67] eingesetzt.
- **Ereignisgesteuerte Simulation [67]**  
Für die ereignisgesteuerte Simulation werden die Zeitpunkte der Ereignisse betrachtet, zu denen sich die Zustände des simulierten Systems ändern. Jedem Ereignis ist ein Zeitpunkt und eine Funktion, die bei Eintreten des Ereignisses ausgeführt werden soll, zugeordnet. Die Ereigniszeitpunkte werden in einem Kalender verwaltet und die zugehörigen Funktionen in chronologischer Reihenfolge abgearbeitet. Durch den Aufruf dieser Funktionen werden ggf. neue Ereignisse erzeugt. Im Gegensatz zur zeitgesteuerten Simulation wird für Zeiten, in denen keine Ereignisse stattfinden, keine Simulationszeit „verbraucht“. Dies vermindert in der Regel die Simulationszeit für ein Modell und erleichtert die Anpassung an unterschiedliche Zeitmaßstäbe.
- **Quasikontinuierliche Simulation**  
Die quasikontinuierliche Simulation ist ein Spezialfall der zeitgesteuerten Simulation. Durch Neuberechnung der Modellgrößen nach sehr kleinen, aber immer noch diskreten Intervallen der Echtzeit, wird eine Approximation des zeitkontinuierlichen Verhaltens erreicht. Die Größe der Zeitintervalle ist stark modellabhängig.

In Bild 4.1 wird der Unterschied zwischen zeitgesteuerter und ereignisgesteuerter Simulation anhand eines einfachen Beispiels gezeigt <sup>(1)</sup>. Durch eine Quelle werden Anforderungen mit dem Ankunftsabstand  $T_A$  erzeugt, die durch eine Bedieneinheit mit der Bedienzeit  $T_H$  verarbeitet werden. Anforderungen, die eine belegte Bedieneinheit vorfinden, werden in einem Warte-speicher der Größe  $W$  zwischengespeichert. Teil b) des Bildes zeigt die Ankunfts- und Bediendeereignisse und die daraus resultierenden Bedien- und Wartezeiten. Für die zeitgesteuerte Simulation zeigt Teil c) die Beziehung zwischen simulierter Zeit für den Bedienprozeß und der durch die CPU verbrauchten Realzeit, die hier in einem festen Verhältnis zueinander stehen.

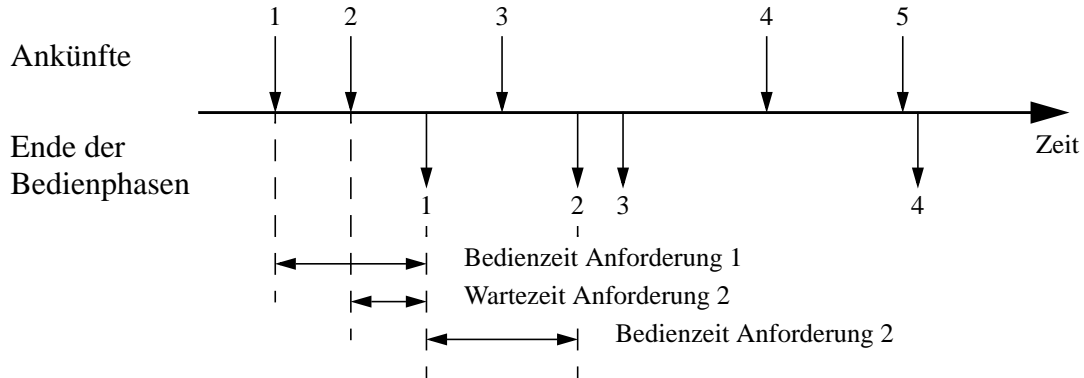
---

(1) Die in Bild 4.1a verwendeten Symbole sind in Tabelle 4.2 erläutert.

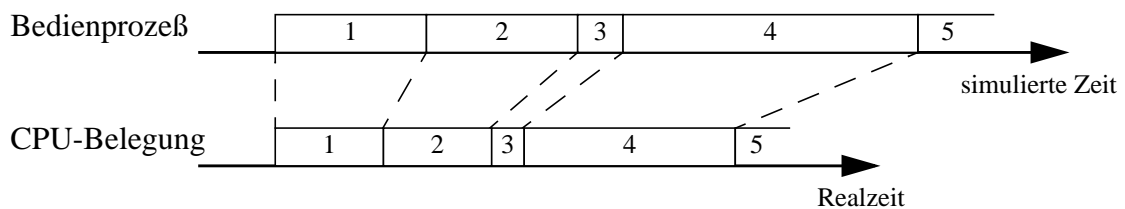
**a) Modell:**



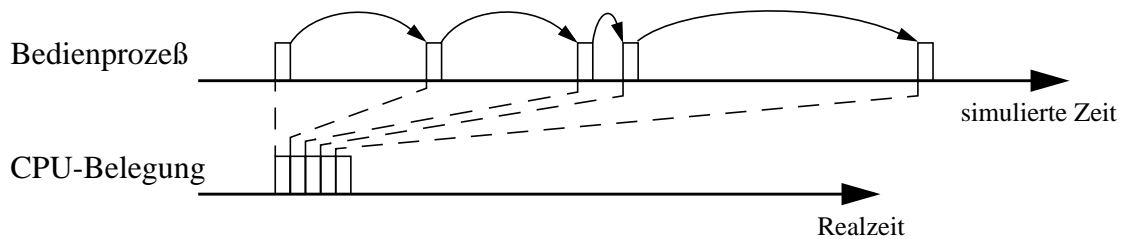
**b) Ereignisse:**



**c) Zeitgesteuerte Simulation:**



**d) Ereignisgesteuerte Simulation:**



**Bild 4.1:** Zeitgesteuerte Simulation vs. Ereignisgesteuerte Simulation

Teil d) zeigt diese Beziehung für die ereignisgesteuerte Simulation. Die Bearbeitung der einzelnen Anforderungen beschränkt sich hier auf die Bestimmung des nächsten Bedienendezeitpunkts und dessen Eintragung in einen Kalender. Die in Bild 4.1 dargestellten Unterschiede in der benötigten CPU-Zeit sind für viele Anwendungen repräsentativ.

Die ereignisgesteuerte Simulation bietet aufgrund ihrer Vorteile gegenüber der zeitgesteuerten Simulation für alle Anwendungsbereiche der Verteilungsprognose die effizienteste Simulationslösung und wird daher für alle Simulationen dieser Arbeit verwendet.

Für die Simulation stochastischer Systeme werden durch ein Simulationsprogramm Zufallsgrößen entsprechend der Verteilungen des Modells erzeugt. Daher ist die Güte der zugrundeliegenden Zufallszahlengeneratoren für die Güte einer Simulation wesentlich. Diese Güte wird unter anderem durch die rechnerinterne Zahlendarstellung eingeschränkt, die sich auf den

Wertebereich und die Auflösung der erzeugten Werte auswirkt. Weitere Einschränkungen bestehen bezüglich der Unabhängigkeit der Zufallsereignisse sowie der maximalen Zykluslänge der Zufallszahlenfolge. Es ist auch nicht möglich, einfach den „besten“ Zufallszahlengenerator zu verwenden, da für einen einzelnen Zufallszahlengenerator nicht nachweisbar ist, daß er in allen Anwendungsfällen gute Ergebnisse liefert. Da speziell Simulationen von Systemen mit Langzeitkorrelationen (s. Abschnitt 2.2.3) sehr große Ereigniszahlen benötigen, ist die Verwendung geeigneter Zufallsgeneratoren, die bei solch großen Ereigniszahlen durch ihre Eigenschaften die Simulationsergebnisse nicht verfälschen, sehr wichtig.

Der Aufbau eines Simulationsmodells erfolgt aus Komponenten, die Anforderungen oder Meldungen erzeugen oder empfangen können und die ggf. beim Eintreten von Ereignissen bestimmte Funktionen ausführen. Die Meldungen können Daten enthalten wie beispielsweise Pakete eines Kommunikationsnetzes oder zur Benachrichtigung anderer Komponenten dienen. Die Semantik dieser Meldungen wird durch das Simulationsmodell festgelegt.

Bei Simulationen im Bereich der Kommunikationstechnik werden Modellkomponenten verwendet, die an die Modellierungskomponenten der Warteschlangentheorie angelehnt sind. Auch hier gibt es Verkehrsquellen, Bedieneinheiten, Warteschlangen, usw. (s. Tabelle 4.2). Allerdings werden diese einfachen Basiskomponenten häufig durch komplexere Komponenten ergänzt, die mit Hilfe der Warteschlangentheorie nicht oder nur schwer analysiert werden können.

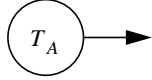
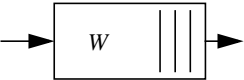
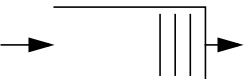
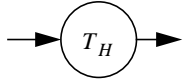
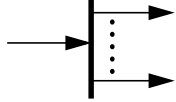
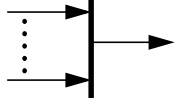
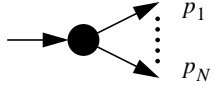
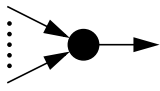

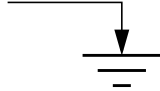
Während eines Simulationslaufs fallen vielfältige Daten an, wie Ankunftsabstände, Warteschlangenlängen, Auslastung von Bedieneinheiten, und andere. Diese Daten können auf unterschiedliche Weise weiter verwendet werden:

- Direkte grafische Anzeige des zeitlichen Verlaufs  
Die direkte grafische Anzeige des zeitlichen Verlaufs der Simulationsergebnisse bietet sich für Simulationen an, die kurze Laufzeiten haben. Bei langandauernden Simulationen, die z. B. Laufzeiten von mehreren Tagen haben, ist dieses Verfahren ungeeignet.
- Speicherung des zeitlichen Verlaufs  
Die während eines Simulationslaufs anfallenden Ergebnisse können gespeichert und nach Beendigung der Simulation geeignet weiterverarbeitet werden. Dieses Verfahren erfordert speziell bei langandauernden Simulationen große Speichermengen, erlaubt aber weitgehend beliebige Auswertungen der Daten.
- Statistikerfassung während des Simulationslaufs  
Die Erfassung von Statistikdaten während des Simulationslaufs reduziert die zu speichernde Datenmengen erheblich und erlaubt daher die größten Simulationsdauern. Allerdings muß vor Start der Simulation die Art der statistischen Auswertung bereits festliegen.

Die genannten Verfahren können auch kombiniert werden.

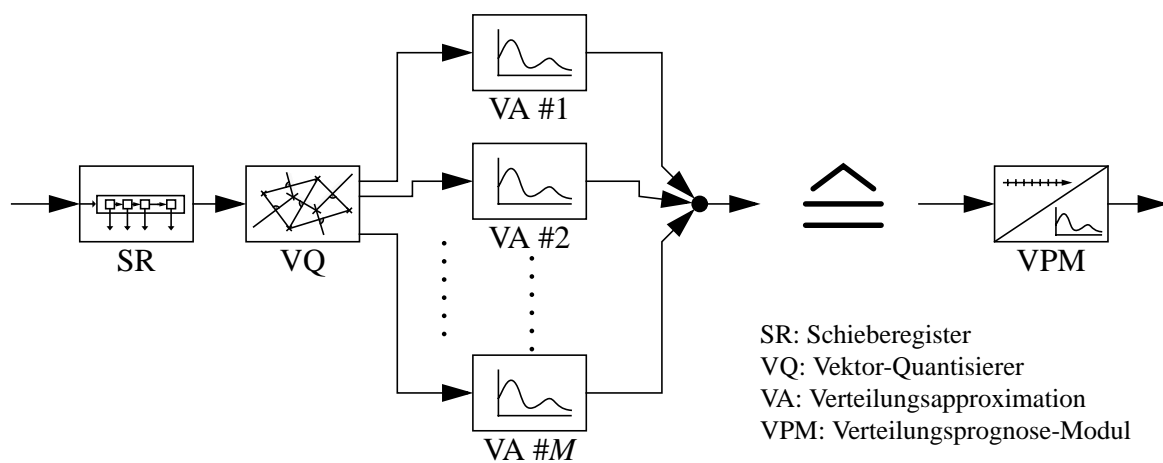
Um den Grad der Zuverlässigkeit der statistischen Simulationsergebnisse quantitativ bewerten zu können, werden für ein bestimmtes Signifikanzniveau Vertrauensintervalle für die Ergebnisse berechnet. Dafür ist die Unterteilung eines Simulationslaufs in mehrere Teiltests erforderlich. Die Ergebnisse der Teiltests erlauben über die Varianz der für eine Meßgröße während der Teiltests ermittelten Mittelwerte die Berechnung von Vertrauensintervallen.

**Tabelle 4.2:** Simulationskomponenten

| Komponente  | Sinnbild  |
|---|---|
| <p>Verkehrsquelle<br/>Die Zeit zwischen zwei Anforderungen ist <math>T_A</math>.<br/><math>A</math> kann beliebig verteilt sein.</p>  |    |
| <p>Wartespeicher endlicher Größe<br/>In dem Wartespeicher haben <math>W</math> Anforderungen Platz. Jede weitere Anforderung geht verloren.</p>   |    |
| <p>Wartespeicher mit unendlich vielen Plätzen<br/>Es können beliebig viele Anforderungen gespeichert werden.</p>  |    |
| <p>Bedieneinheit<br/>Die Bedienzeit pro Anforderung ist <math>T_H</math>. <math>T_H</math> kann beliebig verteilt sein.</p>   |    |
| <p>Aufspaltung<br/>Ankommende Anforderungen werden für jeden abgehenden Weg vervielfältigt.</p>   |    |
| <p>Vereinigung<br/>Mehrere ankommende Anforderungen werden zu einer abgehenden Anforderung zusammengefaßt.</p>  |  |
| <p>Verzweigung<br/>Ankommende Anforderungen nehmen mit Wahrscheinlichkeit <math>p_i</math> den Weg <math>i</math>.</p>  |  |
| <p>Zusammenführung<br/>Ankommende Anforderungen aller Wege werden auf einem gemeinsamen Weg weitergeführt (Multiplexer).</p>  |  |
| <p>Allgemeine Aktion<br/>Durch einen allgemeinen Verarbeitungsblock können Anforderungen in einer Art und Weise manipuliert werden, die über die Möglichkeiten bei Verwendung von Standardkomponenten hinausgeht.</p> |  |
| <p>Senke<br/>In einer Senke werden Anforderungen aus dem System entfernt.</p>   |  |

Wenn in einer Simulation stochastische Prozesse mit LRD-Verhalten simuliert werden, muß in der Regel eine größere Ereigniszahl vorgesehen werden, als wenn kein LRD-Verhalten vorliegt, um dieselben Vertrauensintervalle zu erreichen. Dies liegt daran, daß bei LRD-Verhalten längere Teiltests erforderlich sind, um dieselbe Varianz der Mittelwerte der einzelnen Teiltests zu erhalten, als ohne LRD-Verhalten.

Die Berücksichtigung der in Kapitel 3 eingeführten Verteilungsprognose in einer Simulation ist durch Einführung einer entsprechenden Simulationskomponente problemlos möglich. Die Eingangswerte einer solchen Komponente sind die Werte einer Zeitreihe, die Ausgangswerte sind Verteilungen. Bild 4.2 zeigt, wie die Verteilungsprognose durch Simulationskomponenten realisiert werden kann: Für die Blöcke Schieberegister (SR), Vektor-Quantisierer (VQ) und Verteilungsapproximation werden Symbole für allgemeine Aktionen gewählt, insbesondere, weil diesen Aktionen hier keine Verarbeitungszeit zugeordnet ist. Die Eingangswerte des SR sind z. B. Meldungen, deren Inhalt durch das SR zusammen mit vergangenen Werten in Form einer Meldung an den VQ weitergegeben und dort verarbeitet wird. Der VQ schickt Meldungen an den jeweils ausgewählten VA-Block. Am Ausgang der VA-Blöcke werden Meldungen mit Verteilungen als Inhalt erzeugt, die anschließend wieder zusammengeführt werden. Diese Komponenten können für den Einsatz in komplexeren Modellen zu einer einzigen äquivalenten Komponente zusammengefaßt werden, was ebenfalls in Bild 4.2 gezeigt ist.



**Bild 4.2:** Simulationskomponenten der Verteilungsprognose

## **Kapitel 5**

# **Einsatz der Verteilungsprognose zur Modellierung von Verkehrsquellen**

### **5.1 Einführung**

Für die Untersuchung der Leistungsfähigkeit von Kommunikationsnetzen und -systemen ist die Abstraktion des real auftretenden Verkehrs durch Quellmodelle ein wichtiger Forschungsgegenstand. Dabei muß zwischen Quellmodellen für Analysen und Modellen, die für Systemsimulationen und technische Realisierungen geeignet sind, unterschieden werden. In der Analyse von Warteschlangennetzen werden Quellmodelle verwendet, die einerseits die Analyse nicht zu aufwendig machen und dem Abstraktionsniveau des Systemmodells entsprechen und andererseits ein ausreichend gutes Abbild der Realität bieten. Im Bereich der Systemsimulation und für den Aufbau von Verkehrsgeneratoren für meßtechnische Aufgaben liegt das Interesse bei Verkehrsquellen, die dem Verhalten realer Quellen möglichst gut entsprechen, durch möglichst wenige Parameter beschrieben werden können und die, falls erforderlich, statistisch unabhängig voneinander realisierbar sind.

Die Systemgrößen, die durch Quellmodelle modelliert werden, hängen von der jeweiligen Abstraktion ab. Abhängig von dem betrachteten Systemmodell kann es sich dabei z. B. um Ankunftsabstände, Raten, Paketgrößen oder Mehrfachankünfte handeln.

### **5.2 Bekannte Verfahren**

Bei der Modellierung von Verkehrsquellen werden abhängig von den Eigenschaften, die das jeweilige Modell aufweisen soll, unterschiedliche Methoden eingesetzt. Je nach Anwendungszweck eines Modells kann der Schwerpunkt auf Modellierung der Verteilungsfunktion oder zusätzlich der Autokorrelationsfunktion des Prozesses liegen. Je genauer die Autokorrelationsfunktion modelliert werden soll, desto komplexer wird diese Aufgabe.

Eine starke Abstraktion des realen Quellverhaltens führt zu einfachen Quellmodellen, bei denen zur Modellierung unabhängige (regenerative) oder begrenzt abhängige stochastische Prozesse eingesetzt werden, die üblicherweise durch Angabe ihres Mittelwerts ausreichend charakterisiert sind. Durch derartige Modelle ist eine grobe Approximation der Verteilung der realen Systemgröße möglich.

Um das büschelförmige Verhalten vieler Anwendungen (Audio, Video, Daten) zu charakterisieren, werden häufig Prozesse zugrundegelegt, deren Intensität zeitabhängig moduliert wird. Hierzu zählen zustandsbasierte Modelle, bei denen mit bestimmten Übergangswahrscheinlichkeiten zwischen mehreren Zuständen gewechselt wird. Wenn das Modell in einem bestimmten

Zustand ist, wird Verkehr nach einer für diesen Zustand spezifischen Verteilung generiert. Populäre Modelle sind MMPP-Modelle (Markov Modulated Poisson Process) und GMDP-Modelle (Generally Modulated Deterministic Process) sowie Burst-Silence-Modelle. Durch diese Modelle ist eine gegenüber den einfachsten Quellmodellen verbesserte Nachbildung von Verteilung und Korrelation realer Systemgrößen möglich.

Weitere stochastische Quellmodelle sind die aus der Zeitreihenanalyse bekannten Modelle (s. Abschnitt 2.4). Diesen Modellen liegen normalverteilte Zufallsvariablen zugrunde. Ihre Ausgangswerte können daher beliebige Werte annehmen – auch negative. Sie können also nicht unverändert verwendet werden, wenn ein stochastischer Prozeß keine negativen Werte annehmen kann, wie z. B. Ankunftsabstände. In [37] wird ein ARMA-Modell als Quellmodell für Videoquellen verwendet, das die durch die Videoquelle erzeugte Datenmenge für feste Zeitabschnitte modelliert. Die durch das Modell erzeugten negativen Werte werden dort auf Null gesetzt. Dieses Vorgehen ist in der Regel nur dann sinnvoll, wenn die Wahrscheinlichkeit für das Auftreten negativer Werte sehr gering ist, da sonst die statistischen Kenndaten (Erwartungswert, Varianz, Korrelation, etc.) des Prozesses durch das Abschneiden der negativen Werte zu stark beeinträchtigt werden.

Zeitreihen mit LRD-Verhalten (s. Abschnitt 2.2.3) sind aufgrund ihrer sehr lange andauernden Korrelation durch Quellmodelle schwer nachbildbar. Quellmodelle mit LRD-Verhalten sind in der Regel viel komplexer als andere Quellmodelle aufgebaut [28]. Sie sind in der Warteschlangenanalyse nicht verwendbar und benötigen deutlich mehr Rechenleistung in der Simulation, da zum Erreichen einer bestimmten Ergebnisgüte wesentlich länger simuliert werden muß als bei Quellen ohne die LRD-Eigenschaft. Aufgrund der Komplexität dieser Modelle muß vor der Modellierung des LRD-Verhaltens gründlich abgewogen werden, ob es für ein bestimmtes Simulationsmodell überhaupt relevant ist.

Ein weiterer für die Quellmodellierung problematischer Zeitreihentyp sind chaotische Zeitreihen (s. Abschnitt 2.4.3). Sie heben sich durch ihr bis auf den Startwert völlig deterministisches Verhalten von stochastischen Prozessen ab. Dies führt zu deutlich anderen Anforderungen an die Modellierung, da die Wertefolge ausschließlich vom Startwert abhängt. Die Erzeugung chaotischer Zeitreihen durch ein Quellmodell ist bei bekannter Abbildungsvorschrift relativ einfach. Falls die Abbildungsvorschrift jedoch nicht bekannt ist, was den üblichen Fall darstellt, eignen sich selbstlernende Modelle, z. B. auf der Basis künstlicher neuronaler Netze [26, 74], für die Modellierung. Die Modellierung chaotischer Zeitreihen wird im folgenden nicht weiter betrachtet.

Die meisten Ansätze zur Quellmodellierung erfordern eine mehr oder weniger empirische Bestimmung der Modellparameter. Vor allem kompliziertere Modelle, die reale Quellen besser nachbilden, haben häufig viele Parameter, deren Bestimmung aufwendig und langwierig ist. Aus diesem Grund werden Modelle entwickelt, deren Parameter aus der Beobachtung realer Quellen automatisch abgeleitet werden können. Dazu gehört z. B. das ARMA-Modell nach [37], dessen Parameter z. T. aus einer Analyse der realen Quelle im Frequenzbereich bestimmt werden.

In einer Reihe von Ansätzen werden künstliche neuronale Netze mit ihrem inhärenten Adaptionsvermögen zur Zeitreihenmodellierung eingesetzt. Diese Ansätze eignen sich prinzipiell auch für die iterierte Vorhersage einzelner Werte bzw. zum Aufbau von Quellmodellen [19, 47,

74, 81, 98, 99, 100, 105]. Sie sind für die Quellmodellierung allerdings nur eingeschränkt nutzbar, da die erzeugten Zeitreihen keine stochastischen Komponenten enthalten. Je nach Komplexität des jeweiligen Modells werden mehr oder weniger Details des zeitlichen Verlaufs der vorgegebenen Folge gelernt. In der Regel wird für eine bestimmte Vergangenheit des Prozesses  $x_{k-1}, \dots, x_{k-N}$  ein Wert gelernt, der dem Mittelwert der für diese spezielle Vergangenheit beim Lernvorgang auftretenden Werte entspricht. Der Grund für dieses Verhalten liegt darin, daß für ähnliche Vektoren  $x_{k-1}, \dots, x_{k-N}$  aufgrund des stochastischen Verhaltens während der Lernphase völlig unterschiedliche Werte  $x_k$  auftreten. Da das Modell aufgrund seiner begrenzten Komplexität nicht zwischen allen möglichen Vektoren  $x_{k-1}, \dots, x_{k-N}$  und den dazugehörigen Werten  $x_k$  unterscheiden kann, wird auch für deterministische Folgen der Mittelwert dieser Werte gelernt. Ein so aufgebautes Quellmodell wird die ursprüngliche Folge mehr oder weniger gut reproduzieren, da das Quellmodell nur in der Lage ist, eine deterministische, der ursprünglichen Folge ähnliche Zeitreihe zu erzeugen. Im Vergleich zu der in Kapitel 3 beschriebenen Verteilungsprognose wird anstelle einer Verteilung ein Mittelwert gelernt. Wie bei den in Abschnitt 2.4.3 beschriebenen chaotischen Zeitreihen hängt eine durch ein solches Modell erzeugte Zeitreihe nur von ihrem Anfangswert ab. Dadurch wird immer dieselbe, zeitversetzte Wertefolge generiert, was insbesondere für Simulationen mit mehreren gleichartigen Quellen ungeeignet ist. Diese Quellen sind dann wegen der verschobenen, aber identischen Wertefolge stark korreliert, auch wenn alle von unterschiedlichen Startwerten ausgehen.

Eine Verbesserung muß folglich das Ziel haben, zufälliges Verhalten in das Modell einzubringen. Dies kann durch den Einsatz der Verteilungsprognose aus Kapitel 3 erreicht werden. Im folgenden Abschnitt wird ein Quellmodell beschrieben, das die Verteilungsprognose als Baustein verwendet. Das Ergebnis ist ein selbstlernendes Quellmodell, das auf künstlichen neuronalen Netzen basiert, ohne den Nachteil der oben genannten KNN-Modelle aufzuweisen. Während der automatisch ablaufenden Adaptionsphase paßt sich das Modell so an die Originalquelle an, daß die hinterher durch das Modell erzeugten Daten dasselbe stochastische Verhalten wie die Originalquelle haben.

Durch den Einsatz von KNNs kann eine weitgehend automatische Anpassung an eine gegebene Zeitreihe erreicht werden, was zu Zeit- und damit Kosteneinsparungen führt. Für Analysen können auf KNNs basierende Quellmodelle nur sehr eingeschränkt genutzt werden, da die Modelle selten in geschlossener Form vorliegen und die nach dem Lernvorgang vorliegenden Parametersätze schwer interpretierbar sind.

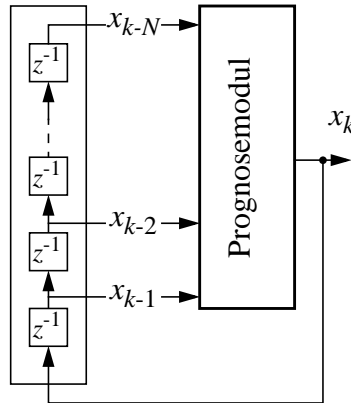
### 5.3 Anwendung der Verteilungsprognose

Bild 5.1 zeigt den prinzipiellen Aufbau eines Quellmodells, das auf der Verwendung der Verteilungsprognose nach Kapitel 3 beruht. Das Prognosemodul zur Generierung der Folge  $\{x_k\}$  hat als Eingangswerte  $N$  rückgekoppelte Werte von  $\{x_k\}$ . Die Rückkopplung erfolgt zeitverzögert, so daß am Eingang des Prognosemoduls der Vektor

$$\mathbf{i}_k = (i_1, i_2, \dots, i_N) = (x_{k-1}, x_{k-2}, \dots, x_{k-N}) \quad (5.1)$$

anliegt. Die Verzögerung der rückgekoppelten Werte läßt sich durch ein reellwertiges Schieberegister erreichen, das das einzige Gedächtnis <sup>(1)</sup> des Modells darstellt. Der Wert  $x_k$  ist der jeweils neu erzeugte Ausgangswert des Modells.

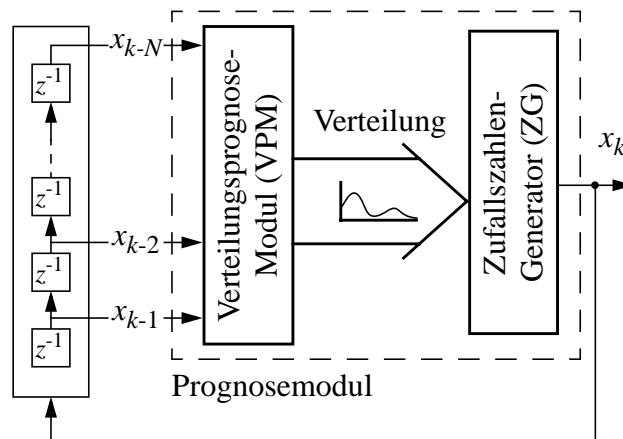




**Bild 5.1:** Quellmodell mit Rückkopplungsprinzip

Die Komponenten von  $\mathbf{i}_k$  bilden die Koordinaten des Eingangsraums mit der Dimension  $N$ . Zur Bestimmung einer Obergrenze von  $N$  kann der in Abschnitt 2.2.2.3 beschriebene Test zur Bestimmung der Dauer der Autokorrelation des zugrundeliegenden stochastischen Prozesses verwendet werden. Je nach Typ des Prozesses, z. B. bei Prozessen mit zyklischem Anteil, kann  $N$  auch deutlich kleiner als der durch den Test ermittelte Wert sein – dies muß empirisch bestimmt werden.

Das Prognosemodul aus Bild 5.1 wird nun in zwei Teile verfeinert, das „Verteilungsprognose-Modul“ (VPM) und den „Zufallszahlengenerator“ (ZG) (s. Bild 5.2). Für jeden Eingangsvektor wird durch den VPM-Block die Verteilung des folgenden Ausgangswerts vorhergesagt. Der Zufallszahlengenerator ermittelt für den Zeitpunkt  $k$  aus der Verteilung zufällig einen neuen Wert. Hierbei wird der als „general inverse-transform method“ bekannte Algorithmus eingesetzt [71].

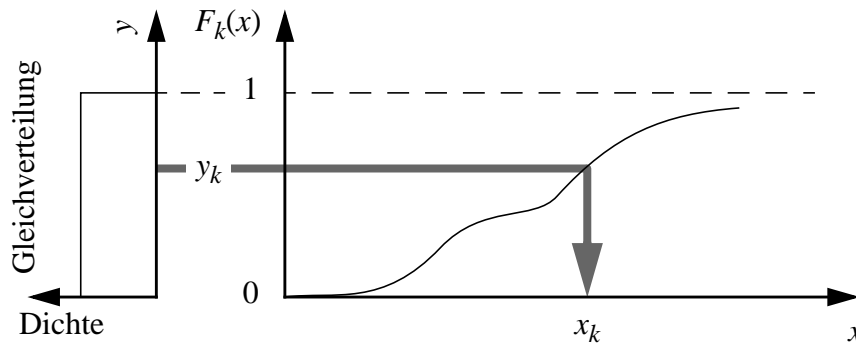


**Bild 5.2:** Quellmodell mit Verteilungsprognose

Wenn  $F_k(x)$  die Verteilungsfunktion am Ausgang der Verteilungsprognose zum Zeitpunkt  $k$  ist, erfolgt die Bestimmung eines zufälligen Werts wie folgt: Zuerst wird ein zufälliger Wert  $y_k$  einer auf dem Intervall  $[0;1]$  gleichverteilten Zufallsvariablen bestimmt. Dann wird der Wert  $x_k$

- (1) Für die Initialisierung der Schieberegisterwerte zum Zeitpunkt  $k=0$  werden die Werte eines der in dem Vektor-Quantisierer der Verteilungsprognose gespeicherten Vektoren verwendet.

gesucht, für den  $F_k(x_k) = y_k$  gilt. Dieser Wert stellt den neuen, nach  $F_k(x)$  verteilten Zufallswert dar. Bild 5.3 verdeutlicht diese Zusammenhänge <sup>(1)</sup>.



**Bild 5.3:** Erzeugung beliebig verteilter Zufallszahlen

## 5.4 Erweiterte Modelle

Das in Bild 5.2 dargestellte Quellmodell kann in verschiedener Hinsicht erweitert werden. Hierfür werden die Erweiterungen der Verteilungsprognose aus Kapitel 3 geeignet eingesetzt.

### 5.4.1 Mehrwertige Quellmodelle

Die Verwendung der Verteilungsprognose mit mehreren Eingängen (s. Abschnitt 3.4) erlaubt die Modellierung mehrwertiger Zeitreihen. Dadurch können Abhängigkeiten zwischen mehreren Zeitreihen berücksichtigt werden.

Die durch  $Q$  Schieberegister realisierten  $Q$  Eingänge werden von  $Q$  Zeitreihen gespeist (vgl. Bild 3.6). Eine dieser Zeitreihen ist in der Regel die durch das Modell erzeugte rückgekoppelte Folge. Die anderen dienen der Steuerung des Modells.

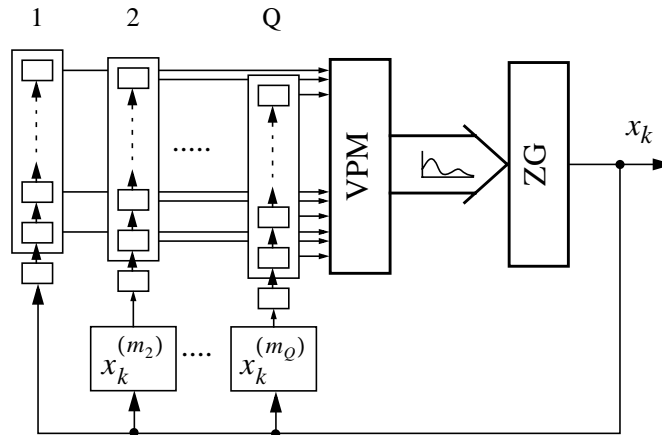
### 5.4.2 Quellmodelle mit aggregierter Rückkopplung

Die Erweiterung des letzten Abschnitts kann auch zur Verbesserung der Modellierungsgüte eines Modells für eine einzige Zeitreihe verwendet werden, indem einzelnen Eingängen nicht vergangene Folgenwerte, sondern Werte des über verschiedene Zeiträume aggregierten Prozesses  $\{x_k^{(m)}\}$  zugeführt werden (s. Bild 5.4). Dieser entsteht durch Mittelung von jeweils  $m$  aufeinanderfolgenden Werten von  $\{x_k\}$  mit

$$x_k^{(m)} = \frac{(x_{km-m+1} + \dots + x_{km})}{m} \quad m \in \mathbb{N}. \quad (5.2)$$

Auf diese Weise kann ohne wesentlichen Zusatzaufwand ein deutlich größerer Zeitbereich für die Vorhersage berücksichtigt werden als ohne diese Erweiterung. Als Konsequenz stimmt die Autokorrelation der modellierten Zeitreihe mit der des Originals besser überein, speziell bei Zeitreihen mit Langzeitkorrelation. Bild 5.4 zeigt den Aufbau eines derartigen Quellmodells.

(1) Dieser Algorithmus stellt eine gegenüber anderen Verfahren effiziente Lösung zur Berechnung beliebig verteilter Zufallswerte dar.

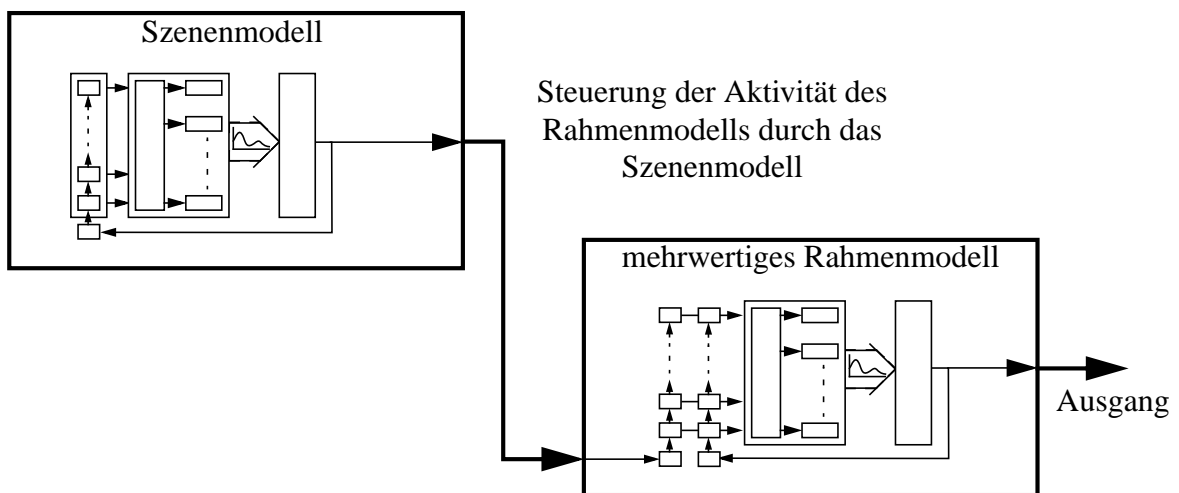


**Bild 5.4:** Quellmodell mit Rückkopplung des aggregierten Ausgangswerts

Die Parameter  $m_i$  bestimmen den Zeitraum der jeweilige Mittelwertbildung, ein Eingang wird direkt durch das rückgekoppelte Ausgangssignal gespeist.

### 5.4.3 Hierarchische Modelle

Der Ansatz des letzten Abschnitts, mehrere Eingänge für die Verteilungsprognose zu verwenden, läßt sich auch zum Aufbau hierarchischer Quellmodelle verwenden (s. Bild 5.5). So kann z. B. das Verhalten eines Quellmodells durch ein weiteres Quellmodell gesteuert werden. Die Modelle auf unterschiedlichen Hierarchiestufen werden sich in der Regel durch den Zeitmaßstab unterscheiden, auf dem sie arbeiten. So sind z. B. Lösungen für hierarchische MPEG-Modelle denkbar, die aus einem Steuermodell auf Szenenebene und einem zweiten Modell auf Rahmenebene, das durch das erste Modell gesteuert wird, bestehen.



**Bild 5.5:** Hierarchisches Quellmodell

## 5.5 Statistische Tests

Die Güte eines Quellmodells kann über qualitative und quantitative Vergleiche zwischen dem für die Adaption verwendeten und dem durch das Quellmodell erzeugten stochastischen Prozeß bestimmt werden. Qualitative Aussagen sind beispielsweise über einen visuellen Vergleich der Korrelogramme und der empirischen Verteilungsfunktionen der beiden Prozesse möglich. Für quantitative Vergleiche werden statistische Tests auf Stichproben der beiden Prozesse angewandt (vgl. Abschnitt 2.2.2).

### 5.5.1 Bewertung der Verteilung

Die Bewertung der Verteilung einer durch ein Quellmodell künstlich erzeugten Zeitreihe kann durch den Kolmogorow-Smirnow-Test (s. Abschnitt 2.2.2.4) erfolgen. Der Test liefert als Ergebnis eine Kennzahl, aufgrund der entschieden wird, ob die Hypothese, daß die Verteilung der modellierten Zeitreihe mit der des Originals übereinstimmt, abgelehnt werden muß oder nicht. Diese Entscheidung erfolgt unter Berücksichtigung einer vorher festzulegenden Irrtumswahrscheinlichkeit (z. B. 0,05) für eine irrtümliche Ablehnung der Hypothese.

### 5.5.2 Bewertung des Korrelogramms

Die aus Abschnitt 2.2.2 bekannten statistischen Tests zur Auswertung von Korrelationen liefern nur Aussagen über das Korrelationsverhalten *einer* Zeitreihe. Für den quantitativen Test der Übereinstimmung der Korrelation zweier Zeitreihen sind ergänzend dazu weitere Tests erforderlich. Diese Tests werden im folgenden vorgestellt. Dabei muß zwischen Prozessen mit bzw. ohne periodischem Anteil unterschieden werden. Ein periodischer Anteil kann z. B. durch Zyklen (bei MPEG-codiertem Video) oder Saisoneffekte (bei Wetterdaten) verursacht werden.

#### Bewertung des Korrelogramms von Zeitreihen ohne periodischen Anteil

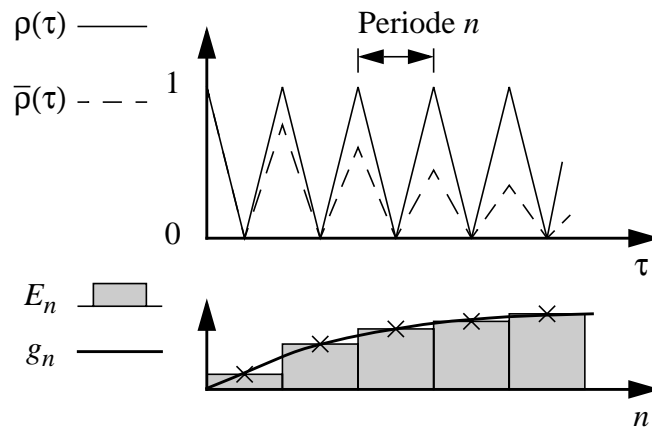
Bei Zeitreihen ohne periodischem Anteil strebt die Autokorrelation  $\rho(\tau)$  für große Werte von  $\tau$  gegen Null. Zu dieser Klasse gehören alle ARMA-Prozesse. Über das Kriterium nach 2.2.2.3 kann ein maximales  $\tau_{\max}$  bestimmt werden, ab dem die Autokorrelationskoeffizienten vernachlässigt werden können. Als Anpassungstest für zwei Autokorrelationsfunktionen wird hier die Summe der quadrierten Abweichungen zwischen den Korrelationskoeffizienten  $\rho(\tau)$  der zu lernenden Zeitreihe und den Korrelationskoeffizienten  $\bar{\rho}(\tau)$  der modellierten Zeitreihe verwendet, die auf die Anzahl der berücksichtigten Koeffizienten normiert werden:

$$E_{\rho} = \frac{1}{\tau_{\max}} \cdot \sum_{\tau=1}^{\tau_{\max}} [\rho(\tau) - \bar{\rho}(\tau)]^2. \quad (5.3)$$

In den Beispielen wird 0,01 als kritischer Wert für  $E_{\rho}$  verwendet. Dieser Wert wurde empirisch ermittelt und liefert einen guten Anhaltspunkt für den Lernerfolg.

### Bewertung des Korrelogramms von Zeitreihen mit periodischem Anteil

Bei Zeitreihen mit periodischem Anteil strebt die Autokorrelation nicht oder nur sehr langsam gegen Null. Deshalb läßt sich kein Maximalwert für  $\tau$  bestimmen. Zu dieser Klasse gehören z. B. MPEG-codierte Videosequenzen. Das vorgestellte Quellmodell ist aufgrund seines Aufbaus nicht in der Lage, beliebig lange Korrelationen zu modellieren, da Vorhersagen immer aufgrund eines begrenzten „Gedächtnisses“ erfolgen. Der Abfall der Langzeitkorrelation der modellierten Zeitreihe ist daher in der Regel signifikant stärker als bei der Original-Zeitreihe. Aus diesem Grund wird hier die Abweichung im Abfall der Autokorrelationsfunktion bewertet. Beobachtungen zeigen, daß die über eine Periode gemittelten Korrelationskoeffizienten der modellierten Zeitreihe nach einem Exponentialgesetz abfallen. Zusammen mit der Annahme, daß die über jeweils eine Periode gemittelten Korrelationskoeffizienten einer zyklischen Zeitreihe zumindest näherungsweise konstant sind, legt dies den Ansatz nahe, den Anstieg des Fehlers pro Periode durch eine Exponentialfunktion zu approximieren.



**Bild 5.6:** Test für periodische Zeitreihen

Bild 5.6 verdeutlicht diese Zusammenhänge anhand eines Beispiels. Im oberen Teil des Bildes sind die Korrelogramme dargestellt. Deutlich zu erkennen ist der Abfall der Autokorrelationskoeffizienten  $\bar{\rho}(\tau)$ . Im unteren Teil des Bildes ist der Anstieg des über jede Periode  $n$  gemittelten Fehlers  $E_n$  zu sehen. Als Fehlermaß für Periode  $n$  wird die Summe der quadrierten Differenzen der Korrelationskoeffizienten über diese Periode gewählt, normiert auf den Faktor  $E_{\max}$ :

$$E_n = \begin{cases} 0 & n = 0 \\ \sum_{\text{Periode } n} \frac{[\rho(\tau) - \bar{\rho}(\tau)]^2}{E_{\max}} & n > 0 \end{cases} \quad (5.4)$$

Der Normierungsfaktor  $E_{\max}$  ist gleich dem Maximalwert, den das nicht normierte  $E_n$  über alle Perioden  $n$  für einen konkreten Test annimmt. Dadurch hat der normierte Fehler  $E_n$  den Maximalwert 1.

Bei Approximation durch eine Exponentialfunktion erfolgt die Bestimmung des Koeffizienten  $\beta$  der Funktion

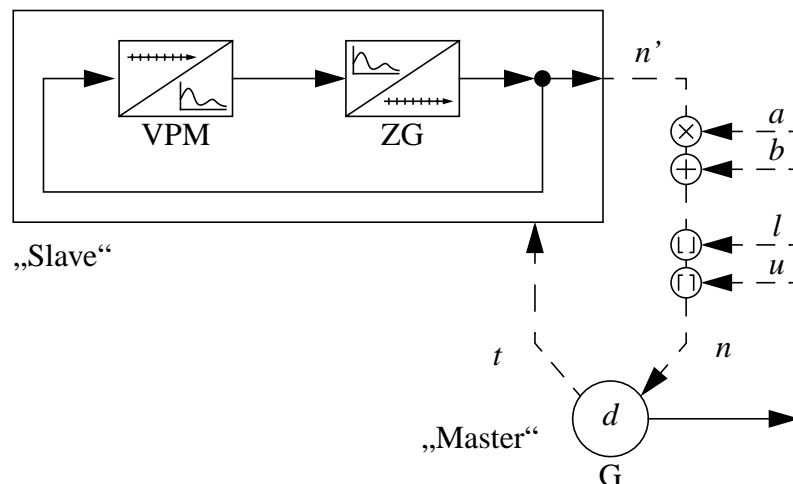
$$g_n = 1 - e^{-\beta \cdot n}, \quad (5.5)$$

so, daß  $g_n$  den Anstieg von  $E_n$  möglichst gut approximiert. Für die Parameteranpassung wird ein numerisches Verfahren zur Minimierung der quadratischen Differenz zwischen  $E_n$  und  $g_n$  eingesetzt.

Zur Bewertung der Übereinstimmung der Korrelogramme wird der Parameter  $\beta$  der angepaßten Exponentialfunktion nach Gl. (5.5) herangezogen. In den Beispielen wird 0,1 als kritischer Wert für  $\beta$  verwendet, oberhalb dessen die Annahme der Übereinstimmung abgelehnt wird. Auch dieser Wert wurde empirisch ermittelt.

## 5.6 Simulationsmodell

Mit Hilfe der in Abschnitt 4.2 eingeführten Simulationskomponenten wird das Simulationsmodell der Quellmodellierung aufgebaut (s. Bild 5.7). Abgebildet ist der Wiedergabevorgang durch ein Quellmodell, dessen interne Parameter durch einen vorangegangenen Lernvorgang bereits bestimmt wurden, wie in Abschnitt 3.2.5 beschrieben. Das Quellmodell besteht aus zwei prinzipiellen Teilen, dem Generator G, der als Steuerung oder „Master“ auftritt, und der Zufallszahlenerzeugung, dem „Slave“. Der „Master“-Teil übernimmt die Steuerung der Zufallszahlenerzeugung, indem durch das Signal  $t$  die Erzeugung einer neuen Zufallszahl  $n$  angefordert wird. Der in Bild 5.7 oben links abgebildete „Slave“-Teil ist eine Kombination aus einem Verteilungsprognose-Modul (VPM) und einem Zufallszahlengenerator (ZG). Das Verteilungsprognose-Modul legt Verteilungen an den Eingang des Zufallszahlengenerators an, der gemäß den Verteilungen einen zufälligen reellen Wert ermittelt. Die so ermittelten Werte werden an den Eingang des Verteilungsprognose-Moduls zurückgekoppelt. Dieser geschlossene Kreis ist in der Lage, Zufallszahlen gemäß den statistischen Kenngrößen des während der Lernphase gelernten Zufallsprozesses zu erzeugen.



**Bild 5.7:** Simulationsmodell des Quellmodells:

Die erzeugten Zufallswerte  $n'$  können durch zusätzliche Parameter  $a$ ,  $b$ ,  $l$  und  $u$  modifiziert werden. Über  $a$  ist eine Skalierung der Varianz und über  $b$  eine Verschiebung des Mittelwerts möglich:

$$n = n' \cdot a + b \tag{5.6}$$

Die Parameter  $l$  und  $u$  können zur Begrenzung der Werte von  $n$  nach oben ( $u$ ) und nach unten ( $l$ ) verwendet werden.

Für die anschließende Weiterverarbeitung der ggf. modifizierten Zufallswerte  $n$  durch G existieren zwei unterschiedliche Betriebsarten. In einer Betriebsart wird  $n$  direkt als Ankunftsabstand  $d$  der von G erzeugten Meldungen interpretiert. In der anderen Betriebsart erzeugt G Meldungen mit einem äquidistanten zeitlichen Abstand  $d$ , wobei die erzeugten Meldungen den Wert  $n$  als Inhalt haben.

Bei den im folgenden betrachteten Beispielen erfolgt nur eine rein statistische Auswertung der erzeugten Werte. Da keine weitere Verarbeitung der Meldungen erfolgt, spielt die Betriebsart des Generators keine Rolle.

Die Bestimmung der internen Parameter des Quellmodells (Lernvorgang) erfolgt *nicht* innerhalb eines Simulationsprogramms, sondern aus Effizienzgründen durch ein separates Lernprogramm.

## 5.7 Beispiele und Leistungsbewertung

In den folgenden Abschnitten werden an einigen Beispielen die Leistungsfähigkeit und die Grenzen des beschriebenen Quellmodells aufgezeigt. Dabei werden für einen Teil der Beispiele künstlich erzeugte Zeitreihen verwendet, deren statistische Eigenschaften bekannt sind. Weitere Beispiele basieren auf Daten, die durch Messungen an realen Systemen ermittelt wurden.

### 5.7.1 Notation

Im weiteren wird als Kurzschreibweise für ein Modell mit  $N$  VQ-Eingängen,  $M$  Verteilungsapproximationen und  $L$  Approximationsbereichen pro Verteilung die Abkürzung  $N/M/L$  verwendet. Die Werte für die Vorhersageweite  $P$  sowie die Anzahl der Eingänge  $Q$  sind außer in Ausnahmefällen immer 1. Die Abkürzung  $N/M/L/P$  wird verwendet, falls  $P$  ungleich 1 ist. Bei Verwendung eines mehrwertigen Modells mit aggregierter Rückkopplung ( $Q$  ungleich 1) nach Abschnitt 5.4.2 wird zusätzlich die Anzahl Schieberegisterplätze pro zusätzlichem Eingang  $N_j$  sowie jeweils die Aggregations- oder Summationskonstante  $m_j$  angegeben, getrennt durch den Buchstaben ‚s‘ für ‚Summe‘. In Tabelle 5.1 sind einige einfache Beispiele für die Notation angegeben.

**Tabelle 5.1:** Beispiele zur Notation der Quellmodell-Parameter

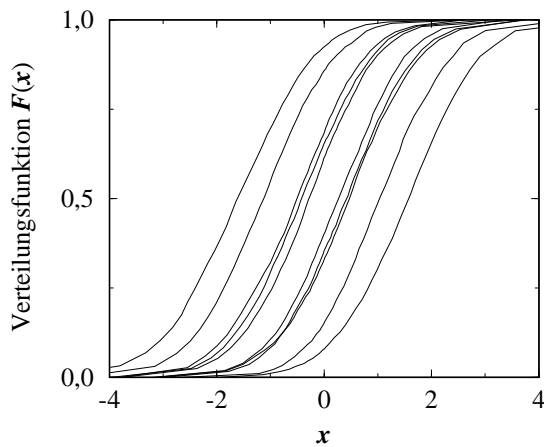
| Parameter                              | Notation    |
|--|-------------|
| $N=4, M=60, L=40$                      | 4/60/40     |
| $Q=2, N_1=9, N_2=5, m_2=9, M=80, L=40$ | 9+5s9/80/50 |
| $N=6, M=60, L=40, P=2$                 | 6/60/40/2   |

## 5.7.2 ARMA-Prozesse

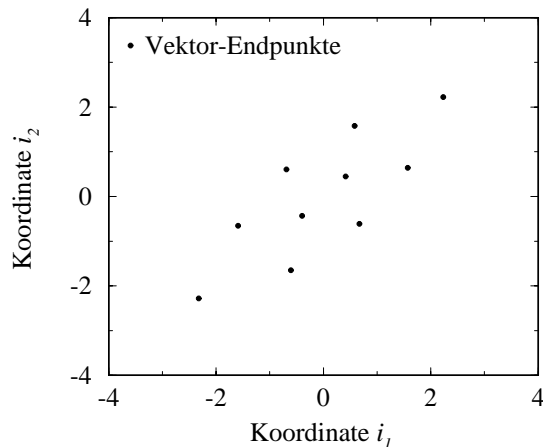
### Autoregressiver Prozeß erster Ordnung, AR(1)

Für dieses Beispiel werden die Daten, an die das Quellmodell angepaßt werden soll, durch ein AR-Modell erster Ordnung (s. Abschnitt 2.4.2.1) mit dem Parameter  $\phi_1=0,7$  erzeugt.

Die Rückkopplung vergangener Daten, die zur Erzeugung dieses Prozesses implizit durchgeführt wird, entspricht in besonderer Weise der Arbeitsweise des Quellmodells. Daher ist zur Modellierung dieses Prozesses ein einfaches 2/10/50-Modell ausreichend. Bild 5.8 zeigt die durch das Quellmodell gelernten 10 Verteilungsfunktionen, von denen jede eine verschobene, für ein bestimmtes „Gedächtnis“ gültige Normalverteilung darstellt. Bild 5.10 zeigt die Korrelogramme des Originalprozesses und des modellierten Prozesses. Die quantitativen Vergleichsgrößen zwischen Original und Quellmodell können Tabelle 5.2 am Ende dieses Kapitels entnommen werden.



**Bild 5.8:** Gelernte Verteilungsfunktionen für AR(1) (2/10/50-Modell)

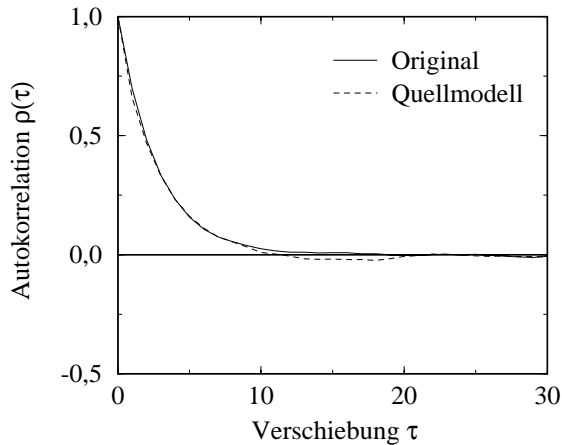


**Bild 5.9:** Interne VQ-Repräsentation für AR(1)-Prozeß und 2/10/50-Modell

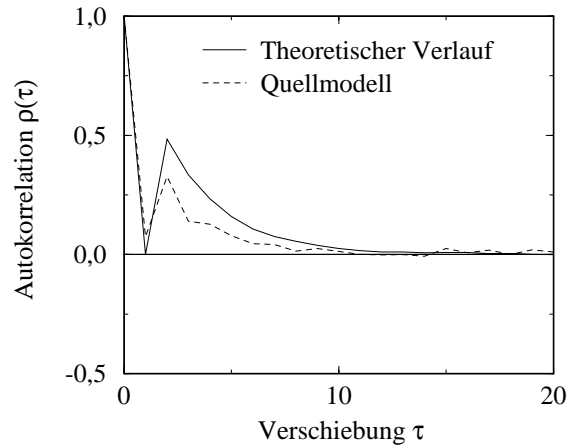
Anhand dieses Beispiels läßt sich gut die Anpassung des Vektor-Quantisierers an die korrelierten Eingangsdaten demonstrieren. Bild 5.9 zeigt eine zweidimensionale Darstellung der Vektorendpunkte des VQs. Als Koordinaten sind die Werte der beiden VQ-Eingänge  $i_1$  und  $i_2$  aufgetragen. Man erkennt die Häufung der Punkte entlang der Hauptdiagonalen des Koordinatensystems. Der Grund ist die serielle Korrelation der Zeitreihe, wodurch die Wahrscheinlichkeit für ähnliche aufeinanderfolgende Werte höher ist als die für stark unterschiedliche Werte. Da die Gebiete des VQ nach abgeschlossenem Lernvorgang gleich häufig angesprochen werden, häufen sich die Vektorendpunkte dort, wo ähnliche aufeinanderfolgende Werte liegen. Darüberhinaus ist durch die normalverteilten Werte eine Häufung der Vektorendpunkte um die Koordinaten (0, 0) zu beobachten.

An diesem einfachen AR(1)-Prozeß kann weiterhin die Vorhersage der Verteilung von Werten demonstriert werden, die mehr als einen Schritt in der Zukunft liegen. Die Realisierung eines Quellmodells mit der Prognose einer Verteilung, die mehrere Zeitschritte in der Zukunft liegt, hat vermutlich keine praktische Anwendung. Dafür kann das Verhalten der in Abschnitt 3.3 vorgestellten Erweiterung der Verteilungsprognose an diesem Beispiel gut demonstriert wer-





**Bild 5.10:** Autokorrelogramme für AR(1)-Original und 2/10/50-Modell



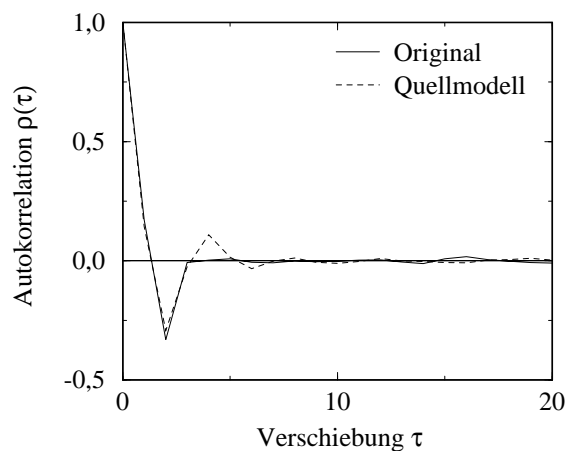
**Bild 5.11:** Autokorrelogramme für AR(1)-Original und 5/80/50/2-Modell bei Vorhersage von zwei Schritten,  $P=2$

den. Für dieses Beispiel wird  $P=2$  gewählt. Um der Theorie zu entsprechen, müßte der Wert des Korrelogramms an der Stelle 1 den Wert Null haben, was das Diagramm in Bild 5.11 bestätigt. Allerdings ist zu beobachten, daß das Korrelogramm der modellierten Zeitreihe relativ stark vom theoretischen Verlauf abweicht, was auf die höhere Vorhersageunsicherheit bei Verwendung der mehrschrittigen Prognose zurückzuführen ist.

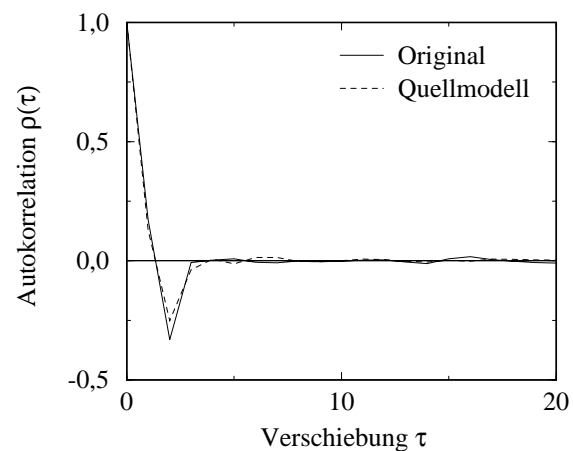
### Moving-Average-Prozeß zweiter Ordnung, MA(2)

Für dieses Beispiel werden die Daten, an die das Quellmodell angepaßt werden soll, durch ein MA-Modell zweiter Ordnung (s. Abschnitt 2.4.2.2) mit den Parametern  $\theta_1=0,5$  und  $\theta_2=-0,5$  erzeugt. Die Autokorrelationsfunktion dieses Prozesses zeigt Bild 5.12.

Moving-Average-Prozesse haben die Eigenschaft, daß die Autokorrelationskoeffizienten ab einer bestimmten Verschiebung Null sind. Diese Eigenschaft und die Tatsache, daß MA-Prozesse nicht intern rückgekoppelt sind, macht es für das Quellmodell aufgrund dessen Rück-



**Bild 5.12:** Autokorrelogramme für MA(2)-Original und 3/50/50-Modell



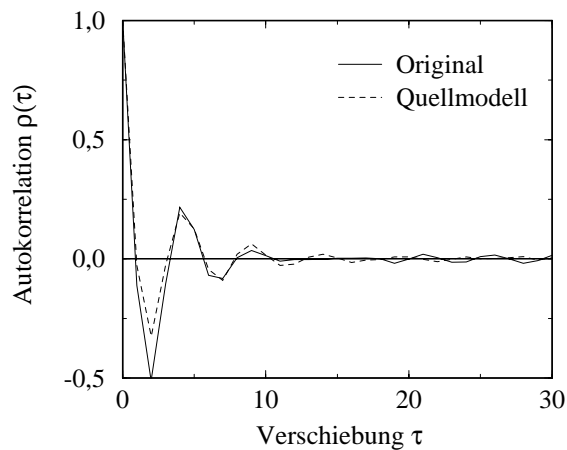
**Bild 5.13:** Autokorrelogramme für MA(2)-Original und 4/70/50-Modell

kopplungsstruktur schwieriger, MA-Prozesse zu lernen. Die Konsequenz ist, daß zum Erreichen derselben Adaptionsgüte wie bei AR-Prozessen komplexere Modelle herangezogen werden müssen.

Bild 5.12 zeigt die Korrelogramme des Originalprozesses und des modellierten Prozesses für ein Quellmodell mit einer Schieberegisterlänge von 3. Man erkennt die gute Übereinstimmung der Korrelogramme bis auf die Verschiebung 4. Um eine noch bessere Übereinstimmung zu erzielen und das Abbrechen der Korrelation gut zu modellieren, kann die Schieberegisterlänge auf 4 erhöht werden (s. Bild 5.13).

### ARMA(1,2)-Prozeß

Für dieses Beispiel werden die Daten, an die das Quellmodell angepaßt werden soll, durch ein ARMA-Modell der Ordnung (1,2) mit den Parametern  $\phi_1=0,3$ ,  $\phi_2=-0,5$  und  $\theta_1=-0,9$  erzeugt (s. Abschnitt 2.4.2.3). Die Autokorrelationsfunktion dieses Prozesses zeigt Bild 5.14.



**Bild 5.14:** Autokorrelogramme für ARMA(1,2) und 6/70/50-Modell

Für dieses Beispiel werden die Anforderungen der beiden vorherigen Beispiele kombiniert. Es stellt daher auch höhere Anforderungen an das Quellmodell, was eine höhere Komplexität zur Folge hat. Die geforderte Genauigkeit kann durch ein 6/70/50-Modell erreicht werden. Bild 5.14 zeigt die Korrelogramme des Originalprozesses und des modellierten Prozesses.

## 5.7.3 MPEG-kodierte Videodaten

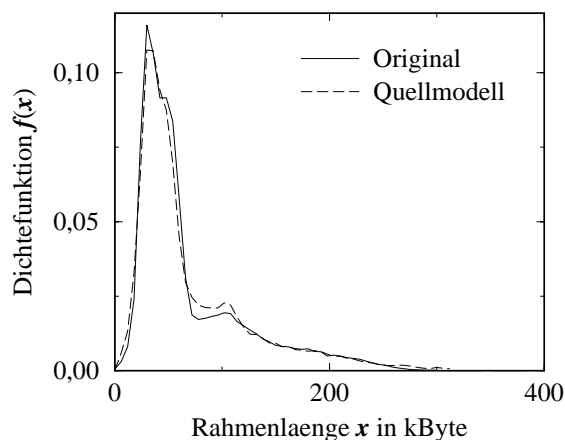
In diesem Beispiel wird das Quellmodell an MPEG-codierte Videodaten angepaßt. Die hohe und langandauernde Korrelation (LRD-Eigenschaft) dieses Prozesses ist die Folge der ausgeprägten Zyklen und der Selbstähnlichkeit des Prozesses auf verschiedenen Zeitebenen (Rahmen- und Szenenebene). Die Eigenschaften von MPEG-codierten Videodaten werden in Abschnitt 3.5 ausführlich beschrieben.

### 5.7.3.1 Korrelation und Verteilung

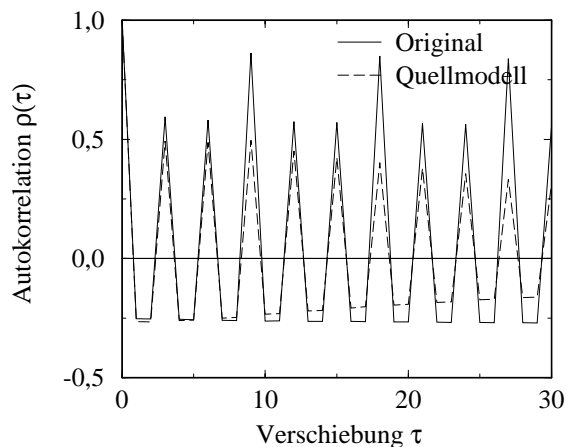
Gute Quellmodelle für MPEG-Daten sind bei der Simulation von Systemen erforderlich, die ein Gedächtnis (z. B. gekennzeichnet durch die Aufenthaltsdauer in einem Wartespeicher) in

der Größenordnung der Autokorrelationsdauer des MPEG-Prozesses haben. Dann wirkt sich die LRD-Eigenschaft dieses Prozesses auf die Systemeigenschaften aus. Bei Systemen, die diese Anforderung nicht erfüllen, sind auch einfachere Quellmodelle ausreichend [28]. Ein simulativer Nachweis dieser Aussage wird in Abschnitt 5.7.3.2 geführt.

Bereits Quellmodelle mit geringer Komplexität ermöglichen eine sehr gute Wiedergabe der Korrelation des Originalprozesses, da die zyklische Eigenschaft dieses Prozesses bereits für kleine Schieberegisterlängen reproduziert werden kann. Die Bilder 5.15 und 5.16 zeigen die Dichtefunktionen und Korrelogramme für ein 6/9/20-Modell. Mit weniger als 6 Eingängen ist die Wiedergabe der Korrelationsstruktur deutlich schlechter, wie in Bild 5.18 für ein Modell mit 4 Eingängen gezeigt wird (s. dazu auch Abschnitt 3.5). Bild 5.17 zeigt die zu diesem Modell gehörenden Dichtefunktionen.



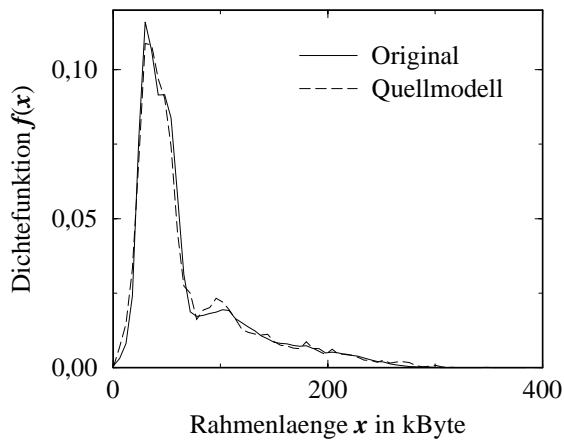
**Bild 5.15:** Verteilungsdichten (Original und 6/9/20-Modell) für MPEG-Daten



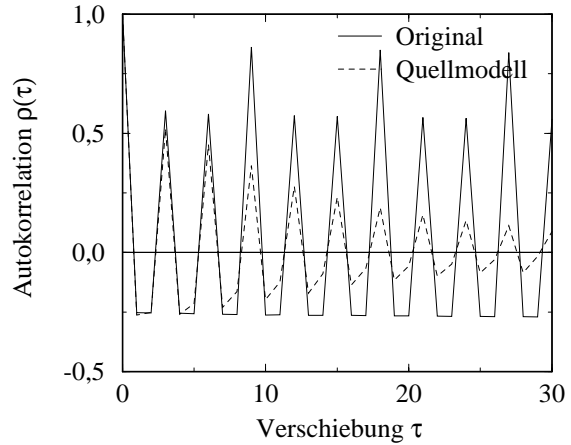
**Bild 5.16:** Autokorrelogramme (Original und 6/9/20-Modell) für MPEG-Daten

Um die geforderten Gütekriterien besser zu erfüllen, ist ein komplexeres 9/80/50-Modell erforderlich: Bild 5.19 zeigt die Dichtefunktionen, Bild 5.20 die Korrelogramme. Deutlich ist hier zu erkennen, daß die Korrelogramme im Vergleich zu Bild 5.16 besser übereinstimmen.

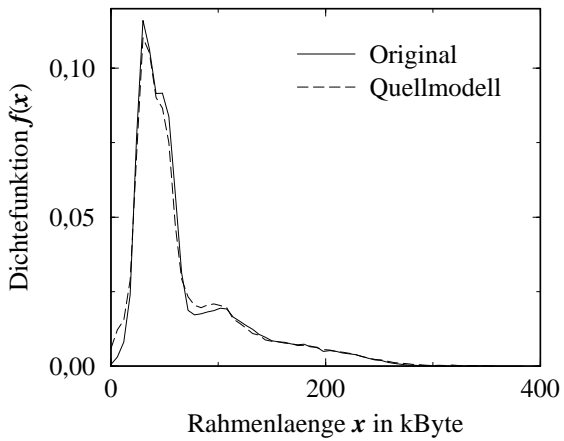
Wie bei dem AR(1)-Beispiel wird auch hier zur Veranschaulichung die Vorhersage von Verteilungen von Werten demonstriert, die mehr als einen Schritt in der Zukunft liegen. Bild 5.21 zeigt die Korrelogramme von Original und Quellmodell für eine Vorhersageweite von  $P=5$ . Im Gegensatz zu einem AR(1)-Prozeß, bei dem es sich um streng stationären Prozeß ohne Zyklen o. ä. handelt, sind hier die Werte des Korrelogramms für  $\tau=1, \dots, P-1$  nicht Null. Dies liegt an dem deterministischen zyklischen Anteil des MPEG-Prozesses, der durch das Modell sehr gut gelernt wird.



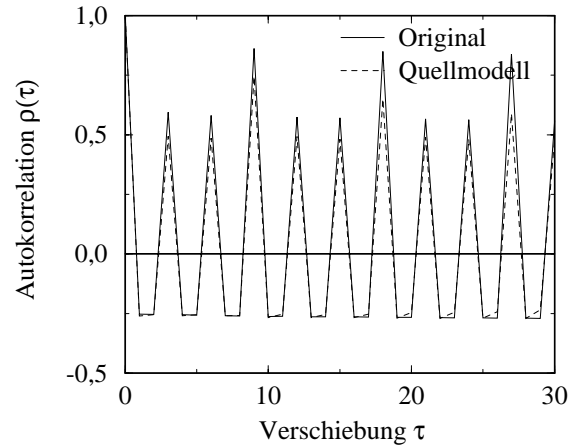
**Bild 5.17:** Verteilungsdichten (Original und 4/9/20-Modell) für MPEG-Daten



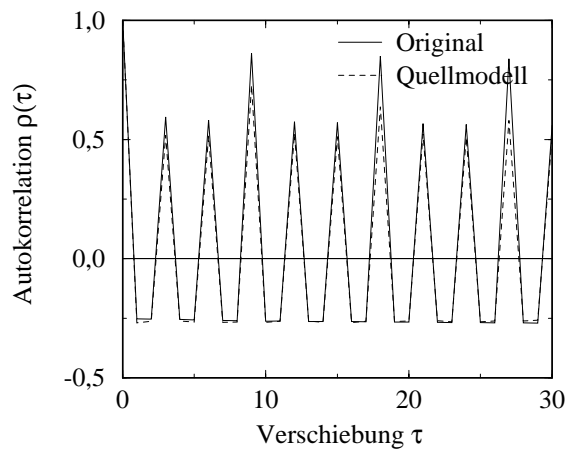
**Bild 5.18:** Autokorrelogramme (Original und 4/9/20-Modell) für MPEG-Daten



**Bild 5.19:** Verteilungsdichten (Original und 9/80/50-Modell) für MPEG-Daten



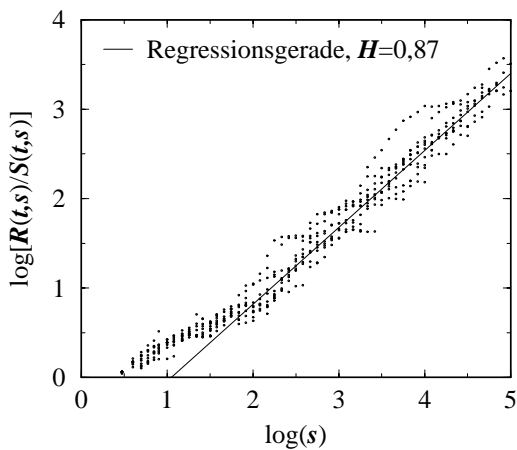
**Bild 5.20:** Autokorrelogramme (Original und 9/80/50-Modell) für MPEG-Daten



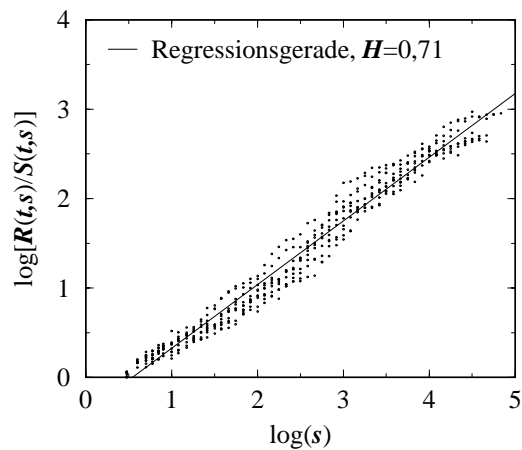
**Bild 5.21:** Autokorrelogramme (Original und 6/9/20/5-Modell) für Vorhersageweite von fünf Schritten,  $P=5$

### 5.7.3.2 LRD-Verhalten

Die bisherigen Ergebnisse zeigen eine gute Reproduktion des zyklischen Verhaltens, der Korrelation und der Verteilung der MPEG-Sequenzen. Im folgenden wird die Modellierung des LRD-Verhaltens durch das Quellmodell untersucht. Eine quantitative Aussage über das LRD-Verhalten ist über die Bestimmung des Hurst-Parameters möglich, der direkt ein Maß für die Langzeitkorrelation darstellt (s. Abschnitt 2.2.3). Die Bestimmung des Hurst-Parameters erfolgt grafisch durch die RS-Analyse. Die Bilder 5.22 und 5.23 zeigen die RS-Analyse für den Originalprozeß und den durch das Quellmodell erzeugten Prozeß sowie jeweils eine Regressionsgerade, deren Steigung eine Schätzung des Hurst-Parameters darstellt. Die ermittelten Werte  $H=0,87$  für den Originalprozeß und  $H=0,71$  für den modellierten Prozeß zeigen einerseits, daß die MPEG-Daten eine sehr starke Langzeitkorrelation aufweisen und andererseits, daß durch das Quellmodell trotz der begrenzten Berücksichtigung vergangener Werte ebenfalls ein signifikantes LRD-Verhalten realisiert werden kann.



**Bild 5.22:** RS-Analyse für Originaldaten



**Bild 5.23:** RS-Analyse für Modelldaten  
(9+5s9/200/70-Modell)

Zur Erzielung dieses Ergebnisses wird ein 9+5s9/200/70-Modell gewählt, d. h. ein Modell mit zwei Eingängen, bei dem neben den 9 vergangenen Folgenwerten auch 5 über jeweils 9 Folgenwerte gemittelte Werte rückgekoppelt werden und der Verteilungsprognose als Eingangswerte zur Verfügung stehen. Durch die gegenüber den bisherigen Modellen höhere Komplexität dieses Modells kann das LRD-Verhalten einer MPEG-Quelle besser reproduziert und damit ein größerer Hurst-Parameter erzielt werden. Im Vergleich dazu kann durch ein 9/80/50-Modell ein Hurst-Parameter von 0,58 und durch ein 9+5s9/80/50-Modell ein Hurst-Parameter von 0,67 erzielt werden. Man erkennt daran, daß durch Verwendung mehrwertiger Quellmodelle mit aggregierter Rückführung eine erhebliche Steigerung der Reproduktionsgüte erzielt wird.

Wie weiter oben erwähnt, ist die Reproduktion des LRD-Verhaltens nicht für alle Systeme erforderlich. Als Anhaltspunkt kann hier die Größe der Puffer dienen, die im zu modellierenden System vorhanden sind. Zur Verdeutlichung wird an einem einfachen Modell gezeigt, wann die Reproduktion des LRD-Verhaltens angezeigt ist und wann nicht. Das betrachtete Modell besteht aus einer Anzahl von  $S$  Verkehrsquellen mit Ankunftsabständen  $T_{A_1}, \dots, T_{A_S}$ , die jeweils einen MPEG-codierten Videodatenstrom repräsentieren und die auf eine gemein-

same Übertragungsleitung gemultiplext werden (s. Bild 5.24). Um ein realistisches Szenario zu erhalten, wird angenommen, daß die Übertragung über ein ATM-Netz erfolgt (s. Abschnitt 6.2), weshalb eine Segmentierung der MPEG-Rahmen in kleine Pakete gleicher Größe (Zellen) erfolgen muß. Die Zellen werden dann über einen Multiplexer einem endlichen Wartespeicher mit  $W$  Wartepätzen zugeführt, wo die Serialisierung gleichzeitig ankommender Zellen für den Weitertransport über eine einzelne Ausgangsleitung erfolgt. Die Sendezeit auf der Ausgangsleitung wird durch die Bedieneinheit mit Bedienzeit  $T_H$  modelliert. Durch Variieren der Werte von  $T_H$  werden unterschiedliche Auslastungen des Systems simuliert. An dem Wartespeicher können aufgrund gleichzeitiger Ankünfte und langsamer Bedienung durch die Bedieneinheit Verluste auftreten.

Die Bilder 5.25 bis 5.28 zeigen die Simulationsergebnisse für dieses Modell <sup>(1)</sup>, wobei für unterschiedliche Wartespeichergrößen  $W$  die Verlustwahrscheinlichkeit in Abhängigkeit von der Auslastung der Bedieneinheit aufgetragen ist. Man erkennt, daß die Verluste an der Warteschlange zwischen Originaldaten und Modelldaten für wenige Wartepätze sehr gut übereinstimmen. Für eine größere Zahl von Wartepätzen wirkt sich das stärkere LRD-Verhalten der Originaldaten aus, was relativ höhere Verluste zur Folge hat. Diese Ergebnisse belegen, daß in Systemen mit vielen Wartepätzen, die also ein großes „Gedächtnis“ haben, an Modelle für Quellen mit LRD-Verhalten höhere Anforderungen gestellt werden müssen, um dieses Verhalten zu modellieren. Man erkennt hieran auch, daß Simulationen, in denen Videoquellen z. B. durch Poisson-Prozesse modelliert werden und bei denen das System viele Wartepätze hat, die tatsächlichen Verluste stark unterschätzt werden können.

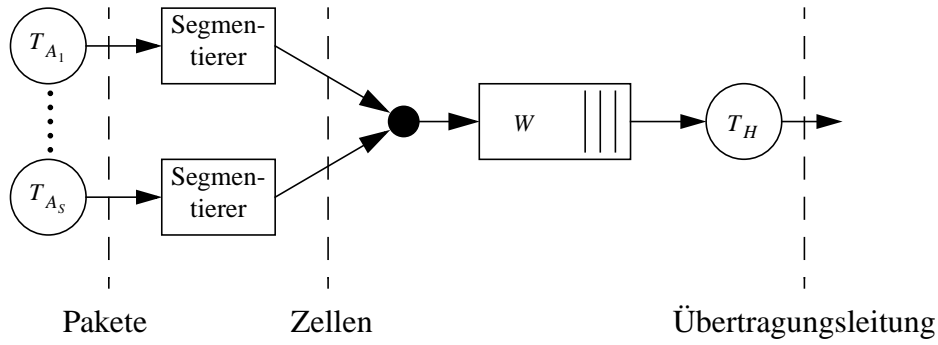
#### 5.7.4 Deterministische Zahlenfolge

In Anknüpfung an das Beispiel in Abschnitt 3.5.2 soll hier gezeigt werden, daß das Quellmodell in der Lage ist, neben stochastischen Prozessen auch deterministische Prozesse zu lernen. Das vorliegende Beispiel besteht aus einer sinusförmigen Folge mit 20 Abtastwerten pro Periode und Amplitude 1. Das Modell hat die Form  $2/20/3$ , d. h. es werden die letzten zwei vergangenen Werte als Gedächtnis zur Vorhersage berücksichtigt. Aufgrund dieser Werte ist für eine Sinusfolge die Vorhersage des jeweils nächsten Werts eindeutig möglich, da daraus der letzte Wert sowie die Tendenz (steigend/fallend) der Zahlenfolge hervorgehen. Das Quellmodell sagt nicht die Werte selbst, sondern deren Dichtefunktion vorher. Diese Dichtefunktionen bestehen im Fall einer ganzzahligen Anzahl von Abtastwerten pro Periode und einer ausreichenden Zahl  $M$  von Verteilungsapproximationen aus je einer Dirac-Distribution an der Stelle des jeweiligen Werts. Eine Verteilungsapproximation besteht aus einer stückweise konstanten Funktion für die Dichte, die die Approximation einer Dirac-Distribution durch  $L=3$  Approximationsbereiche erlaubt (Beschreibung der Verteilungsapproximationen s. Anhang A2). Bild 5.29 zeigt die Repräsentation einer Dirac-Distribution durch 3 Bereiche. Die Höhe des mittleren Bereichs sowie die Breite der äußeren Bereiche streben gegen unendlich, die Breite des mittleren sowie die Höhe der äußeren Bereiche gegen Null.

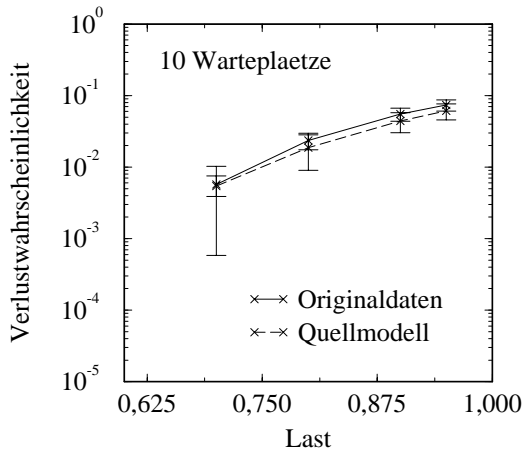
Die Autokorrelationsfunktion einer sinusförmigen Folge hat selbst Sinusform. Bild 5.30 zeigt die Korrelogramme für Original und Modellprozeß. Man erkennt, daß die Korrelation des Modellprozesses das Original sehr gut approximiert, für größere Verschiebungen aber leichte

---

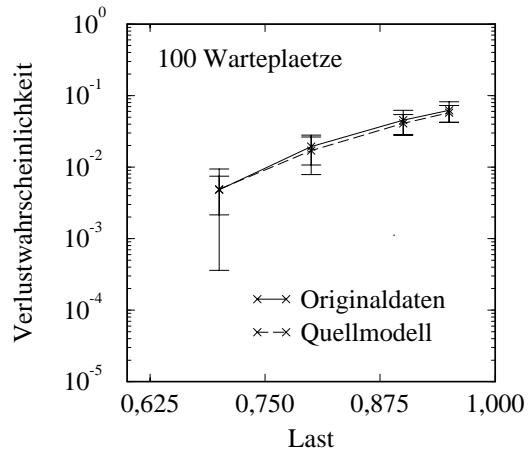
(1) Für jeden Simulationspunkt wird das 95%-Vertrauensintervall angegeben.



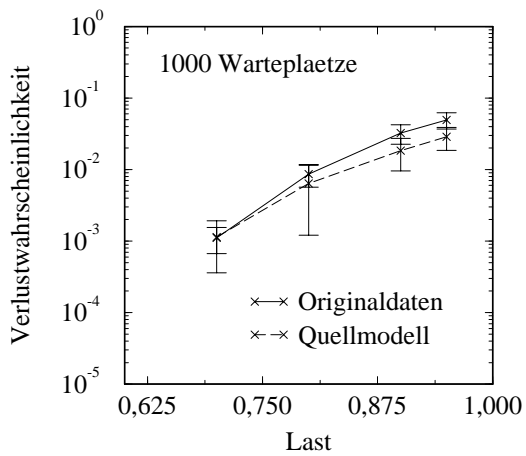
**Bild 5.24:**  $S$  MPEG-Quellen an Zellmultiplexer



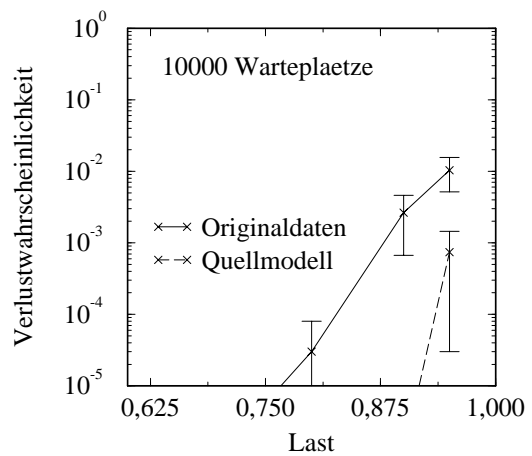
**Bild 5.25:** MPEG-Verkehr an Warteschlange mit 10 Warteplätzen



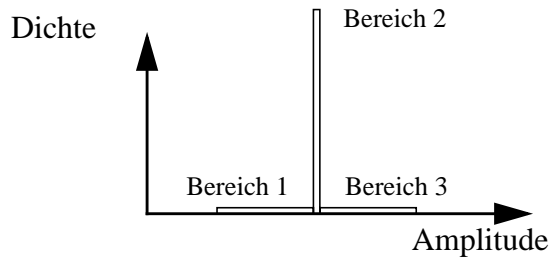
**Bild 5.26:** MPEG-Verkehr an Warteschlange mit 100 Warteplätzen



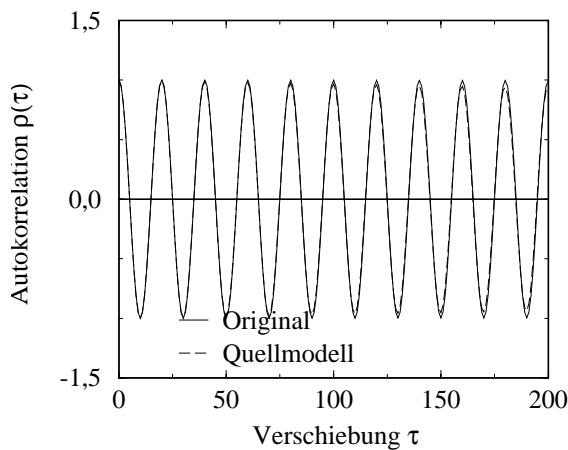
**Bild 5.27:** MPEG-Verkehr an Warteschlange mit 1000 Warteplätzen



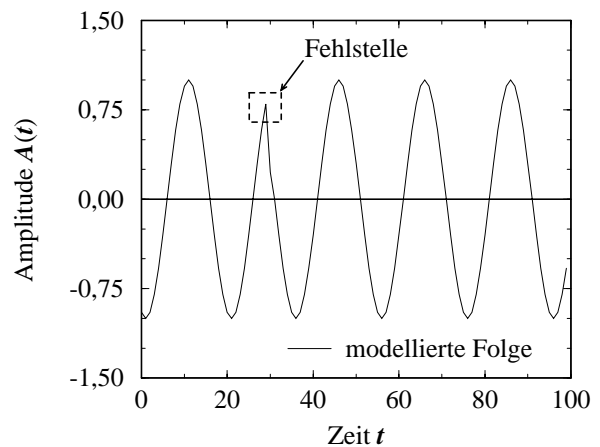
**Bild 5.28:** MPEG-Verkehr an Warteschlange mit 10000 Warteplätzen



**Bild 5.29:** Dichteapproximation einer Dirac-Distribution



**Bild 5.30:** Autokorrelogramme für deterministische Sinusdaten



**Bild 5.31:** Ausschnitt mit Fehlerstelle aus einer modellierten Folge

Abweichungen aufweist. Diese Abweichungen resultieren aus sehr seltenen Fehlern in der modellierten Zeitreihe, wie z. B. in Bild 5.31 dargestellt. Die Ursache solcher Fehler ist die Wahrscheinlichkeit, daß das Quellmodell für ein bestimmtes Gedächtnis einen anderen als den durch die Sinusfolge vorgegebenen Wert berechnet. Diese Wahrscheinlichkeit ist im Idealfall Null. Da in der Realität die Dirac-Distributionen aber nicht ideal nachgebildet werden, bleibt eine sehr kleine Restwahrscheinlichkeit <sup>(1)</sup>, daß durch den Zufallszahlengenerator andere Werte ausgewürfelt werden, die zu Unterbrechungen des zyklischen Verhaltens führen.

## 5.7.5 Ergebnisse

In Tabelle 5.2 sind die wichtigsten Modellparameter sowie die jeweiligen Testergebnisse für alle Beispiele des Abschnitts 5.7 zusammengefaßt.

Die Spalte für den Verteilungstest (Kolmogorow-Smirnow-Test, s. Abschnitt 2.2.2.4) beinhaltet das Ergebnis des Tests sowie den von der Größe der Stichprobe abhängigen Schwellwert, oberhalb dessen nicht von einer Übereinstimmung der Verteilungen ausgegangen werden kann. Man erkennt, daß der Test für MPEG-Daten bei den verwendeten Modellparametern nicht erfüllt wurde. Durch komplexere Modelle könnte auch dieses Ziel erreicht werden. Für das

(1) Der Wert wird mit steigender Lernzyklenzahl immer kleiner. Für die im Beispiel verwendeten Parameter treten Restwahrscheinlichkeiten im Bereich  $10^{-4} \dots 10^{-6}$  auf.



**Tabelle 5.2:** Parameter aller Beispiele zur Quellmodellierung

| Modell    | Parameter |     |     | Testergebnisse                 |  |  |
|-----------|-----------|-----|-----|--------------------------------|--|--|
|           | $N$       | $M$ | $L$ | Verteilungstest <sup>(a)</sup> | Korrelationstest (nicht zyklisch) <sup>(b)</sup> | Korrelationstest (zyklisch) <sup>(c)</sup> |
| AR(1)     | 2         | 10  | 50  | 230 (236)                      | 0,0005   | -  |
| MA(2)     | 3         | 50  | 50  | 217 (236)                      | 0,0015   | -  |
| MA(2)     | 4         | 70  | 50  | 215 (236)                      | 0,0012   | -  |
| ARMA(1,2) | 6         | 70  | 50  | 120 (236)                      | 0,0017   | -  |
| MPEG      | 4         | 9   | 20  | 3079 (430)                     | -  | 0,29                                       |
| MPEG      | 6         | 9   | 20  | 4597 (430)                     | -  | 0,12                                       |
| MPEG      | 9         | 80  | 50  | 2925 (430)                     | -  | 0,05                                       |
| Sinus     | 2         | 20  | 3   | -                              | -  | 0,07                                       |

(a) In Klammern wird die Schwelle für die Ablehnung der Hypothese angegeben, daß beide Verteilungen gleich sind

(b) Schwellwert: 0,01

(c) Schwellwert: 0,1

Beispiel mit deterministischen Sinus-Daten ist kein Ergebnis für den Verteilungstest angegeben, da er für eine diskrete Zahlenfolge nicht durchgeführt werden kann.

Die Ergebnisse der Korrelationstests (s. Tests in Abschnitt 5.5.2) für nicht zyklische Prozesse zeigen, daß die Korrelation der Originalzeitreihen durch das jeweilige Quellmodell sehr gut gelernt wurde und der Testwert weit unter dem Schwellwert 0,01 liegt. Die Testergebnisse für zyklische Prozesse weisen für das 9/80/50-Modell der MPEG-Daten und das Modell für die deterministischen Sinusdaten ein sehr gutes Ergebnis auf. Die Ergebnisse der einfacheren Modelle für MPEG-Daten sind etwas weniger gut, was in Einklang mit der abfallenden Korrelation in den Bildern 5.18 und 5.20 steht.

## 5.8 Zusammenfassung

In diesem Kapitel wird die Anwendung des in Kapitel 3 eingeführten Verfahrens zur Verteilungsprognose am Beispiel der Modellierung von Verkehrsquellen gezeigt. Das aus einer Kombination von Verteilungsprognose und Zufallszahlengenerator bestehende Quellmodell ist in der Lage, sich während einer automatisch ablaufenden Lernphase an sehr unterschiedliche stochastische Prozesse anzupassen. Durch geeignete Parameterwahl kann das stochastische Verhalten der Originalquelle nach Abschluß des Lernprozesses mit beliebiger Genauigkeit durch das Quellmodell reproduziert werden. Obwohl die Leistungsfähigkeit des Verfahrens aufgrund der Vorhersage von Verteilungen naturgemäß bei der Betrachtung von Zufallsprozessen am höchsten ist, können auch deterministische Prozesse durch das Quellmodell nachgebildet werden. Aufgrund dieser Eigenschaft können auch bestimmte nicht-stationäre Prozesse modelliert werden, die sich als Überlagerung eines deterministischen und eines stochastischen Prozesses ansehen lassen. Hierunter fallen z. B. stochastische Prozesse mit Zyklen. Weiterhin

sind auch stochastische Prozesse mit Langzeitkorrelation ohne aufwendige Analyse des Datenmaterials sehr gut modellierbar.

Die auf der Verteilungsprognose basierenden Quellmodelle sind sehr gut für den Einsatz in Simulationen und in Verkehrsgeneratoren für meßtechnische Aufgaben geeignet, da nicht der zeitliche Verlauf einer Zahlenfolge, sondern ihr stochastisches Verhalten nachgebildet wird. Dadurch wird die Abhängigkeit zwischen mehreren gleichartigen Quellen, die bei anderen selbstlernenden Modellen häufig auftritt, vermieden.

## **Kapitel 6**

# **Einsatz der Verteilungsprognose zur Bandbreitereservierung für den ABR-Dienst in ATM-Netzen**

## **6.1 Überblick**

Viele Kommunikationsprotokolle beinhalten Parameter, die für den effizienten und korrekten Betrieb des jeweiligen Protokolls geeignet gewählt werden müssen. Diese Parameter sind entweder konstant oder sie werden während des Betriebs an neue Gegebenheiten angepaßt. Die geeignete Wahl solcher Parameter in Abhängigkeit von Systemzustand und Verkehrsverhalten kann durch Prognosealgorithmen wie der Verteilungsprognose erfolgen.

In diesem Kapitel wird der Einsatz der Verteilungsprognose zur Reservierung von Bandbreite in ATM-Netzen beschrieben. Zu diesem Zweck wird ein Szenario betrachtet, in dem zwei lokale Netze über ein ATM-Netz unter Verwendung des ABR-Protokolls miteinander verbunden werden. Die Bandbreite dieser Verbindung wird dynamisch an Anforderungen und verfügbare Ressourcen angepaßt.

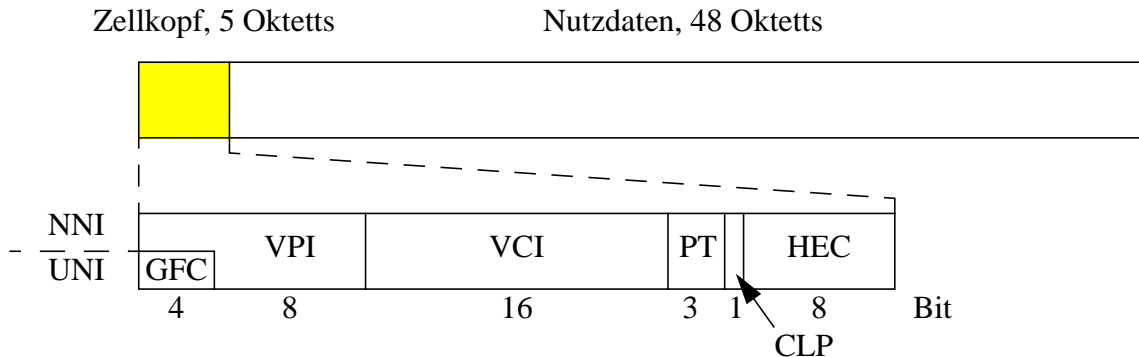
Der erste Teil dieses Kapitels enthält eine Einführung in die ATM-Technologie und die Protokolle, die für die Behandlung des speziellen Problems notwendig sind. Im zweiten Teil werden für das Szenario ein Systemmodell erstellt und die dafür durch Simulation gewonnenen Ergebnisse diskutiert.

## **6.2 Der Asynchrone Transfermode als Transportmechanismus für breitbandige Kommunikationsnetze**

Das Ziel der Entwicklung des Asynchrone Transfermode (ATM) war eine einheitliche Technologie für unterschiedlichste Dienste zur Übertragung von Audio, Video, Daten, etc. sowie die Integration von privaten und öffentlichen Netzen. ATM wurde durch das CCITT (heute ITU-T) als Übertragungstechnologie für das Breitband-ISDN (B-ISDN) gewählt. Die Spezifikation durch ITU-T erfolgt primär mit dem Ziel, ATM in Weitverkehrsnetzen (Wide Area Network, WAN) einzusetzen. Im Gegensatz dazu hat sich 1991 das ATM-Forum als Kooperation verschiedenster Firmen des Kommunikationssektors formiert, die das Ziel hat, durch eigene Standards die Entwicklung von ATM-Produkten und ATM-Diensten zu beschleunigen – speziell im Bereich privater lokaler Netze (Local Area Network, LAN). Durch diese unterschiedlichen Zielsetzungen und Geschwindigkeiten bei der Verabschiedung von Standards haben sich auch einige Unterschiede in der Spezifikation von Protokollen und Diensten ergeben. Da heute hauptsächlich ATM-Produkte aus dem LAN-Bereich auf dem Markt sind, sind die Spezifikationen des ATM-Forums heute von größerer Bedeutung als die der ITU-T.

## 6.2.1 Das ATM-Prinzip

Die Datenübermittlung in ATM erfolgt in Paketen fester Größe, die Zellen genannt werden. Eine ATM-Zelle besteht aus einem Kopf- und einem Nutzdatenfeld (s. Bild 6.1).



**Bild 6.1:** Aufbau einer ATM-Zelle

Das Kopffeld besteht aus den in Tabelle 6.1 beschriebenen Feldern.

**Tabelle 6.1:** Felder des Zellkopfs einer ATM-Zelle

| Feld | Beschreibung  |
|------|---|
| GFC  | Generic Flow Control (4 Bit)<br>Datenflußsteuerung zwischen Endeinrichtung und Vermittlungsknoten. Spezifiziert durch ITU-T [6], nicht verwendet durch das ATM-Forum [53].<br>Dieses Feld entfällt bei Zellen auf Übertragungsabschnitten zwischen Vermittlungsstellen.   |
| VPI  | Virtual Path Identifier (8 bzw. 12 Bit), s. Text.   |
| VCI  | VCI: Virtual Connection Identifier (16 Bit), s. Text.   |
| PT   | Payload Type (3 Bit)<br>Charakterisierung der Nutzlast. Unterscheidung zwischen Benutzerzellen und Zellen zur Netzverwaltung (OAM, Operation, Administration and Management), Unterscheidung zwischen Benutzerzellen, die im Netz Überlast erfahren oder nicht (EFCI, Explicit Forward Congestion Indication), sowie durch höhere Schichten (z. B. durch ATM-Adaptionsschicht) verwendbare Kennung zur Unterscheidung zwischen ATM-SDU-Typen. |
| CLP  | Cell Loss Priority<br>Unterscheidung von Zellen hoher und niederer Priorität. Zellen niederer Priorität können bei Überlast durch das Netz verworfen werden.  |
| HEC  | Header Error Control<br>Prüfbits für den Zellkopf. Mit Hilfe der Prüfbits können ein Bitfehler des Zellkopfs korrigiert und gleichzeitig mehrere Bitfehler erkannt werden. Das HEC-Feld kann ferner zur Erkennung von Zellgrenzen verwendet werden.   |

ATM ist ein verbindungsorientierter Transportdienst, bei dem die Zellen einer Verbindung alle denselben Weg durch das Netz nehmen, der beim Verbindungsaufbau festgelegt wird. Da keine durchgeschalteten Kanäle für die Verbindungen eingerichtet werden und auf den Übertragungsabschnitten die Reihenfolge von Zellen unterschiedlicher Verbindungen beliebig ist, sie also nicht z. B. eine feste Zeitlage zugewiesen bekommen, spricht man von virtuellen Verbindungen (Virtual Connection, VC). Die Kennzeichnung von Zellen unterschiedlicher Verbindungen auf einem Übertragungsabschnitt erfolgt durch die Werte der VPI/VCI-Felder im Zellkopf. Diese Werte werden beim Verbindungsaufbau für jeden Übertragungsabschnitt festgelegt und während der Dauer einer Verbindung durch Vermittlungsknoten auf den für den jeweils nächsten Übertragungsabschnitt gültigen Wert gesetzt. Die abschnittsweise Zuordnung von VPIs und VCIs zu einer Verbindung erfolgt beim Einrichten dieser Verbindung durch das Netzmanagement (Permanent Virtual Connection, PVC) oder durch Signalisierung (Switched Virtual Connection, SVC).

In Ergänzung zum Konzept der virtuellen Verbindungen wird durch ATM das Konzept der virtuellen Pfade realisiert (Virtual Path, VP). Virtuelle Pfade erlauben das Zusammenfassen mehrerer virtueller Verbindungen zu einem Zellstrom. Im allgemeinen existieren auf einem Übertragungsabschnitt mehrere virtuelle Pfade, deren Unterscheidung durch das VPI-Feld des Zellkopfs erfolgt. Wie virtuelle Verbindungen können auch virtuelle Pfade durch Netzmanagement oder Signalisierung eingerichtet werden. Zum Zeitpunkt des Aufbaus eines virtuellen Pfads werden für jeden Übertragungsabschnitt die VPI-Werte vergeben. Das VPI-Feld hat bei Zellen auf Übertragungsabschnitten zwischen Endsystemen und Vermittlungsknoten (User Network Interface, UNI) 8 Bit Länge und zwischen Vermittlungsknoten (Network Node Interface, NNI) 12 Bit Länge, da hier das GFC-Feld entfällt.

Innerhalb eines virtuellen Pfads ist die Unterscheidung von Zellen unterschiedlicher Verbindungen erforderlich und erfolgt durch das VCI-Feld im Zellkopf. In jedem Vermittlungsknoten, an dem ein virtueller Pfad endet, ist die Umsetzung des VCI-Werts für den nächsten VP erforderlich.

Eine wesentliche Anwendung des VP-Konzepts sind VP-Verbindungen zwischen Vermittlungsstellen, die untereinander ein hohes Verkehrsaufkommen haben und über mehrere andere Vermittlungsstellen hinweg miteinander verbunden sind. Da für virtuelle Kanäle ein virtueller Pfad wie ein Übertragungsabschnitt fungiert, muß das VCI-Feld der Zellen nicht in jedem dazwischenliegenden Vermittlungsknoten umgesetzt werden. Dadurch kann dort der Verwaltungs- und Bearbeitungsaufwand für VCIs der betroffenen Verbindungen eingespart werden. Ein weiterer Vorteil des VP-Konzepts ist die relativ einfache Umkonfigurierung des Netzes auf VP-Ebene bei Leitungsausfällen.

## **6.2.2 ATM-Referenzmodell**

Die Beschreibung komplexer Protokolle erfolgt in der Regel hierarchisch mit Hilfe von Schichtenmodellen. Die Bedeutung der einzelnen Schichten und die Beziehungen untereinander werden häufig durch ein Referenzmodell definiert. Das bekannteste Referenzmodell ist das OSI-7-Schichten-Modell der ISO [49].

Das Referenzmodell des B-ISDN wird wie in Bild 6.2 dargestellt. Die dargestellten Schichten werden als Benutzerebene (User Plane) bezeichnet, da hier alle für die Übertragung von

Benutzerdaten relevanten Protokollfunktionen enthalten sind. Häufig erfolgt eine Aufteilung der höheren Schichten in Benutzerebene und Steuerungsebene (Control Plane) sowie eine Fortsetzung des Referenzmodells in die dritte Dimension zur Darstellung einer Management-Ebene, die das Schichten-Management und das Ebenen-Management enthält [52].

|                              |
|------------------------------|
| Höhere Schichten             |
| ATM-Anpassungsschicht (AAL)  |
| ATM-Schicht                  |
| Bitübertragungsschicht (PHY) |

**Bild 6.2:** ATM-Referenzmodell

Die Definition der Schichten des ATM-Referenzmodells unterscheidet sich von den Definitionen des OSI-Modells, daher wird im folgenden die Terminologie des ATM-Referenzmodells verwendet.

Die einzelnen Schichten des Referenzmodells können weiter verfeinert werden (s. Bild 6.3). Die Funktionen der Teilschichten werden in den folgenden Abschnitten beschrieben.

|     |     |  |  |
|-----|-----|--|--|
| AAL | CS  | Dienstspezifische Funktionen,<br>z. B. Ausgleich von Verzögerungsschwankungen,<br>Zellverlusten und Bitfehlern   | } Funktionen des Informationsfeldes,<br>dienstspezifisch |
|     | SAR | Umsetzung der Daten in Zellen und umgekehrt  |  |
| ATM |     | Erzeugung und Entfernen des Kopffeldes,<br>VCI-/VPI-Umsetzung,<br>Multiplexen und Demultiplexen von VC- und VP-<br>Verbindungen,<br>Quellflußkontrolle,<br>Verkehrsklassen | } Funktionen des Zellkopfes,<br>dienstunabhängig         |
| PHY | TC  | Entkoppeln von Zellraten,<br>Erzeugung und Überprüfung der HEC,<br>Zell-Synchronisation,<br>Rahmenerzeugung und -verarbeitung,<br>Rahmenanpassung                          |  |
|     | PM  | Elektrisch-optische Umsetzung<br>Bit-Synchronisation<br>Leitungscodierung  |  |

**Bild 6.3:** Unterteilung und Funktionalität der Schichten des Referenzmodells

### **6.2.2.1 Bitübertragungsschicht**

Die Bitübertragungsschicht (Physical Layer, PHY) wird unterteilt in die PM-Teilschicht (Physical Medium) und die TC-Teilschicht (Transmission Convergence). Die PM-Teilschicht enthält die vom physikalischen Medium abhängigen Funktionen zur Bitübertragung. Die TC-Teilschicht enthält die vom physikalischen Medium unabhängigen Funktionen zur Anpassung der ATM-Zellen an das verwendete physikalische Übertragungssystem (z. B. Synchronous Digital Hierarchy, SDH).

### **6.2.2.2 ATM-Schicht**

Die ATM-Schicht enthält Funktionen zum Erzeugen und Entfernen des Kopffeldes, Funktionen zur Quellflußkontrolle, zum Multiplexen und Demultiplexen von Zellen, sowie Funktionen zur Umsetzung der VCIs und VPIs und ist sowohl vom Medium als auch von Diensten unabhängig. Sie enthält keine Funktionen zur Fehlersicherung der Nutzdaten.

In der ATM-Schicht werden weiterhin die Funktionen der fünf in Tabelle 6.2 aufgeführten Verkehrsklassen realisiert. Sie dienen der Anpassung von Verkehrscharakteristiken an das Netzverhalten. Funktionen wie Verbindungsannahme, Verkehrsflußkontrolle und Belegung von Betriebsmitteln können in der Regel die Wahl einer Verkehrsklasse beeinflussen und durch diese beeinflußt werden. Die Definitionen von ATM-Forum [11] und ITU-T [56] der Verkehrsklassen weisen deutliche Unterschiede auf (s. Tabelle 6.2). In den weiteren Ausführungen wird auf die Definitionen des ATM-Forums Bezug genommen.

Die Verkehrsklassen werden entsprechend ihrer Echtzeitfähigkeit klassifiziert. Echtzeitfähige Verkehrsklassen sind CBR und Echtzeit-VBR (Abkürzungen s. Tabelle 6.2), nicht echtzeitfähig sind nicht-Echtzeit-VBR, UBR und ABR. Dabei wird Echtzeitfähigkeit beschrieben durch strenge Anforderungen an Verzögerungen und Verzögerungsschwankungen einer Verbindung. Die Auswahl einer Verkehrsklasse für eine Verbindung erfolgt beim Verbindungsaufbau in Abhängigkeit von den angegebenen Verkehrsparametern [11, 56] (Spitzenzellrate, mittlere Zellrate, minimale Zellrate, Dienstgüteparameter, etc. ).

In Abschnitt 6.4 wird die ABR-Verkehrsklasse detailliert beschrieben.

### **6.2.2.3 ATM-Anpassungsschicht**

Die ATM-Anpassungsschicht dient zur Anpassung der ATM-Schicht an die unterschiedlichen Dienste höherer Schichten. Die Spezifikation der verschiedenen Typen der Anpassungsschicht (ATM Adaptation Layer, AAL) erfolgte durch die ITU-T [54, 55] und wurde vom ATM-Forum übernommen. Von den ursprünglich geplanten AAL-Typen wurde nur ein Teil spezifiziert. Tabelle 6.3 zeigt die verschiedenen AAL-Typen und ihre Eigenschaften.

Die AALs lassen sich nach drei Kriterien klassifizieren:

- Zeitbeziehung zwischen Sender und Empfänger (fest: AAL 1; nicht fest: AAL 3/4, 5)
- Variabilität der Bitrate (konstant: AAL 1; variabel: AAL3/4, 5)

**Tabelle 6.2:** ATM-Verkehrsklassen

| Verkehrsklasse                               |   | Beschreibung   |
|--|---|--|
| ATM-Forum                                    | ITU-T                                     |  |
| CBR<br>(Constant Bit Rate)                   | DBR<br>(Deterministic Bit Rate)           | Die konstantbitratige Verkehrsklasse wird für Verbindungen verwendet, die während ihrer gesamten Lebensdauer eine feste Bandbreite benötigen (z. B. für unkomprimierte Sprache oder Video, Emulation durchgeschalteter Verbindungen).<br><br>Es wird garantiert, daß für alle Zellen, die mit nicht mehr als der ausgehandelten Spitzenbitrate gesendet werden, die ausgehandelte Dienstgüte eingehalten wird. |
| rt-VBR<br>(real-time Variable Bit Rate)      | nicht spezifiziert<br>(for further study) | Der Echtzeit-VBR-Dienst wird für Verbindungen mit Echtzeitanforderungen (geringe Verzögerungen und Verzögerungsschwankungen im Netz) verwendet, deren Übertragungsrate während der Verbindungsdauer schwankt (z. B. für komprimiertes Video).  |
| nrt-VBR<br>(non-real-time Variable Bit Rate) | SBR<br>(Statistical Bit Rate)             | Im Gegensatz zum Echtzeit-VBR-Dienst wird dieser Dienst für Verbindungen ohne enge Anforderungen an die Verzögerung im Netz verwendet.   |
| ABR<br>(Available Bit Rate)                  | ABR<br>(Available Bit Rate)               | ABR stellt die einzige Verkehrsklasse mit Flußsteuerung dar. Ihr Ziel ist das Auftreten sehr kleiner Verluste und die faire Aufteilung der verfügbaren Bandbreite zwischen allen ABR-Verbindungen. Die Flußsteuerung wird durch verschiedene Rückkopplungsmechanismen zwischen Endgeräten und Vermittlungseinrichtungen realisiert. Es werden keine Echtzeitanwendungen unterstützt.                           |
| UBR<br>(Unspecified Bit Rate)                | kein Äquivalent                           | Dieser Dienst ist für Anwendungen gedacht, die keine Echtzeitanforderungen haben (z. B. Filetransfer, E-Mail). Er gibt keine Garantien und arbeitet auf einer „Best-Effort“-Basis.   |
| kein Äquivalent                              | ABT<br>(ATM Block Transfer)               | Für diesen Dienst werden Netzbetriebsmittel für jeden zu übertragenden Datenblock reserviert, nicht für die Dauer einer Verbindung.  |



**Tabelle 6.3:** Typen der ATM-Anpassungsschicht

| AAL-Typ | Beschreibung   |
|---------|--|
| AAL 0   | nicht spezifiziert<br>Für AAL 0 existieren unterschiedliche Interpretationen: vereinfachter AAL 1 zur Sprachübertragung [23] bzw. leerer AAL zur Übertragung reiner ATM-Zellen.  |
| AAL 1   | Verwendung für Sprache, Video mit konstanter Bitrate oder Emulation von Leitungvermittlung.  |
| AAL 2   | AAL 2 war geplant für variabel-bitratige Verbindungen mit fester Zeitbeziehung zwischen Sender und Empfänger.<br>Wurde inzwischen neu definiert für das vereinfachte Interworking zwischen Mobilkommunikations-Netzen und ATM. |
| AAL 3/4 | Geeignet zur Übertragung verbindungsorientierter (AAL 3) oder verbindungsloser Daten (AAL 4) variabler Bitrate, bei denen keine feste Zeitbeziehung zwischen Sender und Empfänger vorliegt.                                    |
| AAL 5   | Vereinfachte Implementierung von AAL 3/4, nur für verbindungsorientierte Kommunikation. Verwendung in erster Linie zur Rechnerkommunikation.   |

- Verbindungstyp (verbindungsorientiert: AAL 1, 3, 5; verbindungslos: AAL 4)

Die größte praktische Bedeutung hat zur Zeit AAL 5, da er am einfachsten zu implementieren und für den noch dominierenden LAN-Verkehr am besten geeignet ist. Von den anderen AALs wird sich vermutlich aufgrund der Forderung nach echtzeitfähigen Verbindungen für Multimediaanwendungen am ehesten der AAL 1 durchsetzen.

Es existiert keine feste Zuordnung zwischen ATM-Verkehrsklassen und den Typen der ATM-Anpassungsschicht. Diese Zuordnung wird verbindungsindividuell je nach Dienstgüteparametern beim Verbindungsaufbau getroffen.

Die Wahl eines AALs sowie die Wahl einer ATM-Verkehrsklasse für eine bestimmte Verbindung ist weitgehend davon abhängig, ob der AAL bzw. die Verkehrsklasse in allen betroffenen Komponenten (Endsysteme, Vermittlungsknoten) implementiert sind. Dies kann zum gegenwärtigen Zeitpunkt system- und herstellerübergreifend nur für AAL5 und UBR vorausgesetzt werden.

#### **6.2.2.4 Höhere Schichten**

Die höheren Schichten definieren Protokollfunktionen, die es Anwendungen erlauben, über ATM zu kommunizieren. Hier sind z. B. alle Anwendungen, die auf Internet-Protokolle aufbauen (TCP/IP Stack) oder Multimedia-Anwendungen zu sehen. Die zwischen der jeweiligen Anwendung und der AAL-Schicht liegenden Schichten können leer sein, falls die Anwendung direkt einen AAL-Dienst verwendet.

Ausgehend von der Annahme, daß die meisten Anwendungen noch längere Zeit auf den TCP/IP-Protokollen basieren werden, wurden Möglichkeiten zur Anpassung dieser Anwendungen an die AAL-Schicht geschaffen. Hierfür haben sich zwei Lösungen durchgesetzt. Die erste Lösung – IP over ATM [89] – stammt von der IETF (Internet Engineering Task Force) und realisiert auf relativ einfache Weise den Transport von IP-Paketen über ATM. Es gilt die Einschränkung, daß alle an einer Kommunikation beteiligten Geräte direkt am ATM-Netz angeschlossen sein müssen. Bei der zweiten Lösung wird durch eine Anpassungsschicht oberhalb der AAL-Schicht ein Schicht-2-Protokoll emuliert, um weiterhin die vorhandene TCP/IP-Protokollsoftware einsetzen zu können. Diese sehr aufwendige, durch das ATM-Forum standardisierte Lösung, wird daher LAN-Emulation genannt [8, 9, 10]. Im Gegensatz zu IP over ATM erlaubt LAN-Emulation auch den Anschluß von Endgeräten über konventionelle LANs.

### 6.3 Überlastabwehr und -vermeidung

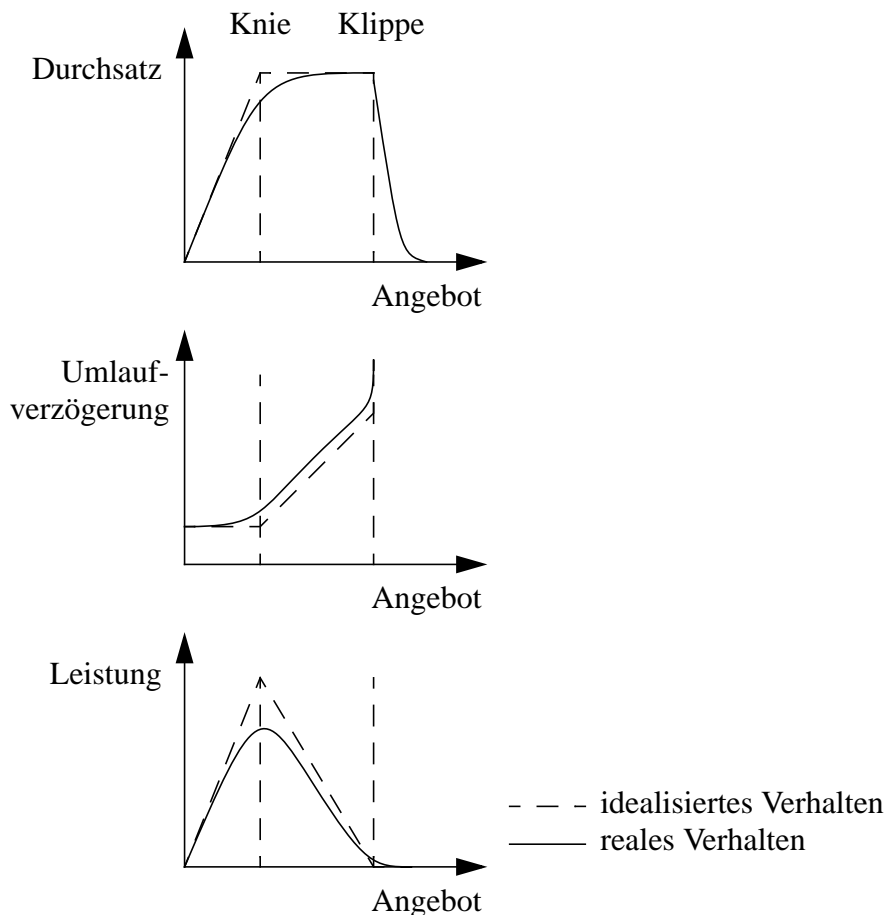
Wenn in Kommunikationsnetzen das Verkehrsangebot, das durch Anwendungen verursacht wird, die Netzkapazität überschreitet, entstehen Überlastsituationen, die zu erhöhten Verzögerungen, abgewiesenen Verbindungswünschen und bis zu einem Zusammenbruch des Durchsatzes führen können. Bild 6.4 zeigt das prinzipielle Netzverhalten [21, 59, 60]: Bei ansteigendem Angebot steigt der Durchsatz relativ linear bis zu einem Knick, dem „Knie“. Ab hier ist der maximale Durchsatz erreicht, das System ist in Sättigung. Ab einem zweiten ausgezeichneten Punkt, der „Klippe“, kommt das System in Überlast und der Durchsatz fällt rapide ab. Ähnliches läßt sich für die Umlaufverzögerung einer Nachricht durch das Netz bis zurück zum Sender beobachten. Diese Verzögerung steigt ab dem „Knie“ leicht und ab der „Klippe“ sehr stark an. Zusätzlich zu diesen Effekten können Hystereseerscheinungen auftreten, durch die sich das Verhalten bei Erreichen vom Verhalten bei Verlassen der Überlastsituation unterscheidet.

Als optimaler Arbeitspunkt für Netzbetreiber und Netznutzer stellt sich der Bereich des „Knies“ dar, da hier das Netz eine hohe Auslastung hat und die Verzögerungen noch nicht stark ansteigen. Zur Bewertung des Arbeitspunkts wird in [59] der Quotient aus Durchsatz und Umlaufverzögerung definiert, die „Leistung“. Sie nimmt am „Knie“ den maximalen Wert an.

Strategien, die die unerwünschten Begleiterscheinungen von Überlastsituationen in Kommunikationssystemen bekämpfen, werden in zwei Kategorien unterteilt, zum einen Strategien zur Überlastabwehr und zum anderen Strategien zur Überlastvermeidung. Üblicherweise versucht man, Kommunikationsnetze nicht in einen Überlastzustand kommen zu lassen, sondern durch gezielte Steuerung oder Regelung im optimalen Bereich um das „Knie“ aus Bild 6.4 zu halten; es werden also Strategien zur Überlastvermeidung eingesetzt.

Zur Optimierung der Leistung einzelner Verbindungen werden z. B. Verfahren zur Flußkontrolle wie Fenstermechanismen oder Ratenregelungen verwendet.

In ATM-Netzen werden sowohl Verfahren zur Überlastvermeidung als auch Verfahren zur Überlastabwehr eingesetzt. Überlastvermeidung erfolgt durch geeignete Verbindungsannahmeverfahren (Connection Admission Control, CAC), Wegesucheverfahren (Routing), Quellflußkontrolle (Usage Parameter Control, UPC), Verkehrsformung (Traffic Shaping) und Priorisierung unterschiedlicher Verkehrsklassen. Durch die genannten Verfahren wird ein Systemzustand angestrebt, der die vorhandenen Ressourcen möglichst gut auslastet und keine



**Bild 6.4:** Überlastvermeidung

Dienstgütevereinbarungen mit Nutzern verletzt. Treten dennoch in seltenen Fällen Überlastsituationen ein, müssen Überlastabwehrverfahren wie das Suchen neuer Wege für bestehende Verbindungen, Signalisierung von Überlast zwischen Netzknoten und Endsystemen (Explicit Congestion Notification) oder Flußkontrollverfahren eingesetzt werden. Dazu gehört auch das Verwerfen von Zellen, deren CLP-Bit gesetzt ist.

## 6.4 Die ABR-Verkehrsklasse

### 6.4.1 Allgemeines

Die ABR-Verkehrsklasse wurde sowohl von der ITU-T als auch vom ATM-Forum spezifiziert. Da die beiden Spezifikationen Unterschiede aufweisen und Implementierungen bisher nur aufgrund der Spezifikation des ATM-Forums vorliegen, wird diese hier im weiteren zugrundegelegt.

ABR ist die einzige ATM-Verkehrsklasse, die Rückkopplungsmechanismen in Form einer Flußkontrolle beinhaltet. Durch die Flußkontrolle passen ABR-Quellen ihre Senderate der momentan im ATM-Netz zur Verfügung stehenden Bandbreite an. Die zur Verfügung stehende Bandbreite wird in den Vermittlungsknoten fair zwischen unterschiedlichen ABR-Verbindungen aufgeteilt. Die Rückkopplung erfolgt in Form von sogenannten RM-Zellen (Resource Management), die die erlaubte Zellrate der Verbindung steuern. Die Verteilung der erlaubten

Senderate auf mehrere ABR-Verbindungen kann durch unterschiedliche Algorithmen erfolgen [11], die meist auf den in [21] angegebenen Fairneßdefinitionen beruhen.

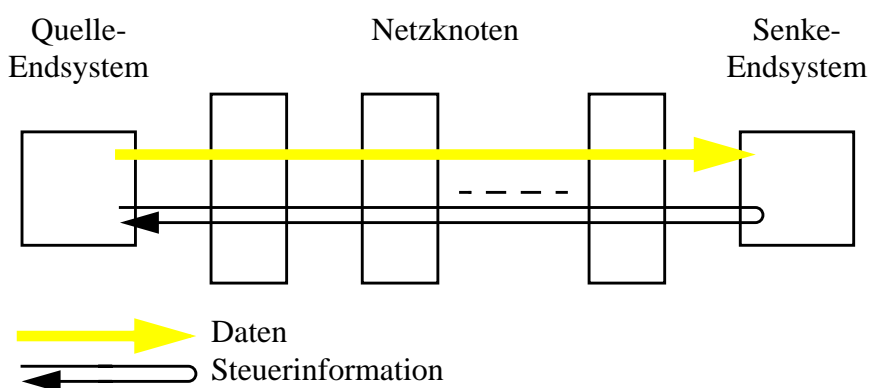
Durch die Rückkopplung kann eine niedrige Zellverlustwahrscheinlichkeit erzielt werden, da bei Bandbreiteengpässen im ATM-Netz die entsprechende Quelle gedrosselt wird. Bezüglich der Zellverzögerungszeit im Netz werden keine Garantien abgegeben; sie kann durch die Verwendung großer Zellpuffer im Netz relativ groß werden. Die Flußkontrolle führt bei ABR dazu, daß Zellen, die ein Endsystem wegen nicht verfügbarer Bandbreite nicht senden kann, im Endsystem gepuffert werden. Im Gegensatz dazu werden bei UBR Zellen immer gesendet und ggf. im Netz vor überlasteten Übertragungsabschnitten zwischengespeichert.

Das ABR-Protokoll beinhaltet eine Vielzahl von Parametern, was einerseits zu komplexen Implementierungen und andererseits zu einer nichttrivialen Verwaltung dieses Dienstes führt. Dies steht im Gegensatz zu den Vorteilen von ABR.

Für den Betreiber eines ATM-Netzes bietet die ABR-Verkehrsklasse den Vorteil, daß ungenutzte Bandbreite schnell verfügbar gemacht wird, daß die Puffer innerhalb des Netzes kleiner dimensioniert werden können als bei UBR und daß trotz hoher Netzauslastung Garantien bezüglich der maximalen Verlustwahrscheinlichkeit abgegeben werden können. Aus Sicht eines Teilnehmers bietet die ABR-Verkehrsklasse ebenfalls Vorteile, da z. B. Verluste eher beim Teilnehmer auftreten und somit die höheren Schichten sofort geeignet reagieren können, wogegen bei UBR die Verluste eher im Netz auftreten und daher eine höhere Verzögerung auftritt.

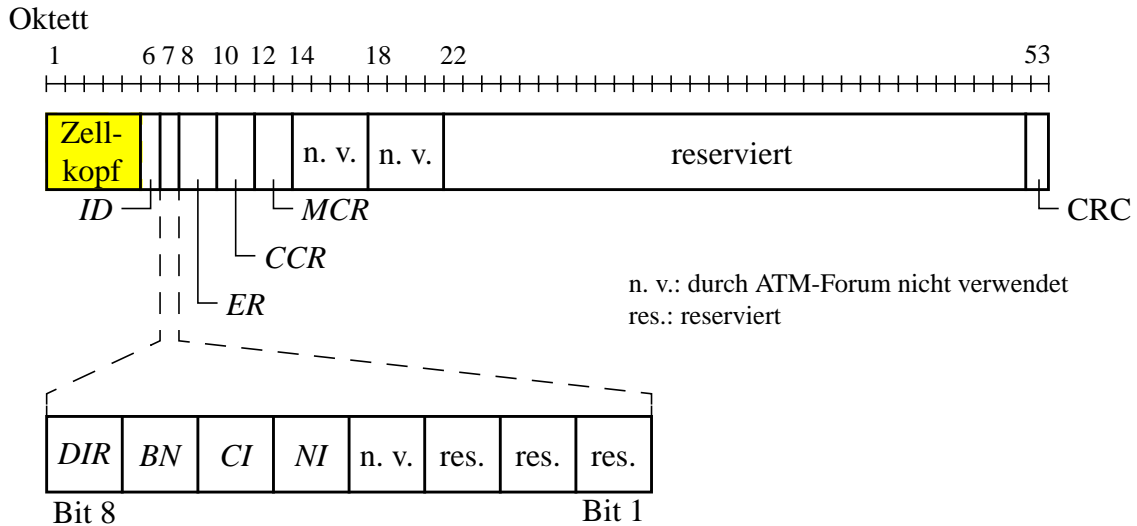
## 6.4.2 Das ABR-Protokoll

Bild 6.5 zeigt einen Überblick über den Informations- und Steuerfluß für eine unidirektionale ABR-Verbindung. Die Daten werden von der Quelle zur Senke über mehrere Vermittlungsknoten übertragen. Parallel dazu werden in beiden Richtungen Steuerinformationen übertragen, die in der Vorwärtsrichtung zwischen die Nutzdaten eingestreut werden.



**Bild 6.5:** Informationsfluß und zugehörige Regelschleife

Die Übertragung der Steuerinformation erfolgt durch sogenannte RM-Zellen (Resource Management Cells) über denselben VCI/VPI wie die Datenzellen. Diese RM-Zellen stellen spezielle OAM-Zellen (Operation, Administration and Management Cells) dar. Bild 6.6 zeigt den Aufbau einer RM-Zelle. Im folgenden wird die Bedeutung der einzelnen Felder kurz erläutert und



- ID:** Identifier  
Gibt den Protokolltyp für die OAM-Zelle an
- DIR:** Direction  
Richtung der RM-Zelle (vorwärts oder rückwärts)
- BN:** Backward Explicit Congestion Notification Cell  
Nicht durch die Quelle generierte RM-Zelle zur Überlastanzeige
- CI:** Congestion Indication
- NI:** No Increase  
Keine Erhöhung der Senderate erlaubt
- ER:** Explicit Cell Rate
- CCR:** Current Cell Rate
- MCR:** Minimum Cell Rate
- CRC:** Cyclic Redundancy Check  
10 Prüfbits zur Sicherung der Nutzdaten

**Bild 6.6:** RM-Zellen

in den nächsten Abschnitten eine kurze Beschreibung des Verhaltens der unterschiedlichen, am ABR-Protokoll beteiligten Komponenten gegeben.

Das ID-Feld gibt den Typ der OAM-Zelle an und hat für RM-Zellen den Wert 1. Der Nachrichtentyp wird im folgenden Oktett kodiert: *DIR* gibt an, ob die RM-Zelle in Richtung Senke oder in Richtung Quelle unterwegs ist, mit *BN=1* werden RM-Zellen gekennzeichnet, die nicht von der Quelle generiert wurden (s. u.), *CI=1* signalisiert eine Überlastsituation und *NI=1* eine Situation, die zu Überlast führen kann. Die Felder *ER*, *CCR*, *MCR* beinhalten jeweils eine Zellrate in Festkommadarstellung: *ER* ist die maximale Rate, mit der die entsprechende Quelle senden darf, *CCR* die momentane Senderate der Quelle und *MCR* die minimale, beim Verbin-

dungsaufbau ausgehandelte Zellrate der Quelle. Eine RM-Zelle wird, wie alle OAM-Zellen, durch einen 10-Bit-CRC abgeschlossen. Einige Felder der RM-Zelle werden in der Spezifikation des ATM-Forums nicht verwendet, wohl aber in der entsprechenden ITU-T-Spezifikation [56]. Andere Felder sind für zukünftige Anwendungen reserviert.

Prinzipiell werden zwei Arten von RM-Zellen unterschieden: RM-Zellen, die zu Zeitpunkten gesendet werden, die durch das ABR-Protokoll vorgesehen sind und die zusammen mit den Datenzellen die erlaubte Rate nicht überschreiten (in-rate cells, IR-Zellen), sowie RM-Zellen, die zu nicht vorgesehenen Zeitpunkten gesendet werden oder zu einem Überschreiten der erlaubten Rate führen (out-of-rate cells, OOR-Zellen). OOR-Zellen werden durch das CLP-Bit des Zellkopfes als niederprior gekennzeichnet und können von Netzkomponenten verworfen werden.

In den folgenden Abschnitten werden die an einer ABR-Verbindung beteiligten Netzkomponenten – Quelle, Senke, Vermittlungsknoten – kurz charakterisiert.

#### 6.4.2.1 Quelle

In diesem Abschnitt wird das Verhalten einer ABR-Quelle beschrieben, allerdings ohne auf alle Details des ABR-Protokolls einzugehen. Tabelle 6.4 zeigt die wesentlichen Parameter einer ABR-Quelle.

Die Zellrate der Quelle muß immer kleiner oder gleich der erlaubten Zellrate  $ACR$  sein, die sich nach der momentanen Netzauslastung richtet. Der Wert von  $ACR$  bewegt sich immer im Bereich zwischen  $MCR$  und  $PCR$ . Sendet eine Quelle mit einer höheren Rate, so sollte der erste Vermittlungsknoten dies durch Überwachung der vereinbarten Verbindungsparameter (Usage Parameter Control, UPC) feststellen und die Zellen durch Setzen des CLP-Bits markieren.

Die Anpassung von  $ACR$  an die momentane Netzsituation wird durch RM-Zellen gesteuert, deren Felder durch Senke und Vermittlungsknoten entsprechend des Lastzustands gesetzt werden. Es bestehen im wesentlichen zwei Möglichkeiten, einer ABR-Quelle zu signalisieren, ob  $ACR$  erhöht oder verringert werden soll, wobei auch Kombinationen möglich sind:

- Signalisierung durch Überlastanzeige

Die Ratenanpassung erfolgt durch Auswertung der CI- und NI-Bits der rücklaufenden RM-Zellen. Ist das CI-Bit gesetzt, wird dadurch eine Überlastsituation einer anderen Netzkomponente signalisiert und die ABR-Quelle muß die erlaubte Senderate,  $ACR$ , um mindestens den Faktor  $RDF$  verringern. Ist  $CI=0$  und  $NI=1$ , so wird von einer Netzkomponente eine möglicherweise beginnende Überlast signalisiert und  $ACR$  darf nicht erhöht werden. Nur wenn  $CI=NI=0$  ist, darf  $ACR$  um maximal den Wert  $RIF*PCR$  erhöht werden.

- Explizite Raten-Signalisierung

Hier erfolgt die Ratenanpassung durch direkte Vorgabe eines maximalen Werts für  $ACR$  durch den ER-Wert rücklaufender RM-Zellen ( $ER_b$ ). In diesem Fall gibt die Quelle einen gewünschten Wert, z. B.  $PCR$ , im ER-Feld der vorwärtslaufenden RM-Zellen vor ( $ER_f$ ). Alle Vermittlungsknoten und die Senke tragen anschließend auf dem Hin- und Rückweg

**Tabelle 6.4:** ABR-Parameter

| Bezeichnung | Beschreibung  |
|-------------|---|
| <i>ICR</i>  | Initial Cell Rate <sup>(a)</sup><br>Direkt nach Verbindungsaufbau gültige Zellrate.   |
| <i>MCR</i>  | Minimum Cell Rate <sup>(a)</sup><br>Garantierte minimale Zellrate einer Verbindung.   |
| <i>PCR</i>  | Peak Cell Rate <sup>(a)</sup><br>Maximale Zellrate einer Verbindung.  |
| <i>ADTF</i> | <i>ACR</i> Decrease Time Factor <sup>(a)</sup><br>Zeit zwischen dem Senden von RM-Zellen, nach deren Überschreitung die Senderate auf <i>ICR</i> reduziert wird.            |
| <i>FRTT</i> | Fixed Round-Trip Time <sup>(a)</sup><br>Summe aller festen Verzögerungen sowie der Übertragungs- und Laufzeiten in Quelle, Senke und Netzkomponenten.                       |
| <i>Nrm</i>  | Maximale Zellenzahl, die eine Quelle senden darf, bevor die nächste RM-Zelle gesendet werden muß <sup>(a)</sup> .   |
| <i>RDF</i>  | Rate Decrease Factor <sup>(a)</sup><br>Parameter zur multiplikativen Verringerung der Zellrate.   |
| <i>RIF</i>  | Rate Increase Factor <sup>(a)</sup><br>Parameter zur additiven Erhöhung der Zellrate.   |
| <i>CRM</i>  | Missing RM-cell count <sup>(b)</sup><br>Anzahl RM-Zellen, die in Vorwärtsrichtung gesendet werden dürfen, bevor auf die erste RM-Zelle in Rückrichtung gewartet werden muß. |
| <i>ACR</i>  | Allowed Cell Rate <sup>(c)</sup><br>Momentan erlaubte maximale Senderate einer Quelle. Nach oben durch <i>PCR</i> beschränkt.   |

(a) wird beim Verbindungsaufbau zwischen Quelle, Netz und Senke ausgehandelt

(b) Wert ist implementierungsspezifisch.

(c) berechnete Größe

der Zelle die von ihnen momentan unter Einhaltung der Dienstgüte maximal verarbeitbare Rate in das ER-Feld ein, wobei nur größere ER-Werte überschrieben werden. Dadurch erhält die Quelle das Minimum aller beteiligten Komponenten zurück. Die Modifikation von *ACR* erfolgt wie bei der Signalisierung durch Überlastanzeige abhängig von *CI* und *NI*. Anschließend wird *ACR* auf das Minimum aus *ACR* und *ER* gesetzt, aber nicht kleiner als *MCR*.

Neben den genannten Regeln für die Erhöhung und Verringerung von *ACR* gibt es eine Reihe weiterer Regeln, die für beide Signalisierarten gelten. Bei langandauernder Niedriglast einer Quelle oder großen Verzögerungen im Netz wird *ACR* verringert (s. *ADTF* bzw. *CRM* in Tabelle 6.4). Die erlaubte Zellrate direkt nach dem Verbindungsaufbau ist durch den Wert von

*ICR* gegeben. Weiterhin kann bei sehr niedrigen Senderaten das Problem auftreten, daß die Quelle trotz im Netz zur Verfügung stehender Bandbreite diese nicht nutzen kann, da keine RM-Zellen unterwegs sind, die die freie Bandbreite signalisieren könnten. In diesem Fall besteht die Möglichkeit, durch außerplanmäßige RM-Zellen das Aufbrechen des Regelkreises zu vermeiden, die dann als Out-of-rate gekennzeichnet sein müssen (niederprior). Die Sendezeitpunkte für diese RM-Zellen werden durch den Standard nicht definiert und sind implementierungsspezifisch.

Wie bereits erwähnt, kann eine ABR-Quelle über das ER-Feld der RM-Zellen eine bestimmte Rate im Bereich zwischen *MCR* und *PCR* anfordern. Bisherige Realisierungen setzen den Wert *PCR* ein, um immer die maximal verfügbare Bandbreite zugeteilt zu bekommen. Hierfür können durch einen geeigneten Algorithmus auch niedrigere Werte berechnet werden, die die momentan benötigte Bandbreite besser repräsentieren (s. Abschnitt 6.5).

#### **6.4.2.2 Senke**

Neben dem im letzten Abschnitt beschriebenen Verhalten einer ABR-Quelle muß auch das Verhalten einer ABR-Senke definiert werden. Die Aufgabe der ABR-Senke besteht primär darin, die in Vorwärtsrichtung laufenden RM-Zellen „herumzudrehen“ und zur Quelle zurückzusenden, nachdem ggf. einige der Felder der RM-Zellen modifiziert wurden (*ER*, *CI*).

Bei ABR-Verbindungen werden üblicherweise Daten in beiden Richtungen übertragen, wodurch in beiden Endgeräten sowohl die Funktionalität einer ABR-Quelle als auch die einer ABR-Senke realisiert werden muß. Der entstehende Zellstrom besteht dann in beiden Richtungen aus Datenzellen und RM-Zellen der beiden Quellen und Senken.

#### **6.4.2.3 Vermittlungsknoten**

Neben der Vermittlung von Zellen und der Überwachung von Verbindungsparametern haben Vermittlungsknoten im Falle von ABR-Verbindungen zusätzlich zu der Vermittlung von Zellen virtueller Verbindungen die Aufgabe, die Regelschleife über RM-Zellen aufrechtzuerhalten. Hierbei unterscheidet der Standard zwei Möglichkeiten:

- Der Vermittlungsknoten wertet die RM-Zellen aller ABR-Verbindungen aus und trägt ggf. abhängig von seinem momentanen Lastzustand neue Werte für die ER-, CI-, NI-Felder ein. Er kann allerdings auch out-of-rate RM-Zellen generieren, falls z. B. die schnelle Signalisierung geänderter Zellraten erforderlich ist.
- Der Vermittlungsknoten unterteilt die ABR-Regelschleife in Segmente, indem in dem Knoten eine sogenannte virtuelle Quelle und Senke realisiert wird. Diese Alternative scheint zwar die Möglichkeit der Entkopplung und der Bildung kürzerer Regelschleifen zu bieten, hat aber den großen Nachteil, daß bei ungünstigen Konstellationen sehr große Puffer benötigt werden. Dies tritt beispielsweise bei einer großen Differenz zwischen Sende- und Empfangsrate von virtueller Quelle und Senke auf.

Generell existieren für das Verhalten von Vermittlungsknoten nur wenige Regeln und es bleibt viel Spielraum für herstellerepezifische Realisierungen. Alle im Standard vorgeschlagenen Algorithmen haben das Ziel gemeinsam, die Bandbreite fair unter allen ABR-Verbindungen zu verteilen, die nach Abzug der für CBR- und VBR-Verkehr benötigten Bandbreite noch verfüg-



bar ist. Dabei muß die Dienstgüte *aller* über den Vermittlungsknoten laufenden Verbindungen eingehalten werden. Die Möglichkeiten, die ein Vermittlungsknoten zur Signalisierung von Lastzuständen und Ratenänderungen hat, lassen sich grob in drei Klassen einteilen:

- Verwendung des EFCI-Bits des ATM-Zellkopfs (EFCI Marking).
- Binäre Signalisierung über die CI- und NI-Bits der RM-Zellen (Relative Rate Marking).
- Verwendung des ER-Felds der RM-Zellen zur expliziten Vorgabe einer maximalen Senderate (Explicit Rate Marking).

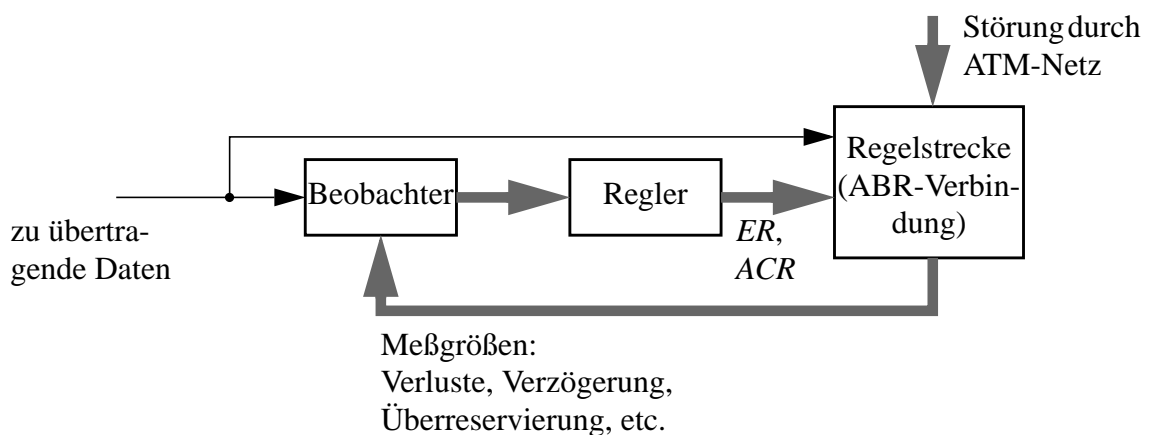
Die jeweiligen Reaktionen einer ABR-Quelle werden in Abschnitt 6.4.2.1 beschrieben. Ein beispielhafter Algorithmus zur Berechnung der verbindungsindividuellen Bandbreite für ABR-Verkehr in einem ATM-Vermittlungsknoten wird in Abschnitt 6.7.1 beschrieben.

### 6.4.3 Regelungstechnische Betrachtung

Die ABR-Verkehrsklasse ist, wie in Bild 6.5 angedeutet, die einzige Verkehrsklasse, die eine Regelschleife zur optimalen Einstellung des Arbeitspunktes realisiert. Dieser Arbeitspunkt soll sowohl für die einzelnen ABR-Verbindungen als auch für das gesamte Netz optimal gewählt werden. Dabei sind unterschiedliche Optimierungskriterien zu erfüllen:

- Aus Betreibersicht soll das Netz möglichst weit ausgelastet werden, ohne daß die Dienstgüte einzelner Verbindungen darunter leidet. Die verfügbare Bandbreite soll auf alle Nutzer des ABR-Dienstes fair verteilt werden. Ziel ist das Erreichen eines optimalen Betriebszustands („Knie“, s. Abschnitt 6.3). Dazu ist im Falle sich anbahnender oder bereits existierender Überlast an einem Knoten die Drosselung der ABR-Quellen erforderlich.
- Aus Benutzersicht sind möglichst geringe Verluste und Verzögerungen das Ziel. Falls der Benutzer Gebühren entrichtet, ist ein weiteres Optimierungskriterium die Minimierung der erlaubten, aber nicht genutzten Zellrate (Überreservierung), da sich dadurch die Kosten für die Bereitstellung dieser Ressourcen reduzieren.

Die genannten Ziele lassen sich als regelungstechnisches Problem formulieren (s. Bild 6.7).



**Bild 6.7:** Regelungstechnisches Modell für eine ABR-Verbindung

Dabei lassen sich folgende Komponenten eines Regelmodells identifizieren:

- **Regler**  
Die Aufgabe des Reglers in diesem Kontext ist die Beeinflussung einer ABR-Verbindung in einer Weise, die die oben formulierten Ziele verfolgt. Die Beeinflussung erfolgt durch geeignete ER-Werte in den zur Quelle zurücklaufenden RM-Zellen, die wiederum eine Änderung des ACR-Werts der Quelle zur Folge haben. Der Regler ist bei ABR über mehrere Netzkomponenten verteilt realisiert. Dazu gehören alle durch eine Verbindung betroffenen ATM-Vermittlungsknoten sowie die Endgeräte. In den Endgeräten beeinflussen z. B. die ABR-Parameter die Veränderung von ACR durch  $CI$ ,  $NI$  und  $ER_b$  und damit die Dynamik und Wirksamkeit des Reglers.
- **Regelstrecke**  
Die Regelstrecke ist durch die eigentliche ABR-Verbindung und ihre Eigenschaften (Verzögerungen, maximale Bitraten der beteiligten Übertragungsabschnitte, etc.) gegeben. Der restliche ATM-Verkehr tritt dabei als Störgröße auf, der z. B. die Verzögerungen der ABR-Verbindung oder die verfügbare Bandbreite beeinflusst.
- **Beobachter**  
Der Beobachter dient zur Aufbereitung der für den Regelalgorithmus benötigten Meßgrößen (verfügbare Bandbreite, benötigte Bandbreite pro ABR-Verbindung, Verluste, etc.). Falls die Bandbreitereservierung der Quelle durch Vorgabe eines ER-Werts in ihren RM-Zellen an den tatsächlichen Bandbreitebedarf angepaßt werden soll, muß auch ein Meßwert zur Bestimmung des momentanen Bandbreitebedarfs durch den Beobachter aufbereitet und dem Regler zugeführt werden.

Der Teil des Beobachters, der den benötigten Bandbreitebedarf für die ABR-Verbindung bestimmt, ist durch die ABR-Spezifikation nicht definiert. Für diesen Teil werden im folgenden Alternativen beschrieben und bezüglich ihrer Leistungsfähigkeit verglichen.

## 6.5 Realisierungsalternativen für ABR-Quellen

In diesem Abschnitt werden verschiedene Möglichkeiten für einen Teil der in einer ABR-Quelle implementierten Algorithmen angegeben. Es handelt sich dabei um die Bestimmung des benötigten Bandbreitebedarfs für eine ABR-Verbindung. Die betrachteten Algorithmen setzen eine explizite Signalisierung der erlaubten Rate über das ER-Feld der RM-Zellen durch ABR-Vermittlungsknoten an ABR-Quellen voraus (Explicit Rate Marking). Weiterhin wird die Möglichkeit des ABR-Protokolls ausgenutzt, das ER-Feld für Reservierungsanforderungen durch eine ABR-Quelle zu verwenden.

Das Ziel der im folgenden beschriebenen Algorithmen ist die Reduzierung einer allgemeinen Kostenfunktion, die neben den tatsächliche Kosten für die Nutzung einer ABR-Verbindung auch Verluste und Wartezeiten berücksichtigt.

Für die tatsächlich auftretenden Kosten einer ABR-Verbindung wird angenommen, daß sie neben einem Anteil, der von der Datenmenge abhängt, auch Kosten für zugeteilte Netzressourcen beinhalten. Daher ist die Minimierung der zugeteilten, aber nicht genutzten Übertragungskapazität ein weiteres Optimierungsziel des ABR-Regelkreises. Prinzipiell können sowohl Messungen in der ABR-Quelle als auch in den Vermittlungsknoten durchgeführt und als Ein-

gangsgößen für den verteilten Regelalgorithmus verwendet werden. In dieser Arbeit erfolgt eine Beschränkung auf Algorithmen in der ABR-Quelle, die mit dem ABR-Protokoll konforme Erweiterungen darstellen.

### 6.5.1 Reservierung der ausgehandelten Maximalrate

Die einfachste und heute in den meisten ABR-fähigen Geräten implementierte Methode zur Bandbreiteanforderung ist die Belegung des ER-Felds von RM-Zellen der Quelle durch den PCR-Wert, der beim Verbindungsaufbau ausgehandelt wurde. Durch den Regelalgorithmus wird dann grundsätzlich versucht, diese Grenze zu erreichen, indem der Quelle abhängig von der momentanen Netzauslastung Bandbreite zugeteilt wird.

### 6.5.2 Reservierung der zuletzt beobachteten Rate

Das im letzten Abschnitt beschriebene Verfahren reserviert stets die für eine Verbindung ausgehandelte maximale Bandbreite. Dadurch werden auch in Perioden, in denen das tatsächliche Verkehrsaufkommen geringer ist, Betriebsmittel für diese Verbindung reserviert, was zu unnötigen Kosten führt.

Um diesen Nachteil zu umgehen, ist eine geeignete Abschätzung des tatsächlich benötigten Bandbreitebedarfs notwendig. Falls die Datenrate der ABR-Verbindung eine starke positive Autokorrelation hat, kann durch Messung der momentanen Datenrate im LAN eine Abschätzung für die Reservierung gewonnen werden. In [94] wird dieser Ansatz zur Kopplung von DQDB-Netzen (Distributed Queue Dual Bus) über ATM verwendet.

### 6.5.3 Anwendung der Verteilungsprognose

Die Reservierung durch Verwendung der zuletzt gemessenen Rate folgt dem Trend dieser Werte. Da dort jedoch die Vergangenheit des beobachteten Prozesses nicht weiter ausgewertet wird, ist durch Prognoseverfahren eine Verbesserung der Resultate zu erwarten.

Die in Kapitel 3 eingeführte Verteilungsprognose ist ein Prognoseverfahren, mit dessen Hilfe unterschiedlichste statistische Kenngrößen zukünftiger Datenraten vorhergesagt werden können. Es kann dazu eingesetzt werden, die Verteilung zukünftiger Datenraten basierend auf vergangenen Meßwerten vorherzusagen und aus diesen Verteilungen einen geeigneten Wert für die Bandbreitereservierung zu berechnen. Die Auswertung erfolgt hier im Gegensatz zu den

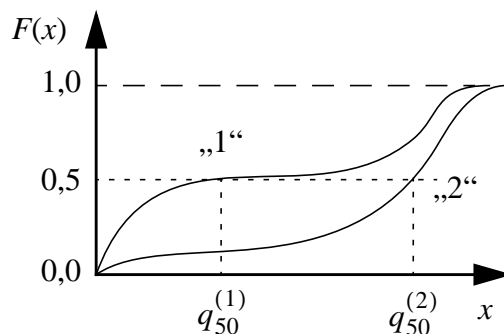


Bild 6.8: Beispiel zur Quantilbestimmung

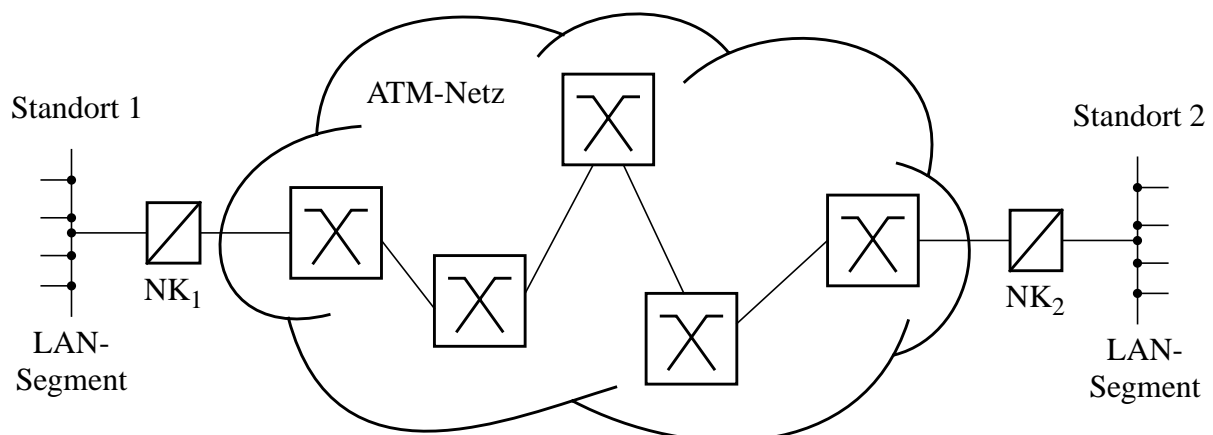
Anwendungen in Kapitel 5 nicht durch Bestimmung eines zufälligen Werts aufgrund der vorhergesagten Verteilung, sondern durch Berechnung eines Quantils. Eine Diskussion verschiedener Auswertemethoden erfolgt in Abschnitt 6.8.1. Bild 6.8 zeigt beispielhaft die Bestimmung von 50%-Quantilen  $q_{50}^{(1)}$  und  $q_{50}^{(2)}$  aus zwei unterschiedlichen Verteilungsfunktionen „1“ und „2“. Die Weiterverwendung der so gewonnenen Werte wird in Abschnitt 6.7.2 im Detail beschrieben.

Der Lernprozeß für die Anpassung der Verteilungsprognose an einen Datenstrom entspricht der Beschreibung in Kapitel 3.

## 6.6 LAN-Kopplung über ABR

ATM wird heute häufig zur Kopplung bestehender LAN-Infrastrukturen (Ethernet, Token Ring) als Backbone eingesetzt. Dadurch werden beispielsweise LAN-Segmente eines Firmennetzes untereinander verbunden, die zu mehreren Standorten der Firma gehören. Je nach Verteilung der Anwendungen und Ressourcen kann der Verkehr über den ATM-Backbone beträchtliche Ausmaße annehmen. Zur flexiblen und effizienten Bandbreitenutzung werden üblicherweise die beiden Verkehrsklassen UBR und ABR eingesetzt.

Im weiteren wird ein Szenario betrachtet, das aus zwei Ethernet-Segmenten besteht, die über ein ATM-Netz gekoppelt sind (s. Bild 6.9). Die Ankopplung der Ethernet-Segmente an das ATM-Netz erfolgt über zwei Netzkoppeleinheiten ( $NK_1$  und  $NK_2$ ), in denen die Anpassung der unterschiedlichen Protokollhierarchien durchgeführt wird. Für dieses Beispiel wird angenommen, daß die Kopplung durch eine bidirektionale AAL-5-Verbindung über ABR erfolgt. Die Verbindung der zwei Standorte erfolgt über eine 50 km lange ATM-Strecke, die über 5 Vermittlungsknoten führt. Dieses ATM-Netz kann entweder durch denselben Betreiber verwaltet werden wie das betrachtete LAN oder durch einen Anbieter von Netzdienstleistungen. Auf jeden Fall wird angenommen, daß sowohl eine Abrechnung der genutzten als auch der reservierten Netzressourcen erfolgt. Alle Übertragungsabschnitte auf dem Weg durch das ATM-Netz sind als SDH-Strecken mit einer Bitübertragungsrate von 155 Mbit/s realisiert, was einer durch ATM-Zellen nutzbaren Übertragungsrate von 149,76 Mbit/s entspricht.



**Bild 6.9:** LAN-Kopplung über ATM

Die für die Simulation des Beispiels verwendeten Daten wurden durch Messung an einem Ethernet-Netz gewonnen, das aus zwei Segmenten besteht, die über eine Brücke gekoppelt sind <sup>(1)</sup>.

Eine Zusammenfassung der Ergebnisse einer repräsentativen Messung ist in Tabelle 6.5 dargestellt.

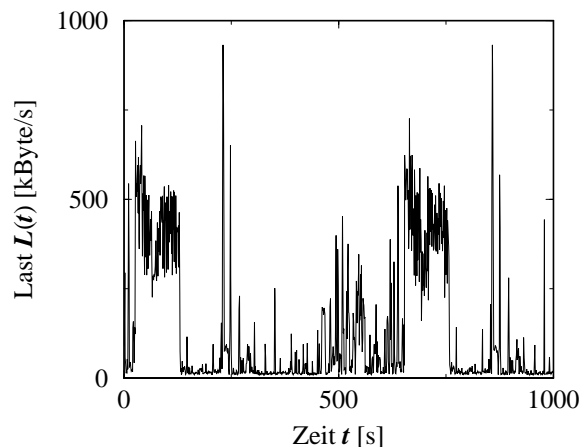
**Tabelle 6.5:** Kenngrößen des Ethernet-Verkehrs zwischen den Segmenten

| Kenngröße                                    | Wert            |
|--|-----------------|
| Dauer des Meßintervalls                      | ca. 10 min      |
| Anzahl Pakete im Meßintervall <sup>(a)</sup> | ca. 165.000     |
| Mittlerer Ankunftsabstand                    | 3,9 ms          |
| Mittlere Paketlänge                          | 434,8 Bytes     |
| Mittlere Auslastung                          | ca. 9,6%        |
| Mittlere Datenrate                           | ca. 121 kByte/s |

(a) Beschränkung durch den Speicherplatz des Meßgerätes

Im weiteren wird angenommen, daß die Brücke durch einen ATM-Backbone ersetzt wird. Daher wird für den Verkehr über den ATM-Backbone die gemessene Charakteristik des Verkehrs über die Brücke zugrundegelegt.

In Bild 6.10 ist der Lastverlauf im LAN über der Zeit dargestellt. Man erkennt die starken Schwankungen im Bandbreitebedarf, die typisch für das Verkehrsaufkommen in lokalen Netzen sind. Wenn aufgrund dieses Lastverhaltens eine feste Bandbreitereservierung durchgeführt werden soll, muß entweder eine viel höhere Übertragungsrate reserviert werden, als im Mittel benötigt wird, oder es müssen bei Reservierung einer niedrigeren Rate starke Verluste in Kauf genommen werden.

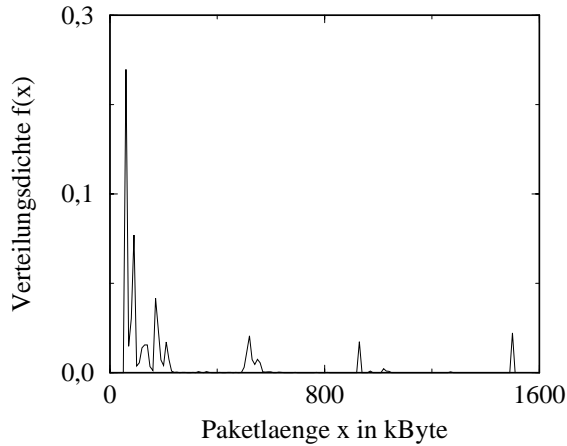


**Bild 6.10:** Lastverlauf im LAN

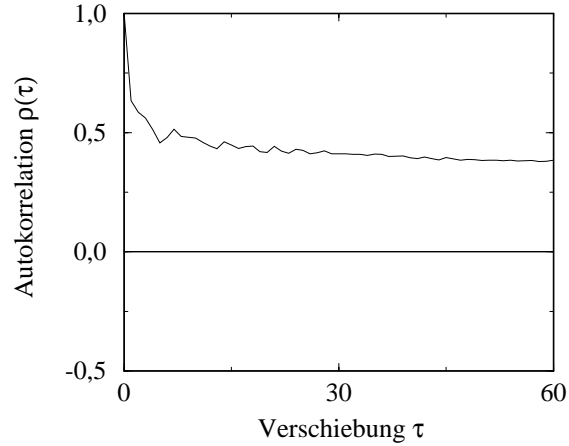
---

(1) Die Messungen wurden am Institutsnetz des Instituts für Nachrichtenvermittlung und Datenverarbeitung der Universität Stuttgart durchgeführt.

Die Bilder 6.11 und 6.12 zeigen die Paketlängenverteilung auf dem Ethernet und das zugehörige Korrelogramm. Man erkennt eine starke Häufung bestimmter Paketgrößen. Die Pakete minimaler Größe sind dabei hauptsächlich auf Interaktiv-Verkehr und die Pakete maximaler Größe auf Filetransfers zurückzuführen. Eine weitere Beobachtung ist die starke Korrelation der Paketgrößen, die weitgehend mit der Korrelation der Last übereinstimmt.



**Bild 6.11:** Dichte der Paketlängenverteilung im LAN



**Bild 6.12:** Korrelation der Paketlängen im LAN

Die Auswertungen der Meßdaten zeigt ein sehr starkes LRD-Verhalten (s. Abschnitt 6.7.2). Dies ist in Übereinstimmung mit an anderer Stelle durchgeführten Beobachtungen, nach denen LAN-Verkehr in großen Netzen stark selbstähnlichen Charakter hat [73]. Die durch die Selbstähnlichkeit implizierte langandauernde serielle Korrelation des LAN-Verkehrs kann für Vorhersagen des stochastischen Verhaltens durch die Verteilungsprognose genutzt werden.

Die LAN-Kopplung über ABR hat je nach Verbindungsparametern und Auslastung des ATM-Netzes Wartezeiten und Verluste zur Folge. Die Wartezeiten werden durch die Zellpuffer in Netzkoppeleinheiten und Vermittlungsknoten sowie durch Übertragungszeiten im Netz verursacht. Für Interaktiv-Verkehr in einem LAN sollten Obergrenzen für feste Verzögerungen im Bereich weniger Millisekunden (<50 ms) eingehalten werden, da dauerhaft höhere Verzögerungen bei interaktiver Tätigkeit als störend empfunden werden. Durch variable Pufferfüllstände entstehen phasenweise höhere Verzögerungen, deren Obergrenze ebenfalls begrenzt sein sollte. Hier wird eine Toleranzschwelle von etwa 0,5 sec angenommen. Falls das ATM-Netz kurzfristig nicht die zur Übertragung aller LAN-Pakete erforderliche Bandbreite zur Verfügung stellen kann, treten an den Netzkoppeleinheiten Paketverluste auf. Solange diese Verluste im Bereich weniger Prozent liegen, können sie toleriert werden, da im LAN selbst ähnliche Verluste auftreten und LAN-Protokolle so ausgelegt sind, daß diese Fehler korrigiert werden. Von den Verlusten im ATM-Netz wird angenommen, daß sie weit unter den Verlusten im LAN und der Netzkoppeleinheit liegen und daher vernachlässigt werden können.

Die Übertragung der LAN-Pakete über ATM ist mit einem geringfügigen Overhead verbunden. Bei Verwendung des AAL-5 Dienstes werden an jedes Paket acht Oktetts zur Fehlersicherung angefügt und das entstehende Paket in ATM-Zellen segmentiert. Dabei ist in der Regel die letzte Zelle unvollständig gefüllt. Das ABR-Protokoll führt zu einer weiteren Erhöhung der benötigten Bandbreite, indem nach rund  $N_{rm}-1$  Datenzellen eine RM-Zelle übertragen wird

(s. Abschnitt 6.4.2.1). Der auf einer AAL-5-Verbindung über ABR zu erwartende Verkehr wird für die beschriebenen LAN-Daten in Tabelle 6.6 zusammengefaßt.

**Tabelle 6.6:** Kenngrößen der ABR-Verbindung

| Kenngröße                                      | Wert                                |
|--|-------------------------------------|
| Mittlere benötigte Zellenzahl pro Paket        | ca. 9,7 cells/packet <sup>(a)</sup> |
| Mittlere Rate der ABR-Datenzellen              | 2662 cells/s <sup>(a)</sup>         |
| Mittlere Rate der ABR-Zellen (incl. RM-Zellen) | 2748 cells/s <sup>(b)</sup>         |
| Mittlere Last durch LAN-Verkehr im ATM-Netz    | 0,78% <sup>(c)</sup>                |
| Mittlere Rate der RM-Zellen                    | 85,9 cells/s <sup>(b)</sup>         |
| Mittlere Zeit zwischen zwei RM-Zellen          | 11,6 ms                             |

(a) 48 Oktetts Nutzdaten pro ABR-Datenzelle

(b) Jede 32te Zelle ist eine RM-Zelle,  $N_{rm}=32$

(c) Bei 155 MBit/s Linkbitrate (149,76 MBit/s nutzbar)

Die ATM-Vermittlungsknoten, die durch die betrachtete Verbindung berührt werden, realisieren eine Priorisierung von Zellen der unterschiedlichen ATM-Verkehrsklassen in folgender Reihenfolge: CBR, rt-VBR, nrt-VBR, ABR (RM-Zellen), ABR (Datenzellen), UBR. Durch diese Priorisierung ist u. a. sichergestellt, daß RM-Zellen der ABR-Verbindung weit seltener verloren gehen als entsprechende Datenzellen. Als weitere Folge werden ABR-Datenzellen durch RM-Zellen in Vermittlungsknoten überholt, wenn sich dort eine Warteschlange für die Datenzellen gebildet hat. Auf diese Weise ist die schnelle Signalisierung innerhalb der ABR-Regelschleife gewährleistet, auch wenn für die ABR-Daten Wartezeiten entstehen.

Für das beschriebene Szenario muß eine geeignete Abstraktion gefunden werden, für die die interessierenden Fragestellungen hinreichend gut beantwortet werden können. Auf der Basis dieser Abstraktion wird anschließend ein Simulationsmodell für das beschriebene Szenario erstellt. In dem vorliegenden Fall ist die Betrachtung auf Zellebene notwendig, da nur dort die detaillierte Beobachtung des ABR-Protokolls möglich ist. Die Simulation eines ATM-Netzes auf Zellebene erfordert üblicherweise wegen der großen Zahl von Verbindungen, durch die eine realistische Auslastung gekennzeichnet ist, einen sehr großen Aufwand, der sich in langen Simulationszeiten ausdrückt. Verschärfend kommt hier das LRD-Verhalten der LAN-Daten hinzu, das zu deutlich höheren Simulationszeiten führen kann als bei Daten ohne LRD-Verhalten, um eine bestimmte Ergebnisgüte zu erhalten (s. Abschnitt 4.2). Um diese Problematik zu vermeiden, wird hier das Netzmodell so gewählt, daß zwar das prinzipielle Netzverhalten nachgebildet wird, die Nachbildung anderer als der betrachteten virtuellen Verbindung aber nicht erforderlich ist. Dadurch ist trotz der gewählten Abstraktion eine ausreichend umfangreiche Simulation bei vertretbarer Rechenzeit möglich.

Das Modell des ATM-Netzes muß daher die variierende Gesamtbelastung eines ATM-Netzes derart berücksichtigen, daß die Auswirkungen auf die betrachtete ABR-Verbindung möglichst realitätsnah sind. Eine genauere Beschreibung des Netzmodells erfolgt in Abschnitt 6.7.1.

Das Modell der Netzübergangsknoten wird im Gegensatz zum Modell des ATM-Netzes sehr detailliert ausgeführt und bildet viele Eigenschaften eines realen ATM-Endgerätes nach. Die Beschreibung erfolgt in Abschnitt 6.7.2.

## 6.7 Modellierung

### 6.7.1 Modell für das ATM-Netz

#### 6.7.1.1 Übersicht

Das Modell des ATM-Netzes muß die tatsächlichen Abläufe in einem realen ATM-Netz stark vereinfachen, ohne daß sich dadurch das Verhalten nach außen – hier gegenüber der betrachteten ABR-Verbindung – stark ändert. Diese Aufgabe ist wegen der sehr vielen Freiheitsgrade bei der Modellierung und der vielen festzulegenden Netzparameter problematisch.

Die Abstraktion des in Bild 6.9 dargestellten ATM-Netzes erfolgt durch das Modell in Bild 6.13. Dabei werden folgende Eigenschaften eines realen ATM-Netzes berücksichtigt:

- Der zeitabhängige Verlauf der gesamten Netzlast wird durch einen Zustandsprozeß (s. Abschnitt 6.7.1.4) und durch die Last der betrachteten ABR-Verbindung modelliert.
- Die zufälligen Verzögerungen der RM-Zellen durch das ATM-Netz werden durch einen festen Anteil und einen negativ-exponentiell verteilten variablen Anteil beschrieben (s. Abschnitt 6.7.1.2).
- Der gleichzeitige Aufenthalt mehrerer RM-Zellen im ATM-Netz wird durch einen unbeschränkten Wartespeicher modelliert. Dies ist dadurch motiviert, daß die Zellpuffer realer ATM-Vermittlungsknoten Platz für mehrere 10.000 Zellen bieten.

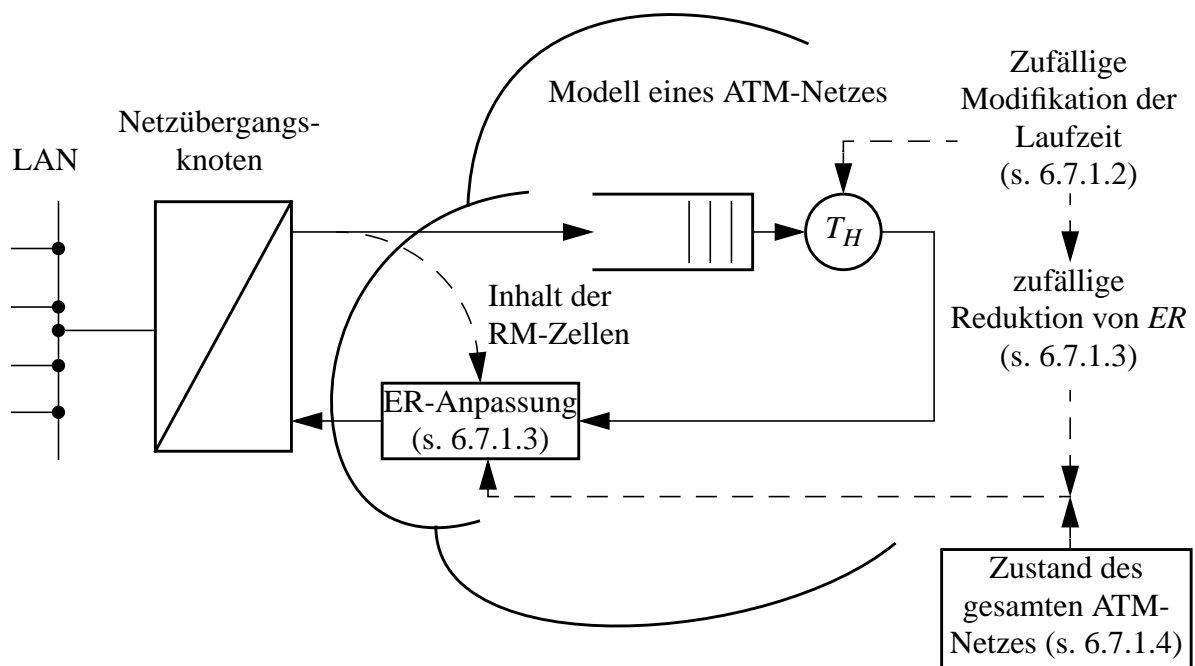


Bild 6.13: Modell des ATM-Netzes



- Die Modifikation des ER-Werts der rückwärtslaufenden RM-Zellen ( $ER_b$ ) erfolgt in Abhängigkeit von der momentanen Netzlast und der Verzögerung der jeweiligen RM-Zellen im Netz (s. Abschnitt 6.7.1.3).
- Der Inhalt vorwärtslaufender RM-Zellen wird durch einen Fairneß-Algorithmus sofort für Berechnungen verwendet. Das bedeutet, daß der Inhalt im Netz ankommender RM-Zellen sofort ausgewertet wird und damit Einfluß auf andere, durch die Quelle lange vorher abgeschickte rücklaufende RM-Zellen hat. Der hier verwendete Fairneß-Algorithmus wird in Abschnitt 6.7.1.3 beschrieben.
- Das Modell berücksichtigt keine Verluste, da die RM-Zellen eine höhere Priorität als die ABR-Datenzellen haben und Datenzellen nicht weiter betrachtet werden, wenn sie einmal in das ATM-Netz eingetreten sind.

### 6.7.1.2 Feste und variable Verzögerung im ATM-Netz

Die Modellierung der Laufzeit von RM-Zellen erfolgt durch einen konstanten und einen variablen Anteil. Der konstante Anteil ergibt sich aus den konstanten Bearbeitungs- und Übertragungszeiten in Netzkomponenten sowie der Laufzeit auf dem Medium. Der variable Anteil entspricht den Schwankungen der Durchlaufverzögerung in Netzkomponenten aufgrund der veränderlichen Last durch andere Verbindungen und den dadurch verursachten Warteschlangen. Er wird durch eine negativ-exponentielle Verteilung mit Mittelwert  $D_v$  modelliert.

In Tabelle 6.7 sind einige Merkmale der Komponenten und der Übertragungswege aufgeführt. Für die feste Verzögerung in Vermittlungsknoten wird von  $60 \mu\text{s}$  ausgegangen, was für heute marktübliche Geräte einen Mittelwert darstellt <sup>(1)</sup>.

**Tabelle 6.7:** ATM-Netz und Komponenten

| Kenngroße   | Wert  |
|---|---|
| Abstand der Zugangspunkte zum ATM-Netz                                      | 50 km ( $\hat{=}$ Laufzeit $250 \mu\text{s}$ <sup>(a)</sup> )           |
| Anzahl dazwischenliegender ATM-Vermittlungsknoten                           | 5   |
| Feste Verzögerung pro Vermittlungsknoten                                    | $60 \mu\text{s}$ ( $\hat{=}$ 21 Zellübertragungszeiten <sup>(b)</sup> ) |
| Prioritäten in den ATM-Vermittlungsknoten (je Priorität eine Warteschlange) | 6 (CBR, rt-VBR, nrt-VBR, ABR (RM), ABR (Data), UBR)                     |

(a) Ausbreitungsgeschwindigkeit auf dem Medium  $2 \times 10^8$  m/s

(b) Die Nettoübertragungszeit einer Zelle bei SDH und  $155,52$  Mbit/s Übertragungsrate beträgt  $2,83 \mu\text{s}$

In der ABR-Senke tritt die Verzögerung  $D_d$  auf, die in erster Linie von der Leistungsfähigkeit des Interfaces bei der ABR-Protokollverarbeitung abhängt. Hier wird eine konstante Zeit von  $10 \mu\text{s}$  angenommen.

(1) Bei Messungen an Vermittlungsknoten, die die SDH-Übertragungstechnik unterstützen, treten z. T. feste Verzögerungen bis weit über  $100 \mu\text{s}$  auf (EXPLOIT-Testbett [30]). Demgegenüber liegen die in Datenblättern angegebenen Verzögerungen für ATM-Vermittlungsknoten im LAN-Bereich um  $10$ - $20 \mu\text{s}$ .

Aufgrund der genannten Werte und unter Berücksichtigung der zweifachen Laufzeit von RM-Zellen durch das Netz ist die gesamte konstante Verzögerung  $D_c$ :

$$D_c = 2 \cdot (250\mu\text{s} + 5 \cdot 60\mu\text{s}) + D_d = 1110\mu\text{s} \quad (6.1)$$

Für die Abschätzung des Mittelwerts der variablen Verzögerung  $D_v$  in einem ATM-Vermittlungsknoten wird angenommen, daß die Warteschlangen für CBR- und VBR-Verkehr zur Beschränkung der Verzögerungen in diesen Verkehrsklassen eine maximale Länge von ca. 100 Zellen haben. Weiterhin wird von einer starken Last für alle Verkehrsklassen ausgegangen, wodurch alle Warteschlangen im Mittel eine Länge größer Null haben. Zusätzlich tritt durch den Fairneß-Algorithmus in jedem Vermittlungsknoten eine Bearbeitungszeit für RM-Zellen auf, die von der Leistungsfähigkeit und Auslastung der CPU des Knotens abhängt. Tabelle 6.8 zeigt die Schätzwerte für die genannten Größen.

**Tabelle 6.8:** Annahmen zur mittleren Belastung eines Netzknotens

| Kenngröße  | Wert             |
|--|------------------|
| Mittlere Verzögerung aufgrund priorisierter CBR- und VBR-Zellen  | 10 Zellen        |
| Mittlere Verzögerung aufgrund von RM-Zellen anderer Verbindungen | 10 Zellen        |
| Bearbeitungszeit einer RM-Zelle                                  | 10 $\mu\text{s}$ |

Der Mittelwert  $D_v$  des variablen Anteils der Zellverzögerung und die gesamte mittlere Laufzeit  $D$  einer RM-Zelle durch das Netz sind damit:

$$D_v = 2 \cdot (5 \cdot (10 + 10) \cdot 2,83\mu\text{s} + 5 \cdot 10\mu\text{s}) = 616\mu\text{s} \quad (6.2)$$

$$D = D_c + D_v = 1,78 \text{ ms}$$

Die mittlere Laufzeit einer RM-Zelle durch das ATM-Netz liegt mit weniger als 2 ms ungefähr eine Größenordnung unter dem mittleren Abstand zweier RM-Zellen.

### 6.7.1.3 Modifikation von $ER_p$ -Werten im Netz

In ATM-Vermittlungsknoten, die das ABR-Protokoll unterstützen, muß die für ABR-Verkehr verfügbare Bandbreite in geeigneter Weise auf die vorhandenen ABR-Verbindungen aufgeteilt werden (s. Abschnitt 6.4.2.3). Das Modell des ATM-Netzes enthält stellvertretend für alle ATM-Vermittlungsknoten eine Instanz eines entsprechenden Algorithmus. Der Algorithmus basiert auf dem EPRCA-Verfahren (Enhanced Proportional Rate Control Algorithm) [11], das um Mechanismen zur Bedarfsschätzung ergänzt wird. Dieses Verfahren wird verwendet, da es im Gegensatz zu anderen Verfahren die Betrachtung einer einzelnen ABR-Verbindung auf einfache Weise ermöglicht.

Das EPRCA-Verfahren berechnet für alle ABR-Verbindungen eine mittlere erlaubte Senderate  $MACR$  (mean  $ACR$ ), die zyklisch so modifiziert wird, daß die gesamte Last (ABR und variable Anteile von VBR, etc.) auf einen vorgegebenen Zielwert (*targetRate*) eingeregelt wird. Dabei

erfolgt im nicht modifizierten Algorithmus die Zuteilung von Bandbreite an unterschiedliche ABR-Verbindungen proportional zu deren momentaner Zellrate ( $CCR$ ):

$$MACR_k' = MACR_{k-1} \cdot (1 - AVF) + CCR_k \cdot AVF \quad (6.3)$$

Dies stellt eine exponentiell gewichtete Mittelung der  $CCR$ -Werte mit dem Parameter  $AVF$  ( $ACR$  variation factor) dar (s. Anhang A3).

Der genannte Zielwert liegt üblicherweise wenige Prozent unterhalb der maximal verarbeitbaren Bandbreite. Zur Bestimmung des momentanen Lastzustands wird der Lastfaktor ( $loadFactor$ ) als Quotient zwischen der als Rate ausgedrückten momentanen Last ( $inputRate$ ) und dem Zielwert ( $targetRate$ ) gebildet:

$$loadFactor = \frac{inputRate}{targetRate} \quad (6.4)$$

Die momentane Last ergibt sich als Summe aus der Last der Netznachbildung  $r_k^{(Net)}$  (s. Gl. (6.15)) und der Last der betrachteten ABR-Verbindung  $r_k^{(ABR)}$ , gemittelt über die letzten  $K_M$  Werte:

$$inputRate_k = \frac{1}{K_M} \cdot \sum_{i=0}^{K_M-1} (r_{k-i}^{(Net)} + r_{k-i}^{(ABR)}) \quad (6.5)$$

Im Fall von Unterlast (Lastfaktor  $< 1$ ) erfolgt die Verteilung von ungenutzter Bandbreite durch Erhöhung von  $MACR$  um einen festen Wert  $MAIR$  ( $MACR$  additive increase rate) gleichmäßig an alle Verbindungen:

$$MACR_k = MACR_k' + MAIR \quad (6.6)$$

Der daraus resultierende Wert von  $MACR_k$  wird in das ER-Feld der RM-Zellen geschrieben, die zu ihrer Quelle zurückgesendet werden ( $ER'_{b,k} = MACR_k$ ).

Im Fall von Überlast wird  $MACR$  nicht erhöht ( $MACR_k = MACR_k'$ ), sondern die an die jeweilige Quelle gesendete explizite Rate um den Faktor  $MRF$  ( $MACR$  reduction factor,  $MRF < 1$ ) reduziert, wodurch über den geschlossenen Regelkreis die Senderate und der  $MACR$ -Wert aller beteiligten ABR-Quellen reduziert wird.

$$ER'_{b,k} = MACR_k \cdot MRF \quad (6.7)$$

Überlast wird in diesem Modell dann festgestellt, wenn der Lastfaktor für mehrere ( $K_{ov}$ ) aufeinander folgende Intervalle den Wert 1 überschreitet.

Das beschriebene Verfahren bietet keine Möglichkeit, Reservierungen durch die ABR-Quelle zu berücksichtigen. Daher werden im folgenden die durch (6.3), (6.6) und (6.7) beschriebenen Schritte so modifiziert, daß Reservierungen durch den  $ER_f$ -Wert der Quellen berücksichtigt werden. Die Feststellung von Überlastsituationen erfolgt weiterhin wie oben beschrieben.

Wenn keine Überlast vorliegt oder eine Reduzierung der erlaubten Senderate gefordert wird, also  $ER'_{f,k} < MACR_{k-1}$  gilt, erfolgt die Anpassung von  $MACR$  an die von der Quelle angefor-

derte Rate durch das modifizierte Verfahren wie in (6.3), wenn dort  $CCR$  durch  $ER_f$  sowie  $AVF$  durch  $RDD$  ersetzt wird:

$$MACR_k = MACR_{k-1} \cdot (1 - RDD) + ER_{f,k} \cdot RDD \quad (6.8)$$

Der Wert des Parameters  $RDD$  (rate demand dynamics) ist ein Maß für die Geschwindigkeit, mit der der Algorithmus bzw. das Netz auf Ratenanforderungen reagiert. In Anhang A3 sind beispielhafte Reaktionszeiten für unterschiedliche Werte des Parameters angegeben.

Im Fall von Überlast wird  $MACR$  nicht verändert:

$$MACR_k = MACR_{k-1} \quad (6.9)$$

Die Bestimmung des  $ER'_b$ -Werts erfolgt wie oben; bei Unterlast wird  $MACR$  kopiert und bei Überlast um den Faktor  $MRF$  reduziert.

Durch den beschriebenen Algorithmus können Reservierungsanforderungen der ABR-Quelle in Abhängigkeit von der momentanen Netzauslastung beantwortet werden.

Im Verhältnis zur Zeit zwischen RM-Zellen stellt der Netzzustand des Modells eine relativ statische Größe dar (Gültigkeitsdauer eines Netzzustands ist z. B. 0,1sec, s. Abschnitt 6.7.1.4), dadurch können kurzfristige Fluktuationen in der Netzauslastung bei der Bandbreitevergabe nicht berücksichtigt werden. Um dies im Modell trotzdem zu erreichen, wird über einen weiteren Mechanismus der ER-Wert der RM-Zellen auf ihrem Weg durch das ATM-Netz zufällig verringert. Dadurch wird modelliert, daß der ER-Wert von RM-Zellen, die im Netz in überlasteten Vermittlungsknoten stark verzögert werden, mit relativ hoher Wahrscheinlichkeit durch diese Vermittlungsknoten reduziert wird.

Die Reduktion des  $ER_b$ -Werts aufgrund von Überlast im Netz wird durch Multiplikation von  $ER'_b$  mit einem Faktor  $F_{ER}$  modelliert. Der Faktor ist eine Funktion eines negativ-exponentiell verteilten Zufallswerts  $r_{ER}$  mit Mittelwert  $R_{ER}$  sowie eine Funktion der Verzögerung  $d_v$  der jeweiligen RM-Zelle durch das Netz.

$$F_{ER} = \frac{1}{1 + \frac{d_v}{D_v} \cdot r_{ER}} \quad (\leq 1) \quad (6.10)$$

$$ER_{b,k} = ER'_{b,k} \cdot F_{ER}$$

( $D_v$  ist der Mittelwert der variablen Verzögerung von RM-Zellen im ATM-Netz,  $r_{ER}$  der negativ-exponentiell verteilte Zufallswert, dessen Mittelwert  $R_{ER}$  z. B. zu 0,01 gewählt wird.)

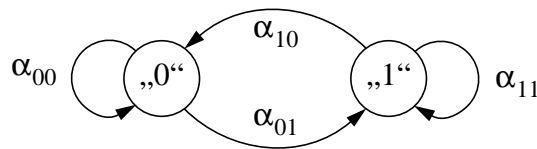
Auf diese Weise wird  $ER_b$  im Mittel um so stärker reduziert, desto größer die Verzögerung im Netz ist.

#### 6.7.1.4 Zustandsprozeß zur Neubildung realer Netzlast

Von der gesamten Kapazität des für die betrachtete ABR-Verbindung relevanten Pfads durch das ATM-Netz steht für ABR-Verbindungen nur ein Teil zur Verfügung. Dieser Teil entsteht nach Abzug der Bandbreite für CBR-Verkehr und der mittleren Bandbreite für VBR-Verkehr.

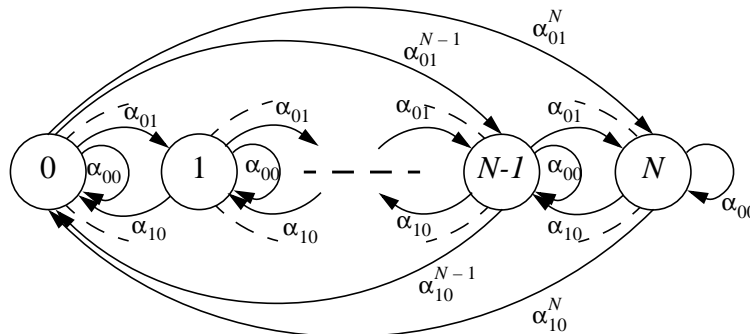
Die verbleibende Bandbreite wird als Rate ausgedrückt und hat die Größe  $r_{max}^{(Net)}$ . Sie dient der Kompensation kurzfristiger Lastfluktuationen durch VBR-Verkehr sowie zur Verwendung durch ABR-Verkehr.

Die Last im ATM-Netz, die die Bandbreitezuteilung der betrachteten ABR-Verbindung beeinflusst, wird als Summe der Last modelliert, die durch  $N$  gleichartige zeitdiskrete Pseudoquellen erzeugt wird. Der Zustand  $i$  dieses Modells ist die Anzahl der gleichzeitig aktiven Pseudoquellen. Da die Last in einem realen ATM-Netz durch z. T. stark korrelierte Verkehre verursacht wird, erfolgt die Modellierung der Pseudoquellen durch ein einfaches Modell mit den Zuständen „0“ und „1“ (On-Off-Quelle), das über die Wahl seiner Parameter die Erzeugung von stark korreliertem Verhalten zuläßt. Die Übergangswahrscheinlichkeiten der Pseudoquellen sind  $\alpha_{00}, \alpha_{01}, \alpha_{10}, \alpha_{11}$  für die jeweiligen Übergänge zwischen „0“ und „1“ (s. Bild 6.14). Als Randbedingung gilt  $\alpha_{i0} + \alpha_{i1} = 1$ .



**Bild 6.14:** Pseudoquelle – Zustands-Übergangsdiagramm

Die Folge der Zahl gleichzeitig aktiver Pseudoquellen bildet eine Markoff-Kette mit  $N+1$  Zuständen (s. Bild 6.15).



**Bild 6.15:** Zustandsprozeß für den Systemzustand

Die Wahrscheinlichkeit für einen Übergang von Zustand  $i$  nach Zustand  $j$  ergibt sich zu

$$p_{ij} = \sum_{a=\max(0, j-i)}^{\min(j, N-i)} \binom{N-i}{a} \binom{i}{i-j+a} \alpha_{01}^a \cdot \alpha_{10}^{(i-j+a)} \cdot \alpha_{11}^{(j-a)} \cdot \alpha_{00}^{(N-a-i)} \quad (6.11)$$

wobei die Bedingung

$$\sum_{i=0}^N \sum_{j=0}^N p_{ij} = 1 \quad (6.12)$$

erfüllt ist. Der mittlere Zustand einer Pseudoquelle ist

$$X = 0 \cdot P\{0\} + 1 \cdot P\{1\} = \frac{\alpha_{01}}{(\alpha_{01} + \alpha_{10})} \quad (6.13)$$

und damit der Erwartungswert für den Systemzustand  $i$ :

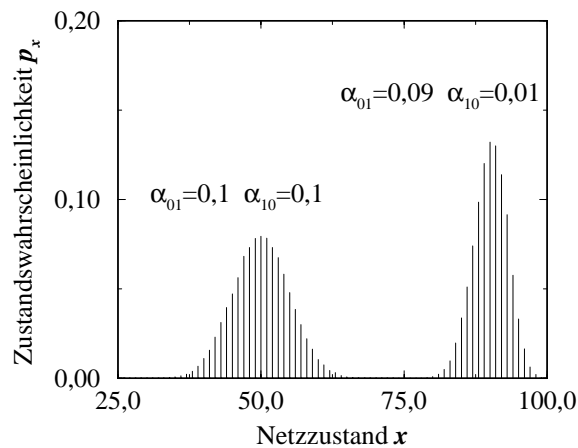
$$E[i] = N \cdot X = N \cdot \frac{\alpha_{01}}{\alpha_{01} + \alpha_{10}} \quad (6.14)$$

Die einem Zustand  $i$  des Netzmodells zum Zeitpunkt  $k$  entsprechende Zellrate ist

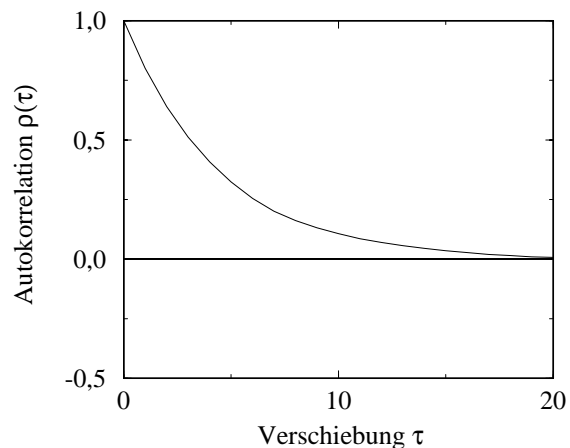
$$r_k^{(Net)} = \frac{i}{N} \cdot r_{max}^{(Net)} \quad (6.15)$$

Dabei ist  $r_{max}^{(Net)}$  die maximale Last, die durch den Zustand  $i=N$  erreicht wird.

Bild 6.16 zeigt die Zustandsverteilung für zwei Parameterkombinationen mit je  $N=100$  Pseudoquellen, Bild 6.17 das beiden Kombinationen gemeinsame Korrelogramm.



**Bild 6.16:** Zustandsverteilung des Netzmodells



**Bild 6.17:** Autokorrelation der Zustände des Netzmodells

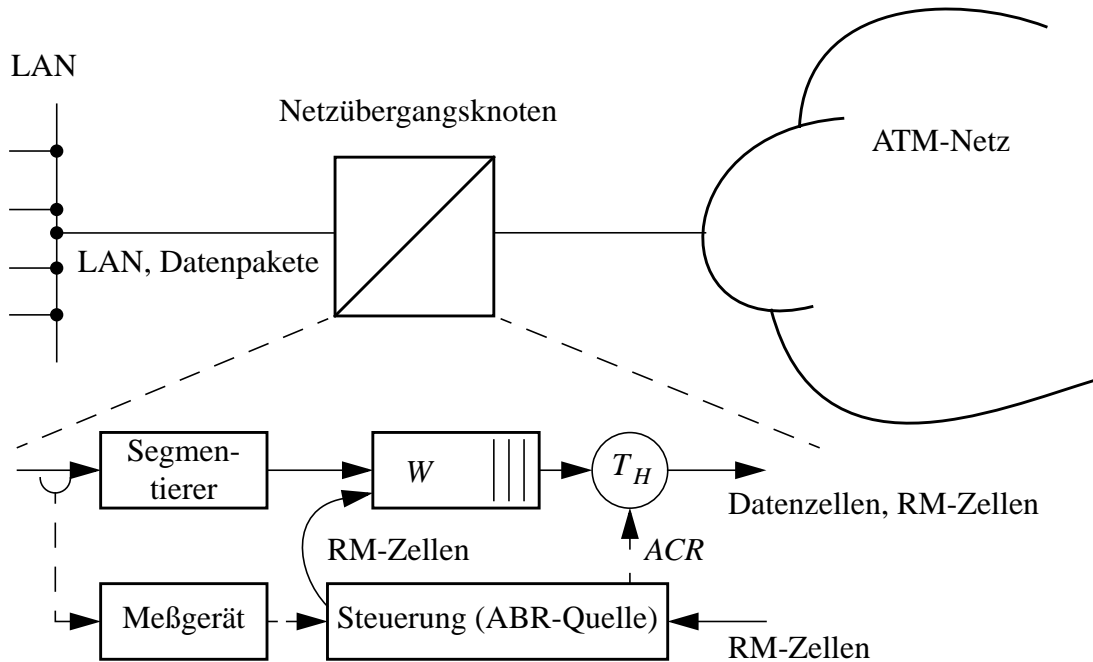
Die Bestimmung eines neuen Netzzustands erfolgt zyklisch mit dem Abstand  $T_N$ .

## 6.7.2 Modell für einen Netzübergangsknoten

### 6.7.2.1 Übersicht

Für die Simulation der Kopplung eines Ethernet-LANs über einen ATM-Backbone wird in diesem Abschnitt das Modell des zugehörigen Netzübergangsknotens beschrieben. Zur besseren Übersichtlichkeit und da dort die Vorhersage der benötigten Übertragungsbandbreite erfolgt, wird hier nur der Teil eines solchen Knotens modelliert, der den Sendeteil einer ABR-Verbindung realisiert. Das hat ein einfacheres Modell und eine einfachere Auswertung der Ergebnisse zur Folge.

Der Aufbau eines Netzübergangsknotens zur Übertragung von LAN-Paketen über einen ATM-Backbone ist prinzipiell wie in Bild 6.18 aufgebaut: Die LAN-Pakete werden in einem Seg-



**Bild 6.18:** Funktion eines Netzübergangsknotens

mentierer, der auch den AAL-5-Overhead berücksichtigt, in ATM-Zellen aufgeteilt, die anschließend durch die Bedieneinheit über das ATM-Netz gesendet werden. Falls die Bedieneinheit momentan belegt ist, werden die Zellen in einem Wartespeicher mit  $W$  Warteplätzen zwischengespeichert. Falls alle  $W$  Warteplätze bei Ankunft einer Zelle belegt sind, wird diese Zelle verworfen und geht verloren. Zusätzlich zu der Zelle, die den Verlust ausgelöst hat, werden alle folgenden, zu demselben LAN-Paket gehörenden ATM-Zellen verworfen, da nach Verlust einer Datenzelle das Paket nicht mehr zusammengesetzt werden kann. Dadurch wird die Anzahl der unnötig über das ATM-Netz übertragenen Zellen verringert (early packet discard).

Die Rate, mit der die ATM-Zellen durch die Bedieneinheit durch das Netz gesendet werden, ist durch den Wert von  $ACR$  nach oben beschränkt, während die tatsächliche Bedienzeit  $T_H$  durch die Bitrate der Übertragungsstrecke bestimmt wird. Da hier von einer Strecke mit einer Bitrate von 155,52 Mbit/s und SDH-Übertragung ausgegangen wird, ergibt sich  $T_H = 2,83 \mu\text{s}$ .

Die durch die Steuerung des Netzübergangsknotens bzw. durch die ABR-Quelle erzeugten RM-Zellen werden ebenfalls über den Wartespeicher der Bedieneinheit zugeführt. Dabei haben jedoch die RM-Zellen gegenüber den Datenzellen eine höhere Priorität und überholen diese im Wartespeicher. Aus dem Netz empfangene RM-Zellen werden asynchron von der ABR-Quelle verarbeitet.

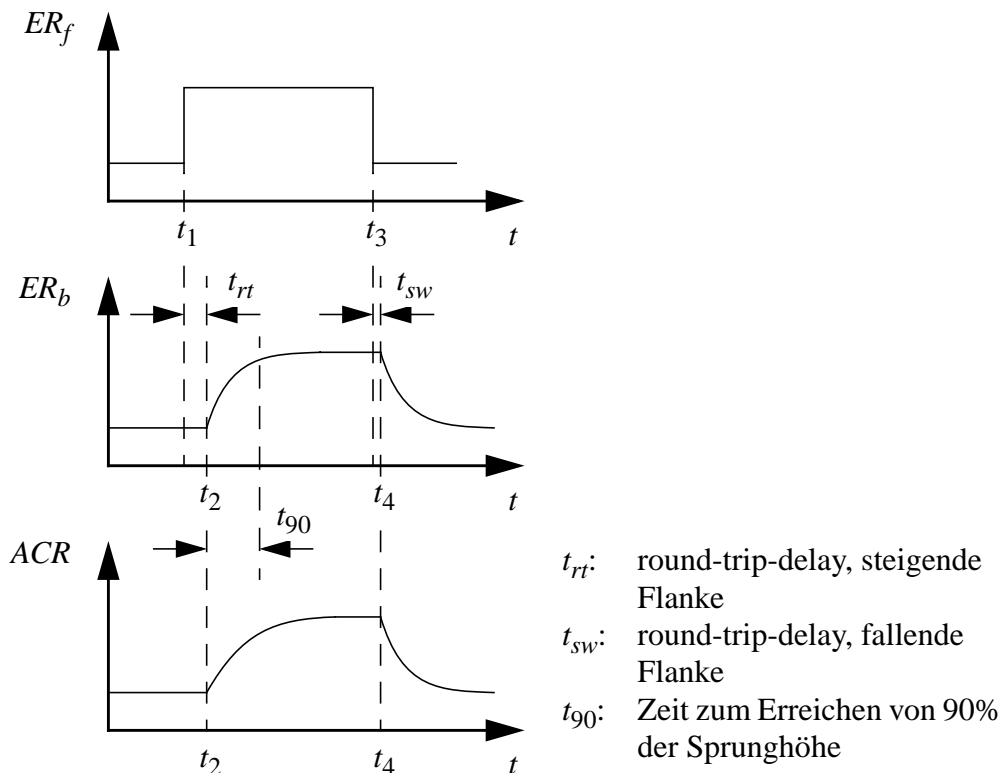
Zum Zweck der Bandbreiteprognose wird der Steuerung darüberhinaus die momentane Bandbreite zugeführt, die durch ein Meßgerät bestimmt wird.

In Abschnitt 6.6 wurden tolerierbare Antwortzeiten für interaktive LAN-Anwendungen angegeben. Davon ausgehend können die Puffergrößen am Netzzugang und im Netz abgeschätzt werden. Bei einer mittleren Zellrate von 2748 Zellen/sec (s. Tabelle 6.6) entspricht ein Warte-

speicher mit 400 Plätzen einer maximalen Verzögerung von 150 ms. Dies führt zusammen mit den im Netz zu erwartenden Verzögerungen ungefähr auf die geforderten Antwortzeiten.

### 6.7.2.2 Zeitverhalten

In Bild 6.19 sind einige zeitliche Zusammenhänge zwischen Größen der ABR-Regelschleife dargestellt. Zur Diskussion der auftretenden Effekte wird vereinfachend von einer sprunghöhen Bandbreitereservierung  $ER_f$  durch vorwärtslaufende RM-Zellen ausgegangen, wie sie im oberen Teil des Bildes dargestellt ist. Weiterhin wird davon ausgegangen, daß das ATM-Netz unbelastet ist, damit sämtliche Reservierungen erfüllt werden können. Die sprunghöhen Erhöhung der Reservierung zum Zeitpunkt  $t_1$  wirkt sich erst nach der Zeit  $t_{rt}$  zum Zeitpunkt  $t_2$  an der Quelle aus. Diese Totzeit ist auf die Umlaufverzögerung der RM-Zelle durch das Netz zurückzuführen und hat den in Gl. (6.2) angegebenen Mittelwert  $D$ . Diese Zeit wird im Mittel benötigt, bis die RM-Zelle eine Erhöhung der erlaubten Rate an die Quelle melden kann, da jeder Vermittlungsknoten und die ABR-Senke eine höhere Rate erst durch Eintrag in das ER-Feld bestätigen müssen. Darüberhinaus wird die angeforderte Rate üblicherweise nicht sofort gewährt, sondern die erlaubte Rate schrittweise erhöht, was in Bild 6.19 durch den exponentiellen Anstieg von  $ER_b$  im mittleren Teil angedeutet wird. Dieses Verhalten ist die Folge der Fairneß-Algorithmen in den Vermittlungsknoten; eine Möglichkeit für einen solchen Algorithmus wird in Abschnitt 6.7.1 beschrieben. Die Zeit bis zum Erreichen von 90% des Endwerts von  $ER_b$  wird als Verzögerung  $t_{90}$  bezeichnet.



**Bild 6.19:** Zeitdiagramme für ABR-Protokoll



Zum Zeitpunkt  $t_3$  wird  $ER_f$  wieder auf den alten niedrigen Wert zurückgenommen. In diesem Fall wird die Änderung sofort durch den ersten Vermittlungsknoten im Netz bestätigt, was nur zu einer kleinen Verzögerung  $t_{sw}$  führt.

Für ein symmetrisches Verhalten bei Erhöhung bzw. Verringerung der angeforderten Rate ist es günstig, wenn die Zeit  $t_{90}$  deutlich größer als  $t_{rt}$  ist. Diese Forderung kann durch geeignete Parameter des Fairneß-Algorithmus erfüllt werden. Durch die Bedingung  $t_{90} > t_{rt}$  wird außerdem vermieden, daß durch eine im Verhältnis zur Totzeit kleine Regler-Zeitkonstante Instabilitäten auftreten können.

Die Bestimmung des ACR-Werts aus  $ER_b$  erfolgt wie in Abschnitt 6.4.2.1 beschrieben. Dabei muß bei der Wahl der Parameter  $RDF$  und  $RIF$  (s. Tabelle 6.4) beachtet werden, daß die Zeit bis zum Erreichen von 90% von  $ACR$  sich nicht zu stark von  $t_{90}$  unterscheidet, da sonst die Überlegungen zur Dynamik des Gesamtsystems nicht mehr gelten. In Bild 6.19 ist im unteren Bildteil ein beispielhafter Verlauf von  $ACR$  gezeigt, bei dem die Verzögerung bis zum Erreichen von 90% der Sprunghöhe durch schlecht gewählte Parameter deutlich größer als  $t_{90}$  ist.

Die RM-Zellen der ABR-Verbindung werden außer in Ausnahmesituationen nach einer festen Anzahl von  $N_{rm}-1$  Datenzellen gesendet und bestimmen die Regelzeitpunkte des ABR-Regelkreises. Diese Zeitpunkte haben keine festen Abstände, sondern hängen von der momentanen Datenrate der ABR-Verbindung ab. Dadurch werden bei hohen Datenraten häufiger RM-Zellen gesendet als bei niedrigen Datenraten. Sowohl die Anpassung der reservierten Bandbreite durch das Netz als auch die Anpassung der erlaubten Senderate ( $ACR$ ) in der Netzkoppeleinheit werden durch RM-Zellen angestoßen.

Die Zeitpunkte, zu denen die momentane LAN-Datenrate gemessen und zu denen Vorhersagen für die Datenrate der ABR-Quelle gemacht werden, müssen diesem Verhalten geeignet angepaßt werden. Da über das ER-Feld der RM-Zellen die Bandbreitereservierung erfolgt, sollten auch die Ratenvorhersagen bei hohen Datenraten häufiger erfolgen als bei niedrigen Datenraten. Aus diesem Grund werden die Meß- bzw. Vorhersagezeitpunkte analog zu den Sendezeitpunkten der RM-Zellen nach einer festen Anzahl von  $N_M$  Datenzellen gewählt. Der Wert von  $N_M$  sollte in der Größenordnung von  $N_{rm}$  liegen, um aktuelle Vorhersagewerte für die Reservierung zu gewährleisten.

Weiterhin ist zu entscheiden, welche Vorhersageweite für die Verteilungsprognose gewählt wird (Parameter  $P$  des Algorithmus). Die Wahl dieses Parameters hängt von der Totzeit im System  $t_{rt}$  ab. Falls diese Zeit größer als  $t_{90}$  ist, muß für  $P$  ein Wert größer 1 gewählt werden, da sonst die vorhergesagten Ratenwerte zu dem Zeitpunkt, zu dem sie an der Quelle verwendet werden können, bereits veraltet sind.

### 6.7.2.3 Reservierungsverfahren

In diesem Abschnitt werden die unterschiedlichen in der ABR-Quelle verwendeten und untersuchten Reservierungsverfahren beschrieben. Sie entsprechen den in Abschnitt 6.5 genannten Realisierungsalternativen.

Die erste Möglichkeit besteht in der festen Reservierung der Spitzen-Zellrate  $PCR$  der Verbindung, ohne Schwankungen des Bedarfs in Betracht zu ziehen (s. Abschnitt 6.5.1). Hierfür ist der im Standard beschriebene Algorithmus einer ABR-Quelle ohne Erweiterung geeignet.

Eine weitere Möglichkeit besteht in der Reservierung einer Rate, die dem Wert der letzten Ratenmessung entspricht (s. Abschnitt 6.5.2):

$$p_k = r_k \quad (6.16)$$

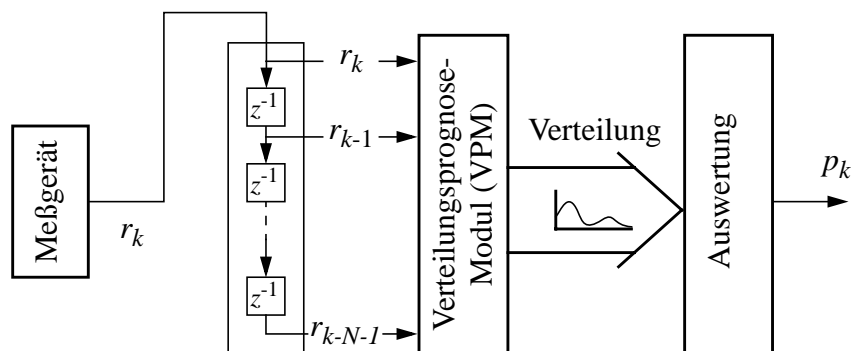
Die Rate der durch den LAN-Verkehr erzeugten ATM-Datenzellen wird bestimmt, indem die Folge der LAN-Pakete beobachtet und der Zeitpunkt  $T_k$  bestimmt wird, zu dem ein Äquivalent zu  $N_M$  Datenzellen erreicht ist. Dieser Zeitpunkt kann auch während der Übertragung eines LAN-Pakets erreicht werden. Das Meßverfahren wird unabhängig von der Paketsegmentierung aus Bild 6.18 realisiert, da dort bei einzelnen Zellverlusten das gesamte Paket verworfen wird (early packet discard) und damit nicht die volle Datenrate des LAN-Verkehrs sichtbar ist. Der Overhead, der durch Segmentierung und AAL-5 entsteht, wird jedoch berücksichtigt.

Das Ergebnis der Ratenmessung ist eine Folge von Ratenwerten  $r_k$  zu den Zeitpunkten  $T_k$ . Die einzelnen Raten zu den Zeitpunkten  $T_k$  haben den Wert:

$$r_k = \frac{N_M}{T_k - T_{k-1}} \quad (6.17)$$

Als dritte Möglichkeit wird die Ratenreservierung über die Verteilungsprognose betrachtet. Dabei wird aufgrund vergangener Ratenbeobachtungen die Verteilung der Rate des nächsten Intervalls vorhergesagt und geeignet ausgewertet (s. Abschnitt 6.5.3).

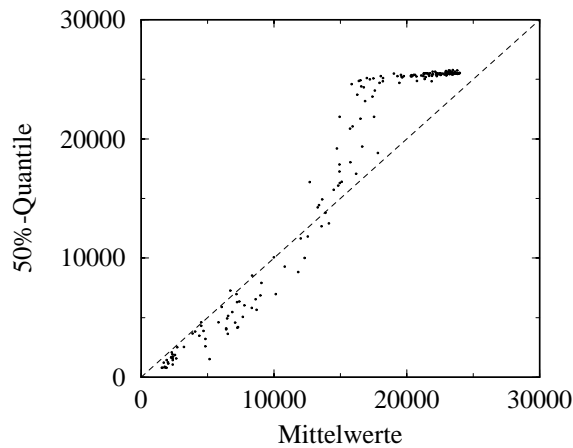
Im Fall der Verteilungsprognose führt dies auf die in Bild 6.20 dargestellte Struktur der Steuerung der Netzkoppeleinheit. Die Ratenmeßwerte werden dem Verteilungsprognose-Modul zugeführt, das daraus die Verteilung des nächsten Werts vorhersagt. Diese Verteilung wird nach einer geeigneten Auswertung als Reservierungswert  $p_k$  in RM-Zellen verwendet.



**Bild 6.20:** Auswertung der Ratenmeßwerte durch die Verteilungsprognose

Einfache Auswertemöglichkeiten für die vorhergesagten Verteilungen sind die Berechnung von Mittelwert, Varianz oder Quantilen. Ein Vergleich zwischen den Mittelwerten und den 50%-Quantilen der prognostizierten Verteilungen <sup>(1)</sup> in Bild 6.21 zeigt, daß die Mittelwerte für kleine Raten über und für große Raten unter den 50%-Quantilen liegen. Der Vergleich dieser zwei Auswertemethoden durch Simulation führt zu dem Ergebnis, daß die Verwendung der

(1) Die Parameter der Verteilungsprognose sind  $10+10s_{10}/200/30$ , d. h. es treten 200 unterschiedliche Verteilungen am Ausgang des Verteilungsprognosemoduls auf.

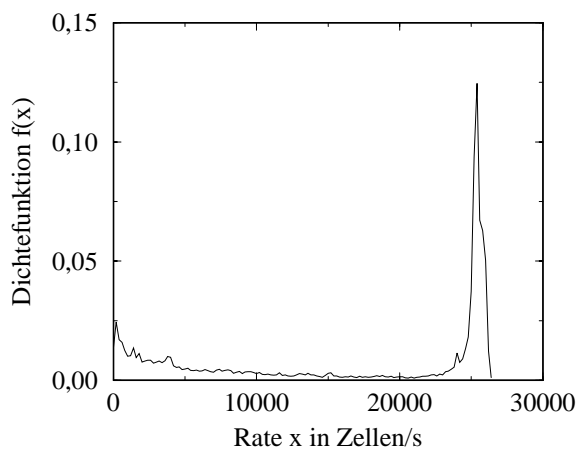


**Bild 6.21:** Darstellung der 50%-Quantile über den Mittelwerten der Verteilungen am Ausgang des Verteilungsprognosemoduls

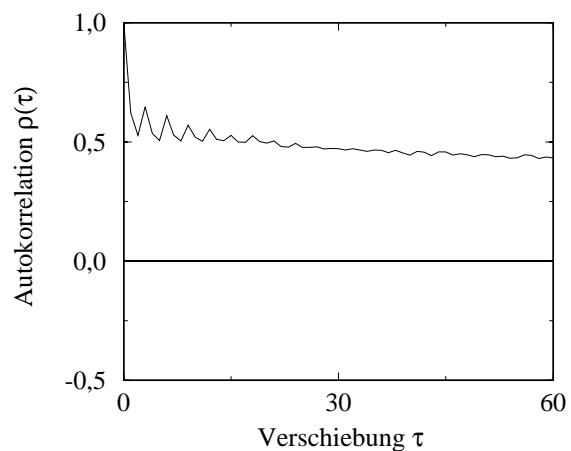
50%-Quantile ein günstigeres Gesamtverhalten des Systems zur Folge hat (s. Abschnitt 6.8). Die Werte, die durch konventionelle Prognoseverfahren geliefert werden, die nicht auf der Verteilungsprognose basieren, entsprechen den hier aus der Verteilung berechneten Mittelwerten. Eine Verwendung von Quantilen ist mit solchen Verfahren jedoch nicht möglich.

Der durch die genannten Auswerteverfahren neu bestimmte Wert wird während des jeweils nächsten Meßintervalls für die Ratenreservierung verwendet. Abhängig von dem Fairneß-Algorithmus im ATM-Netz wird die erlaubte Senderate der ABR-Quelle schrittweise an die angeforderte Rate angepaßt – vorausgesetzt, entsprechende Ressourcen sind verfügbar.

Für die Ratenvorhersage mit Hilfe der Verteilungsprognose ist eine Lernphase erforderlich. Während des Lernvorgangs werden die KNNs der Verteilungsprognose an den stochastischen Prozeß der Raten  $\{r_k\}$  angepaßt. In den Bildern 6.22 und 6.23 sind Verteilungsdichte und Korrelogramm von  $\{r_k\}$  dargestellt. Hervorzuheben sind die Spitze der Dichtefunktion für hohe Datenraten um 25.000 Zellen/s sowie die langandauernde Korrelation des Prozesses.

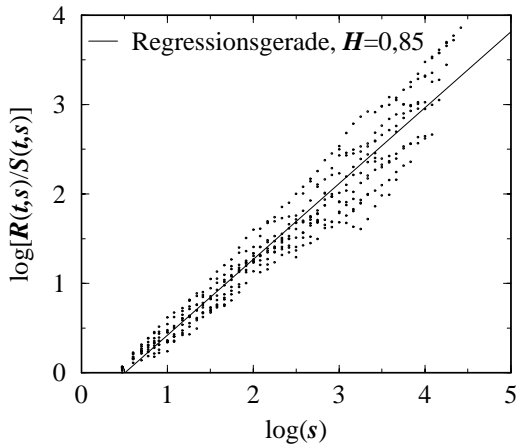


**Bild 6.22:** Verteilungsdichte der Raten-Folge

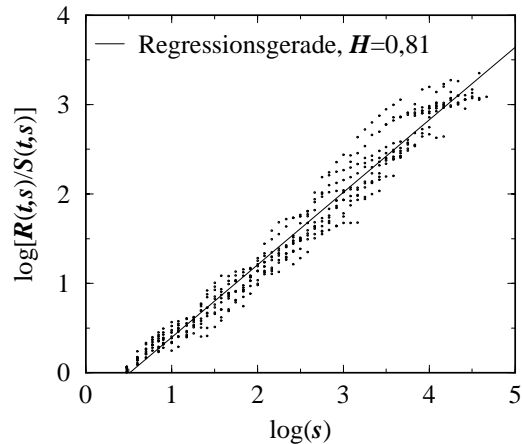


**Bild 6.23:** Korrelogramm der Raten-Folge

Da es sich bei LAN-Verkehr um einen stark selbstähnlichen Prozeß handelt, stellt sich für die Verteilungsprognose die Frage, ob dieses Verhalten ausreichend nachgebildet wird. Die Bilder 6.24 und 6.25 zeigen die Diagramme der RS-Analyse für die gemessenen LAN-Daten und für die Daten, die ein Quellmodell nach Kapitel 5 mit dem gelernten Prognosemodul erzeugt hat. Die RS-Analysen führen auf Hurst-Parameter von 0,85 für den Originalprozeß bzw. 0,81 für das Quellmodell, was auf eine sehr gute Übereinstimmung hinweist.



**Bild 6.24:** RS-Analyse für Originaldaten



**Bild 6.25:** RS-Analyse für Modelldaten  
(10+10s10/200/30-Modell)

Bei den beschriebenen Arten der Ratenreservierung tritt häufig die Situation auf, daß die Rate des Folgeintervalls unterschätzt wird, was sich in einer steigenden Warteschlangenlänge ausdrückt. Um in diesem Fall Verlusten und langen Wartezeiten vorzubeugen, wird bei steigender Warteschlangenlänge  $W_k$  die momentane Reservierung erhöht. Ein Anstieg der Warteschlangenlänge ist durch eine positive Differenz ihrer Länge für zwei aufeinanderfolgende Prognosezeitpunkte gekennzeichnet. Die Erhöhung des Reservierungswerts erfolgt in Abhängigkeit von dieser Differenz:

$$diff = W_k - W_{k-1}$$

$$p'_k = \begin{cases} p_k \cdot \left(1 + F_W \cdot \frac{diff}{N_M}\right) & diff > 0 \\ p_k & \text{sonst} \end{cases} \quad (6.18)$$

Dabei ist  $F_W$  ein Sicherheitsfaktor ( $F_W \geq 1$ ). Für  $F_W = 1$  wird genau die Rate zusätzlich reserviert, die benötigt wird, um den Zuwachs der Warteschlangenlänge seit dem letzten Ereigniszeitpunkt während des nächsten Meßintervalls abzubauen. Dabei gilt die Annahme, daß der Prognosewert  $p_k$  für das nächste Meßintervall korrekt ist.

Der tatsächliche Wert für den nächsten Reservierungswert  $ER_{f,k}$  ergibt sich aus dem zu diesem Zeitpunkt gültigen Prognosewert durch Beschränkung auf die Verbindungsparameter  $MCR$  und  $PCR$ :

$$ER_{f,k} = \min(\max(p'_k, MCR), PCR) \quad (6.19)$$

### 6.7.2.4 Modellparameter

In Tabelle 6.9 sind die Parameter des Modells für die LAN-Kopplung über ABR angegeben, die für alle Simulationen verwendet wurden.

**Tabelle 6.9:** Modellparameter

| Parameter | Wert           | Parameter         | Wert           |
|-----------|----------------|-------------------|----------------|
| $MCR$     | 200 cells/s    | $D_c$             | 1,1 ms         |
| $ICR$     | 2000 cells/s   | $D_v$             | 0,62 ms        |
| $PCR$     | 25.000 cells/s | $K_M$             | 10             |
| $RDF$     | 0,004          | $K_{ov}$          | 3              |
| $RIF$     | 0,0625         | $N$               | 100            |
| $Nrm$     | 32             | $T_N$             | 0,1 s          |
| $Crm$     | 5              | $r_{max}^{(Net)}$ | 50.000 cells/s |
| $AVF$     | 0,0625         | $F_W$             | 5              |
| $MRF$     | 0,95           | $N_M$             | 64             |
| $RDD$     | 0,1            | $R_{ER}$          | 0,01           |

Die ABR-Parameter  $MCR$ ,  $ICR$  und  $PCR$  wurden entsprechend der Charakteristik des LAN-Verkehrs gewählt, die Parameter  $RDF$ ,  $RIF$ ,  $Nrm$  und  $Crm$  entsprechen den im ABR-Standard vorgeschlagenen Werten. Dies gilt auch für die Parameter des Fairneß-Algorithmus ( $AVF$ ,  $MRF$ ).

Der Parameter  $RDD$  des erweiterten Fairneß-Algorithmus wurde auf 0,1 gesetzt, um ein geeignetes Zeitverhalten des Algorithmus zu erreichen. Mit diesem Wert entspricht die Zeit von 21 RM-Zellen der Zeit  $t_{90}$  aus Bild 6.19, die Herleitung dieser Zusammenhänge erfolgt in Anhang A3.

Die Werte für die Zeiten  $D_c$  und  $D_v$  entsprechen den Abschätzungen aus Abschnitt 6.7.1.2.

Die Parameter  $N$ ,  $T_N$ ,  $K_M$ ,  $K_{ov}$  und  $R_{ER}$  der Netznachbildung sowie der Parameter  $N_M$  der Ratenmessung wurden experimentell bestimmt. Die Wahl der maximal für ABR-Verkehr zur Verfügung stehenden Bandbreite  $r_{max}^{(Net)}$  erfolgt so, daß die Last des LAN-Verkehrs einen deutlichen Anteil an der Gesamtlast ausmacht und somit die Effekte, die durch unterschiedliche Reservierungsarten auftreten, deutlich sichtbar sind.

## 6.8 Leistungsuntersuchung

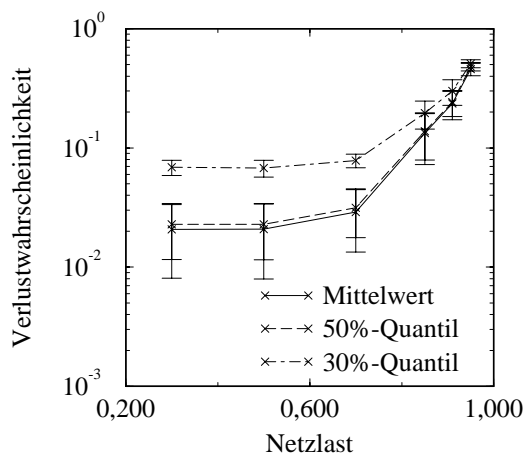
### 6.8.1 Diskussion einiger Modellparameter

Die im Verlauf der Modellbeschreibung angegebenen unterschiedlichen Auswertemethoden für die Verteilungen, die durch die Verteilungsprognose vorhergesagt werden, werden im folgenden verglichen. Die Bewertung der Methoden erfolgt aufgrund der Verluste und der mittleren Wartezeit am Zellpuffer sowie aufgrund der zuviel reservierten Bandbreite. Für die Bewertung der Überreservierung wird der Anteil der während eines Simulationslaufs durch die ABR-Quelle gesendeten Datenmenge an der reservierten Bandbreite bestimmt. Wenn dieser Reservierungsfaktor große Werte annimmt, werden Netzressourcen reserviert, ohne genutzt zu werden. Dies erhöht die Kosten der Verbindung, ohne daß dafür Daten übertragen werden.

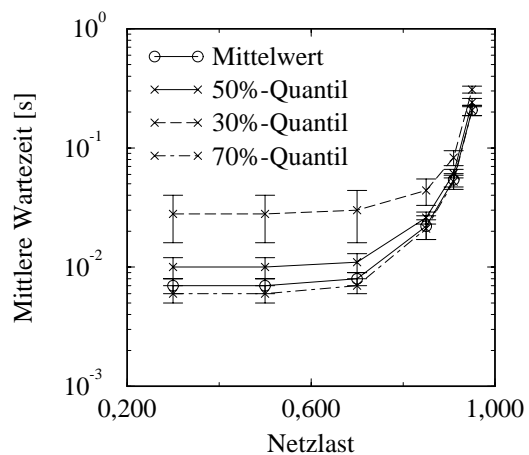
Der Vergleich umfaßt vier einfache Methoden: Reservierung durch den Mittelwert, durch das 30%-Quantil, 50%-Quantil oder durch das 70%-Quantil der prognostizierten Verteilungen. Die Bilder 6.26, 6.27 und 6.28 zeigen die Verlustwahrscheinlichkeiten, die Wartezeiten und den Grad der Überreservierung in Abhängigkeit von der mittleren Netzlast  $r_k^{(Net)}$  für alle vier Auswertemethoden. In Bild 6.26 sind die Kurven für Mittelwert und 70%-Quantil deckungsgleich. Daher ist nur diejenige für die Mittelwertauswertung dargestellt.

Es ist zu erkennen, daß die Ergebnisse bei Verwendung des Mittelwerts und des 70%-Quantils relativ gut übereinstimmen. Hier treten die geringsten Verluste und Wartezeiten auf. Dagegen ist in diesen Fällen die Überreservierung von Bandbreite am höchsten. Im Vergleich dazu treten bei Verwendung des 50%-Quantils um etwa 2-5% höhere Verluste und Wartezeiten auf. Auf der anderen Seite ist die Überreservierung in diesem Fall um etwa 5-10% niedriger. Die Verwendung der 30%-Quantile hat zwar eine geringe Überreservierung zur Folge, die Verluste und Verzögerungen sind aber erheblich höher als bei den anderen Verfahren.

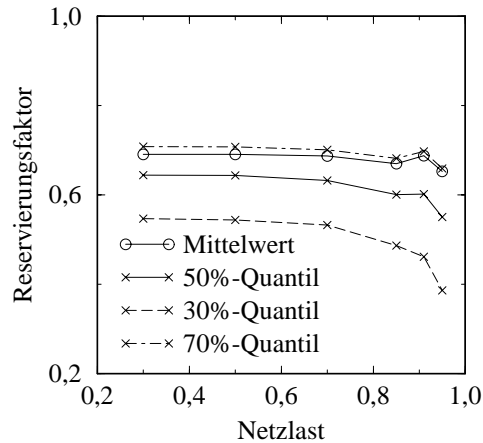
Für die im weiteren präsentierten Ergebnisse werden die Werte für die Bandbreitereservierungen durch Bildung der 50%-Quantile aus den Verteilungen am Ausgang der Verteilungsprognose berechnet, da die hierfür erzielten Ergebnisse die geeignetste Kombination darstellen.



**Bild 6.26:** Verluste für mehrere Verfahren zur Verteilungsauswertung

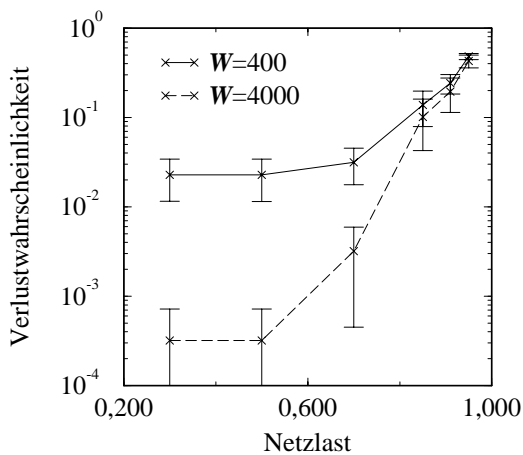


**Bild 6.27:** Wartezeiten für mehrere Verfahren zur Verteilungsauswertung

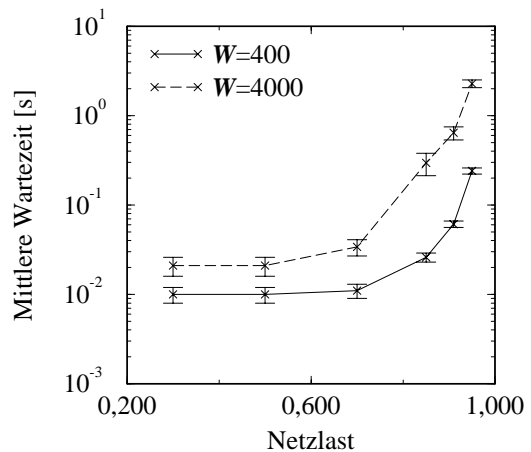


**Bild 6.28:** Überreservierung für mehrere Verfahren zur Verteilungsauswertung

Wie in Abschnitt 6.6 angedeutet, darf die Größe des Zellspeichers der Netzkoppeleinheit nicht zu groß gewählt werden, um die Verzögerungen für Interaktiv-Verkehr zu beschränken. Kleine Zellspeicher vergrößern dagegen die Paketverluste. Die Bilder 6.29 und 6.30 zeigen die Verlustwahrscheinlichkeiten und Wartezeiten an einem Zellpuffer in Abhängigkeit von der mittleren Netzlast für zwei Puffergrößen:  $W = 400$  Zellen und  $W = 4000$  Zellen. Aufgrund dieser Ergebnisse wird  $W = 400$  gewählt, da die Verzögerungen für den größeren Puffer für eine große Netzlast bis zu einer Größenordnung größer sind und nicht toleriert werden können. Die Verluste für  $W = 400$  sind zwar deutlich größer als die für  $W = 4000$ , sie werden aber toleriert, da sie im Bereich der Verluste im LAN liegen.



**Bild 6.29:** Verlustwahrscheinlichkeiten für unterschiedliche Wartespeichergrößen



**Bild 6.30:** Wartezeiten für unterschiedliche Wartespeichergrößen

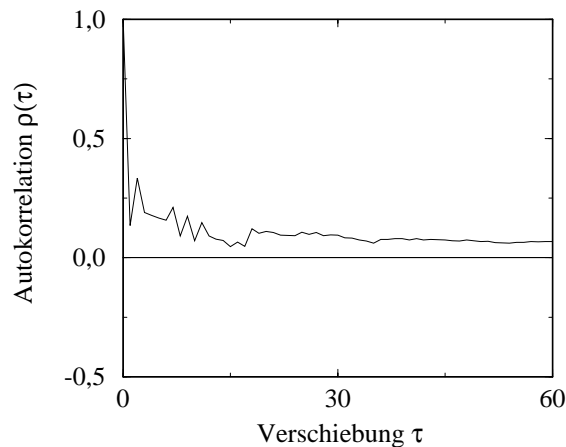
## 6.8.2 Untersuchung der unterschiedlichen Reservierungsverfahren

Die in Abschnitt 6.5 beschriebenen unterschiedlichen Reservierungsverfahren werden im weiteren untersucht:

- Reservierung der Spitzenzellrate der Verbindung
- Reservierung der als letztes gemessenen Datenrate im LAN
- Reservierung von Bandbreite aufgrund von Vorhersagewerten, die durch geeignete Auswertung von Verteilungsprognosen gewonnen werden (50%-Quantile)

Die drei Verfahren werden in den Schaubildern dieses Abschnitts mit den Abkürzungen „PCR“, „letzte Rate“ und „Prognose“ bezeichnet. Die betrachteten Leistungsdaten sind die Verluste und Wartezeiten am Zellpuffer der Netzkoppeleinheit sowie die Überreservierung durch das verwendete Reservierungsverfahren.

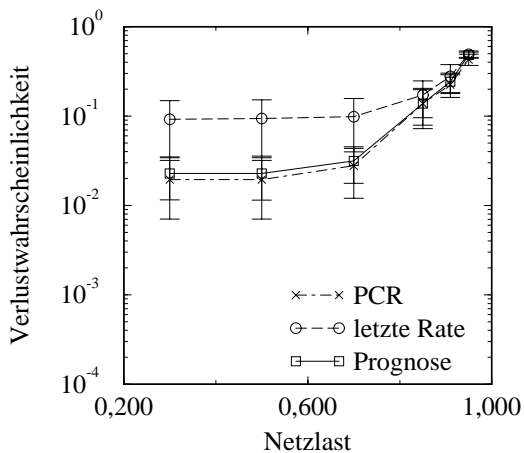
Die adaptiven Elemente der Verteilungsprognose werden in einer Lernphase an einen vorliegenden Datensatz angepaßt. Um zu zeigen, daß dadurch der Einsatz des darauf basierenden Prognoseverfahrens nicht auf diesen speziellen Lerndatensatz beschränkt ist, werden im folgenden auch Ergebnisse für einen zweiten Testdatensatz gezeigt. Dieser Datensatz hat etwas andere Eigenschaften als der Lerndatensatz. Die in Bild 6.31 dargestellte Korrelation ist deutlich geringer als die des Lerndatensatzes (s. Bild 6.12). Andererseits ist die Last im LAN etwa 40% höher als für den Lerndatensatz.



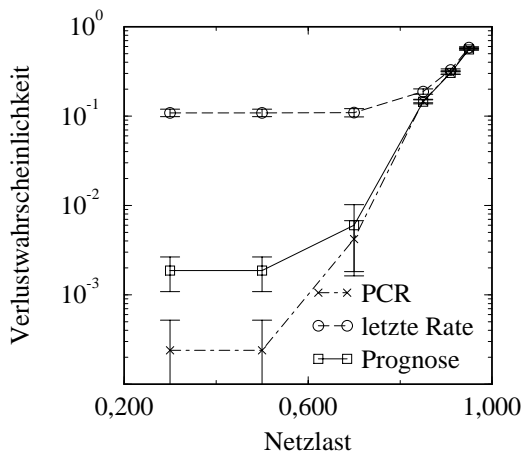
**Bild 6.31:** Korrelation der Paketlängen des Testdatensatzes

Die Bilder 6.32 und 6.33 zeigen für beide Datensätze die Verlustwahrscheinlichkeiten für unterschiedliche Lastzustände des ATM-Netzes. Die Verluste bei Reservierung der maximal möglichen Rate („PCR“) sind erwartungsgemäß am geringsten. Die größten Verluste treten bei Verwendung des letzten Ratenmeßwerts zur Reservierung auf („letzte Rate“). Aufgrund der geringeren Korrelation des zweiten Datensatzes treten dort deutlich geringere Verluste auf, obwohl das Angebot der Verbindung höher ist. Für steigende Last im ATM-Netz können kaum noch ABR-Daten durchgesetzt werden und es treten hohe Verluste auf, die aufgrund des höheren Angebots für den zweiten Datensatz höher ausfallen als für den ersten.

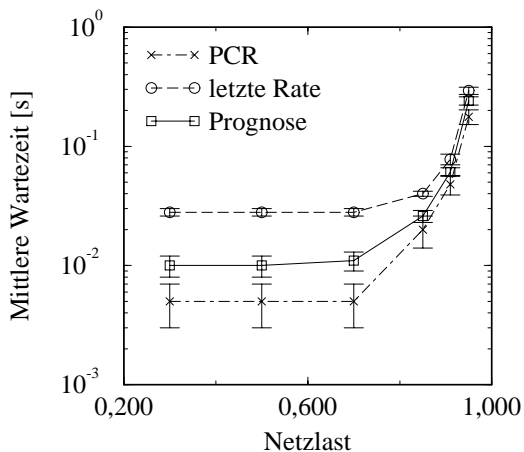




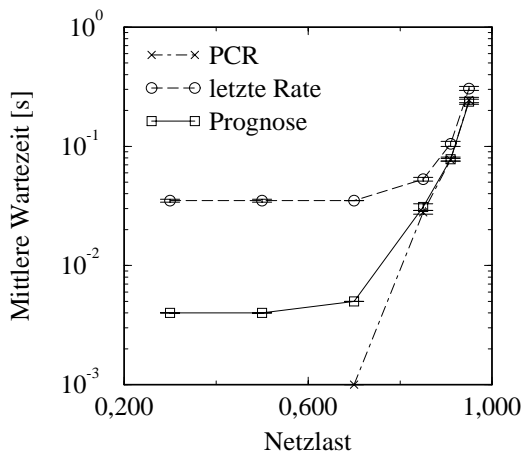
**Bild 6.32:** Verlustwahrscheinlichkeit über der Last (Lerndatensatz)



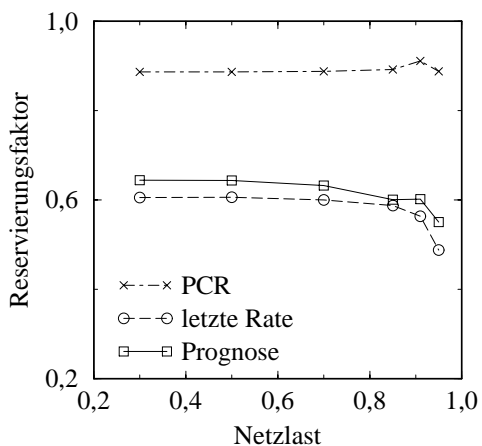
**Bild 6.33:** Verlustwahrscheinlichkeit über der Last (Testdatensatz)



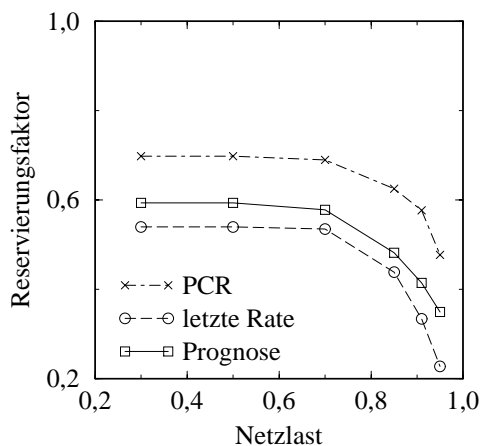
**Bild 6.34:** Wartezeit über der Last (Lerndatensatz)



**Bild 6.35:** Wartezeit über der Last (Testdatensatz)



**Bild 6.36:** Überreservierung über der Last (Lerndatensatz)



**Bild 6.37:** Überreservierung über der Last (Testdatensatz)

Die in den Bildern 6.34 und 6.35 dargestellten Ergebnisse für die Wartezeiten entsprechen in ihrer Tendenz weitgehend den Ergebnissen für die Verluste. Die Unterschiede zwischen der Reservierung von *PCR* und dem Prognoseverfahren sind hier größer, speziell für den zweiten Datensatz treten bei kleinen Netzlasten fast keine Wartezeiten bei der *PCR*-Reservierung auf.

In den Bildern 6.36 und 6.37 ist der Faktor der Überreservierung dargestellt. Hier ist deutlich zu erkennen, daß durch das Prognoseverfahren etwas mehr Bandbreite umsonst reserviert wird als durch Reservierung aufgrund der letzten Rate. Andererseits liegt die *PCR*-Reservierung weit über den beiden anderen Verfahren. Diese Auswertung zeigt deutlich den Unterschied in der Last, die abhängig von den beiden Datensätzen im ATM-Netz verursacht wird: Der Reservierungsfaktor für *PCR*-Reservierung in Bild 6.37 liegt weit niedriger als in Bild 6.36, da von der reservierten Rate ein viel höherer Anteil auch genutzt wird. Ebenso ist der stärkere Abfall der Überreservierung für den zweiten Datensatz zu erklären, da hier die durch die höhere Netzlast reduzierte erlaubte Senderate besser ausgenutzt wird.

## 6.9 Bewertung

In diesem Kapitel wurde ein zweites Anwendungsgebiet für das Prognoseverfahren aus Kapitel 3 beschrieben. Im Gegensatz zu Kapitel 5 werden die durch das Prognoseverfahren vorhergesagten Verteilungen nicht zur Bestimmung von Zufallswerten verwendet, sondern zur Berechnung von Prognosewerten.

Das Beispiel, an dem die Prognose zukünftiger Werte demonstriert wird, besteht aus einer Kopplung von Ethernet-LANs über ein ATM-Netz, für die die benötigte Bandbreite über das ABR-Protokoll reserviert wird. Die Ergebnisse zeigen die Eignung des Verfahrens und seine gegenüber anderen Verfahren höhere Leistungsfähigkeit.

Die Bestimmung von Reservierungswerten aus Verteilungen erfolgt hier durch Berechnung von Mittelwert oder Quantilen dieser Verteilungen. Diese Vorgehensweise kann durch zusätzliche Auswertungen der Verteilungen weiter verbessert werden, was auch eine weitere Verbesserung der Ergebnisse erwarten läßt. Grundsätzlich zeigt sich hier die bei Verwendung der Verteilungsprognose gewonnene Flexibilität, da unterschiedliche Auswertungen der Verteilungen möglich sind. Durch die Wahl einer konkreten Auswertung kann in diesem Beispiel darüberhinaus die Dienstgüte der betrachteten ABR-Verbindung gesteuert werden.

Anhand der Simulationsergebnisse wird deutlich, daß die Verteilungsprognose während ihres Lernprozesses die grundsätzlichen stochastischen Eigenschaften des LAN-Verkehrs sehr gut lernt. Daher lassen sich auch für andere als den zum Lernen verwendeten Datensatz sehr gute Prognoseergebnisse beobachten.

Neben dem Einsatz zur Vorhersage des Bandbreitebedarfs einer Verbindung sind weitere Einsatzmöglichkeiten in dem betrachteten Szenario denkbar. So kann die Verteilungsprognose z. B. auch in ATM-Vermittlungsknoten zur Vorhersage des Verhaltens einzelner Verbindungen oder der aggregierten Last durch einzelne Verkehrsklassen eingesetzt werden.

# Kapitel 7

## Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein neues Prognoseverfahren beschrieben und seine Einsatzfähigkeit an exemplarischen Beispielen demonstriert. Dieses Verfahren erlaubt die Vorhersage der Verteilung eines stochastischen Prozesses aufgrund von beobachteten vergangenen Werten. Der Vorteil liegt hierbei in der Vielzahl von Möglichkeiten für die Auswertung der vorhergesagten Verteilungen, die klassische Prognoseverfahren nicht bieten.

Anhand von zwei Anwendungsgebieten aus der Kommunikationstechnik wurden unterschiedliche Auswertemöglichkeiten gezeigt. Die erste Möglichkeit besteht darin, die vorhergesagten Verteilungen zur Bestimmung von Zufallwerten zu nutzen und auf diese Weise hochwertige Quellmodelle zu erhalten. Die zweite Möglichkeit besteht in der Auswertung der Verteilungen mit dem Ziel, einzelne Werte für bestimmte Vorhersagen zu gewinnen. Diese Methodik wurde beispielhaft zur Bandbreitenvorhersage für ABR-Verbindungen in ATM-Netzen genutzt. Hier zeigt sich die neu gewonnene Flexibilität besonders gut darin, daß sich durch geeignete Wahl der Auswertung die Dienstgüte der betrachteten ATM-Verbindungen gezielt beeinflussen läßt.

Eine wesentliche Eigenschaft der Verteilungsprognose ist ihre Lernfähigkeit, die aus der Verwendung künstlicher neuronaler Netze zu ihrem Aufbau resultiert. Dadurch ist eine automatische Anpassung an eine breite Klasse von stochastischen Prozessen möglich, ohne daß eine längere Analysephase vorangestellt werden muß. Auf diese Weise können Quellmodelle oder allgemeine Prognosemodule weitgehend automatisch und durch Angabe nur weniger Parameter an eine Aufgabenstellung adaptiert werden.

Die in dieser Arbeit angeführten Beispiele realisieren die folgenden Auswertemöglichkeiten für die prognostizierten Verteilungen: Neben der Auswertung zur Zufallszahlenerzeugung wurden Auswertungen zur Bestimmung des Mittelwerts und von Quantilen verwendet.

Für andere Anwendungen kann die statistische Auswertung von Varianz oder höheren Momenten sinnvoll sein, die ebenfalls aus den Verteilungen bestimmt werden können.

Neben den Anwendungen, die in dieser Arbeit gezeigt wurden, gibt es eine Reihe weiterer Gebiete der Kommunikationstechnik, auf denen die Verteilungsprognose gewinnbringend eingesetzt werden kann. Dazu gehört z. B. die Adaption von Protokollparametern an den übertragenen Verkehr, so daß bestimmte Leistungsmerkmale optimiert werden. Weitere mögliche Einsatzgebiete sind Verfahren der Zugangskontrolle (beispielsweise CAC bei ATM) oder Routingstrategien, die durch Berücksichtigung des erwarteten zukünftigen Verkehrsaufkommens optimale Entscheidungen treffen können.

# Kapitel 8

## Literaturverzeichnis

- [1] Ali, K. K. M. and Kamoun, F.: A Neural Network Approach to the Maximum Flow Problem. *Proceedings of GLOBECOM '91*, Volume 1, pp. 130-134.
- [2] Ali, M. M. and Nguyen, H. T.: A Neural Network Controller for a High-Speed Packet Switch. *IEEE International Telecommunications Symposium*, Rio de Janeiro, Brazil, September 3-6, 1990, pp. 493-497.
- [3] Ali, M. M. and Nguyen, H. T.: A Neural Network Implementation of an Input Access Scheme in a High-Speed Packet Switch. *Proceedings of GLOBECOM '89*, pp. 32.7.
- [4] Almeda, L.: Backpropagation in Perceptrons with Feedback. In Eckmiller, R. and von der Malsberg, C. (Eds.): *Neural Computer*, NATO ASI Series, F (41), Springer, Berlin, 1988, pp. 199-208.
- [5] Ansari, N. and Liu, D.: The Performance Evaluation of a New Neural Network Based Traffic Management Scheme for a Satellite Communication Network. *Proceedings of GLOBECOM '91*, Volume 1, pp. 110-114.
- [6] ATM Forum Technical Committee: User-Network Interface (UNI) Specification, Version 3.1. *ATM Forum af-uni-0010.002*, September 1994.
- [7] ATM Forum Technical Committee: ATM User-Network Interface (UNI) Signalling Specification, Version 4.0. *ATM Forum af-sig-0061.000*, July 1996.
- [8] TM Forum Technical Committee: LAN Emulation Over ATM, Version 1.0. *ATM Forum af-lane-0021.000*, January 1995.
- [9] TM Forum Technical Committee: LAN Emulation Client Management Specification v1.0. *ATM Forum af-lane-0044.000*, September 1995.
- [10] ATM Forum Technical Committee: LAN Emulation Servers Management Specification 1.0. *ATM Forum af-lane-0057.000*, March 1996.
- [11] ATM Forum Technical Committee: Traffic Management Specification, Version 4.0. *ATM Forum af-tm-0056.000*, April 1996.
- [12] Bagchi, A.: *Optimal Control of Stochastic Systems*. Prentice Hall, 1993.
- [13] Basso, A. and Kunt, M.: Autoassociative Neural Networks for Image Compression. *European Transactions on Telecommunications*, Vol. 3, No. 6, 1992, pp. 593-598.

- [14] Block, H. D.: The Perceptron: A Model for Brain Functioning. *Rev. Mod. Phys.*, 1962, Vol. 34, No. 123.
- [15] Bradburn, D. S. : Reducing Transmission Error Effects Using a Self-Organizing Network. *Proceedings of International Joint Conference on Neural Networks*, Vol. 2, IJCNN 89, Washington D.C., pp. II-531 - II-537.
- [16] Brillinger, D.: *New Directions in Time Series Analysis, Part I+II*. Springer, 1992.
- [17] Bronstein, I. N. und Semendjajew, K. A.: *Taschenbuch der Mathematik*. Verlag Harry Deutsch, 21. Auflage, 1984.
- [18] Brown, T.: Neural Networks for Switching. *IEEE Communications Magazine*, November 1989, pp. 72-81.
- [19] Chakraborty, K., Mehrotra, K., Mohan, C. K. and Ranka, S.: Forecasting the Behavior of Multivariate Time Series Using Neural Networks. *Neural Networks*, 1992, vol. 5, pp. 961-70.
- [20] Chen, X. and Leslie, L. M.: A Neural Network Approach towards Adaptive Congestion Control in Broadband ATM Networks. *Proceedings of GLOBECOM '91*, Volume 1, pp. 115-119.
- [21] Chiu, D. M. and Jain, R.: Analysis of the Increase and Decrease Algorithm for Congestion Avoidance in Computer Networks. *Computer Networks and ISDN Systems*, Vol. 17, 1989, pp. 1-14.
- [22] Chugo, A. and Iida, I.: Dynamic Path Assignment for Broadband Networks Based on Neural Computation. *IEICE Transactions on Communications*, Vol. E75-B, No. 7, July 1992, pp. 634-640.
- [23] Claus, J. und Siegmund, G.: *Das ATM-Handbuch*. Hüthig Verlag, 1995.
- [24] Coleman, K. G.: Neural Networks – A High Impact Technology with Applications in Telecommunications. *Proceedings of GLOBECOM '91*, pp. 1038-1042.
- [25] Cubero, R. G.: Neural Networks for Water Demand Time Series Forecasting. In: *Artificial Neural Networks* (ed. Prieto, A.), Springer, '91, pp. 401-408.
- [26] Deppisch, J., Bauer, H. U. and Geisel, T.: Hierarchical training of neural networks and prediction of chaotic time series, in *Artificial Neural Networks: Forecasting Time Series* (ed. V. R. Vemuri, R. D. Rogers), IEEE Computer Society Press, 1994, pp. 66-71.
- [27] Dony, R. D. and Haykin, S.: Neural Network Approaches to Image Compression. *Proceedings of the IEEE*, Vol. 83, No. 2, Feb. 1995, pp. 288-303.
- [28] Enssle, J.: *Modellierung und Leistungsuntersuchung eines verteilten Video-On-Demand-Systems für MPEG-codierte Videodatenströme mit variabler Bitrate*. Dissertation, Institut für Nachrichtenvermittlung und Datenverarbeitung, IND, Universität Stuttgart, eingereicht, 1996.

- [29] Enssle, J.: Modelling of Short and Long Term Properties of VBR MPEG Compressed Video in ATM Networks. *Proceedings of the 1995 Silicon Valley Networking Conference (SVNC 95)*, 1995, pp. 95-108.
- [30] EXPLOIT – Workpackage 3.6: *Final Report on Network Performance*. R2061/EXP/SW3/DS/P/048/B1, 12/95.
- [31] Fahlman, S. E.: An Empirical Study of Learning Speed in Back-Propagation Networks. *Research Report CMU-CS-88-162*, School of Computer Science, Carnegie Mellon University, Pittsburg, 1988.
- [32] Fahlman, S. E. and Lebiere, C.: The Cascade-Correlation Learning Architecture. *Research Report CMU-CS-90-100*, School of Computer Science, Carnegie Mellon University, Pittsburg, 1990.
- [33] Fioravanti, F. and Giusto, D. D.: Inter-Block Redundancy Reduction in Vector-Quantized Images by a Neural Predictor. *European Transactions on Telecommunications*, Vol. 3, No. 6, 1992, pp. 605-607.
- [34] Fritsch, T., Mittler, M. and Tran-Gia, P.: Artificial Neural Net Applications in Telecommunication Systems. *Neural Computing and Applications*, Springer, 1993, pp. 124-146.
- [35] Fukushima, K.: Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural Networks*, 1988, vol. 1, pp. 119-130.
- [36] Gibson, G. J. and Cowan, C. F. N.: On the Decision Regions of Multilayer Perceptrons. *Proceedings of the IEEE*, Vol. 78, No. 10, October 1990, pp. 1590-1594.
- [37] Gruenenfelder, R., Cosmas J. P. and Odinma-Okafor, A.: Characterization of Video Codecs as Autoregressive Moving Average Processes and Related Queueing System Performance. *IEEE Journal on Selected Areas in Communications*, 1991, vol. 9, no. 3, pp. 284-93.
- [38] Hamilton, J. D.: *Time Series Analysis*. Princeton University Press, 1994.
- [39] Harvey, A. C.: *Time Series Models*. Harvester Wheatsheaf, 1993.
- [40] Hebb, D. O.: *The Organization of Behavior: A Neurophysiological Theory*. Wiley, New York, 1949.
- [41] Hecht-Nielsen, R.: *Neurocomputing*. Addison-Wesley, 1989.
- [42] Hiramatsu, A.: ATM Communications Network Control by Neural Networks. *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, March 1990, pp. 122-130.
- [43] Hiramatsu, A.: Integration of ATM Call Admission Control And Link Capacity Control By Distributed Neural Networks. *Proceedings of GLOBECOM '90*, pp. 708.6.1-708.6.5.
- [44] Ho, Y. and Cao, X.: *Perturbation Analysis of Discrete Event Dynamic Systems*. Kluwer Academic Publishers, 1991.

- [45] Hopfield, J. J.: Neural Networks and Physical Systems with Emergent Collective Computational Abilities. *Proc. Natl. Acad. Sci. USA*, 1982, Vol. 79, No. 2554.
- [46] Hopfield, J. J. and Tank, D. W.: 'Neural' Computation of Decisions in Optimization Problems. *Biol. Cybern*, 1985, Vol. 52, No. 141.
- [47] Hudson, J. L., Kube, M., Adomaitis, R. A., Kevrekidis, I. G., Lapedes, A. S. and Farber, R. M.: Nonlinear Signal Processing and System Identification: Applications to Time Series from Electrochemical Reactions, in *Artificial Neural Networks: Forecasting Time Series* (ed. V. R. Vemuri, R. D. Rogers), IEEE Computer Society Press, 1994, pp. 36-42.
- [48] Hurst, H. E.: Long-Term Storage Capacity of Reservoirs, *Trans. Amer. Soc. Civil Eng.*, 1951, Vol. 116, pp. 770-799.
- [49] ISO 7498: Information Processing Systems – Open Systems Interconnection – Basic Reference Model. *ISO Standard*, 1984.
- [50] ITU-T G.708: Network Node Interface for the Synchronous Digital Hierarchy. *ITU-T Recommendation G.708*, 03/93.
- [51] ITU-T G.709: Synchronous Multiplexing Structure. *ITU-T Recommendation G.709*, 03/93.
- [52] ITU-T I.321: B-ISDN Protocol Reference Model and its Applications. *ITU-T Recommendation I.321*, 1991.
- [53] ITU-T I.361: B-ISDN ATM Layer Specification. *ITU-T Recommendation I.361*, 11/95.
- [54] ITU-T I.362: B-ISDN ATM Adaptation Layer (AAL) Functional Specification. *ITU-T Recommendation I.362*, 03/93.
- [55] ITU-T I.363: B-ISDN ATM Adaptation Layer (AAL) Specification. *ITU-T Draft Recommendation I.363*, 09/95.
- [56] ITU-T I.371: Traffic Control and Congestion Control in B-ISDN. *ITU-T Draft Recommendation I.371*, 05/96.
- [57] ITU-T I.432: B-ISDN User-Network Interface – Physical Layer Specification. *ITU-T Recommendation I.432*, 03/93.
- [58] Izquierdo, A. C., Sueiro, J. C. and Méndez, J. A. H.: Self-Organizing Feature Maps and their Application to Digital Coding of Information. In: *Artificial Neural Networks* (ed. Prieto, A.), Springer, 1991, pp. 401-408.
- [59] Jain, R.: A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks. *Computer Communications Review*, Vol. 19, No. 5, 1989, pp. 56-71.
- [60] Jain, R.: Congestion Control in Computer Networks; Issues and Trends. *IEEE Network Magazine*, May 1990, pp. 24-30.
- [61] Janacek, G.: *Time Series*. Ellis Horwood, 1993.

- [62] Kameyama, K. and Kosuki, Yukio: Automatic Fusion and Splitting of Artificial Neural Elements in Optimizing the Network Size. *Proceedings of 1991 IEEE intern. Conference on Systems, Man, and Cybernetics*, pp. 1633-1638.
- [63] Kameyama, K. and Kosuki, Yukio: Neural Network Pruning by Fusing Hidden Layer Units. *IEICE Transactions*, Vol. E 74, No. 12, December 1991, pp. 4198-4204.
- [64] Kamoun, F. and Ali, M. K. M.: A Neural Network Shortest Path Algorithm for Optimum Routing in Packet-Switched Communication Networks. *Proceedings of GLOBECOM '91*, Volume 1, pp. 120-124.
- [65] Khasnabish, B., Ahmadi, M. and Shridgar, M.: Congestion Avoidance in Large Supra-High-Speed Packet Switching Networks Using Neural Arbiters. *Proceedings of GLOBECOM '91*, Volume 1, pp. 140-144.
- [66] Kleinrock, L.: *Queuing Systems, Vol. 1+2*. John Wiley & Sons, 1975.
- [67] Kocher, Hartmut: *Entwurf und Implementierung einer Simulationsbibliothek unter Anwendung objektorientierter Methoden*. Dissertation, Institut für Nachrichtenvermittlung und Datenverarbeitung, IND, Universität Stuttgart, 1994.
- [68] Kohonen, T.: The Self-Organizing Map. *Proceedings of the IEEE*, Vol. 78, No. 9, September 1990, pp. 1464-1480.
- [69] Kühn, P. J. K.: *Manuskript zur Vorlesung „Wartezeitprobleme der Daten- und Nachrichtenverkehrstheorie“*, Universität Stuttgart, Institut für Nachrichtenvermittlung und Datenverarbeitung.
- [70] Lancini, R., Perego, F. and Tubaro, S.: Some Experiments on Vector Quantization Using Neural Nets. *Proceedings of GLOBECOM '91*, Volume 1, pp. 135-139.
- [71] Law, A. M. and Kelton, W. D.: *Simulation Modelling & Analysis*. McGraw-Hill, 1991.
- [72] Le Gall, D.: MPEG: A Video Compression Standard for Multimedia Applications, *Communications of the ACM*, 1991, vol. 34, no. 4, 46-58.
- [73] Leland, W. E., Taqqu, M. S., Willinger, W. and Wilson, D. V.: On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Transactions on Networking*, Vol. 2, No. 1, February 1994, pp. 1-15.
- [74] Lowe, D. and Webb, A. R.: Time series prediction by adaptive networks: a dynamical systems perspective, in *Artificial Neural Networks: Forecasting Time Series* (ed. V. R. Vemuri, R. D. Rogers), IEEE Computer Society Press, 1994.
- [75] Mandelbrot, B. B. and Wallis, J. R.: Computer Experiments with Fractional Gaussian Noises. Parts 1-3. *Water Resources Research*, Vol. 5, No. 1, 1969, pp. 228-267.
- [76] Mandelbrot, B. B. and Wallis, J. R.: Robustness of the Rescaled Range R/S in the Measurement of Noncyclic Long Run Statistical Dependence. *Water Resources Research*, Vol. 5, No. 5, 1969, pp. 967-988.



- [77] Marrakchi, A. and Troudet, T.: A Neural Net Arbiter for Large Crossbar Packet-Switches. *Transactions on Circuits and Systems*, Vol. 36, No. 7, July 1989, pp. 1039-1047.
- [78] McCulloch, W. S. and Pitts, W.: A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bull. Math. Biophys.*, 1943, Vol. 5, No. 115.
- [79] Minsky, M., and Papert, S.: *Perceptrons*, MIT Press, 1969.
- [80] Morris, R. J. T.: Prospects for Neural Networks in Broadband Network Resource Management. *Proceedings of the 13th International Teletraffic Congress (ITC)*, pp. 335-340.
- [81] Mozer, M. C.: Neural Net Architectures for Temporal Sequence Processing, in *Time Series Prediction: Forecasting the Future and Understanding the Past* (ed. A. S. Weigend), Addison-Wesley, 1993, pp. 243-64.
- [82] Müller, B., Reinhardt, J.: *Neural Networks, An Introduction*. Springer, 1990.
- [83] Necker, T.: *Verbindungsannahme in ATM-Systemen basierend auf rekurrenten Neuronalen Netzen*. Diplomarbeit, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1993.
- [84] Ortuño, T. , Ortuño, M. and Delgado, J. A.: Neural Networks as Error Correcting Systems in Digital Communications. In: *Artificial Neural Networks* (ed. Prieto, A.), Springer, 1991, pp. 409-414.
- [85] Papoulis, A.: *Probability, Random Variables, and Stochastic Processes*. McGraw-Hill, 1984.
- [86] Rauch, H. E. and Winarske, T.: Neural Networks for Routing Communication Traffic. *IEEE Control Systems Magazine*, April 1988, pp. 26-31.
- [87] Rehm, W. und Sterzing, V.: Ein evolutionstheoretisches Optimierungsverfahren für Multi-Layer-Perceptrons. *Informationstechnik*, 5/92, S. 307-312.
- [88] Renger, T.: *Verbindungsannahme in ATM-Systemen basierend auf Neuronalen Netzen*. Diplomarbeit, Institut für Nachrichtenvermittlung und Datenverarbeitung, Universität Stuttgart, 1993.
- [89] RFC 1577: Classical IP and ARP over ATM. *Request for Comments: 1577, Network Working Group*, January 1994.
- [90] Rosenblatt, F.: *Principles of Neurodynamics*, Spartan, New York, 1962.
- [91] Rumelhart, D. E. and McClelland, J. L.: Parallel Distributed Processing: *Explorations in the Microstructure of Cognition, I & II*, MIT Press, 1986.
- [92] Salomon, R.: Beschleunigtes Lernen durch adaptive Regelung der Lernrate bei back-propagation in feed-forward Netzen. *Konnektionismus in Artificial Intelligence und Kognitionsforschung* (Hrsg.: G. Dorffner), Springer, 1990, S. 173-178.

- [93] Scargle, J. D.: Predictive Deconvolution of Chaotic and Random Processes, in *New Directions in Time Series Analysis, Part I* (ed. D. Brillinger), Springer, 1992, pp. 335-56.
- [94] Schödl, W.: *Kopplung von DQDB-Regionalnetzen mit ATM-Weitverkehrsnetzen: Architektur, Steuerstrategien und Leistungsverhalten*. Dissertation, Institut für Nachrichtenvermittlung und Datenverarbeitung, IND, Universität Stuttgart, 1994.
- [95] Schürmann, B., Hollatz, J. and Ramacher, U.: Adaptive Recurrent Neural Networks And Dynamic Stability. *Lecture Notes in Physics* (ed. L. Garrido), Springer, 1990.
- [96] Silva, F. M. and Almeida, L. B.: Speeding up Backpropagation. *Advanced Neural Computers* (ed. R. Eckmiller), Elsevier, 1990, pp. 151-158.
- [97] Takahashi, T. and Hiramatsu, A.: Integrated ATM Traffic Control by Distributed Neural Networks. *Proceedings of International Switching Symposium (ISS) 1990*, Vol. III, pp. 59-65.
- [98] Tang, Z., de Almeida, C. and Fishwick, P. A.: Time series forecasting using neural networks vs. Box-Jenkins methodology, in *Artificial Neural Networks: Forecasting Time Series* (ed. V. R. Vemuri, R. D. Rogers), IEEE Computer Society Press, 1994.
- [99] Tarraf, A. A., Habib, I. W. and Saadawi, T. N.: Neural Networks for ATM Multimedia Traffic Prediction. *Proceedings of IWANNT '93*, 1993, pp. 85-91.
- [100] Tarraf, A. A., Habib, I. W. and Saadawi, T. N.: A Novel Neural Network Traffic Enforcement Mechanism for ATM Networks. *IEEE Journal on Selected Areas in Communications*, 1994, Vol. 12, No. 6, pp. 1088-96.
- [101] Touretzky, D. S. and Pomerleau, D. A.: What's Hidden in the Hidden Layers? *Byte*, August 1989, pp. 227-233.
- [102] Tran-Gia, P. and Gropp, O.: Structure and Performance of Neural Nets in Broadband System Admission Control. *Research Report Series, Report No. 37*, University of Würzburg, Institute of Computer Science, December 1991.
- [103] Troudet, T. P. and Walters, S. M.: Neural Network Architecture for Crossbar Switch Control. *IEEE Transactions on Circuits and Systems*, Vol 38, No. 1, January 1991, pp. 42-56.
- [104] Vemuri, V. R. and Rogers, R. D.: *Artificial Neural Networks: Forecasting Time Series*. IEEE Computer Society Press, 1994.
- [105] Wan, E. A.: Time Series Prediction Using a Connectionist Network with internal Delay Lines, in: *Time Series Prediction: Forecasting the Future and Understanding the Past* (ed. A. S. Weigend), Addison-Wesley, 1993, pp. 195-217.
- [106] Wang, D. and Arbib, M. A.: Complex Temporal Sequence Learning Based on Short-term Memory. *Proceedings of the IEEE*, Vol. 78, No. 9, September 1990, pp. 1536-1543.

- [107] Weigend, A. S.: *Time Series Prediction: Forecasting the Future and Understanding the Past*. Addison-Wesley, 1993.
- [108] Werbos, P. J.: Backpropagation Through Time: What It Does and How to Do It. *Proceedings of the IEEE*, Vol 78, No. 10, October 1990, pp. 1550-1560.
- [109] Widrow, B. and Hoff, M. E.: Adaptive switching circuits. *1960 IRE WESCON Convention Record*, New York, 1960, pp. 96-104.
- [110] Widrow, B., Rumelhart, D. E. and Lehr, M. L.: Neural Networks: Applications in Industry, Business and Science. *Communications of the ACM*, Vol. 37, No. 3, 1994, pp. 93-105.
- [111] Worster, T.: Neural Networks Based Controllers For Connection Acceptance. *Proceedings of 2nd Race Workshop, Traffic and Performance Aspects in IBCN*, 1992.

# Anhang

## Algorithmen zur Verteilungsprognose

### A1 Vektor-Quantisierer

#### A1.1 Einführung

Die Aufgabe eines Vektor-Quantisierers besteht darin, Vektoren des  $\mathbb{R}^N$  eindeutig auf einen Bereich der natürlichen Zahlen  $N$  abzubilden. Dadurch werden diese Vektoren in  $M$  Klassen klassifiziert. Die Realisierung erfolgt durch die Unterteilung des  $\mathbb{R}^N$  in  $M$  Gebiete. Fällt ein Vektor in eines der Gebiete, wird der Repräsentant dieser Klasse, z. B. der Mittelpunktsvektor, ermittelt. Dieser Repräsentant wird in einem Codebuch aufgesucht und sein Index als Ergebnis der Operation betrachtet. Für die Unterteilung des  $\mathbb{R}^N$  und die Erstellung des Codebuchs können unterschiedliche Algorithmen eingesetzt werden.

Vektor-Quantisierer (VQ) werden häufig zur Verminderung der Komplexität von Signalen, zur Komprimierung oder zur Irrelevanzminderung eingesetzt. Ein bekanntes Einsatzgebiet, auf dem Vektor-Quantisierer erfolgreich zur Signalkomprimierung eingesetzt werden, ist die Übertragung von Audio oder Video. Die Vorgehensweise ist hier, daß für jedes Datenwort mittels einer Übersetzungstabelle (Codebuch) ein zu übertragender Wert bestimmt wird (Quantisierungsvorgang). Bei diesem Vorgang erfolgt die oben erwähnte Abbildung; das Ergebnis ist die Nummer eines der erwähnten  $M$  Gebiete. Diese Nummer wird anschließend übertragen und auf der Empfangsseite wieder in den ursprünglichen Wertebereich umgesetzt. Das Resultat dieser Umsetzung ist der gespeicherte Vektor-Repräsentant.

In diesem Anhang wird ein Algorithmus zur Realisierung eines Vektor-Quantisierers beschrieben, der auf künstlichen neuronalen Netzen beruht. Er ist in einigen Details an das bekannte Kohonen-Netz und den entsprechenden Lernalgorithmus [41] angelehnt.

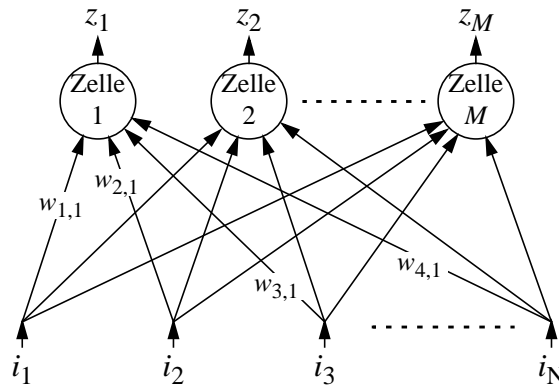
Der hier vorgestellte Algorithmus unterscheidet sich von anderen VQ-Algorithmen (z. B. [70]) dadurch, daß die Adaption an das Datenmaterial, das aus  $N$ -dimensionalen Vektoren besteht, durch einen automatischen Lernvorgang erfolgt („Lernen ohne Lehrer“). Ein weiterer Unterschied besteht darin, daß die Anpassung der  $M$  Gebiete des  $\mathbb{R}^N$  so erfolgt, daß die auftretenden Vektoren mit derselben Häufigkeit in alle Gebiete fallen. Besseres Lernverhalten und höhere Stabilität gegenüber aus der Literatur bekannten Ansätzen werden durch Berücksichtigung der Abhängigkeiten zwischen Parametern des Algorithmus und den Lerndaten erzielt.

Die Anwendung eines VQ erfolgt in der Regel in zwei Phasen: In der ersten Phase wird der VQ an das vorhandene Datenmaterial angepaßt (Lern- oder Adaptionphase), in der zweiten Phase nach Abschluß des Lernvorgangs wird der VQ als „Codebuch“ verwendet und nicht

weiter modifiziert (Wirkbetrieb oder Abrufphase). Im folgenden Abschnitt wird der Lernalgorithmus beschrieben.

## A1.2 Lernalgorithmus

Ein Vektor-Quantisierer ist, wie in Bild A.1 gezeigt, aus  $M$  gleichen Zellen aufgebaut, die alle dieselben Eingangswerte  $i_1, \dots, i_N$  in Form des Eingangsvektors  $\mathbf{i}$  zugeführt bekommen. Im Gegensatz zu vielen anderen KNN-Typen findet hier keine multiplikative Gewichtung der Eingangswerte durch Gewichtungsfaktoren statt, obwohl jedem Eingang einer Zelle ein Gewichtswert zugeordnet ist. Die Verwendung dieser Werte erfolgt allerdings anders als bei anderen Modellen, s. unten. Von den  $M$  Ausgängen des VQ nimmt immer genau einer den Wert 1 an, der Rest den Wert 0. Die Nummer des Ausgangs mit dem Wert 1 entspricht der Nummer der entsprechenden Klasse im Codebuch.



**Bild A.1:** VQ-Vernetzungsstruktur

Die Gewichte einer Zelle  $i$  zum Zeitpunkt  $k$  bilden einen Gewichtsvektor  $\mathbf{w}_{i,k} = (w_{1i,k}, w_{2i,k}, \dots, w_{Ni,k})$ , dessen Abstand vom Vektor  $\mathbf{i}_k = (i_{1,k}, i_{2,k}, \dots, i_{N,k})$  bewertet wird. Als Differenz (Abstandsmaß nach Euklid) ergibt sich

$$d_{i,k} = \|\mathbf{w}_{i,k} - \mathbf{i}_k\| = \sqrt{(w_{1i,k} - i_{1,k})^2 + \dots + (w_{Ni,k} - i_{N,k})^2}. \quad (\text{A.1})$$

Unter den  $M$  Zellen wird für den momentanen Eingangsvektor ein „Gewinner“ ermittelt. Dabei dient der Abstand  $d_{i,k}$  zusammen mit einem Skalierungsfaktor  $b_{i,k}$  als Kriterium. Die Zelle mit dem kleinsten Wert  $d_{i,k} \cdot b_{i,k}$  ist der Gewinner und ihr Ausgangswert  $z_{i,k}$  wird auf 1 gesetzt. Die Ausgänge aller anderen Zellen erhalten den Wert 0. Der Wert  $b_{i,k}$  ist ein Korrekturwert, der sicherstellt, daß alle Zellen nach einer gewissen Adaptionzeit mit derselben Häufigkeit  $1/M$  gewinnen. Er ist auch ein Maß für die Ausdehnung des Gebiets, das Zelle  $i$  repräsentiert: Kleine Werte von  $b_{i,k}$  stehen für ein großes Gebiet und umgekehrt.

Der Korrekturwert wird folgendermaßen bestimmt:

$$b_{i,k} = b_{i,k-1} \cdot (1 - (1 - M \cdot f_{i,k-1}) \cdot \gamma). \quad (\text{A.2})$$

Diese Gleichung verleiht dem System Integraleigenschaften, d. h.  $b_{i,k}$  wächst oder fällt, bis die Differenz  $1/M - f_{i,k}$  gegen Null geht. Über den Parameter  $\gamma$  kann die Adaptiongeschwindigkeit gesteuert werden.

Um dem System größere Stabilität zu verleihen, wird  $b_{i,k}$  auf einen bestimmten Wertebereich  $[b_{i,k,min}, b_{i,k,max}]$  beschränkt. Andernfalls können die  $b_{i,k}$  sehr große oder sehr kleine Werte annehmen, was bei einer Rechnerimplementierung zu internen Darstellungsproblemen führt.

Nachdem der Gewinner bestimmt ist, wird dessen Gewichtsvektor und der Wert  $f_{i,k}$  aller Zellen modifiziert:

$$\mathbf{w}_{i,k} = \mathbf{w}_{i,k-1} + \alpha(\mathbf{i} - \mathbf{w}_{i,k-1}) \quad (\text{A.3})$$

$$f_{i,k} = f_{i,k-1} + \beta(z_{i,k} - f_{i,k-1}) \quad (\text{A.4})$$

Die Hilfsgrößen  $b_i$  und  $f_i$  haben folgenden Einfluß auf die Adaption:  $f_i$  mißt die Häufigkeit, mit der Eingangsvektoren in den Einzugsbereich der Zelle  $i$  fallen und stellt somit eine Art Gedächtnis dar (EWMA-Algorithmus, s. Anhang A3). Das Ziel der Adaption ist ein eingeschwungener Zustand mit  $f_i = 1/M$ . Abhängig von  $f_i$  wird daher  $b_i$  nach Gl. (A.2) so berechnet, daß das Gebiet  $i$  vergrößert wird (Erhöhung der Häufigkeit), wenn  $f_i$  kleiner als  $1/N$  ist, oder verkleinert wird (Verringerung der Häufigkeit), wenn  $f_i$  größer als  $1/N$  ist.

Über die Parameter  $\alpha$  und  $\beta$  kann die Anpassungsgeschwindigkeit des Algorithmus gesteuert werden.

Bild 3.2 in Kapitel 3 zeigt an einem Beispiel mit  $N=2$ , wie die Raumaufteilung prinzipiell erfolgt.

### A1.3 Beispiel

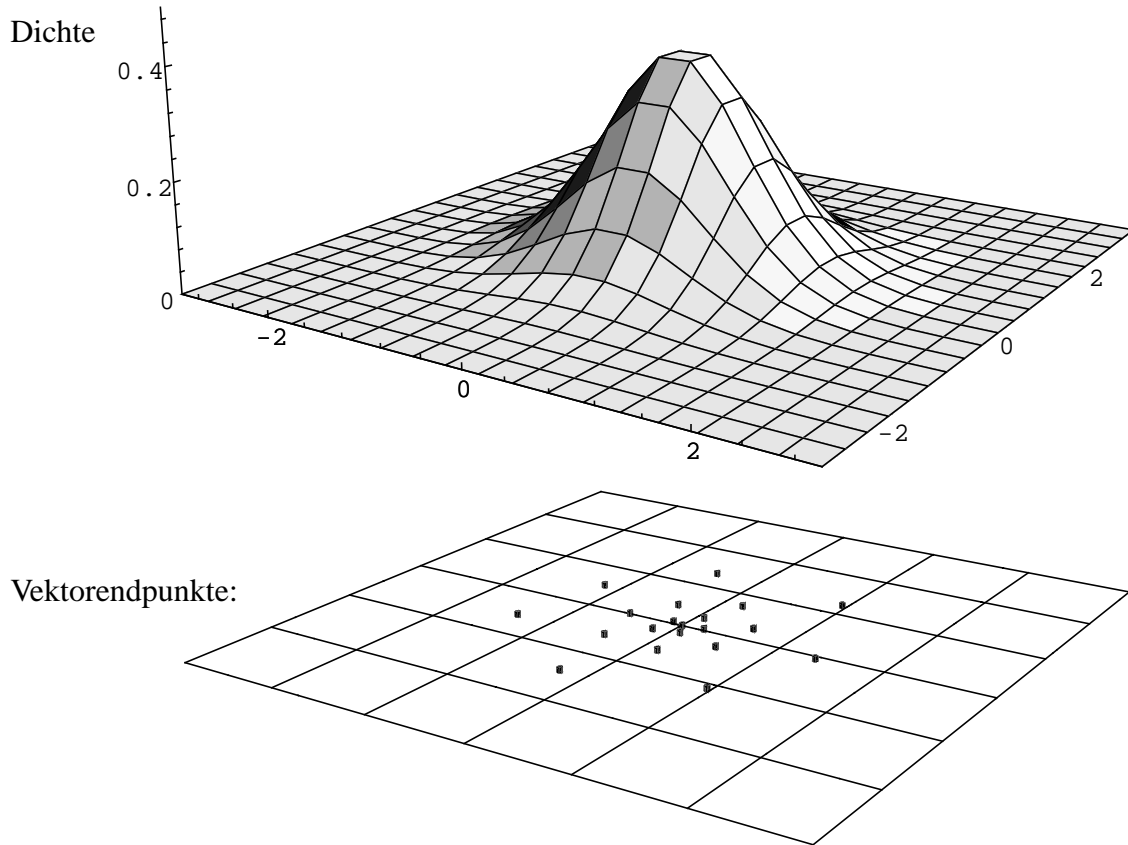
Am Beispiel einer zweidimensionalen Dichtefunktion mit der Form einer rotationssymmetrischer Gausskurve (s. Bild A.2, oben) wird die Verteilung von  $M=20$  Vektoren gezeigt. Aufgrund der Eigenschaft des VQ-Algorithmus, die Vektorendpunkte so zu verteilen, daß jeder Vektor gleich häufig angesprochen wird, häufen sich die Vektorendpunkte aufgrund der höheren Dichte in der Umgebung des Koordinatenursprungs. Bild A.2 zeigt im oberen Teil die zweidimensionale Dichtefunktion und im unteren Teil den Endpunkt jedes Vektors.

### A1.4 Parameterwahl

In Tabelle A.1 sind die Parameter des Algorithmus sowie Richtwerte für die Parameter zusammengefaßt.

### A1.5 Bewertung

Wegen der Verwandtschaft zu künstlichen neuronalen Netzen (viele einfache Verarbeitungselemente, wenig zentrale Operationen) ist der Algorithmus gut parallelisierbar und gut geeignet für eine Hardware-Implementierung.



**Bild A.2:** Beispiel zum Vektor-Quantisierer

**Tabelle A.1:** Parameter des Lernalgorithmus

| Parameter                      | Bedeutung und Bemerkungen  | Richtwert   |
|--------------------------------|--|---|
| $M$                            | Anzahl Zellen des VQ   | problemabhängig   |
| $N$                            | Anzahl Eingänge des VQ   | problemabhängig   |
| $\alpha$                       | Parameter für Adaption der Gewichtsvektoren<br>$\alpha$ wird während der Lernphase exponentiell verringert.  | Startwert: 0,05<br>Endwert: 0,00005                         |
| $b_{i,k,min}$<br>$b_{i,k,max}$ | Bereich, auf den $b$ begrenzt wird   | Intervall [0,1; 10]   |
| $\beta$                        | Faktor bestimmt Dynamik der Anpassung von $f$ (vgl. EWMA-Algorithmus, s. Abschnitt A3)<br>$\beta$ wird während der Lernphase exponentiell verringert | Startwert: 0,1<br>Endwert: 0,0001                           |
| $\gamma$                       | Faktor bestimmt Dynamik der Anpassung von $b$ an $f$<br>$\gamma$ ist fest an $\beta$ gekoppelt: $\gamma = \beta / 10$                                | -   |
| $w_{ij,0}$                     | Initialwert für alle Komponenten der Gewichtsvektoren  | ca. 1/10 des Mittelwerts der zum Training verwendeten Daten |

## A2 Verteilungs-Approximation

Gesucht ist die nicht in Form einer Gleichung vorliegende Verteilungsfunktion oder Verteilungsdichtefunktion eines stochastischen Prozesses, z. B. zur Weiterverwendung in einem Quellmodell. Der stochastische Prozeß ist nur aus Messungen bekannt; es handelt sich also um ein Problem der Zeitreihenanalyse, s. Abschnitt 2.4. Als empirische Methode zur Bestimmung einer Näherung der tatsächlichen Verteilung kommen mehrere Verfahren in Frage, von denen einige, die auf Häufigkeitsmessungen beruhen, im folgenden erwähnt werden. Verfahren, die auf Parameterschätzungen analytischer Funktionen beruhen (z. B. [39]) werden an dieser Stelle nicht berücksichtigt, da bei ihnen bedeutende Vorkenntnisse über den untersuchten Prozeß notwendig sind.

### 1. Häufigkeitsmessung in vordefinierten Klassen

Wenn der Wertebereich der Zeitreihe bekannt ist, kann eine Aufteilung des Wertebereichs in  $L$  Klassen gleicher Breite vorgenommen werden, in denen jeweils die Häufigkeit der Ereignisse gemessen wird.

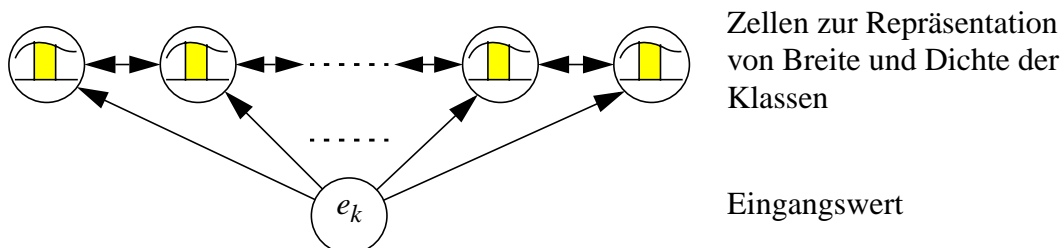
### 2. Häufigkeitsmessung in automatisch angepaßten Klassen

Der erste Ansatz kann erweitert werden, wenn die Grenzen für die Häufigkeitsmessung automatisch aus der Zeitreihe bestimmt werden. Es sind dann zwei Phasen zur Anpassung der Bereiche und Häufigkeiten erforderlich.

### 3. Häufigkeitsmessung in automatisch angepaßten Klassen, Anpassung der Auflösung

Eine weitere Erweiterung verzichtet auf feste Klassenbreiten, um eine bessere Anpassung an die Verteilung des stochastischen Prozesses zu ermöglichen. Dadurch wird der durch die endliche Klassenzahl  $L$  bedingte Anpassungsfehler verringert.

Ein Verfahren der dritten Kategorie wird im folgenden beschrieben. Es setzt für die automatische Anpassung einen Algorithmus ein, der aufgrund seiner Eigenschaften zu den Neuronalen Netzen gerechnet werden kann: Die Anpassung erfolgt automatisch durch einen einfachen verteilten Lernalgorithmus, jeder Klasse kann ein Rechenelement (Zelle, Neuron) zugeordnet werden und die Verbindungen unter den Rechenelementen sind auf Verbindungen zwischen Nachbarerelementen beschränkt (siehe Bild A.3). Dadurch ist der Algorithmus auch sehr gut für eine Parallelimplementierung geeignet.



**Bild A.3:** Verbindungsstruktur der Rechenelemente



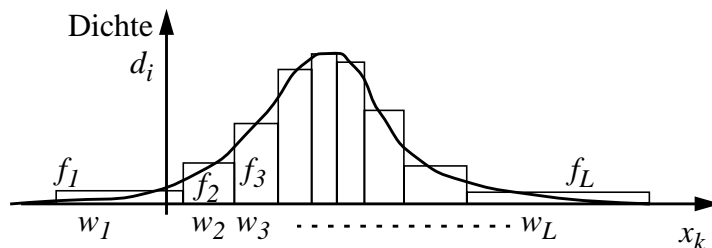
## A2.1 Lernalgorithmus

Im folgenden wird der Ablauf des Lernvorgangs beschrieben und die Formeln zur Anpassung der Klassen der Approximation angegeben. Ziel ist die Approximation der Verteilung der Zeitreihe  $\{e_k\}$ . Für  $\{e_k\}$  wird angenommen:  $e_k=0$  für  $k<0$ . Für jeden Eingangswert erfolgt ein Adaptionsschritt. Die Variablen des Algorithmus werden in Tabelle A.2 und Richtwerte für die Parameter in Tabelle A.3 angegeben.

**Tabelle A.2:** Variablen des Lernalgorithmus

| Variable  | Beschreibung                                   |
|-----------|--|
| $e_k$     | Wert des Ereignisses zum Zeitpunkt $k$         |
| $w_{i,k}$ | Breite der Klasse $i$ zum Zeitpunkt $k$        |
| $l_{i,k}$ | Linke Grenze der Klasse $i$ zum Zeitpunkt $k$  |
| $r_{i,k}$ | Rechte Grenze der Klasse $i$ zum Zeitpunkt $k$ |
| $d_{i,k}$ | Dichte in Klasse $i$ zum Zeitpunkt $k$         |
| $f_{i,k}$ | Häufigkeit in Klasse $i$ zum Zeitpunkt $k$     |

Zwischen Dichte, Breite und Häufigkeit einer Klasse besteht der Zusammenhang  $f_{i,k} = d_{i,k} \cdot w_{i,k}$ . Diese Größen sind in Bild A.4 anhand eines Beispiels dargestellt.



**Bild A.4:** Approximation einer Verteilungsdichtefunktion

Der Lernvorgang wird in zwei Phasen unterteilt. In der ersten Lernphase werden sowohl die Bereichsbreiten  $w_i$  als auch die Höhe  $d_i$  der Dichteapproximation der einzelnen Bereiche angepaßt. In der zweiten Lernphase erfolgt eine Feinanpassung der Dichte, die Bereichsbreiten werden nicht mehr verändert. Die erste Phase dauert  $K_1$  Schritte, die zweite Phase  $K_2$  Schritte.

In jedem Adaptionsschritt wird zuerst die Klasse  $i$  bestimmt, in die das Ereignis fällt. Anschließend werden die Breite der Klasse und ggf. die Häufigkeit angepaßt.

### A2.1.1 Adaption der Klassenbreiten

Das Prinzip der Anpassung der Klassenbreiten beruht auf einem „Häufigkeitsfluß“ zwischen den Klassen. Das Ziel dabei ist die gleiche Häufigkeit aller Klassen. Abhängig von der aktuellen Häufigkeit der Klasse, in die das Ereignis  $e_k$  fällt, wird ein Teil der Häufigkeit an deren Nachbarklassen abgegeben. Dabei sind die Grenzen zwischen Klassen nur „halbdurchlässig“;

d. h. ein Häufigkeitsfluß erfolgt nur in Richtung geringerer Häufigkeit. Dies führt im eingeschwungenen Zustand zu einem Gleichgewicht.

Bei den Randbereichen 1 und  $L$  werden zusätzlich die äußeren Grenzen an den gesamten Wertebereich von  $e_k$  angepaßt. Hier findet ebenfalls die Vorstellung eines „Häufigkeitsflusses“ Anwendung, nur ist die Grenze hier in beide Richtungen durchlässig. Ziel ist hier eine relativ geringe Häufigkeit außerhalb der äußeren Grenzen, um den Approximationsfehler klein zu halten.

Die Anpassung der Klassenbreite erfolgt durch Verteilung eines „Dichteanteils“  $\Delta f_i$  von Klasse  $i$  an die benachbarten Klassen, wie in Bild A.6 gezeigt. Die einzuhaltenden Randbedingungen sind: a) Auswirkung des Adaptionsschritts nur auf die Trefferklasse und ihre direkten Nachbarn und b) Summe der Häufigkeiten in diesen drei Klassen bleibt konstant. Das wird dadurch erreicht, daß die Dichte in Klasse  $i$  sowie die äußeren Grenzen der Nachbargebiete konstant gehalten werden. Die Anpassung erfolgt durch Breitenänderung der Klassen  $i$ ,  $i-1$ ,  $i+1$  und eine Dichteänderung der Klassen  $i-1$  und  $i+1$ . Die Breite von Klasse  $i$  wird reduziert, um die Auflösung in Bereichen höherer Ereignishäufigkeit zu erhöhen. Diese Prozedur führt im Laufe des Anpassungsvorgangs zu einer näherungsweise gleichen Ereignishäufigkeit in allen Klassen.

Für die Behandlung der Klasse  $i$  ergeben sich abhängig von ihrer Lage fünf Fälle:

1.  $2 < i < L - 1$ , „mittlere“ Klassen
2.  $i = 2$ , neben linkem Randbereich
3.  $i = L - 1$ , neben rechtem Randbereich
4.  $i = 1$ , linker Randbereich
5.  $i = L$ , rechter Randbereich

Wenn die Ereignisse außerhalb des Gesamtbereichs fallen, werden sie den Randbereichen zugeschlagen.

**Fall 1 ( $2 < i < L - 1$ ):**

In diesem Fall wird die Breite der Trefferklasse verringert und die Breiten der benachbarten Klassen vergrößert. Die Verringerung der Breite erfolgt indirekt durch eine Verringerung der Klassenhäufigkeit um  $\Delta f_i$  bei gleichzeitig konstanter Dichte. Die Berechnung von  $\Delta f_i$  erfolgt über den Lernparameter  $\lambda$ . Die benachbarten Klassen werden jeweils um die Hälfte der Häufigkeitsdifferenz  $\Delta f_i$  der Trefferklasse vergrößert. Hier muß die Dichte ebenfalls angepaßt werden, um die gesamte Breite der drei betrachteten Klassen nicht zu verändern.

Die Werte für die rechte Grenze  $r_{i,k}$  der Bereiche werden im folgenden nicht überall angegeben, da sie sich immer aus der linken Grenze und der Bereichsbreite ergeben.

Trefferklasse:

$$\begin{aligned}
 \Delta f_{i,k} &= f_{i,k-1} \cdot \lambda \\
 f_{i,k} &= f_{i,k-1} - \Delta f_{i,k} \\
 w_{i,k} &= \frac{f_{i,k}}{d_{i,k}} = \frac{f_{i,k-1} - \Delta f_{i,k}}{d_{i,k}} = w_{i,k-1} \cdot (1 - \lambda) \\
 l_{i,k} &= l_{i,k-1} + \frac{w_{i,k-1} - w_{i,k}}{2} = l_{i,k-1} + \frac{\lambda}{2} \cdot w_{i,k-1} \\
 d_{i,k} &= d_{i,k-1}
 \end{aligned} \tag{A.5}$$

Linker Nachbar:

$$\begin{aligned}
 f_{i-1,k} &= f_{i-1,k-1} + \frac{\Delta f_{i,k}}{2} \\
 w_{i-1,k} &= w_{i-1,k-1} + \frac{\Delta f_{i,k}}{2 \cdot d_{i,k}} \\
 l_{i-1,k} &= l_{i-1,k-1} \\
 d_{i-1,k} &= \frac{f_{i-1,k}}{w_{i-1,k}}
 \end{aligned} \tag{A.6}$$

Rechter Nachbar:

$$\begin{aligned}
 f_{i+1,k} &= f_{i+1,k-1} + \frac{\Delta f_{i,k}}{2} \\
 w_{i+1,k} &= w_{i+1,k-1} + \frac{\Delta f_{i,k}}{2 \cdot d_{i,k}} \\
 l_{i+1,k} &= l_{i+1,k-1} - \frac{\Delta f_{i,k}}{2 \cdot d_{i,k}} \\
 d_{i+1,k} &= \frac{f_{i+1,k}}{w_{i+1,k}}
 \end{aligned} \tag{A.7}$$

**Fall 2 ( $i = 2$ ):**

Außenbereich 1 ist linker Nachbar der Trefferklasse 2. Die Anpassung der Klassen 2 und 3 erfolgt nach (A.5) und (A.7).

$$\begin{aligned}
 f_{1,k} &= f_{1,k-1} + \frac{\Delta f_{2,k}}{2} \\
 w_{1,k} &= w_{1,k-1} + \frac{\Delta f_{2,k}}{2 \cdot d_{2,k}} \cdot \frac{\eta - 1}{\eta} \\
 l_{1,k} &= l_{1,k-1} + \frac{\Delta f_{2,k}}{2 \cdot d_{2,k}} \cdot \frac{1}{\eta}
 \end{aligned} \tag{A.8}$$

**Fall 3 ( $i = L - 1$ ):**

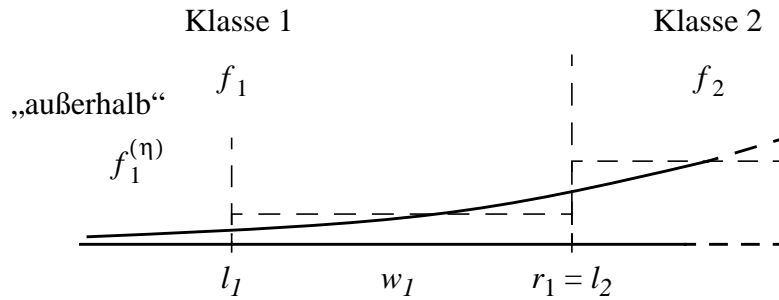
Analog zu Fall 2.

**Fall 4 ( $i = 1$ ):**

In diesem Fall ist die Klassenbreite anders definiert, da die Breite der äußeren Klassen bei der Anpassung an den Wertebereich einer unendlich ausgedehnten Verteilung wie beispielsweise der Normalverteilung nicht beliebig groß werden soll. Um dies zu erreichen, wird ein Fehlanpassungs-Parameter  $\eta$  eingeführt, der eine begrenzte Breite dieser Klassen erlaubt, indem zugelassen wird, daß nicht sämtliche Ereignisse in Klassen fallen. Ein bestimmter Anteil, der

durch den genannten Parameter gegeben ist, fällt in den nicht durch Klassen des Algorithmus abgedeckten „äußeren“ Bereich. In Bild A.5 wird dies verdeutlicht. Es ist die Klasse 1 mit Häufigkeit  $f_1$  dargestellt. Der Parameter  $\eta$  gibt das Verhältnis der Häufigkeit in Klasse 1 zur Häufigkeit links von  $l_1$  an. Die Adaption erfolgt so, daß sich für die Häufigkeit links der Grenze  $l_1$  folgender Wert ergibt:

$$f_1^{(\eta)} = \frac{1}{\eta} \cdot f_1 \quad (\text{A.9})$$



**Bild A.5:** Linke Randklasse

Durch die Fehlanpassung wird die Dichte in den Klassen 1 und  $L$  um den Faktor  $\eta/(\eta - 1)$  zu hoch geschätzt.

Adaption von Klasse 1:

$$\begin{aligned} \Delta f_{1,k} &= f_{1,k-1} \cdot \lambda \\ f_{1,k} &= f_{1,k-1} - \frac{\Delta f_{1,k}}{2} \\ r_{1,k} &= r_{1,k-1} - \frac{\Delta f_{1,k}}{2 \cdot d_{1,k-1}} \end{aligned} \quad (\text{A.10})$$

Die Anpassung der Breite von Klasse 1 muß die Verschiebung beider Grenzen dieser Klasse berücksichtigen und ist daher abhängig von der Position von  $e_k$  relativ zu  $l_1$ , um Gleichung (A.9) zu erfüllen.

Anpassung der Breite für  $e_k$  rechts von  $l_1$ :

$$w_{1,k} = w_{1,k-1} \cdot (1 - \lambda) \cdot (1 - \lambda) \quad (\text{A.11})$$

Anpassung der Breite für  $e_k$  links von  $l_1$ :

$$w_{1,k} = w_{1,k-1} \cdot (1 - \lambda) \cdot (1 + (\eta - 1) \cdot \lambda) \quad (\text{A.12})$$

Der jeweils erste Faktor in den Gleichungen (A.11) und (A.12) resultiert aus der Anpassung der rechten Bereichsgrenze, der zweite Faktor dient zur Anpassung der linken Bereichsgrenze und zur Einhaltung von Gleichung (A.9). Die beiden Schritte zur Anpassung beider Bereichsgrenzen werden als unabhängig angenommen.

Die Anpassung in Klasse 2 erfolgt nach Gleichung (A.7).

**Fall 5 ( $i = L$ ):**

Die Adaption der Klasse  $L$  erfolgt analog zur Klasse 1.

**A2.1.2 Adaption der Klassenhäufigkeiten**

Nach Abschluß der ersten Adaptionphase zur Anpassung der Klassen an den Wertebereich des zugrundeliegenden Prozesses erfolgt die weitere Anpassung der Klassenhäufigkeiten durch exponentiell gewichtete Mittelung (s. Anhang A3). Dies würde ohne Berücksichtigung der ersten Adaptionphase zu folgender Formel führen:

$$f_{i,k} = \alpha \cdot \sum_{j=0}^k (1 - \alpha)^j \cdot t_{i,k-j} \tag{A.13}$$

mit  $t_{i,k} = 1$ , falls das Ereignis  $e_k$  in Bereich  $i$  liegt und  $t_{i,k} = 0$  sonst.

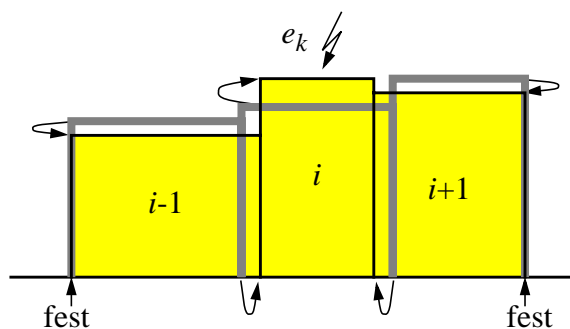
Da die  $t_{i,k}$  für negatives  $k$  Null sind, kann (A.13) als Rekursionsgleichung angegeben werden:

$$f_{i,k} = \alpha \cdot t_{i,k} + (1 - \alpha) \cdot f_{i,k-1} \tag{A.14}$$

Da die  $f_{i,k}$  nach Abschluß der ersten Adaptionphase bereits Werte ungleich Null haben, ist nur (A.14) anwendbar, (A.13) nicht.

Der Faktor  $\alpha$  bestimmt den Beitrag des aktuellen Ereignisses  $t_{i,k}$  zur Häufigkeit. Große Werte von  $\alpha$  betonen neue Ereignisse stärker, d. h. die Auswirkung vergangener Ereignisse schwächt sich schnell ab. Durch kleine Werte von  $\alpha$  behalten vergangene Werte ihren Einfluß auf  $f_{i,k}$  über viele Anpassungsschritte.

Bild A.6 zeigt die kombinierten Einflüsse der beiden Adaptionsschritte auf eine Klasse, in die ein Ereignis fällt.



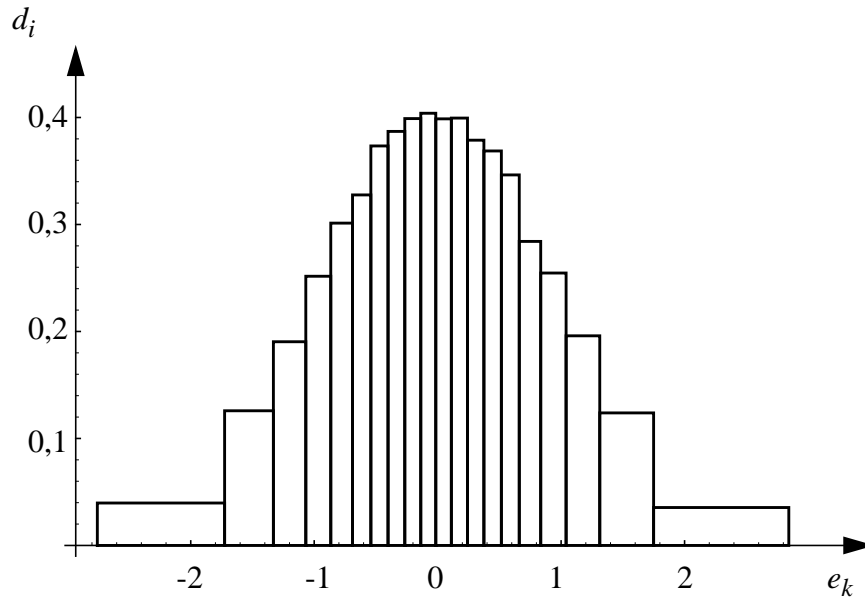
**Bild A.6:** Klassenanpassung

**A2.2 Beispiel**

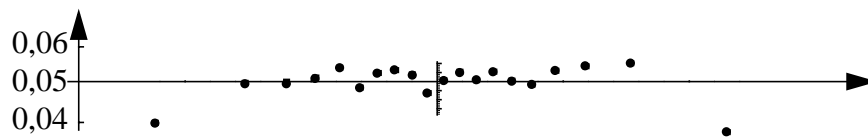
Im folgenden wird anhand eines Beispiels die Anpassungsfähigkeit des Algorithmus gezeigt.

Als Lerndatensatz wurden Werte einer normalverteilten Zeitreihe mit Mittelwert Null und Varianz Eins verwendet. In Tabelle A.3 sind die Parameter des Beispiels angegeben.

Bild A.7 zeigt im oberen Teil die Dichteapproximation. Man erkennt deutlich die abnehmende Breite der Klassen für zunehmende Dichte. Im unteren Teil ist die Abweichung der jeweiligen Klassenhäufigkeit vom Mittelwert  $1/L$  aufgetragen.



a) Approximation der Dichte



b) Abweichung der Klassenhäufigkeiten

**Bild A.7:** Approximierte Normalverteilung

## A2.3 Parameterwahl

Der in Abschnitt A2.1.1 erwähnte „Dichteanteil“ wird durch einen Lernparameter  $\lambda$  bestimmt, der während des Lernprozesses exponentiell verringert wird.

Der Wert für  $\alpha$  während der zweiten Lernphase  $K_2$  hängt von der Größe des Lerndatensatzes  $D$  ab. Da für die Adaption der Häufigkeit die exponentiell gewichtete Mittelung (EWMA, siehe Anhang A3) verwendet wird, ist anzustreben, daß Lerndaten, die um  $D$  in der Vergangenheit liegen, nicht zu gering gewichtet werden. Ansonsten besteht die Gefahr, daß die Verteilungs-Approximation immer nur an Teile der Lerndaten angepaßt wird.

Das Gewicht des um  $D$  zurückliegenden Werts soll  $x$  betragen. Dann resultiert aus (A.16) in Anhang A3 ein Endwert für  $\alpha$  von:

$$\alpha_{end} = 1 - \sqrt[D]{x} \tag{A.15}$$

In Tabelle A.3 sind die Parameter des Algorithmus sowie Richtwerte für die Parameter zusammengefaßt.

**Tabelle A.3:** Parameter des Lernalgorithmus

| Parameter | Bedeutung und Bemerkungen   | Richtwert  | Beispiel (s. A2.2)     |
|-----------|---|--|------------------------|
| $L$       | Anzahl Klassen für Approximation  | problemabhängig <sup>(a)</sup>   | 20                     |
| $\alpha$  | Parameter für Adaption der Häufigkeiten<br>$\alpha$ wird während $K_1$ Schritten exponentiell verringert, danach wird der letzte Wert konstant beibehalten.                       | Startwert 0,01<br>Endwert für $x=0,9$ :<br>$1 - \sqrt[D]{0,9}$ , s. (A.15) | 0,01<br>...<br>0,00001 |
| $\eta$    | Parameter für Fehlanpassung in Randbereichen<br><br>Zu große Werte führen zu instabilem Verhalten.  | 20   | 20                     |
| $K_1$     | Anzahl Adaptionsschritte mit Adaption der Klassenbreiten <i>und</i> Häufigkeiten<br><br>$\lambda$ und $\alpha$ werden exponentiell von ihrem Start- auf ihren Endwert verringert. | problemabhängig <sup>(a)</sup><br>( $10^3, \dots, 10^5$ )                  | 40.000                 |
| $K_2$     | Anzahl Adaptionsschritte nur mit Häufigkeitsadaption<br><br>Es gilt der Endwert von $\alpha$  | wie $K_1$  | 10.000                 |
| $D$       | Größe des Lerndatensatzes<br><br>Für $K_1 + K_2 > D$ wird der Lerndatensatz zyklisch wiederholt.  | problemabhängig <sup>(a)</sup>   | 10.000                 |
| $w_{i,0}$ | Initialwert für alle Klassenbreiten   | ca. $1/(L \cdot 10)$ des Erwartungswerts der Lerndaten                     | 0,005                  |
| $I_1$     | Initialwert für die linke Grenze aller Klassen  | 0  | 0                      |

(a) Die jeweiligen Werte lassen sich empirisch mit Hilfe der in Abschnitt 2.2.2 vorgestellten Anpassungstests ermitteln

## A2.4 Bewertung

Der vorgestellte Algorithmus zur adaptiven Verteilungs-Approximation zeichnet sich durch folgende Eigenschaften aus:

- Automatische Anpassung an die Dichte eines beliebigen stochastischen Prozesses
- Erhöhte Auflösung in Bereichen höherer Dichte
- Parallelisierbarkeit durch Neuronale-Netz-Struktur

Bei der Approximation treten folgende Approximationsfehler auf:

- Fehler aufgrund der endlichen Klassenzahl  
Es ist plausibel, daß dieser Fehler bei steigender Klassenzahl  $L$  gegen Null geht.
- Fehler in den Randbereichen  
Die Größe dieses Fehlers kann durch geeignete Wahl des Parameters  $\eta$  beschränkt werden (siehe Tabelle A.3)



### A3 Exponentiell gewichtete Mittelung

Die exponentiell gewichtete Mittelwertbildung (exponentially weighted moving average – AWMA) stellt eine Form der gleitenden Mittelung dar, in der vergangene Werte mit steigendem Alter immer weniger in den Mittelwert eingehen.

$$a_k = \alpha \cdot \sum_{j=0}^k (1 - \alpha)^j \cdot e_{k-j} \quad (\text{A.16})$$

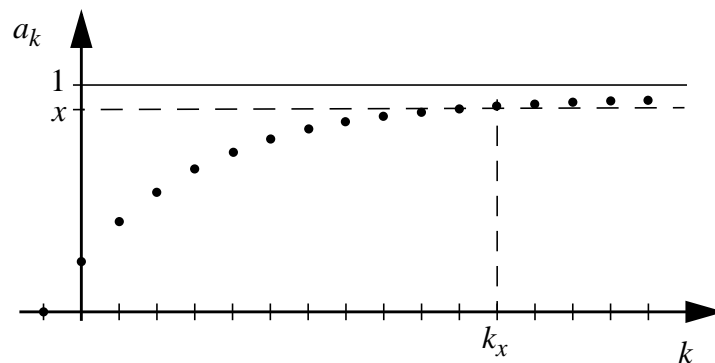
Der Abfall der Wertigkeit erfolgt exponentiell. AWMA lässt sich umformen und als eine sehr einfache Folgendefinition mit einem einzigen Parameter darstellen:

$$a_k = a_{k-1} \cdot (1 - \alpha) + e_k \cdot \alpha. \quad (\text{A.17})$$

Für einige Anwendungen ist insbesondere die Zeitdauer bis zum Erreichen eines vorgegebenen Werts interessant. Hierfür wird die Reaktion der Folge auf eine Sprungfunktion

$$s(t) = \begin{cases} 0 & \text{für } t < 0 \\ 1 & \text{für } t \geq 0 \end{cases} \quad (\text{A.18})$$

untersucht. Das Resultat ist die Folge  $a_k = 1 - (1 - \alpha)^{k+1}$ , die in Abbildung A.8 dargestellt ist.



**Bild A.8:** Sprungantwort bei AWMA

Im folgenden werden für zwei Auswertungen der entstehenden Folge die Formeln und einige beispielhafte Werte angegeben.

### 1. Zeitdauer bis zum Erreichen eines Wertes $x$

Die folgende Tabelle zeigt die zum Überschreiten des Werts  $x$  erforderliche Schrittzahl.

| Formel  | Beispiel ( $x=0,9$ ) |       |
|---|----------------------|-------|
|   | $\alpha$             | $k_x$ |
| $k_x = \left\lceil \frac{\ln(1-x)}{\ln(1-\alpha)} - 1 \right\rceil \quad (\text{A.19})$ | 0,01                 | 229   |
|   | 0,1                  | 21    |
|   | 0,2                  | 10    |
|   | 0,5                  | 3     |

### 2. Parameter $\alpha$

Über die Vorgabe der Schrittzahl bis zum Erreichen des Wertes  $x$  kann der Parameter  $\alpha$  bestimmt werden.

| Formel   | Beispiel ( $x=0,9$ ) |          |
|--|----------------------|----------|
|  | $k$                  | $\alpha$ |
| $\alpha = 1 - \sqrt[k+1]{1-x} \quad (\text{A.20})$ | 5                    | 0,319    |
|  | 30                   | 0,0716   |

