

FORTSCHRITT-
BERICHTE **VDI**

Dipl.-Ing. Werner Henkel, Darmstadt

**Zur Decodierung
algebraischer Blockcodes
über komplexen Alphabeten**

Reihe **10**: Informatik/
Kommunikationstechnik Nr. **109**

VDI VERLAG

Verlag des Vereins Deutscher Ingenieure · Düsseldorf



Henkel, Werner

**Zur Decodierung algebraischer Blockcodes
über komplexen Alphabeten**

Fortschr.-Ber. VDI Reihe 10 Nr. 109. Düsseldorf: VDI-Verlag 1989.
134 Seiten, 29 Bilder.

Für die Dokumentation: RS/BCH-Codes über komplexen Zahlen — Analoge Codes — DFT-Codes — Syndromerzeugung mit der Newton-Interpolation — Berlekamp-Massey-Algorithmus — Levinson-Durbin-Algorithmus — Toeplitz-Matrix/-System — Hankel-Matrix/-System — Lineare Prädiktion — Interpolation/Approximation-Codierung

Die vorliegende Arbeit untersucht Eigenschaften der von Marshall 1981 bzw. Wolf 1983 eingeführten „Analogen Codes“ — RS- bzw. BCH-Codes über komplexen Zahlen — und stellt Verbindungen einerseits zu Interpolation/Approximation und andererseits zu Verfahren der digitalen Signalverarbeitung her. Die neue Codeklasse erlaubt insbesondere die Korrektur von Impulsfehlern unter Tolerierung eines zusätzlich vorhandenen „Hintergrundrauschens“. Konditionsbetrachtungen bilden daher einen Hauptteil dieser Arbeit. Rechenungenauigkeitseffekte werden ebenfalls betrachtet. Wichtige Einzelergebnisse seien kurz in Stichworten zusammengestellt: Syndromerzeugung mittels Newton-Interpolation, neue, didaktisch vorteilhafte Beschreibung des Berlekamp-Massey-Algorithmus, „triangular square-root factorization“ aus dem BMA, Vorgehen zur Bestimmung von Konditionszahlen rekursiv im BMA auftretender Sub-Toeplitz-Matrizen, Erweiterung des Levinson-Durbin-Algorithmus zur Inversion von Toeplitz-Matrizen, Schätzung der Kanalcharakteristik aus dem Decodierungsergebnis.

Die Reihen der FORTSCHRITT-BERICHTE VDI:

- | | |
|--|--|
| 1 Konstruktionstechnik/Maschinenelemente | 12 Verkehrstechnik/Fahrzeugtechnik |
| 2 Fertigungstechnik | 13 Fördertechnik |
| 3 Verfahrenstechnik | 14 Landtechnik/Lebensmitteltechnik |
| 4 Bauingenieurwesen | 15 Umwelttechnik |
| 5 Grund- und Werkstoffe | 16 Technik und Wirtschaft |
| 6 Energieerzeugung | 17 Biotechnik |
| 7 Strömungstechnik | 18 Mechanik/Bruchmechanik |
| 8 Meß-, Steuerungs- und Regelungstechnik | 19 Wärmetechnik/Kältetechnik |
| 9 Elektronik | 20 Rechnerunterstützte Verfahren (CAD, CAM, CAE, CAP, CAQ, CIM,...) |
| 10 Informatik/Kommunikationstechnik | 21 Elektrotechnik |
| 11 Schwingungstechnik | |

D 17

© VDI-Verlag GmbH · Düsseldorf 1989

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe (Photokopie, Mikrokopie), der Speicherung in Datenverarbeitungsanlagen und das der Übersetzung, vorbehalten.

Als Manuskript gedruckt. Printed in Germany.

ISSN 0178-9627

ISBN 3-18-140910-3

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Netzwerk- und Signaltheorie, Fachgebiet Grundlagen der Elektrotechnik, an der Technischen Hochschule Darmstadt.

Herrn Prof. Dr.-Ing. B. Dorsch danke ich für die Übernahme des Referates und besonders für die Freiheit, die gewählte Thematik verfolgen zu können, für begleitende Gespräche und wichtige Anregungen.

Ebenfalls danke ich Herrn Prof. Tzschach, der sich trotz seiner vielseitigen Verpflichtungen sofort bereit fand, das Korreferat zu übernehmen.

Erwähnt sei auch die konstruktive Arbeitsatmosphäre im Institut, für welche allen Kollegen an dieser Stelle gedankt sei. Mein besonderer Dank gilt den Kollegen Dipl.-Ing. J. Cezanne, H. Schneider und Dr.-Ing. E. Schulz für die Zusammenarbeit in theoretischen Fragen sowie den Kollegen Dipl.-Ing. U. Kreßel, B. Liesenfeld und H. Schneider für die kritische Durchsicht des Manuskripts.

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 1 |
| 2 | Analoge RS- und BCH-Codes | 5 |
| 2.1 | Definition von RS- und BCH-Codes | 5 |
| 2.2 | Komplexe RS- und BCH-Codes | 9 |
| 2.3 | Zu erwartende Eigenschaften | 12 |
| 2.4 | Kanalkapazitäten analoger Kanäle | 14 |
| 3 | Interp., Approx. und Decodierung | 23 |
| 3.1 | Decodierung von RS-Codes | 23 |
| 3.2 | Pronys exponentielle Interpolation | 27 |
| 3.3 | Lagrange-Interpolation | 30 |
| 3.4 | Least-Squares-Approximation | 32 |
| 3.5 | Euklidischer Alg. und Approx. mit Kettenbrüchen | 36 |
| 4 | Korrekturfähigkeit analoger Codes | 40 |
| 4.1 | Beweis nach J.K. Wolf | 40 |
| 4.2 | Das Newton-Verfahren | 43 |
| 5 | Konditionierung des Decodierproblems | 48 |
| 5.1 | Simulationsergebnisse | 48 |
| 5.1.1 | Gleichverteiltes Hintergrundrauschen | 49 |

| | | |
|--|--|------------|
| 5.1.2 | Normalverteiltes Hintergrundrauschen | 52 |
| 5.2 | Konditionsbetrachtungen | 54 |
| 5.3 | Neue Beschreibung des BMA | 61 |
| 5.3.1 | Das Schieberegisterproblem nach Massey | 61 |
| 5.3.2 | Neue Matrixbeschreibung | 64 |
| 5.4 | Abschätzung der Konditionierung | 75 |
| 6 | Einfluß der Rechenungenauigkeit | 79 |
| 7 | Zwei weitere Möglichkeiten . . . | 84 |
| 7.1 | Erweiterung des LDA | 84 |
| 7.2 | Toeplitz-Systeme und LMS-Algorithmus | 94 |
| 8 | TP-Filterung versus Decodierung | 96 |
| 8.1 | Simulationsergebnisse | 97 |
| 8.2 | Ein Auswahlkriterium | 100 |
| 9 | Zusammenfassung | 101 |
| Anhang | | 105 |
| A.1 | zu Kapitel 2.3 | 105 |
| A.2 | zu Kapitel 2.4 | 107 |
| A.3 | zu Kapitel 5.1.2 | 112 |
| A.4 | zu Kapitel 5.3 | 116 |
| Verzeichnis der verwendeten Abkürzungen und Formelzeichen | | 119 |
| Literaturverzeichnis | | 123 |

Kapitel 1

Einleitung

“Ich sah, daß die Mathematik in viele Spezialgebiete gespalten war, deren jedes diese kurze uns vergönnte Lebenszeit wegnehmen konnte. So sah ich mich in der Lage von Buridans Esel, der sich nicht für ein besonderes Bündel Heu entschließen konnte.”

— Albert Einstein

Betrachtet man die Entwicklung der Codierungstheorie, so ließe sich obiges Zitat in der folgenden Weise fortsetzen: Ist die Entscheidung zugunsten eines ‘Bündels Heu’ gefallen, so geraten alle anderen in Vergessenheit.

Wegen fehlender technischer Voraussetzungen war die Codierungstheorie lange Zeit eine rein mathematische Disziplin. Als solche blieb sie jedoch auch recht unabhängig von lange etablierten Wissensgebieten, wie der Approximationstheorie und der Numerik. Dies führte dazu, daß Algorithmen zum Teil aufs neue entwickelt wurden, obwohl sie in einer anderen mathematischen Disziplin seit langer Zeit bekannt waren. Bestes Beispiel hierfür ist die Identität zwischen Gorenstein-Zierler-Algorithmus zur Decodierung von Reed-Solomon-Codes und Pronys exponentieller Approximation.

Bis zum heutigen Tage ist nach wie vor eine gewisse Distanzierung zwischen eigentlich sehr verwandten Arbeitsgebieten vorhanden. Als Beispiel für die wenig beachteten Zusammenhänge zwischen Codierung und Approximationstheorie sei die Codierung und Decodierung von RS- und BCH-Codes (**B**ose, **C**haudhuri, **H**ocquenghem) genannt, die ebenso als Interpolations- bzw. Approximationsproblem aufgefaßt werden kann.

Eine ähnliche Separierung ist ebenfalls mit Eindringen der Codierungstheorie in die Ingenieurwissenschaften zu beobachten. So etablierten sich beispielsweise unterschiedliche Algorithmen zur Lösung strukturgleicher Gleichungssysteme in der digitalen Signalverarbeitung und der Codierung.

Das Aufzeigen von Gemeinsamkeiten und Verbindungen zwischen verschiedenen Disziplinen sowohl der Mathematik als auch der Ingenieurwissenschaften bildet einen wichtigen Teilaspekt der vorliegenden Arbeit.

Der Zusammenhang wird hergestellt durch die Definition von Codes über den komplexen Zahlen – im Gegensatz zur Definition über endlichen Zahlkörpern, sogenannten Galois-Feldern. Die Betrachtungen werden hierbei beschränkt auf die wichtigsten Klassen von Blockcodes, die RS- und BCH-Codes. Über den komplexen Zahlen werden diese dann im folgenden als ‘*analoge Codes*’ bezeichnet. Es sind dies dann Codes über einem unendlichen Alphabet, deren Struktur, wie bei herkömmlichen RS-Codes, nur durch die Beschränkung auf einen Teil aller verfügbarer Dimensionen bestimmt wird.

Analoge Codes werden hier keineswegs erstmals eingeführt, jedoch untersucht diese Arbeit erstmals genauer ihre Eigenschaften und stellt Verbindungen zu anderen Arbeitsgebieten her. Die Geschichte der Entwicklung dieser Codes sei im folgenden kurz skizziert.

Erste Überlegungen zur Codierung mit *reellen Signalen* finden sich bereits in einer Veröffentlichung von Tröndle 1978 [57]. Die ersten Untersuchungen zu komplexen Blockcodes sind jedoch wohl auf 1981 (Marshall [40]) zu datieren. 1983 folgten dann zwei Veröffentlichungen von Wolf ([68] und [66]), in welchen erstmals der Begriff ‘*Analog Codes*’ eingeführt wurde. Wolf beschränkte sich jedoch auf die Betrachtung der Einzelfehlerkorrektur. Maekawa und Sakaniwa verwendeten später die Bezeichnung ‘*DFT Codes*’ (1985 [38]). Detailliertere Untersuchungen zur Mehrfehlerkorrektur fehlten jedoch bislang und werden erstmals in dieser Arbeit durchgeführt.

Der Vollständigkeit halber sei noch erwähnt, daß auch noch völlig andere Ansätze zur ‘analogen’ Korrektur impulsförmiger Störungen existieren, wie z.B. die Anwendung von LPC-Modellen (**L**inear **P**redictive **C**oding) für den Einsatz bei Sprachsignalen [60], welche die Quell-Eigenschaften mit einbeziehen und so eine gewisse Verbindung zwischen Quellencodierung und Kanalcodierung herstellen. Diese Verfahren bleiben jedoch auf spezielle Anwendungen beschränkt.

Analoge Codes unterscheiden sich durch eine Reihe interessanter Eigenschaften von herkömmlichen RS-Codes. Vorwegnehmend seien insbesondere zwei hervorgehoben. *Analoge Codes* besitzen eine größere Korrekturfähigkeit als Codes über endlichen Körpern (wenn man realistische Galoisfeldgrößen zugrunde legt), und sie ermögli-

chen es, Fehler unterschiedlicher Amplitude zu unterscheiden, d.h. Fehler großer Amplitude zu korrigieren und dabei diejenigen kleiner Amplitude unberücksichtigt zu lassen. Es besteht somit eine gewisse Toleranz gegenüber ‘Hintergrundrauschen’.

Besonders wichtig ist auch, daß mit eventuell geringfügigen Änderungen alle herkömmlichen Decodieralgorithmen anwendbar sind, und außerdem weitere Algorithmen aus anderen Disziplinen hinzukommen.

In die Untersuchung komplexer Blockcodes eingebettet sind Einzelergebnisse, die, für sich genommen, einerseits auch nützlich für die herkömmliche Codierung über endlichen Zahlkörpern, andererseits ebenfalls für andere Arbeitsgebiete von besonderer Bedeutung sind. Diese seien kurz herausgestellt:

- Als ein zentraler Punkt der Arbeit wird zur Untersuchung des Korrekturverhaltens der herkömmlichen Algorithmen eine neue Beschreibung des **Berlekamp-Massey-Algorithmus** (BMA) – eines der wichtigsten Decodierverfahren – entwickelt, die im Vergleich zu Masseys Darstellung bedeutend leichter verständlich ist und es erlaubt, wichtige Gesetzmäßigkeiten zu erkennen. So führte diese Beschreibung unter anderem zu der Erkenntnis, daß der BMA eine rekursive ‘triangular square-root factorization’ darstellt, d.h. eine Produktzerlegung der Koeffizientenmatrix in Dreiecks- und Diagonalmatrizen. Diese Zerlegung wiederum ermöglicht rekursive Konditionierungsabschätzungen.
- Ein weiteres wichtiges Teilergebnis wird gebildet durch eine Erweiterung des, aus der digitalen Signalverarbeitung bekannten, **Levinson-Durbin-Algorithmus**. Hierdurch wird die Inversion von Toeplitz-Matrizen, nicht nur die Lösung von Toeplitz-Systemen mit bestimmter rechter Seite, ermöglicht.
- Weiterhin folgte aus einer Betrachtung des Newton-Verfahrens zur Polynominterpolation eine neue Definition eines Syndroms¹ zur Decodierung von RS-Codes. Dieses weist sehr vielversprechende Eigenschaften auf, wie z.B. die Erkennbarkeit fehlerfreier Bereiche eines Codewortes, ohne die Anwendung irgendwelcher Decodieralgorithmen.

¹Ein ‘Syndrom’ bezeichnet Terme, die nur vom Fehlermuster, jedoch nicht von der übermittelten Information abhängen.

Die vorliegende Arbeit ist auch etwas im Zusammenhang mit Bestrebungen zu sehen, Modulation und Kanalcodierung als eine Einheit zu betrachten. Im Bereich der Faltungscodes ist dies durch die Trellis-codierte Modulation nach Ungerböck [58] gelungen. Erst in letzter Zeit entstehen erste allgemeine Ansätze zur Kombination von Modulation und Blockcodierung ([69], [9]). Ein komplexes Alphabet zur Codierung vermeidet schon per Prinzip eine Unterscheidung in Modulations- und Codewortsymbole.

Es wird nun im folgenden noch ein Kurzüberblick über die vorliegende Arbeit gegeben.

Nach einigen grundlegenden Definitionen und der Erläuterung der zu erwartenden Eigenschaften *analoger Codes* wird zunächst auf einige Zusammenhänge zwischen Interpolation bzw. Approximation und der Decodierung (analoger) Blockcodes hingewiesen. Es folgen dann Beweise zur höheren Korrekturfähigkeit, die auch gewisse konstruktive Ansätze zur Decodierung liefern. Es schließt sich eine detaillierte Untersuchung des Einflusses von ‘Hintergrundrauschen’ in der Form nicht zu korrigierender Fehler ‘kleiner’ Amplitude an. Insbesondere wird hier die neue Beschreibung des BMA vorgestellt. Anschließend wird beispielhaft der Einfluß von Rechenungenauigkeiten im BMA ermittelt, d.h. es wird die Frage erörtert, wie stabil das Verfahren bei Verringerung der Rechengenauigkeit arbeitet. Nachdem der BMA als geeignetes Verfahren auch für komplexe Toeplitz-Systeme ausgewiesen ist, werden exemplarisch zwei weitere interessante Gemeinsamkeiten verschiedener Arbeitsgebiete aufgezeigt. Zum einen wird hier die schon angesprochene Erweiterung des Levinson-Durbin-Algorithmus vorgestellt, zum anderen werden Zusammenhänge zum Anwendungsbereich des sogenannten LMS-Algorithmus (**L**east **M**ean **S**quare) angedeutet. Dieser bietet hier eine iterative Lösungsmöglichkeit für Toeplitz-Systeme. Abschließend werden Simulationsergebnisse angegeben, die einen Vergleich der Qualität zwischen einfacher Tiefpaßfilterung und analoger Fehlerkorrektur unter Variation der Fehlercharakteristik von reinem Rauschen bis zu reinen Impulsstörungen zulassen. Insbesondere wird dabei ein einfaches Verfahren gefunden, wie man nach einem Decodierungsversuch Aussagen über die Fehlerart bzw. die Störcharakteristik des Kanals treffen und so die geeignete Vorgehensweise bei der endgültigen Korrektur wählen kann.

Kapitel 2

Analoge RS- und BCH-Codes und ihre Eigenschaften

Zur Einleitung in die Thematik dieser Arbeit werden in diesem Kapitel grundlegende Definitionen eingeführt und zu erwartende Eigenschaften analoger Codierung skizziert, die in den Folgekapiteln näher untersucht werden. Im ersten Abschnitt werden zunächst RS- und BCH-Codes (**R**eed, **S**olomon; **B**ose, **C**haudhuri, **H**ocquenghem) über endlichen Zahlkörpern beschrieben.

2.1 Definition von RS- und BCH-Codes

Blockcodes über endlichen Zahlkörpern kennzeichnen Teilmengen aller N -Tupel bzw. Vektoren aus N Elementen, die gewissen Restriktionen genügen müssen. Die Elemente der Vektoren sind dabei aus $GF(q^s)$, wobei q eine Primzahl oder auch wiederum Potenz einer Primzahl ($q = p^m$) ist.

Als Hamming-Distanz $d_h(\vec{a}, \vec{b})$ bezeichnet man die Anzahl unterschiedlicher Komponenten zweier Vektoren \vec{a} und \vec{b}

$$d_h(\vec{a}, \vec{b}) = wt(\vec{a} - \vec{b}),$$

wobei $wt(\vec{a})$ das Gewicht kennzeichnet, d.h. die Anzahl der Komponenten des Vektors \vec{a} ungleich Null.

Die geringste Hamming-Distanz zweier Codewörter aus dem Code \mathcal{C} wird durch die Minstdistanz d_h gekennzeichnet:

$$d_h = \min_{\substack{\forall \vec{a}, \vec{b} \in \mathcal{C} \\ \vec{a} \neq \vec{b}}} d_h(\vec{a}, \vec{b}).$$

Die im folgenden definierten RS-Codes gehören zu den zyklischen Codes.

Definition 2.1 *Ein Code \mathcal{C} wird dann als zyklisch bezeichnet, wenn aus $(c_0, c_1, \dots, c_{N-1}) \in \mathcal{C}$ folgt, daß auch $(c_1, c_2, \dots, c_{N-1}, c_0) \in \mathcal{C}$ ist.*

RS-Codes sind Codes der (Maximal-) Länge $N = q^s - 1$, wobei die Komponenten der Codewörter aus $GF(q^s)$ sind. Sie besitzen die wichtige Eigenschaft, daß sie ‘*maximum distance separable*’ (*MDS*) sind, womit man die Tatsache bezeichnet, daß sie die sogenannte Singleton-Schranke mit Gleichheit erfüllen.

Satz 2.1 (Singleton-Schranke) *Gegeben sei ein Code \mathcal{C} der Länge N , der Dimension K (Anzahl der Informationszeichen) und der Hamming-Distanz d_h . Es gilt dann $d_h \leq N - K + 1$.*

RS-Codes lassen sich als Abtastwerte eines auf den Grad $K - 1$ beschränkten Polynoms über $GF(q^s)$ verstehen. Zunächst sollen jedoch einige grundlegende Bemerkungen vorausgeschickt werden.

Jede zyklische multiplikative Gruppe enthält ein Element einer multiplikativen *Ordnung* N , wobei diese den Modul des Exponenten kennzeichnet und der Anzahl der Elemente der multiplikativen Gruppe entspricht. Umfaßt die multiplikative Gruppe alle betrachteten Körperelemente des $GF(q^s)$ außer der Null, so bezeichnet man das Element als *primitiv*.

Definition 2.2 *Ein primitives Element z eines Körpers ist dadurch gekennzeichnet, daß jedes Körperelement außer der Null als Potenz dieses Elementes darstellbar ist.*

Nach diesen Vorbemerkungen ist es nun möglich, die Definition von RS-Codes über ein gradbeschränktes Polynom anzugeben.

Definition 2.3 Ein RS-Codes der Länge N , der Dimension K und der Mindestdistanz $d_h = N - K + 1$ besteht aus der Menge aller Vektoren $(c_0, c_1, \dots, c_{N-1})$, die der folgenden Bedingung genügen:

$$\mathcal{C} = \left\{ \vec{c} \mid c_i = \frac{1}{N} C(z^{-i}), \text{ Grad } C(x) \leq K - 1 \right\},$$

wobei z ein Element der multiplikativen Ordnung N (meist ein primitives Element) des zugrundeliegenden Körpers darstellt.

Betrachtet man die Komponenten c_i wiederum als Koeffizienten eines Polynoms $c(x)$, so erhält man umgekehrt die Koeffizienten des Polynoms $C(x)$ aus

$$C_j = c(z^j), \quad j = 0, 1, \dots, N - 1. \quad (2.1)$$

Die Definition 2.3 zusammen mit Gleichung 2.1 stellt nichts anderes als die **diskrete Fourier-Transformation (DFT)** dar:¹

$$\boxed{\begin{aligned} c_i &= \frac{1}{N} C(z^{-i}) = \frac{1}{N} \sum_{k=0}^{N-1} C_k \cdot (z^{-i})^k \\ C_k &= c(z^k) = \sum_{i=0}^{N-1} c_i \cdot (z^k)^i \end{aligned}} \quad (2.2)$$

Eine andere Möglichkeit zur Definition ergibt sich unter Verwendung des Generatorpolynoms $g(x) = \prod_{i=K}^{N-1} (x - z^i) = \prod_{i=1}^{N-K} (x - z^{-i})$ in der Form

$$c(x) = i(x) \cdot g(x) \quad (2.3)$$

mit einem frei wählbaren ‘Informationspolynom’ $i(x)$ ($\text{Grad } i(x) \leq K - 1$).

Entsprechend definiert man ebenfalls ein Prüfpolynom $h(x) = \prod_{i=0}^{K-1} (x - z^i)$, mit welchem sich dann ergibt

$$c(x) \cdot h(x) = i(x) \cdot \underbrace{g(x) \cdot h(x)}_{x^N - 1} = 0 \pmod{x^N - 1}.$$

Eine etwas verallgemeinerte Definition von RS-Codes mit den gleichen Eigenschaften bezüglich Fehlerkorrekturvermögen lautet:

$$\mathcal{C} = \left\{ \vec{c} \mid c_i = \frac{1}{N} C(z^{-i}) \cdot (z^{-i})^m, \text{ Grad } C(x) \leq K - 1, m \in [0, N - 1] \right\}, \quad (2.4)$$

¹Die DFT läßt sich natürlich auch mit vertauschten Vorzeichen im Exponenten des primitiven Elementes oder mit $\frac{1}{N}$ bei der Transformation in den ‘Frequenz’-Bereich definieren.

was einer Verschiebung im ‘Frequenzbereich’ (besser ‘DFT-Bereich’) entspricht.

Eine weitergehende Verallgemeinerung stellen die **generalized RS-Codes** – wie der Name schon sagt – dar.

Definition 2.4 *Generalized RS-Codes \mathcal{C}_{GRS} bestehen aus Vektoren der Form*

$$(v_0 F(\alpha_0), v_1 F(\alpha_1), v_2 F(\alpha_2), \dots, v_{N-1} F(\alpha_{N-1})) , \quad \text{Grad } F(x) \leq K - 1 ,$$

wobei alle α_i paarweise verschiedene Elemente aus $GF(q^s)$ und alle v_i beliebige Elemente aus $\{GF(q^s) \setminus 0\}$ darstellen.

Generalized RS-Codes besitzen ebenfalls die ‘MDS’-Eigenschaft.

RS-Codes erlauben mit den üblichen algebraischen Verfahren eine Fehlerzahl von $e \leq E = \lfloor (d_h - 1)/2 \rfloor$ zu korrigieren (als ‘Bounded Minimum Distance Decoding’ bezeichnet), wobei ja gilt $d_h = N - K + 1 = M + 1$ (siehe Singleton-Schranke). M kennzeichnet dabei im folgenden immer die Anzahl der Prüf- oder Parity-Zeichen.

Bei **BCH-Codes** wird für das Codewort im Zeitbereich die zusätzliche Bedingung gefordert, daß dessen Komponenten aus einem Unterkörper des $GF(q^s)$ stammen.

Definition 2.5 *Ein Codewort $(c_0, c_1, \dots, c_{N-1})$ eines ‘primitiven’ BCH-Codes der Länge N und der Dimension K folgt aus*

$$\mathcal{C} = \left\{ \vec{c} \mid c_i = \frac{1}{N} C(z^{-i}), \text{Grad } C(x) \leq K - 1 , \right.$$

$$\left. C_i \in \mathbb{B}_1 = GF(q^s), c_i \in \mathbb{B}_2, \mathbb{B}_2 \subset \mathbb{B}_1 \right\} ,$$

wobei z ein primitives Element, d.h. ein Element der Ordnung $N = q^s - 1$, des zugrundeliegenden Körpers $\mathbb{B}_1 = GF(q^s)$ darstellt.

Anders ausgedrückt, bedeutet dies: $C_i \in GF(q^s), c_i \in GF(q^m), GF(q^m) \subset GF(q^s)$, d.h. $m|s$. d_h stellt jetzt lediglich eine ‘designed distance’ dar. Die wahre Mindestdistanz ist häufig größer. Die Eigenschaft ‘MDS’ entfällt.

Die Länge ‘nichtprimitiver’ BCH-Codes ist ungleich $q^s - 1$. Sie wird als Teiler von $q^s - 1$ gewählt. Zur Definition benötigt man dann auch ein Element der multiplikativen Ordnung $N|(q^s - 1)$.

BCH-Codes im engeren Sinne, sind Codes mit $c_i \in GF(q = p)$ (p : Primzahl), d.h. Elemente des zugrundeliegenden Primkörpers mit den Elementen $0, \dots, p - 1$.

Aus den Restriktionen im ‘Zeitbereich’ folgen im ‘Frequenzbereich’ Konjugiertheitsbedingungen in der Form

$$c_i \in GF(q = p^m) \implies C_{k,q} = C_k^q, \quad k = 0, \dots, N-1. \quad (2.5)$$

Weitere Angaben zu Codekonstruktionen und deren Eigenschaften finden sich in den einschlägigen Standardwerken, wie z.B. [65], [46] und [6].

2.2 Komplexe RS- und BCH-Codes

Wie schon in der Einleitung bemerkt, wurde der Begriff ‘analoge Codes’ von J.K. Wolf 1983 ([68] u. [66]) eingeführt und steht für RS- oder BCH-Codes über den komplexen Zahlen (später auch einfach als DFT-Codes bezeichnet [38]). In dieser Arbeit wird dieser Begriff beibehalten.

Die Möglichkeit, RS-Codes auch über komplexen Zahlen zu definieren, gründet in der Tatsache, daß lediglich die Existenz eines primitiven Elementes gefordert werden muß, d.h. eines Elementes der Ordnung N . Im Falle der komplexen Zahlen ist dies die Zahl

$$z = e^{j2\pi/N}.$$

Aus dieser Analogie heraus, bezeichnet man Galois-Felder ja auch als Kreisteilungskörper.

Die Definition **analoger RS-Codes** erfolgt nun entsprechend derjenigen für endliche Zahlkörper (Def. 2.2), die einer DFT entspricht, wobei $d_h - 1 = N - K$ aufeinanderfolgende Komponenten im ‘Frequenzbereich’ verschwinden.

Definition 2.6 *Analoge RS-Codes sind Vektoren der Form*

$$(c_0, c_1, \dots, c_{N-1})$$

mit $c_i, C_i \in \mathbb{C}$, wobei

$$\begin{aligned} c_i &= \frac{1}{N} C(z^{-i}) = \frac{1}{N} \sum_{k=0}^{K-1} C_k \cdot (z^{-i})^k \\ C_k &= c(z^k) = \sum_{i=0}^{N-1} c_i \cdot (z^k)^i \\ \text{mit } z &= e^{j2\pi/N}. \end{aligned}$$

Die oben gemachten Verallgemeinerungen in Form einer Verschiebung im Frequenzbereich oder gemäß Definition 2.4, sind entsprechend auch hier möglich.

Wichtig ist noch zu bemerken, daß in der Definition keine Restriktionen bezüglich des Wertebereichs der komplexen ‘Zeit’- und ‘Frequenzbereichskomponenten’ eingeführt sind, d.h. insbesondere, daß keine Diskretisierung in irgendeiner Form angenommen wird. An einigen Stellen wird eine solche untersucht, wie z.B. im noch folgenden Abschnitt über Kanalkapazitäten; im allgemeinen jedoch, wird hiervon abgesehen. Analoge RS-Codes sind also Codes über einem unendlichen Alphabet, deren Struktur einzig durch das Nichtverwenden von $N - K$ Dimensionen gebildet wird.

Analoge BCH-Codes lassen sich ebenfalls entsprechend definieren.

Definition 2.7 *Analogue BCH-Codes sind analoge RS-Codes der Form*

$$(c_0, c_1, \dots, c_{N-1})$$

mit der zusätzlichen Einschränkung $c_i \in \mathbb{R}$.

Satz 2.2 *Die Komponenten C_i der ‘Frequenzbereichsvektoren’ analoger BCH-Codes weisen folgende Eigenschaft auf:*

$$C_{N-k} = C_{-k} = C_k^*$$

$$C_0 \in \mathbb{R}$$

$$C_{N/2} \in \mathbb{R} \text{ für } N \text{ gerade.}$$

Beweis: Der Satz ergibt sich direkt aus der Tatsache, daß

$$c_i = \frac{1}{N} \sum_{k=0}^{N-1} C_k e^{-j2\pi i k / N} \in \mathbb{R}.$$

Im komplexen Fall existieren somit zu Gleichung 2.5 entsprechende Konjugiertheitsbedingungen.

Die folgende Abbildung soll den Sachverhalt am Beispiel $N=8, M=2$ veranschaulichen:

| | | | | | | | |
|------------------|---|---|-------|------------------|---------|---|---|
| $\in \mathbb{R}$ | 0 | 0 | C_3 | $\in \mathbb{R}$ | C_3^* | 0 | 0 |
|------------------|---|---|-------|------------------|---------|---|---|

oder auch verschoben definiert:

| | | | | | | | |
|---|---|-------|-------|------------------|---------|---------|---|
| 0 | 0 | C_2 | C_3 | $\in \mathbb{R}$ | C_3^* | C_2^* | 0 |
|---|---|-------|-------|------------------|---------|---------|---|

In dieser Arbeit soll nun insbesondere der Fall untersucht werden, daß additiv überlagerte, impulsartige Fehler in einigen Komponenten des ‘Zeitbereichsvektors’ zu korrigieren sind, wobei jedoch alle Komponenten in gewissem Maße fehlerbehaftet seien, d.h. ein ‘Hintergrundrauschen’ sei überlagert. Gefordert werden muß dabei natürlich, wie später noch gezeigt wird, daß ein deutlicher Amplitudenunterschied zwischen beiden Fehlerarten besteht.

Für analoge Codes, besonders auch im Hinblick auf eine Korrektur impulsartiger Fehler, müssen nun geeignete Abstandsmaße definiert werden. Die Hamming-Distanz ist dabei in der strengen Form nicht mehr anwendbar. Es sei daher eine, dem Problem angepaßte Definition der Hamming-Distanz gegeben, die im folgenden der Einfachheit halber mit d bezeichnet wird.

Definition 2.8 Die ‘analoge’ Hamming-Distanz $d = d_{a,b}$ gebe die Anzahl der Komponenten wieder, in denen sich zwei Vektoren \vec{y} und \vec{z} um einen Betrag größer als a unterscheiden. Alle übrigen Komponenten dürfen dabei nicht um einen größeren Betrag als b voneinander abweichen. Formal läßt sich dies wie folgt ausdrücken:

$$d = d_{a,b} = |\mathcal{ID}| \text{ mit } \mathcal{ID} = \{i \mid |y_i - z_i| > a\}, \text{ falls } i \notin \mathcal{ID} \Rightarrow |y_i - z_i| < b, \ a > b.$$

Die hiermit definierte *analoge Hamming-Distanz* ist somit gleich der Anzahl der Komponenten mit Abweichungen großer Amplitude. Eine geeignete Wahl der Grenzen a und b wird im folgenden vorausgesetzt.

Diese Definition soll auch noch einmal unterstreichen, daß es in dieser Arbeit um die Korrektur impulsförmiger Störungen geht. Unter Vernachlässigung des eventuell zusätzlich vorhandenen ‘Hintergrundrauschens’, ist für die Korrekturfähigkeit analoger RS-Codes die Hamming-Distanz von Bedeutung. Diese steht in direkter Beziehung zu der Anzahl der vom Code nicht genutzten Dimensionen und der Anzahl korrigierbarer Fehler. Die Art des zugrundeliegenden Körpers hat hier keinen Einfluß.

Das ‘analoge’ Gewicht läßt sich entsprechend definieren

Definition 2.9 Das ‘analoge’ Gewicht eines Codewortes sei die analoge Hamming-Distanz d zum Nullwort.

Ein weiteres Abstandsmaß ist natürlich die *Euklidische Distanz* d_e .

Definition 2.10 Die euklidische Distanz d_e zweier Vektoren $\vec{y} = (y_0, \dots, y_{N-1})$ und $\vec{z} = (z_0, \dots, z_{N-1})$ ist gegeben durch

$$d_e = \sqrt{\sum_{i=0}^{N-1} |y_i - z_i|^2}.$$

Nachdem nun die wichtigsten grundlegenden Definitionen bereitgestellt sind, sollen im nächsten Abschnitt die erwarteten Eigenschaften kurz dargestellt werden, deren genauere Untersuchung Gegenstand des Hauptteils dieser Arbeit sein soll. In Verbindung mit diesem kurzen Abschnitt ist der letzte Teil dieses Kapitels mit Betrachtungen zur Kanalkapazität zu sehen, der diskrete und analoge Kanäle bei diskretem Eingang vergleicht.

2.3 Zu erwartende Eigenschaften analoger Codes

Zunächst stellen analoge Codes ein Bindeglied dar, mit dessen Hilfe Gemeinsamkeiten verschiedener Arbeitsgebiete im Laufe der Arbeit offenbar werden. Insbesondere die Tatsache, daß Algorithmen aus dem Bereich der Codierung mit denen der digitalen Signalverarbeitung (z.B. Interpolation, lineare Prädiktion etc.) mit geringfügigen Änderungen austauschbar sind, ist hervorzuheben.

Aus dem vorigen Kapitel ist bereits eine wichtige Eigenschaft analoger Codes zu erkennen. Es existieren keine Längenrestriktionen. Zu jeder Länge N gibt es ein entsprechendes primitives Element der Ordnung N ($z = e^{j2\pi/N}$). Hingegen gilt bei endlichen Zahlkörpern $N = q^s - 1$, d.h. es stehen nur bestimmte Längen zur Verfügung, wenn man einmal von den etwas eingeschränkten Möglichkeiten der Verkürzung oder Erweiterung absieht.

Wichtig ist natürlich auch, daß keine speziellen algebraischen Strukturen benötigt werden, d.h. zur Realisierung ist ebenfalls keine spezielle Hardware nötig. Ganz im Gegenteil wird gezeigt, daß Gleitkomma-Prozessoren mit wenigen Exponentenstellen oder bei geringen Codewortlängen sogar Festkomma-Prozessoren genügen, um den Rechengenauigkeitsansprüchen zu entsprechen. Es ist somit eine gewisse Vereinheitlichung der erforderlichen Hardwarestrukturen bei analoger Codierung und sonstiger digitaler Signalverarbeitung möglich.

Analoge Codes verwenden den analogen Empfangswert zur Decodierung. Geht man speziell von einem diskreten Sendetalphabet aus, so ist bekanntlich eine höhere Kanal-kapazität bei Kanälen mit analogem Ausgang zu erwarten. Untersuchungen hierzu finden sich im nächsten Abschnitt. Die dort wiedergegebenen Ergebnisse lassen auch erkennen, daß aus dem analogen Ausgangswert auf den Kanalzustand geschlossen werden kann und sich hierdurch die Kanalkapazität, selbst bei steigender Fehler-wahrscheinlichkeit in einem Kanalzustand (hier von zwei), erhöht. Es besteht dabei eine gewisse Beziehung zum letzten Kapitel, wo eine Vorgehensweise angegeben wird, die es erlaubt, aus einem Decodierergebnis Rückschlüsse auf die Kanalcharakteristik zu ziehen. Es sei jedoch betont, daß sonst in dieser Arbeit im allgemeinen keine Wertdiskretisierung vorausgesetzt wird.

Im Kapitel 4.1 wird gezeigt, daß analoge RS-Codes es theoretisch ermöglichen, (nahezu) alle Fehlermuster vom Gewicht $e \leq M - 1$ zu korrigieren, wohingegen praktisch bei RS-Codes über Galois-Feldern diese maximal zulässige Fehlerzahl auf $e \leq \lfloor M/2 \rfloor$ beschränkt bleibt, wenn man von den üblichen algebraischen Decodier-verfahren ausgeht (Bounded Minimum Distance Decoding). Die Korrekturfähigkeit analoger Codes (ohne Wertquantisierung) entspricht derjenigen bei einem unendlich großen Galois-Feld.

Von besonderer Bedeutung ist das Vorhandensein von Größenrelationen in \mathbb{C} , die bei endlichen Zahlkörpern fehlen. Die Decodierung bleibt daher nicht nur auf die binäre Entscheidung, ob ein Fehler an einer gewissen Codewortstelle aufgetreten ist, beschränkt; die Amplitude von Fehlern geht nun ebenfalls ein. Es wird daher möglich sein, zwischen ‘großen’ und ‘kleinen’ Fehlern zu unterscheiden. Inwieweit dies möglich ist, wird in Kapitel 5 ausführlich untersucht. Es wird dort die Frage ge-stellt, wie groß die Amplitude eines ‘Hintergrundrauschens’, das allen Komponenten eines Codewortes überlagert ist, werden darf, um das Auffinden von Fehlern großer Amplitude, die im folgenden grundsätzlich als Impulsfehler bezeichnet werden, noch sicherzustellen. In diesem Zusammenhang erfolgt eine neue Beschreibung einer der wichtigsten Decodieralgorithmen, dem Berlekamp-Massey-Algorithmus (BMA), und eine Untersuchung seiner Eigenschaften bei komplexen Toeplitz-Systemen. Die hier gemachten Aussagen sind nicht nur auf Codierungsaspekte beschränkt und stellen eigentlich den zentralen Punkt der Arbeit dar.

Es sei noch auf eine Eigenschaft hingewiesen, die sich ergibt, wenn man von K wertdiskreten Informationsstellen im ‘Zeitbereich’ ausgeht (beispielsweise von den m möglichen Zuständen der m -PSK) und die restlichen Stellen so berechnet, daß der ‘Frequenzbereichsvektor’ die gewünschten M Nullstellen aufweist (*systematische Co-dierung*). Diese M Parity-Komponenten liegen zum Teil nicht auf den vorgegebenen möglichen Zuständen der Informationsstellen. Es ergibt sich hiermit eine Möglich-

keit zur Rahmensynchronisation, da nur die Stellen detektiert werden müssen, die deutlich von dem vorgegebenen Raster abweichen. Dabei muß man jedoch so vorgehen, daß nicht eine einzelne impulsartige Störung zum Synchronisationsausfall führen kann (z.B. Betrachtung mehrerer Rahmen). Zur Verdeutlichung finden sich einige Abbildungen im Anhang A.1.

In diesem Abschnitt wurden Besonderheiten analoger Codes hervorgehoben, jedoch sind auch viele Gemeinsamkeiten mit Codes über endlichen Zahlkörpern zu nennen. Deutlich wird dies in dieser Arbeit insbesondere dann, wenn, der Anschaulichkeit halber, Beispiele über Galoisfeldern gewählt werden. Beispielsweise sind herkömmliche Decodieralgorithmen ebenfalls bei analogen Codes anwendbar (eventuell nach geringfügigen Änderungen).

Der nun folgende Abschnitt geht, wie bereits in diesem Abschnitt angesprochen, sendeseitig von diskreten Signalzuständen aus und vergleicht Kanalkapazitäten bei analogem und diskretisiertem Ausgang bei normalverteiltem Rauschen.

2.4 Kanalkapazitäten analoger Kanäle bei diskretem Eingang

Im folgenden soll ein Vergleich der Kanalkapazität bei analogem und diskretisiertem (binärem) Ausgang und binärem (z.B. PSK) Eingang durchgeführt werden, wobei normalverteiltes Rauschen als Störung angenommen wird. Die graphische Darstellung erfolgt jeweils in Abhängigkeit der Bitfehlerwahrscheinlichkeit bei diskretem Ausgang.

Die hier angegebenen Ergebnisse sind, um es noch einmal zu betonen, auf den Fall eines wertdiskreten Eingangsalphabetes bezogen, welches jedoch in der übrigen Arbeit nicht zwingend vorausgesetzt wird. Die Untersuchung ist, neben der allgemeinen Motivation, daß bei analogem Ausgang (und diskretem Eingang) mit einer höheren Kanalkapazität zu rechnen ist, wegen zweier Einzelergebnisse interessant. Es wird dort ersichtlich, daß unter der Voraussetzung zweier Kanalzustände, gekennzeichnet durch unterschiedliche Standardabweichungen des Gauss'schen Rauschens (unterschiedlicher Störabstand), der Kanalzustand in gewissem Maße aus dem Analogwert des Ausgangs geschlossen werden kann. Erkennbar wird dies an einer Erhöhung der Kanalkapazität trotz Steigerung der Fehlerwahrscheinlichkeit in *einem* Kanalzustand. In Kapitel 8 wird konkret ein Verfahren angegeben, wie man nach analoger Decodierung Aufschluß über den Kanalzustand erhält.

Da die Kanalkapazitätsberechnungen zum Teil sehr umfangreich werden und dieser Abschnitt lediglich im Rahmen einer Einleitung zu sehen ist, wird an dieser Stelle nur für den ersten, vergleichsweise einfachen Fall, die Herleitung angegeben. Für die übrigen Fälle wird dies detailliert im Anhang A.2 geschehen. Auch wird hier gänzlich auf die Darstellung von Grundlagen zur Kanalkapazität verzichtet (siehe hierzu z.B. [45], [23]).

Im folgenden bezeichnen x_i die (beiden) Eingangszeichen, y_i die (beiden) Ausgangszeichen nach Schwellwertentscheid und y den analogen Ausgangswert. Die Eingangszeichen werden normiert verwendet (+1 und -1) und die Dichte des normalverteilten Rauschens ergibt sich zu

$$p_y(y|\pm 1) = \frac{1}{\sqrt{2\pi}\sigma_y} e^{-(y\mp 1)^2/2\sigma_y^2} \quad \sigma_y = \sqrt{\frac{N_0}{2E_b}}, \quad (2.6)$$

wobei N_0 die einseitige Rauschleistungsdichte und E_b die Energie pro Bit kennzeichnet, die hier der Energie pro Zeichen (oder Symbol) E_s entspricht.

Hieraus folgt die Bitfehlerwahrscheinlichkeit für den entsprechenden diskreten Kanal zu

$$P_e = \int_0^\infty \frac{1}{\sqrt{2\pi}\sigma_y} e^{-(y+1)^2/2\sigma_y^2} = \frac{1}{2} \left(1 - \operatorname{erf} \sqrt{\frac{E_b}{N_0}} \right). \quad (2.7)$$

Diesen Zusammenhang und das Kanalmodell des symmetrischen Binärkanals zeigen die Abbildungen 2.1 und 2.2.

Die Kanalkapazität eines diskreten Kanals ergibt sich allgemein zu

$$C_{DMC} = \max_{P(x_j)} \sum_{j=0}^{Q_j-1} \sum_{i=0}^{Q_i-1} P(x_j) P(y_i|x_j) \log \frac{P(y_i|x_j)}{P(y_i)}. \quad (2.8)$$

Entsprechend folgt für die Kanalkapazität eines Kanals mit analogem Ausgang

$$C_{AWGN} = \max_{P(x_j)} \sum_{j=0}^{Q_j-1} \int_{-\infty}^{+\infty} P(x_j) p(y|x_j) \log \frac{p(y|x_j)}{p(y)} dy. \quad (2.9)$$

Folgende Fälle werden üblicherweise unterschieden:

- I** ein einziger Stöorzustand bestimmt durch E_b/N_0 ,
- II** mehrere (zwei) Stöorzustände a - nur $P(a)$ bekannt, jedoch nicht a selbst -,
- III** mehrere (zwei) Stöorzustände a - $P(a)$ und a selbst bekannt -,
- IV** mehrere (zwei) Stöorzustände a - $P(a)$ bekannt, mit Kanalzustandsschätzung \hat{a} , Schätzfehlerwahrscheinlichkeit $P(\hat{a}|a)$ bekannt -.

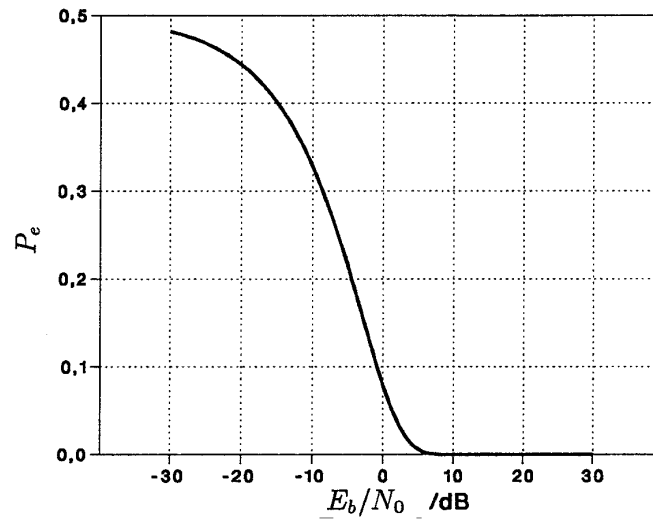


Abbildung 2.1: Bitfehlerwahrscheinlichkeit P_e in Abhängigkeit des Störabstands E_b/N_0 in dB

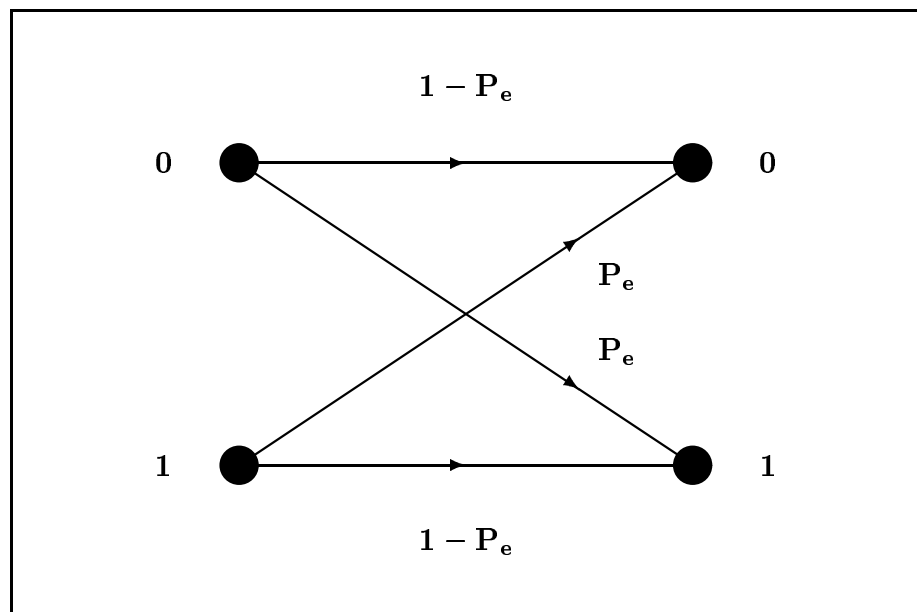


Abbildung 2.2: Symmetrischer Binärkanal

Modellhaft werden dabei, wie bereits angedeutet, für die Fälle **II** bis **IV** zwei normalverteilte Störzustände mit unterschiedlicher Standardabweichung angenommen - im folgenden als 'guten' (G) und 'schlechten' (B) Zustand bezeichnet -, um zwei verschiedene Störabstände wiederzugeben. Zur Beschreibung werden die folgenden Parameter herangezogen:

- b : Wahrscheinlichkeit, daß sich der Kanal im 'schlechten' Zust. befindet,
- $(1 - b)$: Wahrscheinlichkeit, daß sich der Kanal im 'guten' Zustand befindet,
- $E_b/N_0|_b$: Störabstand im 'schlechten' Zustand,
- $E_b/N_0|_g$: Störabstand im 'guten' Zustand,
- P_{eb} : Bitfehlerwahrscheinlichkeit im 'schlechten' Zustand,
- P_{eg} : Bitfehlerwahrscheinlichkeit im 'guten' Zustand.

Interessant sind besonders die Fälle **II** und **IV**, in welchen sich eine Erhöhung der Kanalkapazität im analogen Fall trotz steigender Bitfehlerwahrscheinlichkeit im 'schlechten' Kanalzustand nahe $P_{eb} = 0,5$ ergibt (siehe Abbildungen 2.5 und 2.10). Obwohl der Kanalzustand in diesen beiden Fällen nicht exakt bekannt ist, ist aus dem Analogwert bei hoher Fehlerwahrscheinlichkeit P_{eb} der Kanalzustand erkennbar. Bei $P_{eb} = 0,5$ entsprechen sich die Ergebnisse aus **II**, **IV** und **III** (Kanalzustand exakt bekannt). Man vergleiche die Abbildungen 2.5, 2.10 und 2.7.

I.) Für den Fall nur eines Störzustands ergibt sich die Kanalkapazität bei diskretem Ausgang (DMC, **d**iscrete **m**emoryless **c**hannel) bekanntlich zu

$$C_{DMC} = 1 + P_e \lg P_e + (1 - P_e) \lg(1 - P_e), \quad (2.10)$$

worin die Tatsache eingeht, daß bei einem symmetrischen binären Kanal das Maximum für $P(x_j) = 1/2$ erreicht wird ($\Rightarrow P(y_i) = 1/2$).

Bei analogem Ausgang (AWGN, **a**dditive **w**hite **g**aussian **n**oise) folgt

$$C_{AWGN} = \underbrace{\frac{1}{2}}_{P(-1)} \int_{-\infty}^{\infty} p(y|+1) \lg \frac{p(y|+1)}{p(y)} dy + \underbrace{\frac{1}{2}}_{P(+1)} \int_{-\infty}^{\infty} p(y|-1) \lg \frac{p(y|-1)}{p(y)} dy \quad (2.11)$$

$$\text{mit } p(y) = \frac{1}{2} \frac{1}{\sqrt{2\pi}\sigma_y} e^{-(y-1)^2/2\sigma_y^2} + \frac{1}{2} \frac{1}{\sqrt{2\pi}\sigma_y} e^{-(y+1)^2/2\sigma_y^2}.$$

Abbildung 2.3 zeigt im Vergleich die beiden Kanalkapazitäten in Abhängigkeit der Bitfehlerwahrscheinlichkeit P_e . (Für den AWGN ergibt sich E_b/N_0 aus (2.7) bzw. aus Bild 2.1.)

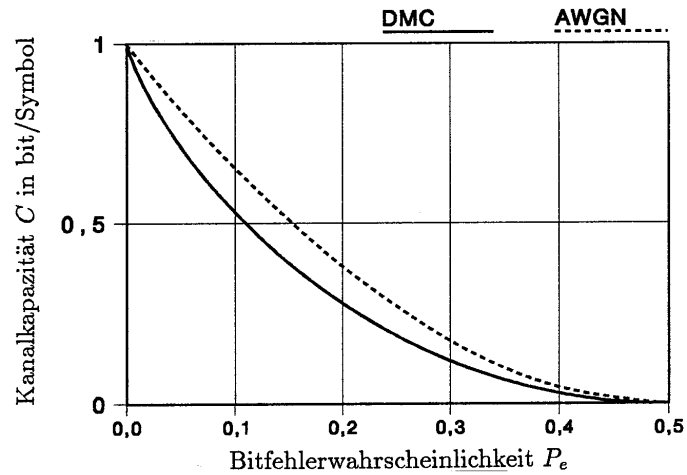


Abbildung 2.3: Kanalkapazitäten bei analogem und binärem Ausgang in Abhängigkeit der Bitfehlerwahrscheinlichkeit P_e

Für die weiteren Fälle seien hier nur in Kürze die Ergebnisse in Form einiger Diagramme angegeben (Herleitung siehe Anhang).

II.) — Zwei Störzustände a, nur $P(a)$ bekannt, jedoch nicht a selbst —

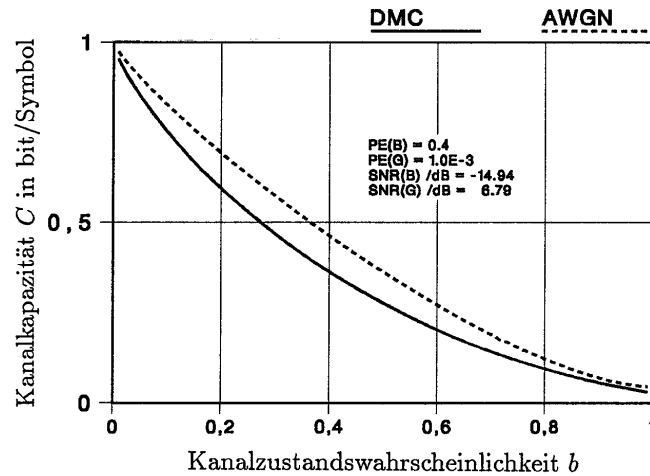


Abbildung 2.4: Kanalkapazitäten in Abhängigkeit der Kanalzustandswahrscheinlichkeit b

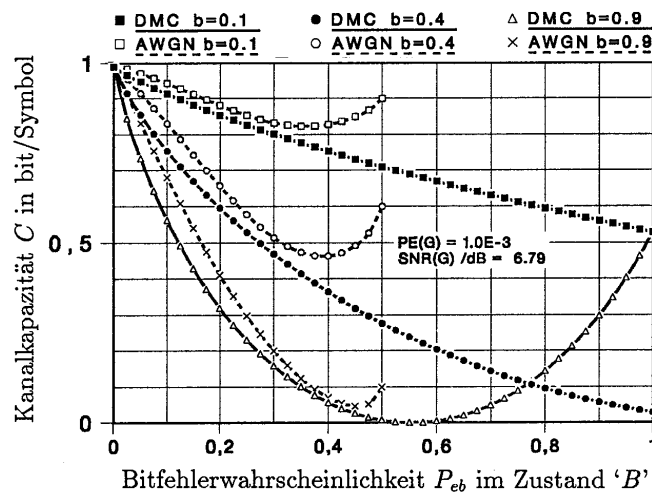


Abbildung 2.5: Kanalkapazitäten in Abhängigkeit der Bitfehlerwahrscheinlichkeit P_{eb} im Zustand B

III.) — Zwei Störzustände a, $P(a)$ und a selbst bekannt —

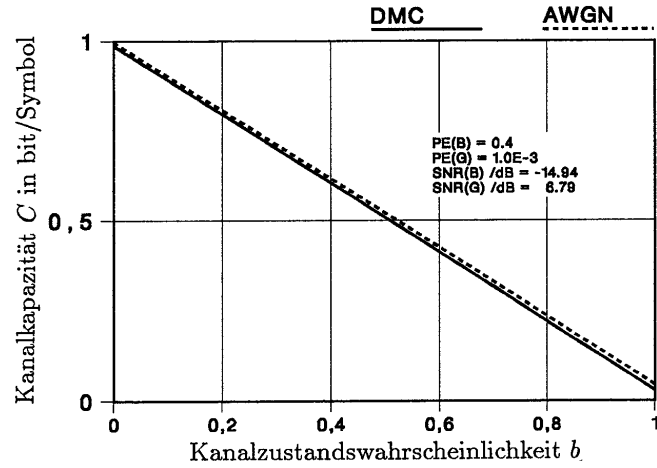


Abbildung 2.6: Kanalkapazitäten in Abhängigkeit der Kanalzustandswahrscheinlichkeit b

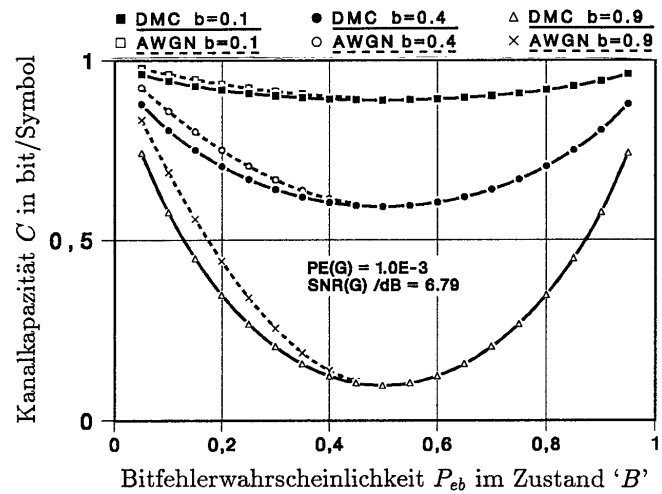


Abbildung 2.7: Kanalkapazitäten in Abhängigkeit der Bitfehlerwahrscheinlichkeit P_{eb} im Zustand B

IV.) — Zwei Störzustände a , $P(a)$ bekannt, mit Kanalzustandsschätzung \hat{a} , Schätzfehlerwahrscheinlichkeit $P(\hat{a}|a)$ bekannt —

Der Kanalzustandsschätzung liegt folgendes Modell mit den Schätzfehlerwahrscheinlichkeiten ϵ_1 und ϵ_2 zugrunde (nach [23]):

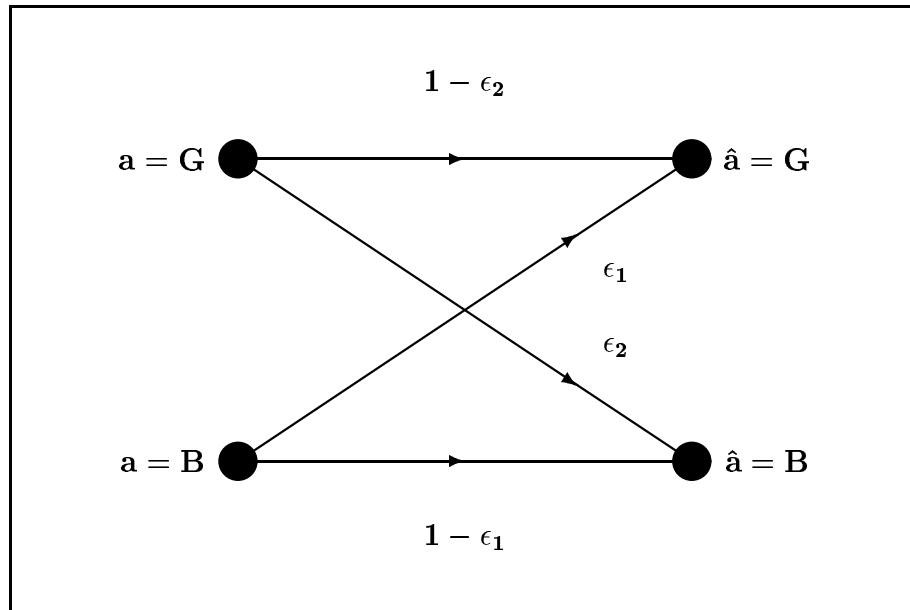


Abbildung 2.8: Modell der Kanalzustandsschätzung

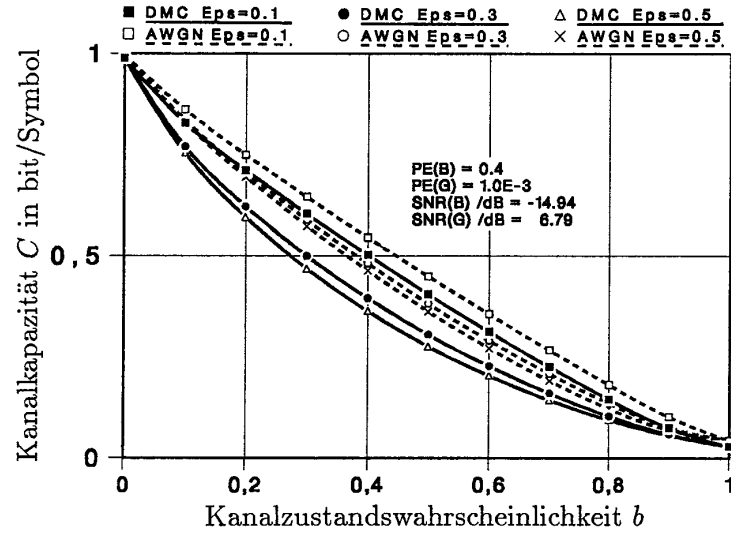


Abbildung 2.9: Kanalkapazitäten in Abhängigkeit der Kanalzustandswahrscheinlichkeit b

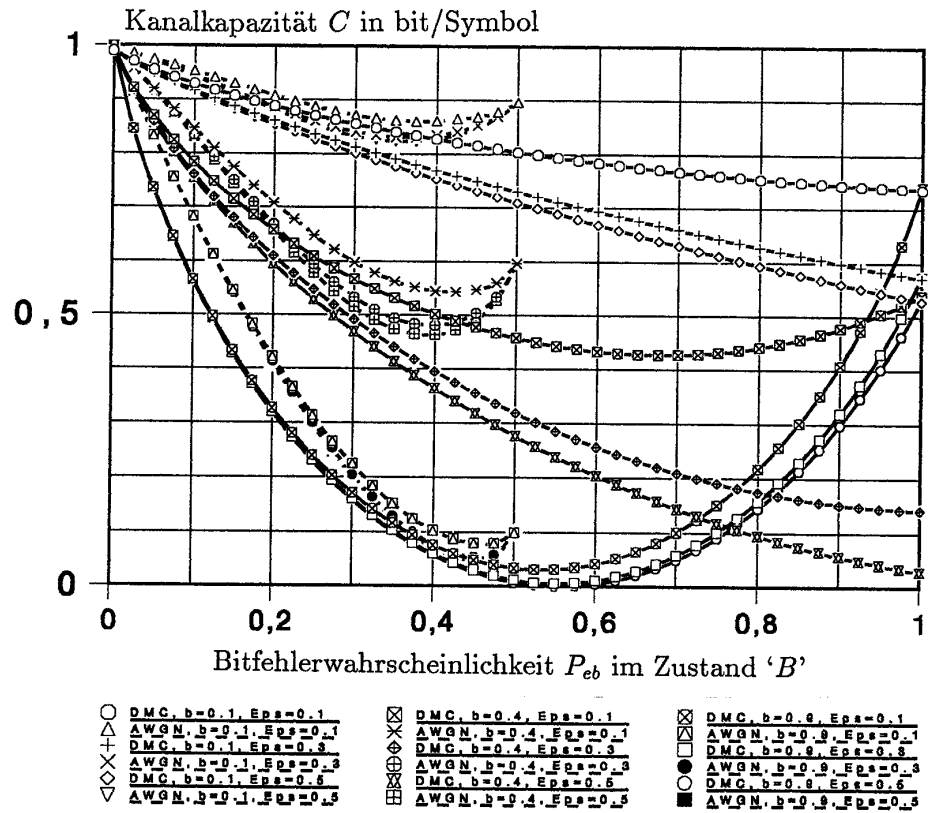


Abbildung 2.10: Kanalkapazitäten in Abhängigkeit der Bitfehlerwahrscheinlichkeit P_{eb} im Zustand B

Kapitel 3

Interpolation, Approximation und Decodierung

In diesem Kapitel werden einige Zusammenhänge zwischen Interpolations-, Approximationsverfahren und der Decodierung von RS-Codes angegeben. Die Decodierung wird dabei erkennbar als Anwendung von exponentieller und Polynominterpolation. Als Grundlage für die Betrachtungen wird zunächst in einem einleitenden Abschnitt die Decodierung von RS-Codes erläutert.

3.1 Decodierung von RS-Codes

Wie schon bemerkt, soll in diesem Abschnitt kurz die Vorgehensweise bei der Decodierung von RS-Codes skizziert werden.

Die Fehlerkorrektur erfolgt grundsätzlich in zwei Schritten:

- Bestimmung der Fehlerstellen,
- Bestimmung der Fehlerwerte.

Die Fehlerwerte seien im folgenden mit f_i bzw. in Polynomschreibweise mit $f(x)$ bezeichnet. Im Frequenzbereich resultiert daraus F_i bzw. $F(x)$. Ein empfangenes Zeichen r_i ergibt sich somit als $r_i = c_i + f_i$, $c_i \in \mathcal{C}$.

Zunächst definiert man ein sogenanntes **Fehlerstellenpolynom**, dessen Nullstellen die Fehlerstellen kennzeichnen.

Definition 3.1 Das Fehlerstellenpolynom $\Gamma(x)$ sei gegeben durch

$$\Gamma(x) = \prod_{i \in \mathcal{IF}} (x - z^{-i}),$$

wobei \mathcal{IF} die Indexmenge der Fehlerstellen darstellt ($\mathcal{IF} = \{i | f_i \neq 0\}$).

Entsprechend sei ein Nichtfehlerstellenpolynom definiert durch

$$\bar{\Gamma}(x) = \prod_{i \in \overline{\mathcal{IF}}} (x - z^{-i}),$$

wobei $\overline{\mathcal{IF}}$ die Indexmenge der fehlerfreien Stellen darstellt ($\overline{\mathcal{IF}} = \{i | f_i = 0\}$).

Aus $x^N - 1 = \prod_{i=0}^{N-1} (x - z^{-i})$ folgt $\bar{\Gamma}(x) = (x^N - 1)/\Gamma(x)$.

Mit obiger Definition ergibt sich die als **Schlüsselgleichung** bekannte Beziehung

$$\Gamma(x) \cdot F(x) = \Gamma(x) \cdot T(x) \cdot \bar{\Gamma}(x) = T(x) \cdot (x^N - 1) = 0 \pmod{(x^N - 1)}. \quad (3.1)$$

Die Koeffizienten des Produkts der Polynome $\Gamma(x)$ und $F(x)$ ergeben sich aus einer (zyklischen) Faltung der Koeffizientenvektoren. Sei e die aufgetretene Fehleranzahl, so folgt

$$\sum_{i=0}^{N-1} \Gamma_i \cdot F_{j-i} = \sum_{i=0}^e \Gamma_i \cdot F_{j-i} = 0 \quad j = 0, \dots, N-1. \quad (3.2)$$

Der Fehleranteil im Frequenzbereich ist jedoch nur an den Stellen des sogenannten **Syndroms** S_i direkt ablesbar, das allgemein wie folgt definiert ist:

Definition 3.2 Als Syndrom bezeichnet man aus dem Empfangswort (Vektor \vec{r}) ermittelte Terme, die nur vom Fehlervektor \vec{f} , jedoch nicht vom gesendeten Codewort \vec{c} , bzw. der zugehörigen Information abhängen.

Aus Definition 2.3 ergibt sich beispielsweise die übliche Syndromdefinition als Koeffizienten von $R(x)$ der Grade $K, \dots, N-1$, die bei einem gültigen Codewort gleich Null sind.

In Gleichung 2.4 wurde eine etwas allgemeinere Definition von RS-Codes angegeben. Diese ermöglicht es, das Syndrom auch an anderer Stelle im Frequenzbereich vorzusehen. Die entsprechenden Koeffizienten müssen lediglich (zyklisch) direkt aufeinanderfolgen. Man spricht insbesondere von einem *untenliegenden Syndrom*, wenn die niedrigsten Koeffizienten (Stellen 0 bis $2E-1$ im Frequenzbereich) verwendet werden. Definition 2.3 kennzeichnet somit ein *obenliegendes Syndrom*.

Im Falle eines untenliegenden Syndroms ergibt sich aus Gleichung 3.2

$$\sum_{i=0}^e \Gamma_i \cdot F_{j-i} = \sum_{i=0}^e \Gamma_i \cdot S_{j-i} = 0 \quad j = e, \dots, 2E - 1. \quad (3.3)$$

Normiert man das Fehlerstellenpolynom auf den niedrigsten Koeffizienten ($C_0 = 1$), so ergibt sich hiermit folgendes Gleichungssystem

$$(1, \Gamma_1, \Gamma_2, \dots, \Gamma_e) \begin{pmatrix} S_e & & S_{2e} \\ & \ddots & \\ S_0 & & S_e \end{pmatrix} = (0, \dots, 0) \quad (3.4)$$

mit einer sogenannten *Toeplitz*-Matrix, die wie folgt definiert wird:

Definition 3.3 Man spricht von einer *Toeplitz-Matrix* \underline{A} , wenn die Koeffizienten folgender Bedingung genügen:

$$a_{i,j} = a_{i+k,j+k}.$$

Die bekannten Decodieralgorithmen, wie z.B. Berlekamp-Massey-Algorithmus und Euklidischer Algorithmus, basieren auf dieser Struktur mit gleichen Komponenten in allen Diagonalen. Es ist ebenfalls noch eine andere Schreibweise üblich – als *Hankel*-System bezeichnet:

$$\begin{pmatrix} S_{2e} & S_e \\ & \\ S_e & S_0 \end{pmatrix} \begin{pmatrix} 1 \\ \Gamma_1 \\ \vdots \\ \Gamma_e \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \quad (3.5)$$

Hankel-Matrizen sind dabei nichts anderes als gespiegelte Toeplitz-Matrizen.

Definition 3.4 Man spricht von einer *Hankel-Matrix* \underline{A} , wenn die Koeffizienten folgender Bedingung genügen:

$$a_{i,j} = a_{i-k,j+k}.$$

Die Auflösung der Schlüsselgleichung liefert schließlich die Koeffizienten des Fehlerstellenpolynoms, dessen Nullstellen die Fehlerstellen bezeichnen. Dies erfolgt durch

rekursive Graderhöhung des Fehlerstellenpolynoms, dessen wahrer Grad e , welcher der Fehleranzahl entspricht, ja nicht bekannt ist.

An Stelle einer Nullstellensuche zur Fehlerortbestimmung, kann im diskreten Fall auch eine DFT-Rücktransformation erfolgen, wobei automatisch alle möglichen z^{-i} überprüft werden. Die verschwindenden Komponenten γ_i des rücktransformierten Fehlerstellenpolynoms kennzeichnen also die Fehlerstellen. Diese Vorgehensweise wird im Verlaufe dieser Arbeit auch auf analoge Codes übertragen.

Das Auflösen des Toeplitz-Systems ist der eigentlich zentrale Teil der Decodierung. Das Ermitteln der Fehlerwerte offenbart sich dann als nichts anderes als eine Aufgabe der Interpolation durch Polynome. Im Falle komplexer Zahlen kann dies auch als Approximationsproblem gesehen werden. Diese Aspekte werden insbesondere in den noch folgenden Abschnitten über die Lagrange-Interpolation und die Least-Squares-Approximation verdeutlicht.

Es lassen sich die Fehlerwerte z.B. durch rekursive Fortsetzung der Beziehung 3.3 mit $j = 2E, 2E+1, \dots$ gewinnen. Fernerhin steht eine als *Forney-Algorithmus* bekannte Vorschrift zur Verfügung:

$$f_i = \frac{z^i T(z^{-i})}{\Gamma'(z^{-i})} \quad \text{mit } T_j = - \sum_{i=0}^j \Gamma_i S_{j-i} \quad (3.6)$$

$$j = 0, 1, \dots, e-1, \quad T(x) = T_0 + T_1 x + \dots T_{e-1} x^{e-1}.$$

Diese folgt unter Zuhilfenahme von Gleichung 3.1 und der Regel von l'Hospital aus

$$f_i = \frac{1}{N} F(x) \Big|_{x=z^{-i}} = \frac{1}{N} \frac{T(x) \cdot (x^N - 1)}{\Gamma(x)} \Big|_{x=z^{-i}}.$$

Im übrigen erhält man die Fehlerwerte bzw. das korrekte Codewort auch durch Lösen eines *Vandermondschen* Gleichungssystems. Dabei sind zwei Fälle zu unterscheiden:

- ausgehend von einer Auswahl K fehlerfreier Stellen im Zeitbereich ermittelt man das Codewort im Frequenzbereich,
- ausgehend von e Syndromstellen ermittelt man die Fehlerwerte im Zeitbereich.

Erste Möglichkeit:

Bei obenliegendem Syndrom ergibt sich für die ausgewählten Stellen i_0, i_1, \dots, i_{K-1} das Gleichungssystem

$$\frac{1}{N} \begin{pmatrix} 1 & z^{-i_0} & z^{-2i_0} & \dots & z^{-(K-1)i_0} \\ 1 & z^{-i_1} & z^{-2i_1} & \dots & z^{-(K-1)i_1} \\ 1 & & & & \vdots \\ \vdots & & & & \vdots \\ 1 & z^{-i_{K-1}} & z^{-2i_{K-1}} & \dots & z^{-(K-1)i_{K-1}} \end{pmatrix} \begin{pmatrix} C_0 \\ C_1 \\ \vdots \\ C_{K-1} \end{pmatrix} = \begin{pmatrix} r_{i_0} \\ r_{i_1} \\ \vdots \\ r_{i_{K-1}} \end{pmatrix}. \quad (3.7)$$

Zweite Möglichkeit:

Bei untenliegendem Syndrom ergibt sich bei Betrachtung der e ersten Syndromstellen

$$\begin{pmatrix} (z^0)^{i_0} & (z^0)^{i_1} & \dots & (z^0)^{i_{e-1}} \\ (z^1)^{i_0} & (z^1)^{i_1} & \dots & (z^1)^{i_{e-1}} \\ \vdots & \vdots & & \vdots \\ (z^{e-1})^{i_0} & (z^{e-1})^{i_1} & \dots & (z^{e-1})^{i_{e-1}} \end{pmatrix} \begin{pmatrix} f_{i_0} \\ f_{i_1} \\ \vdots \\ f_{i_{e-1}} \end{pmatrix} = \begin{pmatrix} S_0 \\ S_1 \\ \vdots \\ S_{e-1} \end{pmatrix}. \quad (3.8)$$

Nachdem nun die prinzipielle Vorgehensweise bei der Decodierung von RS-Codes dargestellt wurde, sollen in den folgenden Abschnitten Parallelen zwischen Interpolation (Approximation) und der Decodierung aufgezeigt werden. Diese sind zwar nicht völlig neu, haben jedoch in der Codierungstheorie bislang nur wenig Beachtung gefunden. Zusammenhänge zwischen Codierung und Interpolation wurden von Mandelbaum 1979 [39] untersucht. Hier soll jedoch das Augenmerk besonders auf die Decodierung gerichtet sein. Die Betrachtung von Interpolationsverfahren führt insbesondere zu einem vielversprechenden Ansatz zur Definition eines ‘neuen’ Syndroms (siehe Kapitel 4.2).

Der folgende Abschnitt mag auch als Indiz für die Separierung der Codierungstheorie von der Approximationstheorie gelten. So entwickelten Gorenstein und Zierler einen Algorithmus, der eigentlich unter dem Namen *Prony's curve fitting method* längst bekannt war.

3.2 Pronys exponentielle Interpolation – – Gorenstein-Zierler-Algorithmus

1961 veröffentlichten Gorenstein und Zierler [22] (siehe auch [46]) einen Algorithmus, der identisch ist zu einem Verfahren, das Prony in [49] bereits 1795 vorstellte.

In der Literatur aus den Bereichen Approximationstheorie und Codierung wird seither, sogar noch bis zum heutigen Tage, der Algorithmus unter diesen verschiedenen Namen geführt. Die Gleichheit wurde erst 1967 von Wolf in [67] registriert.

Es soll nun im folgenden kurz das Verfahren der exponentiellen Approximation bzw. Interpolation nach Prony erläutert und der Bezug zur Codierungstheorie hergestellt werden. Die Darstellung orientiert sich dabei an [31].

Es wird eine Approximation in der Form

$$f(x) = A_0 e^{a_0 x} + A_1 e^{a_1 x} + \dots + A_{n-1} e^{a_{n-1} x} \quad (3.9)$$

bzw.

$$f(x) = A_0 \mu_0^x + A_1 \mu_1^x + \dots + A_{n-1} \mu_{n-1}^x, \quad \mu_k = e^{a_k} \quad (3.10)$$

gesucht.

Bei der Decodierung entspricht dieses der Beziehung für das Syndrom als Funktion der Fehlerwerte (und -stellen):

$$S_k = f_{i_0} (z^{i_0})^k + f_{i_1} (z^{i_1})^k + \dots + f_{i_{e-1}} (z^{i_{e-1}})^k. \quad (3.11)$$

Für den Fall äquidistanter Stützwerte $x = 0, 1, \dots, N_S - 1$ folgt das Gleichungssystem

$$\begin{aligned} A_0 + A_1 + \dots + A_{n-1} &= f_0 \\ A_0 \mu_0^1 + A_1 \mu_1^1 + \dots + A_{n-1} \mu_{n-1}^1 &= f_1 \\ A_0 \mu_0^2 + A_1 \mu_1^2 + \dots + A_{n-1} \mu_{n-1}^2 &= f_2 \\ &\vdots \\ A_0 \mu_0^{N_S-1} + A_1 \mu_1^{N_S-1} + \dots + A_{n-1} \mu_{n-1}^{N_S-1} &= f_{N_S-1} \end{aligned} \quad (3.12)$$

Analog hierzu:

$$\begin{aligned}
f_{i_0} + f_{i_1} + \dots + f_{i_{e-1}} &= S_0 \\
f_{i_0}(z^{i_0})^1 + f_{i_1}(z^{i_1})^1 + \dots + f_{i_{e-1}}(z^{i_{e-1}})^1 &= S_1 \\
f_{i_0}(z^{i_0})^2 + f_{i_1}(z^{i_1})^2 + \dots + f_{i_{e-1}}(z^{i_{e-1}})^2 &= S_2 \\
&\vdots \\
f_{i_0}(z^{i_0})^{2E-1} + f_{i_1}(z^{i_1})^{2E-1} + \dots + f_{i_{e-1}}(z^{i_{e-1}})^{2E-1} &= S_{2E-1}
\end{aligned} \tag{3.13}$$

Für $N_S = 2n$ bzw. $e = E$ läßt sich das Gleichungssystem eindeutig lösen – auf welche Weise, wird Inhalt dieses Abschnittes sein. Überbestimmte Systeme mit $N_S \geq 2n$ bzw. $E \geq e$ können mit ‘Least Squares’-Methoden behandelt werden (siehe Kapitel 3.4).

Wären die Exponenten bzw. die Terme μ_i (bzw. z^{i_j}) bekannt, so verbliebe nur die vergleichsweise einfache Aufgabe der Auflösung eines Vandermonde-Systems, die schon im vorigen Abschnitt angesprochen wurde.

Hier handelt es sich jedoch um ein nichtlineares Gleichungssystem in den μ_i . Die Lösung wird ermöglicht durch die Definition eines Polynoms

$$\mu^n - \alpha_1 \mu^{n-1} - \alpha_2 \mu^{n-2} - \dots - \alpha_{n-1} \mu - \alpha_n = 0 \tag{3.14}$$

mit den Nullstellen μ_i . Diese Definition entspricht derjenigen des Fehlerstellenpolynoms in der Codierung.

Multipliziert man nun die erste Gleichung aus 3.12 mit α_n , die zweite mit α_{n-1} , ... und die $n+1$ -te mit -1 und summiert all diese Gleichungen auf, so erhält man die erste Gleichung des folgenden linearen Gleichungssystems:

$$\begin{aligned}
f_{n-1}\alpha_1 + f_{n-2}\alpha_2 + \dots + f_0\alpha_n &= f_n \\
f_n\alpha_1 + f_{n-1}\alpha_2 + \dots + f_1\alpha_n &= f_{n+1} \\
&\vdots \\
f_{N_S-2}\alpha_1 + f_{N_S-3}\alpha_2 + \dots + f_{N_S-n-1}\alpha_n &= f_{N_S-1}
\end{aligned} \tag{3.15}$$

Die weiteren Gleichungen ergeben sich, wenn man sukzessive bei der zweiten, dritten, usw. Gleichung beginnt. Das Gleichungssystem ist für $N_S = 2n$ direkt lösbar und weist Toeplitz-Struktur auf. Eben dieses Gleichungssystem wurde schon im vorigen Abschnitt zur Ermittlung des Fehlerstellenpolynoms angegeben.

Es folgt dann die Ermittlung der Nullstellen des eingeführten Polynoms und anschließend die Berechnung der Koeffizienten A_i (Fehlerwerte) aus dem Vandermonde-Sys-

tem.

Exponentielle Interpolation nach Prony ist somit völlig identisch zum Problem der Fehlerkorrektur.

3.3 Lagrange-Interpolation zur Fehlerwertberechnung

Die Lagrange-Interpolation ist eine Darstellungsform der Polynominterpolation unter Verwendung sogenannter Kronecker-Deltas. Diese sind wie folgt definiert:

Definition 3.5 *Unter Kronecker-Delta δ_{ij} versteht man eine (diskrete) Funktion mit der Eigenschaft*

$$\delta_{ji} = \begin{cases} 1 & \text{falls } i = j \\ 0 & \text{falls } i \neq j. \end{cases}$$

Für die hier beabsichtigte Anwendung sei die Delta-Funktion in ‘kontinuierlicher’ Form jedoch wie folgt festgelegt:

$$\delta_j(\xi) = \begin{cases} 1 & \text{falls } \xi = z^{-i_j} \\ 0 & \text{falls } \xi \neq z^{-i_j}. \end{cases} \quad (3.16)$$

Hiermit ergibt sich die Interpolation für das Frequenzbereichspolynom ausgehend von K fehlerfreien Stützwerten (erste Möglichkeit in Abschnitt 3.1):

$$C(\xi) = \sum_{j=0}^{K-1} \delta_j(\xi) \cdot r_j. \quad (3.17)$$

Da $C(\xi)$ bei obenliegendem Syndrom als gradbeschränktes Polynom mit einem Maximalgrad von $K - 1$ festgelegt ist, muß das Kronecker-Delta $\delta_j(\xi)$ noch der Bedingung

$$\text{Grad } \{\delta_j(\xi)\} = K - 1$$

genügen. Hiermit folgt

$$\delta_j(\xi) = \prod_{\substack{k=0, \dots, K-1 \\ k \neq j}} (\xi - z^{-i_k}) \bigg/ \prod_{\substack{k=0, \dots, K-1 \\ k \neq j}} (z^{-i_j} - z^{-i_k}) =$$

$$= \frac{\prod_{k=0}^{K-1} (\xi - z^{-i_k})}{(\xi - z^{-i_j}) \cdot \left[\frac{\partial}{\partial \xi} \prod_{k=0}^{K-1} (\xi - z^{-i_k}) \right]_{\xi=z^{-i_j}}} . \quad (3.18)$$

Durch Einsetzen von $\xi = z^{-i}$ in 3.17 erhält man die übrigen Codewortstellen (im Zeitbereich).

Die zweite Möglichkeit in Abschnitt 3.1 ist einer Lagrange-Beschreibung nicht zugänglich, da das Polynom $f(x)$, dessen Koeffizienten die Fehlerwerte selbst darstellen, nicht gradbegrenzt ist. Solche Polynome werden in der angloamerikanischen Literatur als ‘sparse’ (dünnbesetzt) bezeichnet. Hierfür existieren spezielle Interpolationsalgorithmen. Der interessierte Leser sei hierzu auf [35] und die dort zitierte Literatur verwiesen.

Um die Problematik jedoch noch etwas genauer zu erläutern, sei einmal der – natürlich zum Scheitern verurteilte – Lagrange-Ansatz hierfür angegeben:

$$\begin{aligned} f(\xi) &= \sum_{i=0}^{e-1} \delta_j(\xi) S_j \\ \text{Grad } \{\delta_j(\xi)\} &= \max_{k \in [0, e-1]} \{i_k\} \\ \delta_j(\xi) &= \frac{\prod_{k=0}^{e-1} (\xi - z^k)}{(\xi - z^j) \cdot \left[\frac{\partial}{\partial \xi} \prod_{k=0}^{e-1} (\xi - z^k) \right]_{\xi=z^j}} \cdot L_j(\xi) . \end{aligned}$$

Es erscheint hier ein unbestimmter Anteil $L_j(x)$ der aus der Gradbedingung folgt ($\text{Grad } \{L_j(\xi)\} = \max_{k \in [0, e-1]} \{i_k\} - e + 1$).

Eine zu Lagrange äquivalente Polynominterpolation liefert das sogenannte *Newton-Verfahren*, dessen Beschreibung jedoch auf das nächste Kapitel 4.2 zurückgestellt werden soll. Dort wird unter Zuhilfenahme dieses Verfahrens ein neues Syndrom definiert, das vielversprechende Eigenschaften aufweist.

(Zusammenhänge zwischen Codierung und Lagrange-Interpolation wurden auch von Hsueh, Wang und Lee in [32] betrachtet.)

3.4 Least-Squares-Approximation zur Fehlerwertberechnung

Die Anwendung von Methoden der kleinsten Quadrate (Least Squares) gewinnt natürlich bei der Untersuchung analoger Codes an Bedeutung. Insbesondere eröffnet sich hiermit die Möglichkeit der Ausnutzung aller zur Verfügung stehender Gleichungen zur Fehlerwertbestimmung. Bei endlichen Zahlkörpern hingegen, sind überbestimmte Gleichungssysteme wegen des Fehlens von Größenrelationen von untergeordneter Bedeutung (lediglich zusätzliche Kontrollmöglichkeit). Weiterhin ist eine Wichtung einzelner Gleichungen möglich, was wiederum das Einarbeiten von Zusatzinformation, die beispielsweise den Kanalzustand kennzeichnet, erlaubt.

Beschreibungen zur Least-Squares-Approximation finden sich eigentlich in jedem Lehrbuch der Numerik, ‘Praktischen’ Mathematik und Approximationstheorie (insbesondere erwähnt seien hier [4], [31], [12] und [45]). Die Herleitung der sogenannten Normalgleichungen für den komplexen Fall wird im folgenden kurz skizziert.

Minimierungsproblem: Eine Funktion $f(x)$, vorgegeben an endlich vielen Stützstellen x_i (Stützwerte f_i , Anzahl N_S), soll im Sinne der kleinsten Quadrate an diesen Stützstellen durch eine Linearkombination von $n < N_S$ Elementarfunktionen $\phi_j(x)$ angenähert werden.

$$\sum_i w_i \left| f_i - \sum_{j=0}^{n-1} a_j \cdot \phi_j(x_i) \right|^2 \rightarrow \min \quad (3.19)$$

Hierbei stellt $w_i \geq 0$ eine Wichtung der jeweiligen Stützstelle dar.

Dieses Minimum wird dann erreicht, wenn die Normalgleichungen

$$\left\langle w_i \left[\sum_{j=0}^{n-1} a_j \cdot \phi_j(x_i) - f_i \right], \phi_k^*(x_i) \right\rangle_i = 0, \quad k = 0, \dots, n-1 \quad (3.20)$$

($*$: konjugiert komplex) erfüllt sind. Hierbei kennzeichnet $\langle a_i, b_i \rangle_i$ das Skalarprodukt von Vektoren \vec{a} und \vec{b} , deren Komponenten mit i indiziert sind. (Die Vektoren werden dabei oft mit $\text{tab}(f)$ bezeichnet.) Umstellen obiger Gleichung ergibt

$$\sum_{j=0}^{n-1} \langle w_i \phi_j(x_i), \phi_k^*(x_i) \rangle_i \cdot a_j = \langle w_i f_i, \phi_k^*(x_i) \rangle_i. \quad (3.21)$$

Die Normalgleichungen sollen hier nicht bewiesen werden – statt dessen soll beispielhaft eine Veranschaulichung im dreidimensionalen Raum angegeben werden.

Soll ein Vektor im dreidimensionalen Raum durch zwei Vektoren so approximiert werden, daß der Fehler im Sinne des Betragsquadrates minimal wird, so muß der Fehler $(\sum a_j \vec{\phi}_j - \vec{f})$ selbst orthogonal zu der Ebene stehen, die durch die beiden Repräsentanten (Vektoren $\vec{\phi}_1$ und $\vec{\phi}_2$) aufgespannt wird. (siehe Bild 3.1)

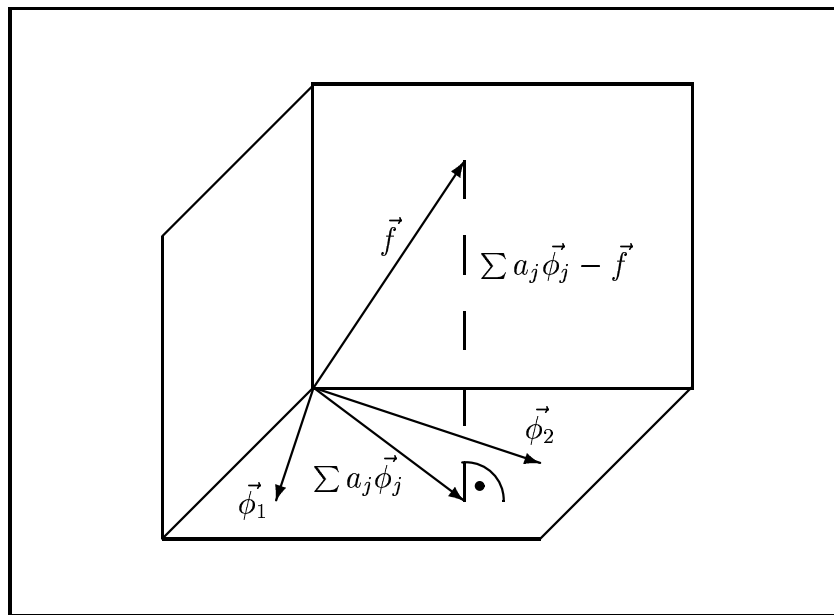


Abbildung 3.1: Veranschaulichung der Normalgleichungen

Die Normalgleichungen werden nun für die beiden schon oben angesprochenen Fälle spezialisiert. Diese sind:

- I** ausgehend von *allen* fehlerfreien Stellen im Zeitbereich, ermittelt man das Codewort im Frequenzbereich,
- II** ausgehend von *allen* Syndromstellen, ermittelt man die Fehlerwerte im Zeitbereich.

I.) Die Menge der Fehlerstellen sei im folgenden wieder mit \mathcal{F} , diejenige der Nichtfehlerstellen mit $\overline{\mathcal{F}}$ bezeichnet.

Die fehlerfreien Stellen r_i , $i \in \overline{\mathcal{F}}$ ergeben sich gemäß der DFT zu

$$r_i = \frac{1}{N} \sum_{k=0}^{K-1} C_k (z^{-i})^k. \quad (3.22)$$

Die Elementarfunktionen sind somit

$$\phi_k(x) = x^k / N. \quad (3.23)$$

Hiermit ergeben sich die Normalgleichungen zu

$$\sum_{k=0}^{K-1} C_k / N \left[\sum_{i \in \overline{\mathcal{F}}} w_i (z^{-i})^k \cdot (z^{-i})^{-\nu} \right] = \sum_{i \in \overline{\mathcal{F}}} w_i r_i (z^{-i})^{-\nu}, \quad \nu = 0, \dots, K-1. \quad (3.24)$$

In Matrixschreibweise:

$$\begin{aligned} \frac{1}{N} \begin{bmatrix} \sum_{i \in \overline{\mathcal{F}}} w_i & \sum w_i z^{i(-1+0)} & \dots & \sum w_i z^{i(-K+1+0)} \\ \sum w_i z^{i(0+1)} & \sum w_i & & \vdots \\ \vdots & & \ddots & \vdots \\ \sum w_i z^{i(0+K-1)} & \dots & \dots & \sum w_i \end{bmatrix} \begin{bmatrix} C_0 \\ C_1 \\ \vdots \\ C_{K-1} \end{bmatrix} = \\ = \begin{bmatrix} \sum_{i \in \overline{\mathcal{F}}} w_i r_i \\ \sum w_i r_i z^{1i} \\ \vdots \\ \sum w_i r_i z^{(K-1)i} \end{bmatrix}. \end{aligned} \quad (3.25)$$

Auflösen dieses *hermiteschen* Gleichungssystems liefert das Codewort im Frequenzbereich.

II.) Das Syndrom ergibt sich aus den Fehlerwerten gemäß der DFT zu

$$S_k = \sum_{i \in \mathcal{F}} f_i (z^k)^i. \quad (3.26)$$

Die Elementarfunktionen sind somit

$$\phi_i(x) = x^i. \quad (3.27)$$

Hiermit ergeben sich die Normalgleichungen zu

$$\sum_{i \in \mathbb{IF}} f_i \left[\sum_k w_k (z^k)^i \cdot (z^k)^{-\nu} \right] = \sum_k w_k S_k (z^k)^{-\nu}, \quad \nu \in \mathbb{IF}. \quad (3.28)$$

In Matrixschreibweise:

$$\begin{bmatrix} \sum_k w_k & \sum w_k z^{k(i_2-i_1)} & \dots & \sum w_k z^{k(i_e-i_1)} \\ \sum w_k z^{k(i_1-i_2)} & \sum w_k & & \vdots \\ \vdots & & \ddots & \vdots \\ \sum w_k z^{k(i_1-i_e)} & \dots & \dots & \sum w_k \end{bmatrix} \begin{bmatrix} f_{i_1} \\ f_{i_2} \\ \vdots \\ f_{i_e} \end{bmatrix} = \begin{bmatrix} \sum_k w_k S_k z^{-ki_1} \\ \sum w_k S_k z^{-ki_2} \\ \vdots \\ \sum w_k S_k z^{-ki_e} \end{bmatrix}. \quad (3.29)$$

Auflösen dieses *hermiteschen* Gleichungssystems liefert nunmehr die Fehlerwerte im Zeitbereich.

Die oben eingeführten Gewichtungsfaktoren erlauben auf einfache Weise, Kanalzustandsinformation zu berücksichtigen, was bei herkömmlichen Blockcodes nur sehr begrenzt möglich ist (siehe z.B. [8]).

In einem späteren Kapitel zum Vergleich von Tiefpaßfilterung und Fehlerkorrektur finden sich Simulationsergebnisse, die unter anderem mit Least-Squares-Korrektur nach **II.**) erhalten wurden. Es soll daher an dieser Stelle auf Beispielrechnungen verzichtet werden.

Im folgenden, dem letzten Abschnitt dieses Kapitels wird noch der Zusammenhang zwischen der Approximation durch Kettenbrüche (continued fractions) und dem Euklidischen Algorithmus, der anstelle des BMA als Decodieralgorithmus Verwendung findet, kurz angesprochen.

3.5 Euklidischer Algorithmus und Approximation mit Kettenbrüchen

Die Approximation durch Kettenbrüche entstand ehemals als Vorgehensweise zur Darstellung von reellen Zahlen durch ganze Zahlen.

Diese Entwicklung ergibt sich dann wie folgt¹:

$$s \in \mathbb{R}, \quad a_i \in \mathbb{Z}, \quad r_i \in \mathbb{R} \wedge 0 < |r_i| < 1$$

$$\begin{aligned} s &= a_0 + r_0 \\ &= a_0 + \frac{1}{a_1 + r_1} \\ &= a_0 + \frac{1}{a_1 + \frac{1}{a_2 + r_2}} \\ &= a_0 + (a_1 + (a_2 + \dots + (a_{n-1} + (a_n + r_n)^{-1})^{-1} \dots)^{-1} \end{aligned}$$

$$\begin{aligned} a_0 &= \lfloor s \rfloor \\ a_n &= \left\lfloor \frac{1}{r_{n-1}} \right\rfloor, \quad n \geq 1 \quad (\text{falls } r_{n-1} \neq 0) \\ r_0 &= s - a_0 \\ r_n &= \frac{1}{r_{n-1}} - a_n \end{aligned}$$

Die Kettenbruchentwicklung ergibt sich somit aus fortgesetztem Abspalten des nicht ganzzahligen Restes und dessen Darstellung durch seinen Kehrwert.

Bringt man den Kettenbruch jeweils in die rationale Darstellung

$$s = \frac{A_n(a_n + r_n) + B_n}{C_n(a_n + r_n) + D_n}, \quad (3.30)$$

so ergeben sich – wenn man die Reste zu Null setzt – Rekursionen für den Zähler, den Nenner und den Approximationsfehler.

¹Die Darstellung orientiert sich dabei zunächst an einer Veröffentlichung von Welch und Scholz 1979 ([63]).

Der rekursiv auftretende Quotient sei

$$s_n = \frac{P_n}{Q_n} \quad \text{bei } r_n = 0. \quad (3.31)$$

$$P_n = A_n a_n + B_n \quad Q_n = C_n a_n + D_n \quad (3.32)$$

Vergleicht man Gleichung 3.30, wobei n ersetzt wird durch $n+1$ mit Gleichung 3.30, wobei r_n ersetzt wird durch $(a_{n+1} + r_{n+1})^{-1}$, so erhält man

$$A_{n+1} = P_n \quad B_{n+1} = P_{n-1} \quad C_{n+1} = Q_n \quad D_{n+1} = Q_{n-1}, \quad (3.33)$$

und mit den Gleichungen 3.32 folgen schließlich die Rekursionsbeziehungen für Zähler und Nenner

$$P_{n+1} = a_{n+1} P_n + P_{n-1} \quad (3.34)$$

$$Q_{n+1} = a_{n+1} Q_n + Q_{n-1} \quad (3.35)$$

mit den Anfangsbedingungen

$$P_0 = a_0 \quad Q_0 = 1 \quad P_{-1} = 1 \quad Q_{-1} = 0. \quad (3.36)$$

Ebenso läßt sich eine Rekursion bei Betrachtung des Approximationsfehlers angeben:

$$s - \frac{P_n}{Q_n} = \frac{R_n}{Q_n} \quad R_n = s \cdot Q_n - P_n \quad (3.37)$$

$$\implies R_{n+1} = a_{n+1} R_n + R_{n-1} \quad \text{mit } R_{-1} = -1, \quad R_0 = s - a_0 = r_0 \quad (3.38)$$

$$a_n = \left\lfloor -\frac{R_{n-2}}{R_{n-1}} \right\rfloor \quad (3.39)$$

(näheres hierzu siehe [63]).

Überträgt man die zuvor dargestellte Approximation mit Kettenbrüchen auf Potenzreihen, so entspräche beispielsweise der reellen Zahl s die Potenzreihe

$$s(x) = \sum_{j=0}^{\infty} s_j x^{d-j}, \quad s_0 \neq 0. \quad (3.40)$$

Vergleichbar den ganzen Zahlen wäre dann

$$p(x) = \sum_{j=0}^d p_j x^{d-j}. \quad (3.41)$$

Der *Euklidische Algorithmus* ermittelt bekanntlich den *GGT* zweier Polynome $m(x)$ und $n(x)$ (oder ganzer Zahlen m und n). Die Vorgehensweise ist dabei wie folgt:

$$\begin{array}{ll} & r'_{-2} = 1m + 0n \\ & r'_{-1} = 0m + 1n \\ r'_{-2} = a_0 r'_{-1} + r'_0 & r'_0 = 1m - a_0 n \\ r'_{-1} = a_1 r'_0 + r'_1 & r'_1 = q_1 m + p_1 n \\ r'_0 = a_2 r'_1 + r'_2 & \vdots \\ \vdots & \vdots \\ r'_{k-2} = a_k r'_{k-1} + r'_k & r'_k = q_k m + p_k n \\ \vdots & \vdots \\ r'_{t-2} = a_t r'_{t-1} + r'_t & r'_t : \text{letzter Rest} \neq 0 \longrightarrow GGT(m, n) \\ r'_{t-1} = a_{t+1} r'_t + 0 & \end{array} \quad (3.42)$$

$$a_i = \left\lfloor \frac{r'_{i-2}}{r'_{i-1}} \right\rfloor \quad (3.43)$$

Für $\frac{r'_{-2}}{r'_{-1}}$ ergibt sich folgender Kettenbruch

$$\frac{r'_{-2}}{r'_{-1}} = a_0 + \frac{1}{r'_{-1}/r'_0} = a_0 + \frac{1}{a_1 + \frac{1}{r'_0/r'_1}} = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{r'_1/r'_2}}} = \dots \quad (3.44)$$

Vergleicht man mit der Darstellung oben, so entsprechen sich

$$\frac{r'_{i-2}}{r'_{i-1}} \equiv s \quad a_i = \left\lfloor \frac{r'_{i-2}}{r'_{i-1}} \right\rfloor \equiv \left\lfloor \frac{1}{r'_{i-1}} \right\rfloor. \quad (3.45)$$

Der Euklidsche Algorithmus, der bei der Decodierung ein mögliches Lösungsverfahren des Toeplitz-Systems darstellt, liefert somit die Kettenbruchapproximation des Quotienten zweier Polynome.

Bemerkungen

- Neuere Veröffentlichungen zeigen die Äquivalenz von Berlekamp-Massey-Algorithmus und Euklidischem Algorithmus ([13], insbesondere aber [17])
- Es bestehen weitere Zusammenhänge zwischen Euklidischem Algorithmus und der sogenannten Padé-Approximation, einem Verfahren zur Approximation von Potenzreihen durch Quotienten von Polynomen bestimmten Grades. Stimmen Zähler- und Nennergrad überein, so führt dies auf ein Toeplitz-System, das Gleichung 3.4 entspricht (siehe [10]).

Kapitel 4

Konstruktive Beweise zur Korrekturfähigkeit analoger Codes

In Kapitel 2.3 wurde bereits die höhere Korrekturfähigkeit analoger Codes im Vergleich zu herkömmlichen RS-Codes angesprochen. Die folgenden beiden Abschnitte sollen der näheren Betrachtung dieser Eigenschaft gewidmet sein. Mit zwei unterschiedlichen konstruktiven Beweisen wird gezeigt, daß analoge RS-Codes es erlauben, ‘nahezu’ alle Fehlermuster vom Gewicht $M - 1$ zu korrigieren, wenn M Parity-Stellen vorhanden sind. Insbesondere der zweite, in dieser Arbeit vorgestellte Beweis liefert eine vielversprechende Definition eines neuen Syndroms, indem Eigenschaften der Newton-Interpolation ausgenutzt werden. Der erste Abschnitt soll jedoch zunächst den ursprünglichen Beweis aus der schon zitierten Veröffentlichung von Wolf ([68]) wiedergeben.

4.1 Beweis nach J.K. Wolf

Der Beweis von Wolf geht davon aus, daß an den e Fehlerstellen keine diskreten, sondern zufällige kontinuierliche Fehlerwerte auftreten. Die übrigen Stellen des Codewortes seien fehlerfrei.

Zunächst ist ohne Beweis einsichtig, daß eine Anzahl von bis zu $\lfloor (d_h - 1)/2 \rfloor = \lfloor M/2 \rfloor$ Fehlern mit herkömmlichen Decodierverfahren korrigiert werden können. Dieses Ergebnis ist von herkömmlichen RS-Codes her bekannt. Hierbei ist der betrachtete Zahlenkörper ohne Belang, lediglich die Anzahl der Fehler in Relation zur Hamming-Distanz ist von Interesse. Man bedenke dabei, daß vorausgesetzt wurde, alle übrigen Stellen, außer den Impulsfehlerstellen, seien völlig fehlerfrei. Es wurde an dieser Stel-

le auch absichtlich nicht die ebenfalls in Kapitel 2 eingeführte ‘analoge’ Distanz $d_{a,b}$ verwendet, obwohl sie hier bei geeigneter Wahl der Parameter a und b zu d_h äquivalent ist. Man beachte ebenfalls, daß $d_h - 1$ die Anzahl der von einem Codewort nicht verwendeten Dimensionen kennzeichnet. Diese Bedeutung ist nicht von der Wahl des Zahlkörpers abhängig. Es sollte vielleicht auch betont werden, daß Euklidische Distanzen hier ebenfalls nicht betrachtet werden, da an die Aufgabe einer Impulsfehlerkorrektur gedacht ist. Ferner wurde auch keine Wertdiskretisierung des Codewortalphabetes angenommen, die Voraussetzung für eine Korrektur hinsichtlich geringer Euklidischer Distanzen wäre.

RS-Codes haben nun die Eigenschaft, daß ausgehend von K beliebigen fehlerfreien Codewortsymbolen das gesamte Codewort rekonstruiert werden kann (siehe auch die Ausführungen zur Lagrange-Interpolation, ‘MDS’). Sind nun $K + 1$ Stellen fehlerfrei, so führt jede der $\binom{K+1}{K} = K + 1$ Auswahlen von K richtigen Stellen zum ursprünglichen Codewort. Sind die Fehlerwerte kontinuierlich und zufällig, so geht die Wahrscheinlichkeit gegen Null, daß zwei fehlerbehaftete Auswahlen zu demselben Codewort gehören. Zur Fehlerkorrektur müssen also lediglich zwei der insgesamt $\binom{N}{K}$ Auswahlen das gleiche Codewort – das fehlerfreie – ergeben. Somit ist der folgende Satz bewiesen.

Satz 4.1 *Analoge RS-Codes mit M Prüfstellen erlauben die Korrektur von (nahezu) allen Fehlermustern des Gewichts $M - 1$, wenn die Fehler zufällige kontinuierliche Werte annehmen.*

Der zitierte Beweis ist zwar konstruktiv, jedoch als Decodierverfahren aus Aufwandsgründen ungeeignet. Um dies zu demonstrieren, wird kurz die Wahrscheinlichkeit des Korrekturversagens in Abhängigkeit der Anzahl der Stichproben und der Fehleranzahl e berechnet.

Sind e Fehler aufgetreten, so ist die Wahrscheinlichkeit, bei einer Stichprobe K richtige Stellen zu entnehmen, gegeben durch

$$P = \frac{\binom{N-e}{K}}{\binom{N}{K}}. \quad (4.1)$$

Es werde im folgenden die Tatsache unberücksichtigt gelassen, daß gleiche Stichproben nicht mehrmals entnommen werden, d.h. die Wahrscheinlichkeit eine i -te korrekte als j -te Stichprobe zu erhalten

$$P = \frac{\binom{N-e}{K} - (i-1)}{\binom{N}{K} - (j-1)}, \quad (4.2)$$

wird genähert durch P gemäß Gleichung 4.1.

Die Wahrscheinlichkeit, daß i von r Stichproben K fehlerfreie Stellen erfassen, beträgt

$$P_i = \binom{r}{i} P^i (1-P)^{r-i}. \quad (4.3)$$

Die Gesamtwahrscheinlichkeit mit r Stichproben das ursprüngliche Codewort zu ermitteln, ergibt sich hiermit zu

$$P(r) = \sum_{i=2}^{\min(r, \binom{N-e}{K})} \binom{r}{i} P^i (1-P)^{r-i} \quad (2 \leq r \leq \binom{N}{K} - \binom{N-e}{K} + 2). \quad (4.4)$$

Bild 4.1 zeigt beispielhaft die Wahrscheinlichkeit für Korrekturversagen ($1 - P(r)$) in Abhängigkeit der Anzahl r der Stichproben und der Fehleranzahl e . Man erkennt, daß die Anzahl der benötigten Stichproben stark mit der Anzahl der Fehler zunimmt.

Dieser Abschnitt weist die höhere Korrekturfähigkeit analoger gegenüber der herkömmlicher Codes (bei praktikablen Galoisfeldgrößen) nach, allerdings fehlt bislang ein praktikabler Algorithmus, um diese Eigenschaft zu nutzen. Auch der folgende Abschnitt wird letztlich kein Decodierverfahren liefern, jedoch wird dort ein Syndrom definiert, das näheren Bezug sowohl zum Zeitbereichscodewort – fehlerfreie Bereiche sind sofort als solche zu erkennen –, als auch zum herkömmlichen, durch ‘DFT’ (im Sinne von *Generalized RS-Codes*) ermittelten Syndrom hat. Dieses ‘neue’ Syndrom stellt sicherlich zumindest einen vielversprechenden Denkansatz zur Entwicklung eines Decodierverfahrens dar.

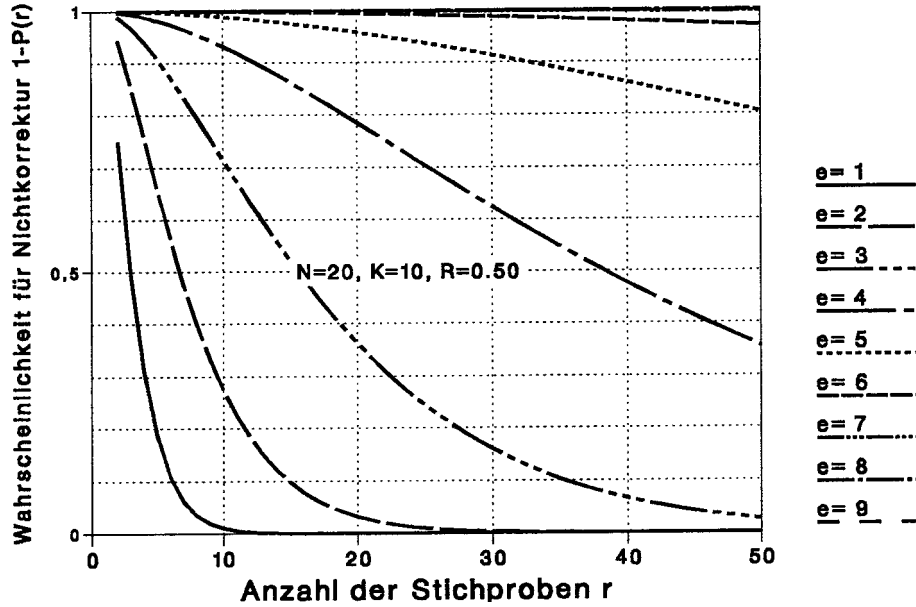


Abbildung 4.1: Wahrscheinlichkeit für Korrekturversagen ($1-P(r)$) in Abhängigkeit der Anzahl r der Stichproben und der Fehleranzahl e

4.2 Das Newton-Verfahren zur Interpolation und Definition eines ‘neuen’ Syndroms

Das Newton-Verfahren erweist sich für Decodieranwendungen als sehr vielversprechende Darstellungsform der Polynominterpolation. Bei gegebenem Maximalgrad n (Stützstellenzahl $n+1$) wird folgender Ansatz verwendet:

$$f(x) = f[x_0] + (x - x_0)f[x_0, x_1] + (x - x_0)(x - x_1)f[x_0, x_1, x_2] + \dots \\ \dots + (x - x_0) \cdots (x - x_{n-1})f[x_0, \dots, x_n] \quad (+E(x)) \quad (4.5)$$

Die sogenannten *Divided Differences* $f[x_0, \dots, x_k]$ sind dabei gegeben durch:

$$\begin{aligned} f[x_0] &= f(x_0) \\ f[x_0, x_1] &= \frac{f[x_1] - f[x_0]}{x_1 - x_0} \\ &\vdots \\ f[x_0, \dots, x_k] &= \frac{f[x_1, \dots, x_k] - f[x_0, \dots, x_{k-1}]}{x_k - x_0} \end{aligned} \quad (4.6)$$

Permutationen der Stützstellen bleiben ohne Einfluß auf das Interpolationspolynom.

Die Komponenten eines RS-Codewortes wurden oben (Definition 2.6) als Funktionswerte eines Polynoms vom Grad $K-1$ an äquidistanten Stellen auf dem Einheitskreis

$$c_i = \frac{1}{N} C(z^{-i}) = \frac{1}{N} \sum_{k=0}^{K-1} C_k \cdot (z^{-i})^k, \quad z = e^{j \frac{2\pi}{N}}$$

beschrieben. Daher bietet es sich an, eine Operatorschreibweise einzuführen:

$$\Delta^1 c_i = \frac{\frac{1}{N} C(z^{-(i+1)}) - \frac{1}{N} C(z^{-i})}{z^{-(i+1)} - z^{-i}} \quad (4.7)$$

$$\Delta^k c_i = \frac{\Delta^{k-1} c_{i+1} - \Delta^{k-1} c_i}{z^{-(i+k)} - z^{-i}}. \quad (4.8)$$

Analog zur Differentiation von Polynomen vom Grad $K-1$ gilt

$$\textbf{Satz 4.2} \quad f[c_i, \dots, c_{i+K}] = 0 \quad \text{bzw.} \quad \Delta^K c_i = 0$$

Diese Eigenschaft folgt sofort aus obigem Interpolationsansatz (Gleichung 4.5).

Beweis der Korrekturfähigkeit

Bei zufälligen kontinuierlichen Fehlern geht die Wahrscheinlichkeit gegen Null, daß die fehlerbehafteten Stellen des empfangenen Wortes genau einem Polynom vom Grad $K-1$ entsprechen. Betrachtet man nun das Schema der Erzeugung der Divided Differences in Bild 4.2, so erkennt man, daß eine K -te Divided Difference nur dann verschwindet, wenn $K+1$ Stützstellen zu einem Polynom vom Grad $K-1$ gehören. Sind $K+1 = N - M + 1$ Stellen fehlerfrei, so erhält man nach maximal $\binom{N}{K+1}$ Permutationen der Stützstellen eine Null an einer der K -ten Divided Differences.

Dies beweist die Korrekturfähigkeit analoger RS-Codes und zeigt gleichzeitig die Möglichkeit auf, ein ‘neues’ Syndrom zu definieren. Es eröffnen sich hierzu zwei Alternativen.

Zunächst kann man die K -ten Divided Differences direkt als Syndrom verwenden (siehe Bild 4.2). Sie erlauben das *sofortige Erkennen fehlerfreier Bereiche* der Länge $K+1$. Hieraus ist dann das gesamte Codewort rekonstruierbar. Dies bedeutet, daß ein eventuell sich anschließender Decodieralgorithmus entfallen kann, sobald das so definierte Syndrom eine Nullstelle aufweist. Im Falle maximaler Fehlerzahl $e_{max} = M-1$ führt ein Anteil von $N e_{max}! (N - e_{max})! / N! = N / \binom{N}{e_{max}}$ aller möglichen

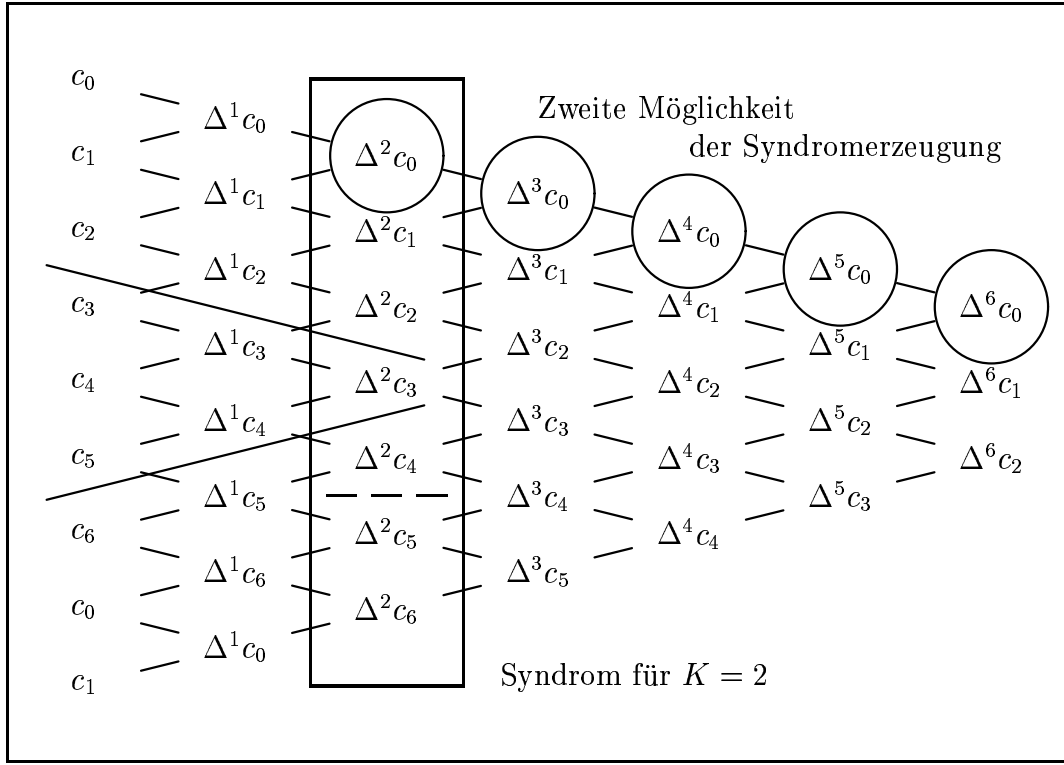


Abbildung 4.2: Newton-Schema zur Erzeugung der Divided Differences

Permutationen auf eine verschwindende Syndromstelle. Aus Aufwandsgründen ist jedoch der Einsatz von Permutationen zur Korrektur bei größerer Fehlerzahl nicht von Interesse (ebenso wie im vorigen Abschnitt).

Die zweite Alternative der Syndromdefinition geht von den Divided Differences aus, die auch in die Interpolationsvorschrift, Gleichung 4.5, eingehen, wenn man sie bis zum Grad $N - 1$ fortsetzt. Die verwendeten Elemente des Newton-Schemas zeigt ebenfalls Bild 4.2. Dieses Syndrom weist einfache Zusammenhänge zu einem Syndrom auf, das über eine ‘DFT’ im Sinne von *Generalized RS-Codes* gebildet wird:

$$\Psi_j = \sum_{i=0}^{N-1} \frac{r_i}{q_i'} \beta_i^j, \quad j = 0, 1, \dots, M - 1. \quad (4.9)$$

Mit $q_i' = 1$ und $\beta_i = z^i$ erhielte man die übliche DFT. Es läßt sich zeigen, daß $f[x_0, \dots, x_{n-1}]$ genau Ψ_0 entspricht. Die übrigen Syndrome Ψ_i folgen rekursiv aus den Divided Differences.

Der Zusammenhang mit dem verallgemeinerten DFT-Syndrom wurde nicht von dem Verfasser erarbeitet. Daher wird auf eine genaue Darstellung hier verzichtet. Stattdessen sei auf die Veröffentlichung von B. Dorsch [19] und die in Kürze erscheinende Arbeit meines Kollegen H. Schneider verwiesen.

Stichpunktartig sollen noch einige wichtige vorteilhafte Eigenschaften des Newton-Verfahrens zusammengestellt werden.

- Ausgelöschte (nicht oder zu unsicher empfangene) Stellen können im Newton-Schema einfach weggelassen werden. (Vergleiche hierzu das Vorgehen bei der Bildung des herkömmlichen DFT-Syndroms im Falle von Auslöschungen in z.B. [6], das doch etwas mehr Aufwand bedeutet.)
- Sukzessives Anfügen nachträglich empfangener Stellen ist leicht möglich — adaptive Codierung. (Eine vergleichende Betrachtung für das herkömmliche Syndrom wird ebenfalls Inhalt der Dissertation meines Kollegen sein.)
- Das Newton-Schema ist durch Parallel- und Pipeline-Strukturen darstellbar.

Auch wenn in diesem Abschnitt kein Decodieralgorithmus auf der Basis des Newton-Verfahrens vorgestellt werden konnte, so sind die hier gemachten Überlegungen sicherlich ein Denkansatz zur Entwicklung eines solchen.

Zum Abschluß dieses Abschnitts sollen noch einige Bemerkungen angefügt werden.

Parallelen der hier dargestellten Beweisidee zu derjenigen nach J.K. Wolf bestehen insofern, als $\Delta^K r(i) = 0$ gleichzeitig bedeutet, daß $\Delta^{K-1} r(i) = \Delta^{K-1} r(i+1)$. Anders ausgedrückt heißt das, daß zwei Entnahmen einer Anzahl von K Stellen die gleiche Divided Difference liefern müssen um $K+1$ fehlerfreie Stellen auszuweisen. Bei dem Beweis nach Wolf mußten mindestens zwei Entnahmen nach Neucodierung das gleiche Gesamtcodewort ergeben.

Vergleich Error Trapping – Newton-Verfahren

Permutationsdecodierer erreichen, wie der Name schon sagt, durch Permutationen, daß Fehlerstellen in den ‘Prüfteil’ eines Codewortes gelegt werden. Anwendung der Prüfmatrix \underline{H} liefert dann sofort das Fehlermuster (siehe z.B. [65]).

Mit Error Trapping bezeichnet man ein Decodierverfahren, das als Permutationen nur zyklische Verschiebungen zuläßt (Definition nach [65]). Es müssen daher K aufeinanderfolgende Stellen fehlerfrei sein. Er folgt daraus das maximale Fehlergewicht, bei dem alle Fehlermuster korrigiert werden können, zu

$$e_{max} = \left\lfloor \frac{N-1}{K} \right\rfloor < \frac{N}{K}. \quad (4.10)$$

Bei der Syndromerzeugung mit dem Newton-Verfahren ist die Decodierung dann leicht durchführbar, wenn ein fehlerfreier Bereich der Länge $K + 1$ vorhanden ist, was dann erfüllt ist, wenn

$$e < \frac{N}{K+1} \quad (4.11)$$

Das Newton-Verfahren erlaubt jedoch zusätzlich beliebige Permutationen, ohne auf Zugehörigkeit zur Automorphismengruppe achten zu müssen, wie dies bei der Permutationsdecodierung der Fall ist.

Mit dem hier vorgestellten interessanten Verfahren zur Syndromerzeugung werden die Betrachtungen zum ‘Umfeld’ analoger Codierung in Form von Interpolation und Approximation vorerst verlassen. Im nächsten und zentralen Kapitel dieser Arbeit folgen Untersuchungen zu den Eigenschaften analoger Codes bei Anwendung herkömmlicher Decodierverfahren.

Kapitel 5

Einfluß von ‘Hintergrundrauschen’ — Konditionierung des Decodierproblems

In diesem Abschnitt wird die Ermittlung der Fehlerstellen ausführlich untersucht. Die Fehlerstellen werden dabei auf herkömmliche Weise durch Auflösen des Schlüsselgleichungssystems 3.4 und Rücktransformation des Fehlerstellenpolynoms bestimmt.

Es wird im folgenden zwischen zu korrigierenden Fehlern, als Impulsfehler bezeichnet, und zusätzlichen Fehlern kleinerer Amplitude, die allen Codewortstellen überlagert sind (‘Hintergrundrauschen’), unterschieden. Der Einfluß dieses Hintergrundrauschens auf die Auffindbarkeit der Impulsfehler-Stellen soll hier Gegenstand genauerer Betrachtungen sein.

Zusätzlich wird in Abschnitt 5.3.2 eine besonders anschauliche und leicht verständliche Beschreibung des Berlekamp-Massey-Algorithmus angegeben, die insbesondere große didaktische Vorzüge gegenüber Masseys Darstellung besitzt. Viele, auch für die durchgeführten Konditionsbetrachtungen nützliche Eigenschaften, sind nur aus dieser Art der Formulierung leicht erkennbar.

5.1 Simulationsergebnisse

Ergebnisse einiger Simulationen sollen zunächst einen ersten Einblick in die Problematik vermitteln.

Wie schon angesprochen, steht die Ermittlung der Fehlerstellen im Vordergrund.

Nach Auflösen des Toeplitz-Systems werden jedoch, um es noch einmal zu betonen, nicht etwa die komplexen Nullstellen des Fehlerstellenpolynoms gesucht, sondern dieses wird, wie im diskreten Fall, einer DFT-Rücktransformation unterworfen. Eine Nullstellensuche wäre auch praktisch nicht durchführbar.

Anstelle der Nullstellen des rücktransformierten Fehlerstellenpolynoms, die die Fehlerstellen kennzeichneten, sind bei Vorhandensein von Hintergrundrauschen jedoch bestenfalls Betragsminima an diesen Stellen zu erwarten. Solange die Betragsminima auch wirklich den Fehlerstellen entsprechen, ist eine Korrektur möglich. Ein Maß für die Güte der Auffindbarkeit der Fehlerstellen sei wie folgt definiert:

Definition 5.1 *Ein Maß für die Güte der Fehlerauffindbarkeit sei gegeben durch*

$$Q = \frac{\min \{ |\gamma_i|_{\text{fehlerfreie Stellen}} \}}{\max \{ |\gamma_i|_{\text{Fehlerstellen}} \}}.$$

Ist $Q > 1$, so sind alle aufgetretenen Impulsfehler auffindbar, und $|\gamma_i|$ an Impulsfehler-freien Stellen unterschreitet nicht die Beträge an Impulsfehler-Stellen.

Es werden nun die Ergebnisse zweier Simulationen dargestellt, wobei im ersten Fall gleichverteiltes (in einem Intervall), im zweiten normalverteiltes Hintergrundrauschen zugrunde gelegt wurde.

5.1.1 Gleichverteiltes Hintergrundrauschen

Die Simulation mit in einem Intervall gleichverteiltem Hintergrundrauschen zeigt die Abhängigkeit des oben definierten Gütemaßes Q vom jeweils aufgetretenen Maximalwert des Rauschens (nicht der variierten Intervallgrenze) und der Anzahl der Impulsfehler e .

Für die Impulsfehler wurden eine konstante Amplitude $|B|$ und gleichverteilte Phase gewählt. Die Abbildungen zeigen jeweils 5000 Simulationsergebnisse.

Da ein doppelt logarithmischer Maßstab verwendet wurde, weist der linear verlaufende Mittelwert in den Darstellungen auf ein prinzipiell *hyperbolisches Verhalten* hin. Man könnte diese allmähliche, asymptotische Annäherung an $Q = 0$ mit steigender Rauschamplitude als Hinweis auf ein vergleichsweise stabiles Verhalten ansehen (kein abruptes Versagen).

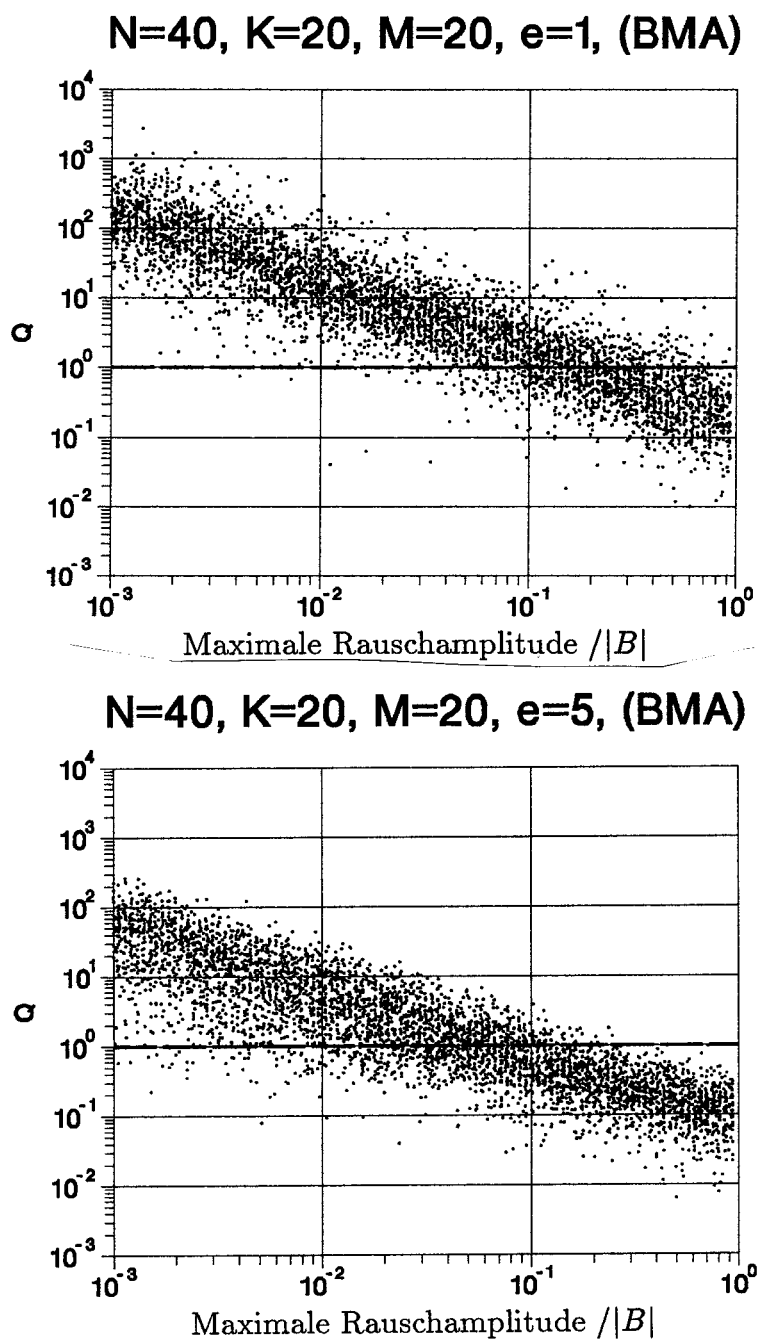
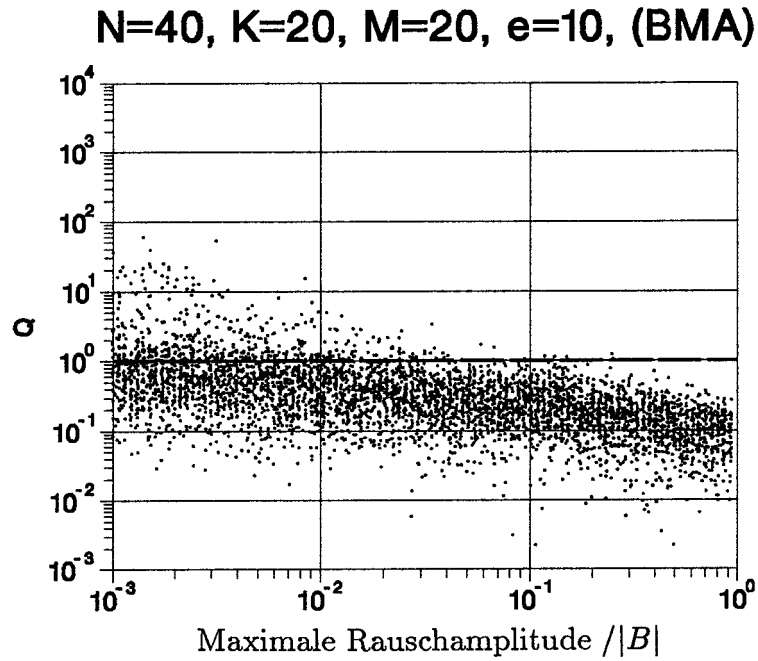


Abbildung 5.1: Das Gütemaß Q in Abhängigkeit der, auf die Impulsfehler-Amplitude $|B|$ normierten, maximalen Rauschamplitude am Beispiel $N = 40, K = 20$



Lineare Regression - N=40, K=20 (BMA)

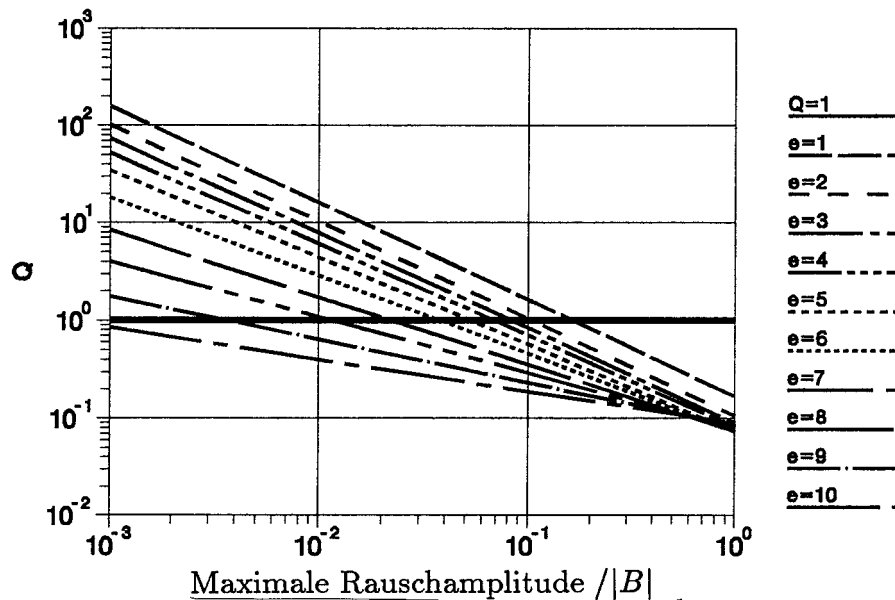


Abbildung 5.2: Das Gütemaß Q in Abhängigkeit der, auf die Impulsfehler-Amplitude $|B|$ normierten, maximalen Rauschamplitude am Beispiel $N = 40, K = 20$ – Fortsetzung –

5.1.2 Normalverteiltes Hintergrundrauschen

Normalverteiltes Hintergrundrauschen bereitet größere Schwierigkeiten für die Fehlerlokalisierung.

Die folgende Abbildung 5.3 zeigt die Abhängigkeit des Anteils auffindbarer Fehlerstellen von der Standardabweichung σ des zweidimensionalen normalverteilten Rauschens und der Impulsfehler-Anzahl e . Die Impulsfehler besitzen wieder konstante Amplitude $|B|$ und gleichverteilte Phase.

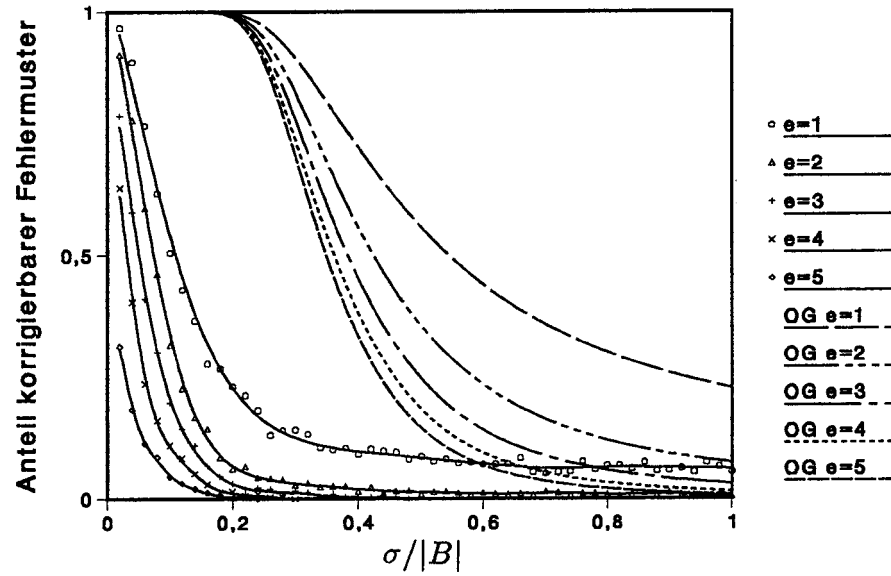


Abbildung 5.3: Anteil korrigierbarer Fehlermuster bei normalverteiltem Hintergrundrauschen in Abhängigkeit der auf $|B|$ normierten Standardabweichung σ

In das Diagramm wurden ebenfalls Obergrenzen für den Anteil korrigierbarer Fehlermuster (mit OG gekennzeichnet) eingetragen, die aus der Verteilung des Rauschens zu schließen sind. Zur Veranschaulichung betrachte man Bild 5.4.

Die Obergrenzen (im folgenden mit P_d bezeichnet) ergeben sich aus der Wahrscheinlichkeit, daß Impulsfehler-Stellen und lediglich durch Hintergrundrauschen gestörte Stellen noch unterscheidbar sind, d.h.

$$P_d = P(\min\{\mathbf{r}_i | i \in IF\} > \max\{\mathbf{r}_i | i \in \overline{IF}\}), \quad (5.1)$$

wobei IF wieder die Indexmenge der Impulsfehler, \overline{IF} die verbleibenden Stellen und \mathbf{r}_i die Zufallsvariable an Stelle i bezeichnet. Es sei kurz die Berechnung von P_d

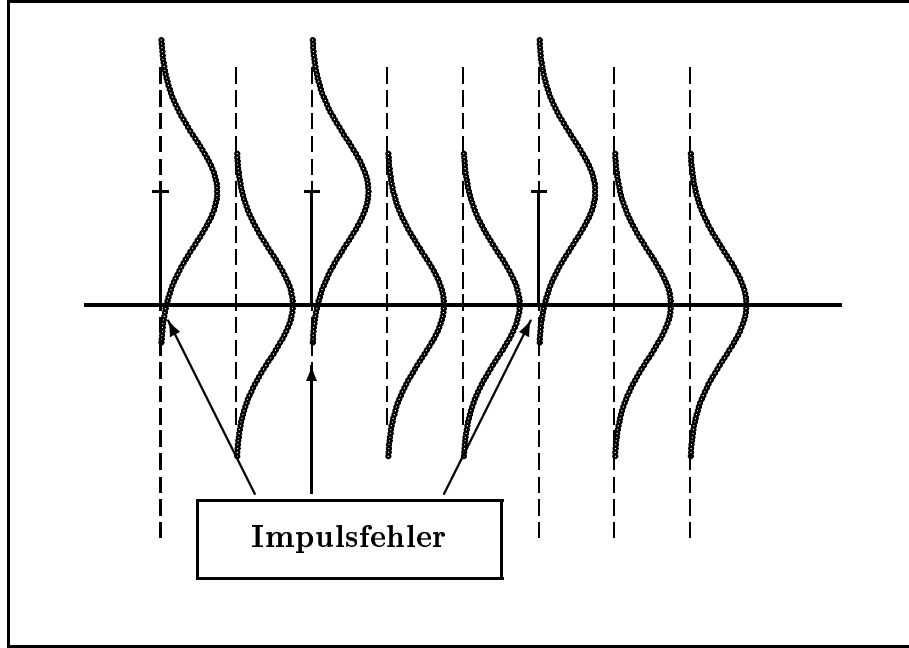


Abbildung 5.4: Illustration zur Herleitung der Obergrenzen in der vorigen Abbildung

angegeben:

$$\begin{aligned}
 P_d &= \int_{u=-\infty}^{\infty} P\left(\min\{\mathbf{r}_i | i \in IF\} > \max\{\mathbf{r}_i | i \in \overline{IF}\} \mid \min\{\mathbf{r}_i | i \in IF\} = u\right) \cdot \\
 &\quad \cdot f_{\mathbf{u}}(u) du = \int_{u=-\infty}^{\infty} P\left(u > \max\{\mathbf{r}_i | i \in \overline{IF}\}\right) \cdot f_{\mathbf{u}}(u) du = \\
 &\quad P\left(u > \max\{\mathbf{r}_i | i \in \overline{IF}\}\right) = \prod_{i \in \overline{IF}} P(u > \mathbf{r}_i) = \left(F_{\mathbf{r}_{\overline{IF}}}(u)\right)^{N-e}
 \end{aligned}$$

Berechnung von $f_{\mathbf{u}}(u)$:

$$P(\mathbf{u} < u) = P(\min\{\mathbf{r}_i | i \in IF\} < u) = 1 - P(\min\{\mathbf{r}_i | i \in IF\} > u)$$

$$\begin{aligned}
 \Rightarrow F_{\mathbf{u}}(u) &= 1 - \left[1 - F_{\mathbf{r}_{IF}}(u)\right]^e \\
 \Rightarrow f_{\mathbf{u}}(u) &= e \cdot \left[1 - F_{\mathbf{r}_{IF}}(u)\right]^{e-1} \cdot f_{\mathbf{r}_{IF}}(u)
 \end{aligned}$$

Es ergibt sich somit P_d zu:

$$P_d = \int_{u=-\infty}^{\infty} \left(F_{\mathbf{r}_{\overline{IF}}}(u) \right)^{N-e} \cdot e \cdot \left(1 - F_{\mathbf{r}_{IF}}(u) \right)^{e-1} \cdot f_{\mathbf{r}_{IF}}(u) du \quad (5.2)$$

$$F_{\mathbf{r}_{IF}}(u) = \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{u-1}{\sqrt{2}\sigma} \right) \right) \quad (5.3)$$

$$F_{\mathbf{r}_{\overline{IF}}}(u) = \frac{1}{2} \left(1 + \operatorname{erf} \left(\frac{u}{\sqrt{2}\sigma} \right) \right) \quad (5.4)$$

$$f_{\mathbf{r}_{IF}}(u) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(u-1)^2/(2\sigma^2)} \quad (5.5)$$

Die Simulationsergebnisse liegen vergleichsweise weit von der angegebenen Obergrenze entfernt. Dies spricht dafür, daß normalverteiltes Rauschen grundsätzlich ungünstige Auswirkungen auf das Korrekturverhalten hat.

Allgemein läßt sich sagen, daß die Fehlerkorrektur dann günstige Ergebnisse liefert, wenn der Impulscharakter der Störungen *deutlich* überwiegt. Wie Simulationen in Kapitel 8 belegen, ist dies für $N = 20$, $K = 10$ bereits dann der Fall, wenn die Amplitude der Impulsfehler um mindestens einen Faktor 10 bis 20 (abhängig von der Impulsfehler-Anzahl) über der Standardabweichung des Gauss'schen Rauschens liegt. Auch aus den zuvor angesprochenen Simulationen mit (in einem Intervall) gleichverteiltem Rauschen sind ähnliche Aussagen zu gewinnen.

Man könnte vielleicht annehmen, der Fall normalverteilten Rauschens wäre noch einer mathematischen Beschreibung zugänglich. Im Anhang A.3 wird jedoch gezeigt, daß insbesondere die Beschreibung der Lösung des Toeplitz-Systems durch Dichtefunktionen ungemein komplex und daher praktisch unlösbar ist.

In den Folgeabschnitten sollen nun die Fortpflanzungseffekte des überlagerten Hintergrundrauschens genauer mathematisch untersucht werden.

5.2 Konditionsbetrachtungen zur Decodierung

Bevor auf die Auswirkungen des Hintergrundrauschens mit der Hilfe von Konditionsbetrachtungen eingegangen wird, soll eine Veranschaulichung mit rekursiven Filtern versucht werden, die nahelegt, daß das untersuchte Decodierproblem grundsätzlich als kritisch bezüglich der Konditionierung einzuschätzen ist. Dies, obwohl die obigen Simulationen ein doch recht stabiles Verhalten nachweisen.

Fehlerstellensuche und rekursive Filter

Auf den ersten Blick scheinen die beiden Themen nicht viel gemein zu haben. Betrachtet man jedoch die Schlüsselgleichung 3.3, so entspricht diese ja bekanntlich (siehe [41]) einem rückgekoppelten Schieberegister gemäß Bild 5.5, welches man ebenso als rekursive Filterstruktur begreifen kann.

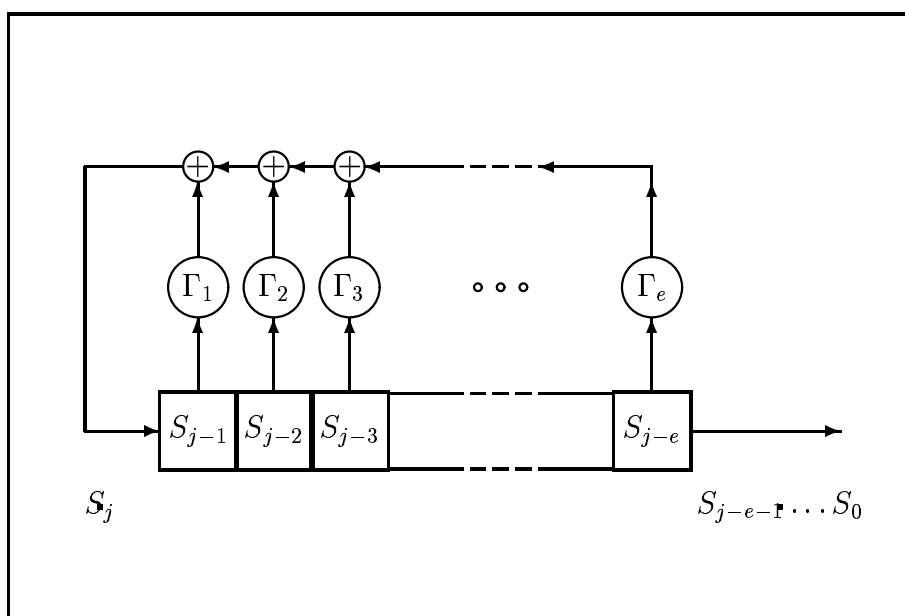


Abbildung 5.5: Schieberegisterveranschaulichung der Schlüsselgleichung

Betrachtet man das Fehlerstellenpolynom

$$\Gamma(1/x) = 1 + \sum_{i=1}^e \Gamma_i x^{-i}, \quad (5.6)$$

so sucht man Γ_i , so daß die Nullstellen, die sich aus den Fehlerstellen ergeben, auf dem Einheitskreis liegen, wenn kein Hintergrundrauschen vorhanden ist. Es wurde hierbei absichtlich die Substitution ' $x \rightarrow 1/x$ ' durchgeführt, um deutlicher den Aufbau des Schieberegisters aus Verzögerungselementen hervortreten zu lassen. Nullstellen auf dem Einheitskreis ändern ihre Lage dabei nur bezüglich des Winkels. Der Kehrwert des Polynoms stellt den Nenner der Übertragungsfunktion eines rekursiven digitalen Filters dar. Die genannten Nullstellen sind dann die Pole dieser Übertragungsfunktion. Pole auf dem Einheitskreis bedeuten *Grenzstabilität* (siehe hierzu Bild 5.6). Bei Vorhandensein von zusätzlichem Rauschen werden die Pole zumindest nahe dem Einheitskreis liegen, was rein anschaulich einen Hinweis auf zu erwartendes 'problematisches' Verhalten gibt.

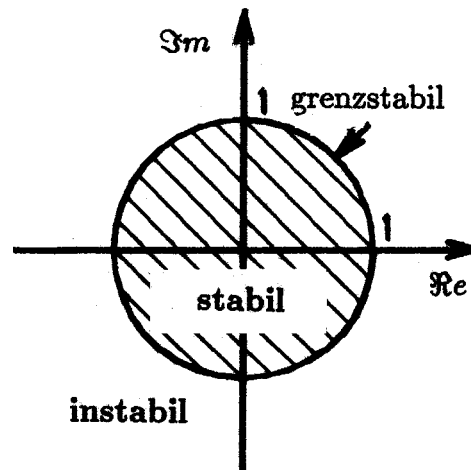


Abbildung 5.6: Bereiche für Stabilität und Instabilität der z-Transformierten

Nach diesen Vorbemerkungen wird nun die Konditionierung der Fehlerlokalisierung detailliert untersucht.

Zur Ermittlung der Fehlerstellen erfolgt (wie schon in Kapitel 3 erläutert) in drei Schritten:

- I** Transformation des empfangenen Zeitbereichsworts (\vec{r}) in den Frequenzbereich (\vec{R})

– *DFT* –

$$\vec{R} = \underline{Z}_{N \times N} \vec{r} \quad \underline{Z}_{N \times N} = \begin{pmatrix} 1 & 1 & 1 & \cdots \\ 1 & z^1 & z^2 & \\ 1 & z^2 & z^4 & \\ \vdots & & & \end{pmatrix} \quad (5.7)$$

- II** Entnahme des Syndroms ($S_0, S_1, \dots, S_{2e-1}$) aus \vec{R} und Lösen des Hankel- (Toeplitz-) Systems

$$\begin{pmatrix} S_{2e-2} & S_{2e-3} & \cdots & S_{e-1} \\ S_{2e-3} & & & \vdots \\ \vdots & & & \vdots \\ S_{e-1} & \cdots & \cdots & S_0 \end{pmatrix} \begin{pmatrix} \Gamma_1 \\ \Gamma_2 \\ \vdots \\ \Gamma_e \end{pmatrix} = - \begin{pmatrix} S_{2e-1} \\ S_{2e-2} \\ \vdots \\ S_e \end{pmatrix}. \quad (5.8)$$

In abgekürzter Schreibweise

$$\underline{S} \cdot \vec{\Gamma'} = -\vec{s}.$$

Man beachte, daß hier, entgegen Gleichung 3.5, $\vec{\Gamma'}$ ohne führende Komponente ‘1’ verwendet wird.

- III** Rücktransformation des Fehlerstellenpolynoms bzw. des entsprechenden Vektors $\vec{\Gamma}$ (mit ‘1’ als erster Komponente erweitert) in den Zeitbereich $\vec{\gamma}$

– *DFT*⁻¹ –

$$\vec{\gamma} = \underline{Z}_{N \times (e+1)}^{-1} \vec{\Gamma} \quad \underline{Z}_{N \times (e+1)}^{-1} = \frac{1}{N} \left\{ \overbrace{\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & z^{-1} & z^{-2} & & \\ 1 & z^{-2} & z^{-4} & & \\ \vdots & & & & \\ \vdots & & & & \end{pmatrix}}^{e+1} \right\} N \quad (5.9)$$

Diese drei Operationen werden nun genauer mit dem Ziel untersucht, Abschätzungen für die Beziehung zwischen relativem Fehler $\rho_{\vec{\gamma}}$ des Fehlerstellenvektors $\vec{\gamma}$ und relativem Fehler $\rho_{\vec{r}}$ des Zeitbereichs-Codewortes \vec{r} zu gewinnen. Die relativen Fehler werden dabei unter Zuhilfenahme von L_∞ -Normen¹ (Maximumnormen) definiert:

$$\rho_{\vec{\gamma}} = \frac{\|\Delta\vec{\gamma}\|_\infty}{\|\vec{\gamma}_i\|_\infty}, \quad \rho_{\vec{r}} = \frac{\|\Delta\vec{r}\|_\infty}{\|\vec{r}_i\|_\infty} \quad (5.10)$$

Der Index i ('ideal') steht hier und im folgenden für Impulsfehler-behaftete Vektoren ohne Hintergrundrauschen.

Zunächst werden die Normen der DFT -Matrizen angegeben, da diese später benötigt werden.

$$\begin{aligned} \|\underline{Z}_{N \times N}\|_\infty &= N & \|\underline{Z}_{N \times N}^{-1}\|_\infty &= \frac{N}{N} = 1 \\ \|\underline{Z}_{N \times (e+1)}^{-1}\|_\infty &= \frac{e+1}{N} & \|\underline{Z}_{(e+1) \times N}\|_\infty &= N \end{aligned} \quad (5.11)$$

Erster Schritt (I)

Als Voraussetzung für die folgenden Konditionsbetrachtungen sei angenommen, daß die Amplitude des Hintergrundrauschens wesentlich kleiner als diejenige der Impulsfehler ist. Der Vektor des Rauschens im Zeitbereich werde repräsentiert durch $(\epsilon \cdot \vec{e})$. Die DFT liefert

$$\vec{R} = \underline{Z}_{N \times N} \vec{r} = \underline{Z}_{N \times N} (\vec{r}_i + \epsilon \vec{e}) \quad (5.12)$$

$$\implies \Delta \vec{R} = \underline{Z}_{N \times N} \cdot \epsilon \vec{e} \quad (5.13)$$

$$\implies \|\Delta \vec{R}\| \leq \|\underline{Z}_{N \times N}\| \cdot \epsilon \cdot \|\vec{e}\|. \quad (5.14)$$

Weiterhin gilt

$$\vec{r}_i = \underline{Z}_{N \times N}^{-1} \vec{R}_i \quad (5.15)$$

$$\implies \|\vec{r}_i\| \leq \|\underline{Z}_{N \times N}^{-1}\| \cdot \|\vec{R}_i\| \implies \frac{\|\underline{Z}_{N \times N}^{-1}\| \cdot \|\vec{R}_i\|}{\|\vec{r}_i\|} \geq 1. \quad (5.16)$$

¹Vektornorm: $\|\vec{a}\|_\infty = \max_\nu |a_\nu|$, Matrixnorm: $\|\underline{A}\|_\infty = \max_\nu \left(\sum_\mu |A_{\nu,\mu}| \right)$

Die Gleichungen 5.14 und 5.16 ergeben zusammen

$$\| \Delta \vec{R}_i \| \leq \| \underline{Z}_{N \times N} \| \cdot \epsilon \cdot \| \vec{e} \| \frac{\| \underline{Z}_{N \times N}^{-1} \| \cdot \| \vec{R}_i \|}{\| \vec{r}_i \|} \quad (5.17)$$

$$\Rightarrow \frac{\| \Delta \vec{R} \|}{\| \vec{R}_i \|} \leq \underbrace{\| \underline{Z}_{N \times N} \| \cdot \| \underline{Z}_{N \times N}^{-1} \|}_{\kappa\{\underline{Z}_{N \times N}\}} \cdot \frac{\epsilon \| \vec{e} \|}{\| \vec{r}_i \|}, \quad (5.18)$$

oder in Kurzform geschrieben

$$\rho_{\vec{R}} \leq \kappa\{\underline{Z}_{N \times N}\} \cdot \rho_{\vec{r}}, \quad (5.19)$$

wobei $\kappa\{\underline{Z}_{N \times N}\}$ die Konditionszahl der *DFT*-Matrix $\underline{Z}_{N \times N}$ bezeichnet.

Zweiter Schritt (II)

Betrachtet man das Hankel- (Toeplitz-) System

$$\underline{S} \cdot \vec{\Gamma}' = -\vec{s}, \quad (5.20)$$

so erhält man unter Verwendung von [21], S. 25

$$\frac{\| \Delta \vec{\Gamma}' \|}{\| \vec{\Gamma}'_i \|} \leq \underbrace{\| \underline{S} \| \cdot \| \underline{S}^{-1} \|}_{\kappa\{\underline{S}\}} \cdot (\rho_A + \rho_B) \left(+O(\epsilon^2) \right) \quad (5.21)$$

mit

$$\rho_A = \frac{\| \Delta \underline{S} \|}{\| \underline{S}_i \|} \quad \text{and} \quad \rho_B = \frac{\| \Delta \vec{s} \|}{\| \vec{s}_i \|}.$$

Berücksichtigt man, daß \vec{s} einen Ausschnitt aus \vec{R} darstellt, so folgt unter Verwendung der L_∞ -Norm

$$\rho_B = \frac{\| \Delta \vec{s} \|_\infty}{\| \vec{s}_i \|_\infty} = \frac{\| \Delta \vec{s} \|_\infty}{\| \vec{R}_i \|_\infty} \cdot \frac{\| \vec{R}_i \|_\infty}{\| \vec{s}_i \|_\infty} \leq \rho_{\vec{R}} \cdot \frac{\| \vec{R}_i \|_\infty}{\| \vec{s}_i \|_\infty}. \quad (5.22)$$

Entsprechend ergibt sich für ρ_A

$$\rho_A = \frac{\| \Delta \underline{S} \|_\infty}{\| \underline{S}_i \|_\infty} = \frac{\| \Delta \underline{S} \|_\infty}{\| \vec{R}_i \|_\infty} \cdot \frac{\| \vec{R}_i \|_\infty}{\| \underline{S}_i \|_\infty} \leq e \cdot \rho_{\vec{R}} \cdot \frac{\| \vec{R}_i \|_\infty}{\| \underline{S}_i \|_\infty}. \quad (5.23)$$

Dritter Schritt (III)

Wie unter **I**, führt die Rücktransformation des Vektors des Fehlerstellenpolynoms

$$\vec{\gamma} = \underline{Z}_{N \times (e+1)}^{-1} \vec{\Gamma} \quad (5.24)$$

zu der Beziehung

$$\frac{\|\Delta \vec{\gamma}\|}{\|\vec{\gamma}_i\|} \leq \|\underline{Z}_{N \times (e+1)}^{-1}\| \cdot \|\underline{Z}_{(e+1) \times N}\| \cdot \frac{\|\Delta \vec{\Gamma}\|}{\|\vec{\Gamma}_i\|}. \quad (5.25)$$

berücksichtigt man, daß $\|\Delta \vec{\Gamma}\| = \|\Delta \vec{\Gamma}'\|$ und $\|\vec{\Gamma}_i\| = \max\{1, \|\vec{\Gamma}'_i\|\}$, so ergibt sich der relative Fehler des Fehlerstellenvektors im Zeitbereich aus den Gleichungen 5.18, 5.21 und 5.25 zu (überall L_∞ -Normen)

$$\begin{aligned} \frac{\|\Delta \vec{\gamma}\|}{\|\vec{\gamma}_i\|} &\leq \underbrace{\frac{\|\vec{\Gamma}'_i\|}{\max\{1, \|\vec{\Gamma}'_i\|\}}}_{\leq 1} \cdot \|\underline{Z}_{N \times (e+1)}^{-1}\| \cdot \|\underline{Z}_{(e+1) \times N}\| \cdot \|\underline{S}\| \cdot \|\underline{S}^{-1}\| \cdot \\ &\cdot \left(\frac{\|\vec{R}_i\|}{\|\vec{s}_i\|} + e \frac{\|\vec{R}_i\|}{\|\underline{S}_i\|} \right) \cdot \|\underline{Z}_{N \times N}\| \cdot \|\underline{Z}_{N \times N}^{-1}\| \cdot \frac{\epsilon \|\vec{e}\|}{\|\vec{r}_i\|} = \end{aligned} \quad (5.26)$$

$$\leq \frac{e+1}{N} \cdot N \cdot \kappa\{\underline{S}\} \cdot \left(\frac{\|\vec{R}_i\|}{\|\vec{s}_i\|} + e \frac{\|\vec{R}_i\|}{\|\underline{S}_i\|} \right) \cdot N \cdot 1 \cdot \frac{\epsilon \|\vec{e}\|}{\|\vec{r}_i\|}.$$

$$\Rightarrow \rho_{\vec{\gamma}} \leq (e+1) \cdot N \cdot \kappa\{\underline{S}\} \cdot \left(\frac{\|\vec{R}_i\|_\infty}{\|\vec{s}_i\|_\infty} + e \frac{\|\vec{R}_i\|_\infty}{\|\underline{S}_i\|_\infty} \right) \cdot \rho_{\vec{r}} \quad (5.27)$$

Konditionsbetrachtungen liefern zwar einerseits sehr nützliche Abschätzungen, sind jedoch bekanntlich auch vergleichsweise grob. Simulationenergebnisse, auf deren Wiedergabe hier verzichtet werden soll, wiesen die Gültigkeit folgender Näherung nach:

$$\boxed{\rho_{\vec{\gamma}} \leq 2 \cdot (e+1) \cdot N \cdot \kappa\{\underline{S}\} \cdot \rho_{\vec{r}}} \quad (5.28)$$

Von besonderer Bedeutung in Gleichung 5.28 ist natürlich die Konditionszahl der Syndrommatrix $\kappa\{\underline{S}\}$.

In den folgenden Abschnitten wird nun untersucht, inwieweit sich diese während des Ablaufs des rekursiven Berlekamp-Massey-Algorithmus (beispielhaft als eines der wichtigsten Decodierverfahren) abschätzen läßt. Dies unter anderem auch, um zu erkennen, welche Größe sinnvollerweise in ein Abbruchkriterium eingehen sollte.

Die genannten Abschätzungen lassen sich insbesondere mit einer im Folgeabschnitt vorgestellten neuen Beschreibung des Algorithmus leicht gewinnen.

5.3 Neue Beschreibung des Berlekamp-Massey-Algorithmus

Wie schon erwähnt, dient dieser Abschnitt einerseits zur Vorbereitung der Abschätzung der Konditionierung der rekursiv im Berlekamp-Massey-Algorithmus verwendeten Sub-Toeplitz-Systeme. Andererseits wird hier jedoch eine Art der Erläuterung des BMA angegeben, die sowohl didaktische Vorzüge gegenüber der Masseyschen Darstellung besitzt, als auch wichtige Eigenschaften sofort erkennen läßt.

Die Operationen im BMA werden als logische Folge erkennbar, ohne jeweils auf umständliche Beweisansätze zurückgreifen zu müssen.

Zur Orientierung sei jedoch zunächst Masseys Beschreibung als Schieberegisterproblem kurz wiedergegeben.

5.3.1 Das Schieberegisterproblem nach Massey

In Kapitel 5.2 wurde bereits der Zusammenhang zwischen Schlüsselgleichung 3.3 und einem rückgekoppelten Schieberegister aufgezeigt.

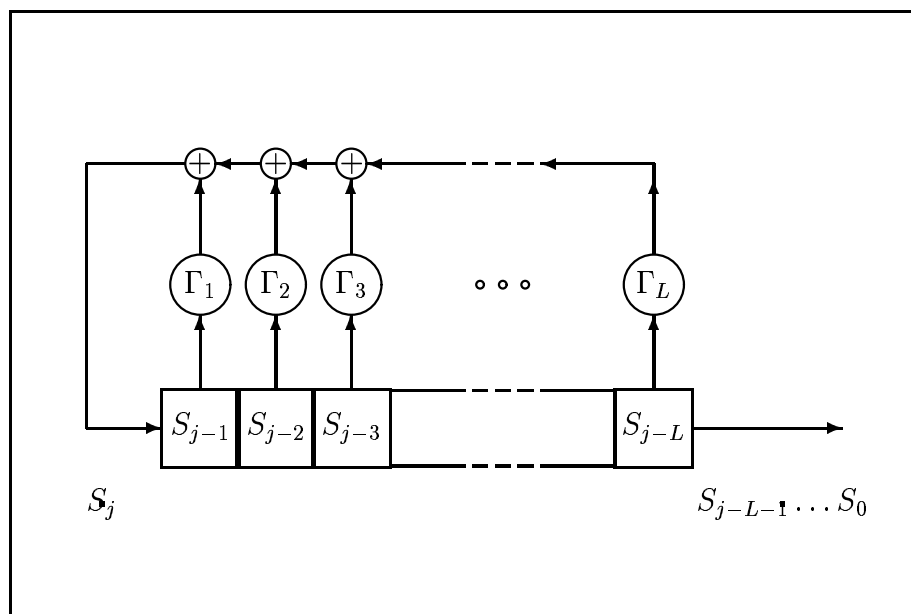
Mit einem Fehlerstellenpolynom des Grades L

$$\Gamma(x) = 1 + \Gamma_1 x^1 + \Gamma_2 x^2 + \dots + \Gamma_L x^L \quad (5.29)$$

folgt die Schlüsselgleichung zu

$$S_j = - \sum_{i=1}^L \Gamma_i S_{j-i}, \quad j = L, L+1, \dots, M-1. \quad (5.30)$$

Das Schieberegister sei hier noch einmal zur Veranschaulichung in Bild 5.7 wiedergegeben.



Gesucht wird ein Schieberegister kürzester Länge, d.h. ein Fehlerstellenpolynom geringsten Grades L , das die Schlüsselgleichung erfüllt. Daher wird im BMA sukzessive, beginnend bei Null ($L = 0$), die Länge des Schieberegisters vergrößert, bis L der wirklich vorhandenen Fehlerzahl e entspricht.

$$d_n = S_n + \sum_{i=1}^{L_n} \Gamma_i S_{n-i} \quad (5.31)$$

Die neuen Schieberegisterkoeffizienten ergeben sich aus den vorherigen durch eine Addition mit einem früheren Koeffizientensatz *vor der letzten Längenänderung*. Diese Koeffizienten werden so weit in die von dem Schieberegister vorgegebene Richtung

verschoben, daß sie wieder auf dieselben Syndromstellen treffen, wie zuvor, als sie die Diskrepanz d_m erzeugten. Entsprechend ergibt sich mit dem so verschobenen Koeffizientensatz wieder die gleiche Diskrepanz d_m , die zur vorigen Längenänderung führte. Die Rekursionsbeziehung des BMA für die Koeffizienten des Fehlerstellenpolynoms in der Form

$$\Gamma_{n+1}(x) = \Gamma_n(x) - \frac{d_n}{d_m} x^{n-m} \Gamma_m(x) \quad (5.32)$$

- $n - m$: Verschiebelänge
- d_n : aktuelle Diskrepanz
- d_m : Diskrepanz vor der letzten Längenänderung

zwingt die Diskrepanz zu Null.

Eine Längenänderung ist nur erforderlich, wenn $d_n \neq 0 \wedge 2L_n \leq n$ ist.

Der Beweis, daß obige Rekursion das Schieberegister geringster Länge liefert, findet sich in der Veröffentlichung von Massey [41], aus welcher ebenfalls im folgenden eine programmähnliche Beschreibung des Algorithmus zitiert wird. Die dort verwendeten Bezeichnungen wurden in die hier eingeführten übersetzt ($t = n - m$).

| | |
|--|--------|
| <ol style="list-style-type: none"> 1) $1 \rightarrow \Gamma(x) \quad 1 \rightarrow \Gamma_v(x) \quad 1 \rightarrow t$ $0 \rightarrow L \quad 1 \rightarrow d_m \quad 0 \rightarrow n$ 2) IF $n = M - 1$, STOP. OTHERWISE COMPUTE $d_n = s_n + \sum_{i=1}^L \gamma_i s_{n-i}.$ 3) IF $d_n = 0$, THEN $t + 1 \rightarrow t$, GO TO 6). 4) IF $d_n \neq 0$ AND $2L > n$, THEN $\Gamma(x) - d_n d_m^{-1} x^t \Gamma_v(x) \rightarrow \Gamma(x)$ $t + 1 \rightarrow t$ GO TO 6). 5) IF $d_n \neq 0$ AND $2L \leq n$, THEN $\Gamma(x) \rightarrow T(x)$ $\Gamma(x) - d_n d_m^{-1} x^t \Gamma_v(x) \rightarrow \Gamma(x)$ $n + 1 - L \rightarrow L$ $T(x) \rightarrow \Gamma_v(x)$ $d_n \rightarrow d_m$ $1 \rightarrow t.$ 6) $n + 1 \rightarrow n$ RETURN TO 2). | (5.33) |
|--|--------|

Nach dieser kurzen Darstellung von Masseys Veranschaulichung der Schlüsselgleichung wird im Folgeabschnitt eine neue Beschreibung vorgestellt, die noch um vieles anschaulicher ist. Ein Beweis, daß der BMA das Schieberegister geringster Länge ermittelt, kann wegen dieser besonderen Anschaulichkeit, insbesondere im Fall nicht-singulärer Untermatrizen, völlig entfallen.

5.3.2 Neue Matrixbeschreibung

Wie die Überschrift bereits vermuten läßt, wird das Schlüsselgleichungssystem zunächst in Matrixschreibweise angegeben.

$$S_j = - \sum_{i=1}^e \Gamma_i S_{j-i} \quad (5.34)$$

entspricht dem Gleichungssystem

$$(1, \Gamma_1, \Gamma_2, \dots, \Gamma_e) \begin{pmatrix} S_e & S_{e+1} & \cdot & \cdot & S_{2e} & \cdot & S_{M-1} \\ S_{e-1} & S_e & \ddots & & \cdot & & \cdot \\ \cdot & S_{e-1} & \ddots & \ddots & \cdot & & \cdot \\ \cdot & & \ddots & \ddots & S_{e+1} & & \cdot \\ S_0 & \cdot & \cdot & S_{e-1} & S_e & \cdot & S_{M-1-e} \end{pmatrix} = (0, \dots, 0), \quad (5.35)$$

wobei die Matrix, wie schon mehrfach erwähnt, eine Toeplitz-Struktur besitzt.

Es soll nun im folgenden der BMA als logische Folge aus der Matrixstruktur und der Forderung nach dem Fehlerstellenpolynom geringst möglichen Grades hergeleitet werden.

Der Algorithmus beginnt somit bei $L = 0$ unter Verwendung der kleinstmöglichen Submatrix

$$(1) \cdot (S_0) = (S_0) =: (d_0^I). \quad (5.36)$$

Die erste auftretende Diskrepanz d_0^I ist somit gleich S_0 . Selbst wenn hier die Diskrepanz verschwinden sollte, ist es nötig, das nächst größere Subsystem auf seine Gültigkeit zu überprüfen.

Da das Fehlerstellenpolynom geringsten Grades gesucht wird, ist es naheliegend, den Vektor $\vec{\Gamma}$ durch rechtsseitiges Anfügen einer Null zu erweitern, um das nächst

größere Subsystem zu untersuchen. Dies vergrößert den Grad des zugehörigen Fehlerstellenpolynoms nicht. Es folgt das Gleichungssystem

$$(1, 0) \begin{pmatrix} S_1 & S_2 \\ S_0 & S_1 \end{pmatrix} = (d_0^{II}, d_1^{II}). \quad (5.37)$$

Gesucht wird eine Lösung, für welche die rechte Seite verschwindet. Sind d_0^{II} und d_1^{II} ungleich Null, so ist es zumindest möglich, die erste Stelle zu Null zu zwingen, wenn auch d_0^I von Null verschieden war.

Verwendet man anstelle des Vektors (1) in Gleichung 5.36 nunmehr einen durch Null links erweiterten Vektor (0, 1) zusammen mit der 2×2 -Matrix, so erscheint auf der rechten Seite als erste Komponente wieder die Diskrepanz $d_0^I = S_0$.

Durch Linearkombination der beiden von links bzw. rechts mit Null ergänzten Vektoren ist es nun möglich, die erste Komponente der rechten Seite zu Null zu zwingen:

$$[(1, 0) - \frac{d_0^{II}}{d_0^I}(0, 1)] \cdot \begin{pmatrix} S_1 & S_2 \\ S_0 & S_1 \end{pmatrix} = (0, d_1^{III}) \quad (5.38)$$

Ist $d_1^{III} \neq 0$, so ist die Submatrix nicht singulär ($\det \underline{S}_1 \neq 0$) und eine Längenänderung, zusammen mit der Betrachtung der wiederum nächst größeren Submatrix, wird nötig. Auch wenn d_1^{III} gleich Null sein sollte, ist die Überprüfung der nächst größeren Submatrix auf Singularität unerlässlich.

An dieser Stelle sei darauf hingewiesen, daß eine Längenänderung im Sinne des BMA nur gegeben ist, wenn die Diskrepanz ungleich Null war, und damit die nächste Operation eine Erweiterung des Lösungsvektors mit Komponenten ungleich Null bedeutet.

Nachdem die ersten Schritte des Algorithmus (und damit die Initialisierung) dargestellt wurden, sollen nun die Hauptoperationen, die auf der Toeplitz-Struktur beruhen, klar herausgestellt werden.

Mit dem bisher Gesagten ist bereits deutlich geworden, daß das Erweitern des Lösungsvektors mit Nullen von links oder rechts, zusammen mit einer Vergrößerung der betrachteten Sub-Toeplitz-Matrix, von besonderem Interesse ist. Diese Erweiterungen stellen auch die zentralen Operationen dar. Im folgenden wird ihr Einfluß auf eine beliebige rechte Seite genauer aufgezeigt.

Erweitern durch Anfügen von Nullen rechts:

Erweitern des Vektors $\vec{\Gamma}$ durch Anfügen von Nullen rechts, führt zu einer Verschiebung der rechten Seite um eine Stelle nach links und dem Anfügen zweier neuer Komponenten rechts.

$$(1, \Gamma_1, \Gamma_2, \dots, \Gamma_l) \cdot \begin{pmatrix} S_l & S_{2l} \\ & \ddots \\ S_0 & S_l \end{pmatrix} = (\rho_0, \rho_1, \rho_2, \dots, \rho_l) \quad (5.39)$$

$$(1, \Gamma_1, \Gamma_2, \dots, \Gamma_l, 0) \cdot \begin{pmatrix} S_{l+1} & S_{2(l+1)} \\ & \ddots \\ S_0 & S_{l+1} \end{pmatrix} = (\rho_1, \rho_2, \dots, \rho_l, \rho_{l+1}, \rho_{l+2}) \quad (5.40)$$

Erweitern durch Anfügen von Nullen links:

Erweitern des Vektors $\vec{\Gamma}$ durch Anfügen von Nullen links, erhält die rechte Seite und fügt lediglich eine neue Komponente rechts hinzu.

$$(1, \Gamma_1, \Gamma_2, \dots, \Gamma_l) \cdot \begin{pmatrix} S_l & S_{2l} \\ & \ddots \\ S_0 & S_l \end{pmatrix} = (\rho_0, \rho_1, \rho_2, \dots, \rho_l) \quad (5.41)$$

$$(0, 1, \Gamma_1, \Gamma_2, \dots, \Gamma_l) \cdot \begin{pmatrix} S_{l+1} & S_{2(l+1)} \\ & \ddots \\ S_0 & S_{l+1} \end{pmatrix} = (\rho_0, \rho_1, \rho_2, \dots, \rho_l, \rho_{l+1}) \quad (5.42)$$

Es wird nun die Eigenschaft genutzt, daß ein zuvor aufgetretener Lösungsvektor (*vor der letzten Längenänderung*) auf der rechten Seite ab einer gewissen Komponente nach links nur Nullen erzeugt. Das Ergänzen mit Nullen von links bewahrt diese rechte Seite bis auf eine neu hinzukommende Komponente rechts. Der sich hieraus ergebende Vektor sei $\vec{\Gamma}_v$.

Der aktuelle Lösungsvektor $\vec{\Gamma}$ wird nun so lange mit Nullen von rechts erweitert und hiermit die rechte Seite des Gleichungssystems nach links geschoben, bis die erste nicht verschwindende Komponente genau an der Stelle zu liegen kommt, die durch die erste Komponente ungleich Null unter Verwendung von $\vec{\Gamma}_v$ gegeben ist.

Die beiden Gleichungssysteme besitzen dann die Struktur

$$(0, \dots, 0, 1, \Gamma_{v1}, \Gamma_{v2}, \dots) \cdot (\cdot) = (\underbrace{0, \dots, 0}_{\text{Länge}}, d_{vi}, d_{vi+1}, d_{vi+2}, \dots) \quad (5.43)$$

$$(1, \Gamma_1, \Gamma_2, \dots, 0, \dots, 0) \cdot (\cdot) = (\overbrace{0, \dots, 0}^{d_i}, d_i, d_{i+1}, d_{i+2}, \dots) \quad (5.44)$$

Mit einer Linearkombination beider Lösungsvektoren zwingt man die erste nichtverschwindende Komponente d_i zu Null. Der neue Lösungsvektor folgt hiermit zu

$$\vec{\Gamma} := (1, \Gamma_1, \Gamma_2, \dots, 0, \dots, 0) - \underbrace{\frac{d_i}{d_{vi}}}_{=d_n/d_m} (0, \dots, 0, 1, \Gamma_{v1}, \Gamma_{v2}, \dots) \quad (5.45)$$

Diese Beziehung entspricht der oben angegebenen Rekursion 5.32.

Die erste nicht verschwindende Komponente der rechten Seite des Gleichungssystems *vor der letzten Längenänderung* gleicht einer ‘Marke’, die durch rechtsseitiges Anfügen von Nullen an den aktuellen Lösungsvektor erreicht werden muß. Was bei der neuen Beschreibung als Längenänderung im Sinne des BMA anzusehen ist und warum es sinnvoll (bzw. zwingend) ist, den Vektor vor der letzten Längenänderung zu verwenden, wird in den ‘Bemerkungen’ am Ende dieses Abschnitts erläutert.

Zwei Beispiele² seien zur Veranschaulichung herangezogen. Das erste Beispiel könnte auch als Einführung dienen, da hier die grundlegende Vorgehensweise des BMA gut ersichtlich ist. Beispiel 2 behandelt den speziellen Fall singulärer Untermatrizen. Man erkennt dort, daß es nötig sein kann, die Syndrommatrix noch um unbekannte Komponenten zu erweitern, um den Algorithmus fortzuführen. Diese werden jedoch bei nachfolgenden Schritten nicht mehr verwendet.

Beiden Beispielen liegt eine ungerade Anzahl von Syndromelementen zugrunde. Der Leser wird jedoch feststellen, daß die letzte Syndromkomponente eigentlich nicht benötigt wird. Sie kann lediglich einer zusätzlichen Kontrolle dienen, ob alle Fehler gefunden wurden.

Die Beispiele lassen auch erkennen, daß eine Längenänderung im Sinne des BMA nur erfolgt, wenn $d_n \neq 0 \wedge 2L_n \leq n$.

²Den Beispielen liegt die umgekehrte Definition der *DFT*, verglichen mit derjenigen in Kapitel 2.2, zugrunde.

Beispiel 1

GF(7), primitives Element: 5, Syndromlänge: 5

Fehlervektor: $(1, 0, 0, 0, 1, 0)$ Syndrom: $(5, 2, 4, 5, 2)$

$$(1, C_1, C_2) \begin{pmatrix} 4 & 5 & 2 \\ 2 & 4 & 5 \\ 5 & 2 & 4 \end{pmatrix} \stackrel{!}{=} (0, 0, 0)$$

$$(1) \cdot (5) = (5)$$

$$(1, 0) \begin{pmatrix} 2 & 4 \\ 5 & 2 \end{pmatrix} = (2, 4)$$

$$(1, 0) - \frac{2}{5}(0, 1) = (1, 1)$$

$$(1, 1) \begin{pmatrix} 2 & 4 \\ 5 & 2 \end{pmatrix} = (0, 6)$$

$$(1, 1, 0) \begin{pmatrix} 4 & 5 & 2 \\ 2 & 4 & 5 \\ 5 & 2 & 4 \end{pmatrix} = (6, 2, 0)$$

$$(1, 1, 0) - \frac{6}{5}(0, 0, 1) = (1, 1, 3)$$

$$(1, 1, 3) \begin{pmatrix} 4 & 5 & 2 \\ 2 & 4 & 5 \\ 5 & 2 & 4 \end{pmatrix} = (0, 1, 5)$$

$$(1, 1, 3) - \frac{1}{6}(0, 1, 1) = (1, 2, 4)$$

$$(1, 2, 4) \begin{pmatrix} 4 & 5 & 2 \\ 2 & 4 & 5 \\ 5 & 2 & 4 \end{pmatrix} = (0, 0, 0)$$

$$C(x) = 1 + 2x + 4x^2$$

Fehlerstellen: 0 and 4

$$C(5^0) = 1 + 2 + 4 = 0$$

$$C(5^4) = 1 + 2 \cdot 2 + 4 \cdot 4 = 0$$

Beispiel 2

GF(11), primitives Element: 6, Syndromlänge: 7

Fehlervektor: $(1, 0, 1, 0, 4, 0, 0, 0, 0, 0)$ Syndrom: $(5, 8, 4, 7, 4, 5, 8)$

$$(1, C_1, C_2, C_3) \begin{pmatrix} 7 & 4 & 5 & 8 \\ 4 & 7 & 4 & 5 \\ 8 & 4 & 7 & 4 \\ 5 & 8 & 4 & 7 \end{pmatrix} \stackrel{!}{=} (0, 0, 0, 0)$$

$$(1) \cdot (5) = (5)$$

$$(1, 0) \begin{pmatrix} 8 & 4 \\ 5 & 8 \end{pmatrix} = (8, 4)$$

$$(1, 0) - \frac{8}{5}(0, 1) = (1, 5)$$

$$(1, 5) \begin{pmatrix} 8 & 4 \\ 5 & 8 \end{pmatrix} = (0, 0)$$

$$(1, 5, 0) \begin{pmatrix} 4 & 7 & 4 \\ 8 & 4 & 7 \\ 5 & 8 & 4 \end{pmatrix} = (0, 5, 6)$$

$$(1, 5, 0, 0) \begin{pmatrix} 7 & 4 & 5 & 8 \\ 4 & 7 & 4 & 5 \\ 8 & 4 & 7 & 4 \\ 5 & 8 & 4 & 7 \end{pmatrix} = (5, 6, 3, 0)$$

$$(1, 5, 0, 0) - \frac{5}{5}(0, 0, 0, 1) = (1, 5, 0, 10)$$

$$(1, 5, 0, 10) \begin{pmatrix} 7 & 4 & 5 & 8 \\ 4 & 7 & 4 & 5 \\ 8 & 4 & 7 & 4 \\ 5 & 8 & 4 & 7 \end{pmatrix} = (0, 9, 10, 4)$$

$$(1, 5, 0, 10) - \frac{9}{5}(0, 1, 5, 0) = (1, 1, 2, 10)$$

$$(1, 1, 2, 10) \begin{pmatrix} 7 & 4 & 5 & 8 \\ 4 & 7 & 4 & 5 \\ 8 & 4 & 7 & 4 \\ 5 & 8 & 4 & 7 \end{pmatrix} = (0, 0, 8, 3)$$



$$(1, 1, 2, 10, 0) \begin{pmatrix} 4 & 5 & 8 & ? & ? \\ 7 & 4 & 5 & 8 & ? \\ 4 & 7 & 4 & 5 & 8 \\ 8 & 4 & 7 & 4 & 5 \\ 5 & 8 & 4 & 7 & 4 \end{pmatrix} = (0, 8, 3, ?, ?)$$

$$(1, 1, 2, 10, 0) - \frac{8}{5}(0, 0, 1, 5, 0) = (1, 1, 7, 2, 0)$$

$$(1, 1, 7, 2, 0) \begin{pmatrix} 4 & 5 & 8 & ? & ? \\ 7 & 4 & 5 & 8 & ? \\ 4 & 7 & 4 & 5 & 8 \\ 8 & 4 & 7 & 4 & 5 \\ 5 & 8 & 4 & 7 & 4 \end{pmatrix} = (0, 0, 0, ?, ?)$$

$$(1, 1, 7, 2) \begin{pmatrix} 7 & 4 & 5 & 8 \\ 4 & 7 & 4 & 5 \\ 8 & 4 & 7 & 4 \\ 5 & 8 & 4 & 7 \end{pmatrix} = (0, 0, 0, 0)$$

$$C(x) = 1 + x + 7x^2 + 2x^3$$

Fehlerstellen: 0, 2 and 4

$$C(6^0) = C(1) = 0$$

$$C(6^2) = C(3) = 0$$

$$C(6^4) = C(9) = 0$$

Nach diesen Beispielen seien noch einige Bemerkungen zum tieferen Verständnis der neuen Beschreibung des BMA angefügt.

Bemerkungen

1.) Zur Längenänderung im Sinne des BMA

Eine Längenänderung (des zugehörigen Schieberegisters) im Sinne des BMA ist nicht schon durch das bloße rechtsseitige Nullanfügen zum Überprüfen der nächsten Diskrepanz gegeben, sondern erst, wenn diese Diskrepanz ungleich Null war, und sich eine Operation zur Änderung des Lösungsvektors anschließt. Aber selbst eine Änderungsoperation muß nicht zwangsläufig auch immer eine Längenänderung mit sich bringen. Dies ist deutlich an Beispiel 2 zu erkennen, als unbekannte Matrixelemente

eingeführt wurden, und trotz Änderungsoperation der Grad des zugehörigen Fehlerstellenpolynoms gleich blieb. Die hier eliminierte Diskrepanz war jedoch hier eigentlich auch diejenige an zweitletzter Stelle des vormaligen 4×4 -Systems. Eine Längenänderung wird immer erforderlich, wenn nur noch die rechte Komponente der rechten Seite ungleich Null ist, denn mit hier drei wählbaren Komponenten des Lösungsvektors (neben der '1' an erster Stelle) ist es zumindest möglich, drei Komponenten der rechten Seite zu Null zu zwingen (im singulären Fall alle). Das Erweitern mit einer zusätzlichen Null rechts und das Einführen von unbekannten Matrix-Koeffizienten war nur zur konsequenten Weiterentwicklung nach der neuen Beschreibung nötig. Als Gegenbeispiel für eine nötige Längenänderung ohne Graderhöhung ist in Anhang A.4 das Originalbeispiel von Massey, zusammen mit der neuen Beschreibung wiedergegeben. Hier wird im letzten Schritt eine Längenänderung durchgeführt, da im Falle des vorherigen 3×3 -Systems die rechte Seite bis auf die rechte Komponente bereits zu Null gezwungen war.

Man erkennt, daß es bei der neuen Beschreibung unter Umständen etwas schwieriger sein kann, Längenänderungen zu erkennen. Zum Ermitteln des Fehlerstellenpolynoms eines korrigierbaren Fehlermusters ist dies aber auch nicht von so großer Bedeutung, da nach Durchlauf des BMA auf jeden Fall das Polynom geringsten Grades gefunden wird, das der Syndromfolge und damit der Toeplitz-Matrix entspricht; durch die Nullstellen sind dann auch die Fehlerstellen bestimmt. Auf jeden Fall sind Längenänderungen durch Prüfen der Bedingung $d_n \neq 0 \wedge 2L_n \leq n$ festzustellen, sollte man etwas unsicher sein.

Im Fall nichtsingulären Untermatrizen ist es jedoch besonders einfach, anzugeben, ob eine Längenänderung eingetreten ist. Sobald nur noch die rechte Komponente der rechten Seite ungleich Null ist, folgt zwangsläufig eine Vergrößerung der Länge. Hier ist die Längenänderung auch direkt an der Länge des Lösungsvektors abzulesen.

2.) *Zur Eindeutigkeit des zugehörigen Schieberegisters bei Vorgabe einer bestimmten Folgenlänge*

Eindeutig ist der Lösungsvektor immer dann, wenn maximal nur noch eine Komponente rechts ungleich Null verbleibt. Im nichtsingulären Fall ist dann nach obiger Bemerkung eine Längenänderung erforderlich.

Nach Massey ergibt sich ein eindeutiges Rückkoppelpolynom des Schieberegisters dann, wenn

$$2L_n \leq n.$$

3.) *Verwendung des Vektors 'vor der letzten Längenänderung'*

Bei Betrachtung der neuen Beschreibung wird klar, daß zur Änderung des Lösungsvektors nur die Verwendung desjenigen Vektors ‘vor der letzten Längenänderung’ in Betracht kommt. Das Anfügen mit Nullen rechts bewirkt ein Verschieben der rechten Seite nach links. Der Vektor vor der letzten Längenänderung bildet als rechte Seite die erste erreichbare ‘Marke’, d.h. die am weitesten rechts liegende erste Komponente ungleich Null. Bei Verwendung weiter zurückliegender Zwischenergebnisse, läge diese ‘Marke’ weiter links, was die Notwendigkeit mit sich brächte, die rechte Seite der aktuellen Lösung weiter nach links zu schieben. Dies wiederum bedeutete weiteres rechtsseitiges Null-Anfügen am aktuellen Lösungsvektor und entsprechendes linksseitiges Null-Anfügen am gewählten vormaligen Vektor. Nach Linearkombination der beiden Vektoren würde dies i.a. zu einem höheren Grad des, dem Ergebnisvektor entsprechenden, Polynoms führen. Gemäß der Forderung nach dem Fehlerstellenpolynom geringsten Grades, ist dies jedoch nicht zulässig.

Ein formaler Beweis, daß die Verwendung des Vektors ‘vor der letzten Längenänderung’ zum Schieberegister geringster Länge und zum Fehlerstellenpolynom geringsten Grades führt, findet sich in dem Aufsatz von Massey.

4.) *Der BMA führt auf das Schieberegister geringster Länge*

Sind alle Sub-Toeplitz-Matrizen bis zum Erreichen der, zur gesuchten Endlänge des Vektors passenden, Untermatrix nichtsingulär (was später für komplexe Koeffizienten vorausgesetzt wird), so wird aus der neuen Beschreibung sofort ersichtlich, daß der BMA den Lösungsvektor geringster Länge liefert. Diese stimmt (nach den Linearkombinationsoperationen) in diesem Fall mit der Länge des zugehörigen Schieberegisters überein. In Bemerkung 1 wurde bereits ausgeführt, daß eine Längenänderung nur erfolgt, wenn nur noch eine Komponente der rechten Seite ungleich Null verbleibt. Eine größere Anzahl von Nullstellen ist auch bei gegebener Länge nicht möglich. Die Länge des Lösungsvektors wird schrittweise auch nur um Eins erhöht. Ist die Endlösung gefunden (Erreichen des ersten singulären Subsystems), so werden lediglich zur Überprüfung der nächsten Diskrepanzen noch Nullen rechts an den Lösungsvektor angefügt, die aber, wie bereits bemerkt, keine Längenänderung mehr für das zugehörige Schieberegister bedeuten.

Sukzessive wird somit im BMA jede Vektorlänge (Schieberegisterlänge) erprobt, bis die Lösung gefunden ist. Bei jedem Schritt wird versucht, auf der rechten Seite eine Maximalanzahl von Nullen zu generieren. Damit ist für den nichtsingulären Fall bewiesen, daß das Schieberegister geringster Länge gefunden wird.

Sind Untermatrizen singulär, auch wenn sie noch nicht die Lösung des Gesamtsystems liefern, so ist zumindest aus der Beschreibung zu erkennen, daß nach Erreichen des singulären Subsystems zunächst nur Nullen von rechts an den Lösungsvektor an-

gefügt werden, was vorerst keine Längenänderung (des Schieberegisters) bedeutet. Weiterhin wird nach Überwinden einer Singularität, mit Erreichen der neuen Länge, wieder eine rechte Seite erzeugt, bei welcher nur maximal die rechte Komponente ungleich Null ist. (Dies läßt sich leicht aus etwas größeren Beispielen ersehen.) Ab dieser Stelle gilt dann wieder die Argumentation des nichtsingulären Falles.

Daß die Größe des bei Singularitäten auftretenden Längensprungs grundsätzlich zum Schieberegister geringster Länge führt, ist jedoch wohl leichter Masseys formalem Beweis zu entnehmen.

Bei der hier vorgestellten Beschreibung folgt die Längenänderung zwangsweise aus der Struktur des Algorithmus.

5.) Zur Initialisierung des BMA

Betrachtet man Masseys Darstellung etwas genauer, so stellt man einen geringfügigen Unterschied in den ersten Operationen zu der hier eingeführten Beschreibung fest. Zur Erläuterung sei daher der Anfang des ersten Beispiels noch einmal – Masseys Aufsatz folgend – wiedergegeben:

$$\begin{array}{ll} \Gamma(x) = 1 & t = 1 \\ \Gamma_v(x) = 1 & L = 0 \\ & d_m = 1 \\ & n = 0 \end{array}$$

$$(1) \cdot (5) = (5)$$

$$(1, 0) - \frac{5}{1}(0, 1) = (1, 2) \quad (\star)$$

$$\begin{array}{ll} \Gamma(x) = 1 + 2x & t = 1 \\ \Gamma_v(x) = 1 & L = 1 \\ & d_m = 5 \\ & n = 1 \end{array}$$

$$(1, 2) \begin{pmatrix} 2 & 4 \\ 5 & 2 \end{pmatrix} = (5, 1)$$

$$(1, 2) - \frac{5}{5}(0, 1) = (1, 1)$$

$$\begin{array}{ll} \Gamma(x) = 1 + x & t = 2 \\ \Gamma_v(x) = 1 & L = 1 \\ & d_m = 5 \\ & n = 2 \end{array}$$

⋮

Vergleicht man beide Darstellungsformen, so erkennt man, daß die Operation (\star) eigentlich völlig unnötig ist. Andererseits hat dies jedoch den Vorteil, daß L immer die Länge des zugehörigen Schieberegisters liefert, was nicht mehr der Fall wäre, wenn der Schritt wie folgt ausgeblendet würde:

$$\begin{array}{c}
 \vdots \\
 5) \quad \text{IF } d \neq 0 \text{ AND } 2L \leq N, \text{ THEN} \\
 \quad C(D) \rightarrow T(D) \\
 \quad \underline{\text{IF } N \neq 0 \text{ THEN } C(D) - db^{-1}D^x B(D) \rightarrow C(D)} \\
 \quad N + 1 - L \rightarrow L \\
 \quad T(D) \rightarrow B(D) \\
 \quad d \rightarrow b \\
 \quad 1 \rightarrow x. \\
 \vdots
 \end{array}$$

Um die Anschaulichkeit der neuen Beschreibung noch einmal aufzuzeigen, wird im folgenden noch der Fall der *Auslöschungs- und Fehlerkorrektur* behandelt (siehe hierzu [6]). Auslöschungen können bei der Decodierung als Fehler an bekannter Stelle gesehen werden. Analog zum Fehlerstellenpolynom wird ein Auslöschungspolynom

$$\Lambda(x) = \prod_{i \in \mathcal{IE}} (x - z^{-i}), \quad (5.46)$$

definiert, wobei \mathcal{IE} die Indexmenge der Ausfallstellen bezeichnet ($|\mathcal{IE}| = a_E$).

Mit der neuen Beschreibung ist leicht ersichtlich, daß der Algorithmus nun mit dem Sub-Gleichungssystem

$$(1, \Lambda_1, \Lambda_2, \dots, \Lambda_{a_E}) \cdot \begin{pmatrix} S_{a_E} & & S_{2a_E} \\ & \ddots & \\ S_0 & & S_{a_E} \end{pmatrix} = (\rho_0, \rho_1, \rho_2, \dots, \rho_{a_E}) \quad (5.47)$$

beginnt. D.h. der Vektor $\vec{\Gamma}$ wird mit $\vec{\Lambda}$ initialisiert. Es folgt dann das Erweitern des Lösungsvektors mit Null von rechts ($\rightarrow \Gamma(x)$) und von links ($\rightarrow \Gamma_v(x)$) und so fort. Die Nullstellen des Auslöschungspolynoms $\Lambda(x)$ bleiben dabei erhalten, wenn man den BMA damit initialisiert. Bei der Initialisierung geht $\Lambda(x)$ sowohl in $\Gamma(x)$ als auch in $\Gamma_v(x)$ ein, woraus folgt, daß $\Lambda(x)$ als gemeinsamer Faktor bei Linearkombinationen der beiden Polynome nicht verändert wird.

Die erste nutzbare Diskrepanz in obigem Gleichungssystem ist ρ_0 . Die Syndromlänge, die zur Bestimmung der restlichen Fehler verbleibt, ist damit $M - a_E$. Die Anzahl noch korrigierbarer Fehler ist dann, wie bekannt, $\left\lfloor \frac{M-a_E}{2} \right\rfloor$. Die Verkürzung des nutzbaren Syndroms ist hier sofort erkennbar, was sonst einer Herleitung durch Faltung bedurfte (wie z.B. in der zuvor zitierten Literaturstelle).

Im nächsten Abschnitt dient die neue Beschreibung der Abschätzung der Konditionierung der Sub-Syndrommatrizen im BMA.

5.4 Abschätzung der Konditionierung der rekursiv im BMA auftretenden Sub-Systeme

In Abschnitt 5.2 wurde klar, daß für die Konditionierung der gesamten Decodierung die Konditionszahl der Syndrommatrix als wesentlich bestimmende Größe eingeht.

Exemplarisch werden im folgenden die rekursiv im BMA auftretenden Sub-Toeplitz-Matrizen untersucht. Der BMA wurde einerseits deshalb gewählt, weil er einen der wichtigsten Decodieralgorithmen darstellt, andererseits, weil Konditionsbetrachtungen hierfür noch nicht durchgeführt wurden. Dies liegt z.T. einfach in der Tatsache begründet, daß der BMA bislang nur für Probleme in endlichen Zahlkörpern eingesetzt wurde. Andererseits ist die im vorigen Abschnitt eingeführte neue Beschreibung des BMA von grundlegender Bedeutung für die Konditionsabschätzungen. Für Toeplitz-Algorithmen, die insbesondere in der digitalen Signalverarbeitung Anwendung finden, wie z.B. dem Levinson-Durbin-Algorithmus (siehe Abschnitt 7.1), dessen Struktur um einiges übersichtlicher ist, finden sich Betrachtungen zur Konditionierung bereits in der Literatur (z.B. [16]).

Die hier gemachten Aussagen beschränken sich nicht nur auf die Anwendung bei analogen Codes, sie gelten vielmehr für jede reelle oder komplexe Toeplitz-Problematik.

Von besonderem Interesse ist die (nach Wissen des Autors) erstmals erkannte Eigenschaft des BMA, die betrachteten Sub-Toeplitz-Matrizen durch eine sogenannte *triangular 'square root'-factorization* (nach [16], deutscher Begriff dem Verfasser nicht bekannt) zu zerlegen. Hiermit wird ersichtlich, daß rekursive Zwischenergebnisse keineswegs nutzlos sind, wie so oft angenommen.

Um den BMA auf komplexe Toeplitz-Systeme anwenden zu können, muß lediglich eine geringfügige Änderung vorgenommen werden: die Abfrage auf Diskrepanz gleich Null in 5.33 muß durch eine Schwellabfrage gemäß

$$\text{IF } |d_n| < d_S, \text{ THEN}$$

ersetzt werden. Wie zunächst vielleicht anschaulich naheliegend, besteht ein enger Zusammenhang zwischen dieser Abfrage bzw. der dort untersuchten Diskrepanz und der Konditionierung der Sub-Toeplitz-Matrix. Dies wird am Ende dieses Abschnitts noch klarer ersichtlich.

Nimmt man nun an, $|d_n|$ sei grundsätzlich größer als d_S , so beschreibt der BMA einen regelmäßigen Ablauf, wobei die Schritte 4) und 5) in 5.33 jeweils im Wechsel durchlaufen werden. Es wird somit im folgenden der Fall betrachtet, daß die Abfrage nach der Diskrepanz gänzlich entfällt.

Bei Durchsicht der Beispiele im vorigen Abschnitt erkennt man, daß vor einer Längenänderung, die nun nach jedem zweiten Durchlauf – jedem geradzahligen – erfolgt, das betrachtete Sub-Toeplitz-System folgende Form aufweist:

$$(1, \Gamma_1, \Gamma_2, \dots, \Gamma_l) \begin{pmatrix} S_l & & S_{2l} \\ & \ddots & \\ S_0 & & S_l \end{pmatrix} = (0, \dots, 0, d_{2l}) \quad (5.48)$$

$$\Leftrightarrow \underbrace{\begin{pmatrix} S_{2l} & S_l \\ S_l & S_0 \end{pmatrix}}_{=: \underline{S}_l} \begin{pmatrix} 1 \\ \Gamma_1 \\ \vdots \\ \Gamma_l \end{pmatrix} = \begin{pmatrix} d_{2l} \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \quad (5.49)$$

wobei d_{2l} die Diskrepanz vor einer Längenänderung bezeichnet, und der Vektor $(1, \Gamma_1, \dots, \Gamma_l)$ aus den entsprechenden Koeffizienten des Fehlerstellenpolynoms besteht. Sind $\Gamma_{i,j}$, $j = 1, \dots, i$ die zu d_{2i} zugehörigen Komponenten des Fehlerstellenvektors, so folgt hieraus (L kennzeichnet die gerade erreichte Länge)

$$\underline{S}_L \cdot \underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ \Gamma_{L,1} & 1 & 0 & 0 \\ \vdots & & \ddots & 0 \\ \Gamma_{L,L} & \Gamma_{L-1,L-1} & \cdots & 1 \end{pmatrix}}_{=: \underline{\Gamma}_L} = \begin{pmatrix} d_{2L} & \cdot & \cdot & \cdot \\ 0 & \ddots & \cdot & \cdot \\ 0 & 0 & d_2 & \cdot \\ 0 & 0 & 0 & d_0 \end{pmatrix} \quad (5.50)$$

$$\underbrace{\begin{pmatrix} 1 & \Gamma_{L,1} & \cdots & \Gamma_{L,L} \\ 0 & 1 & & \vdots \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{=: \underline{\Gamma}_L^T} \cdot \underline{\mathcal{S}}_L = \begin{pmatrix} d_{2L} & 0 & 0 & 0 \\ \cdot & \ddots & 0 & 0 \\ \cdot & \cdot & d_2 & 0 \\ \cdot & \cdot & \cdot & d_0 \end{pmatrix} \quad (5.51)$$

$$\Rightarrow \underline{\Gamma}_L^T \cdot \underline{\mathcal{S}}_L \cdot \underline{\Gamma}_L = \begin{pmatrix} d_{2L} & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & d_2 & 0 \\ 0 & 0 & 0 & d_0 \end{pmatrix}. \quad (5.52)$$

Es ergibt sich die schon angesprochene *triangular ‘square root’-factorization*

$$\boxed{\underline{\mathcal{S}}_L^{-1} = \underline{\Gamma}_L \underline{D}_L^{-1} \underline{\Gamma}_L^T \quad \text{oder} \quad \underline{\mathcal{S}}_L = (\underline{\Gamma}_L^T)^{-1} \underline{D}_L \underline{\Gamma}_L^{-1}} \quad (5.53)$$

$$\underline{\Gamma}_L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ \Gamma_{L,1} & 1 & 0 & 0 \\ \vdots & & \ddots & 0 \\ \Gamma_{L,L} & \Gamma_{L-1,L-1} & \cdots & 1 \end{pmatrix} \quad \underline{\Gamma}_L^T = \begin{pmatrix} 1 & \Gamma_{L,1} & \cdots & \Gamma_{L,L} \\ 0 & 1 & & \vdots \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (5.54)$$

$$\underline{D}_L = \begin{pmatrix} d_{2L} & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & d_2 & 0 \\ 0 & 0 & 0 & d_0 \end{pmatrix} \quad \underline{D}_L^{-1} = \begin{pmatrix} 1/d_{2L} & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 \\ 0 & 0 & 1/d_2 & 0 \\ 0 & 0 & 0 & 1/d_0 \end{pmatrix}. \quad (5.55)$$

In diese gehen somit die rekursiv auftretenden Zwischenlösungen des Fehlerstellenvektors mit den zugehörigen Diskrepanzen ein.

Mit Hilfe der Faktorisierung kann man leicht Grenzen für $\| \underline{\mathcal{S}}_L^{-1} \|_\infty$, $\| \underline{\mathcal{S}}_L \|_\infty$ und die Konditionszahl $\kappa\{\underline{\mathcal{S}}_L\}$ angeben. Betrachtet man die erste Zeile von $\underline{\mathcal{S}}_L^{-1}$ und die letzte Zeile von $\underline{\mathcal{S}}_L$, so erhält man

$$\| \underline{\mathcal{S}}_L^{-1} \|_\infty \geq \frac{1}{|d_{2L}|} (1 + |\Gamma_{L,1}| + \cdots + |\Gamma_{L,L}|) \quad (5.56)$$

$$\| \underline{\mathcal{S}}_L \|_\infty \geq |d_0| \quad (5.57)$$

$$\kappa\{\underline{S}_L\} \geq \frac{|d_0|}{|d_{2L}|}(1 + |\Gamma_{L,1}| + \cdots + |\Gamma_{L,L}|) \geq \frac{|d_0|}{|d_{2L}|} \quad (5.58)$$

Eine Obergrenze liefert

$$\kappa\{\underline{S}_L\} \leq \kappa\{\underline{\Gamma}_L\} \kappa\{\underline{\Gamma}_L^T\} \frac{\max |d_{2i}|}{\min |d_{2i}|}. \quad (5.59)$$

Zwischenergebnisse im BMA ermöglichen es also, sofort Aussagen über die Konditionszahl der zugrundeliegenden Sub-Matrix zu machen. Insbesondere bestätigt sich, daß die Diskrepanz dabei von entscheidender Bedeutung ist.

Interessant ist, daß auch andere Algorithmen, wie Levinson-Durbin-Algorithmus (siehe [16]) und Cholesky-Verfahren, auf ähnliche Faktorisierungen führen.

Bemerkungen

Zum Schluß des Kapitels sei noch einmal betont, daß die hier untersuchten analogen RS-Codes bzw. deren Korrektoreigenschaften auf (analogen) Hamming-, nicht etwa auf Euklidischen Distanzen beruhen.

Es wurden hier keine Rechengenauigkeiten einbezogen. Die Betrachtungen wurden vorerst auf den Einfluß von zusätzlich überlagertem Hintergrundrauschen bei den Empfangssymbolen beschränkt.

Rechengenauigkeitseffekte werden im Folgekapitel untersucht.

Kapitel 6

Einfluß der Rechenungenauigkeit beim BMA

Dieses Kapitel vermittelt einen Eindruck, wie stark sich Rechenungenauigkeiten, hervorgerufen durch Darstellung der Operanden mit einer endlichen Stellenzahl, bei der Ermittlung der Fehlerstellen auswirken. Die Untersuchung erfolgt am Beispiel des BMA. Es sollen dabei insbesondere die Parameter herausgestellt werden, die als bestimmende Größen für die Fortpflanzung von Rechenungenauigkeiten zu sehen sind.

Die Fehlereinflüsse bei Fest- oder Gleitkomma-Arithmetik lassen sich wie folgt modellieren (siehe z.B. [21] und [16]):

Festkomma-Arithmetik (*fixed point* – gekennzeichnet durch fx):

$$\text{fx}(a + b) = a + b \quad (6.1)$$

$$\text{fx}(a \cdot b) = a \cdot b + \mu_{fx} \quad (6.2)$$

$$\text{fx}(a/b) = a/b + \nu_{fx} \quad (6.3)$$

Gleitkomma-Arithmetik (*floating point* – gekennzeichnet durch fl):

$$\text{fl}(a + b) = (a + b)(1 + \sigma_{fl}) \quad (6.4)$$

$$\text{fl}(a \cdot b) = (a \cdot b)(1 + \mu_{fl}) \quad (6.5)$$

$$\text{fl}(a/b) = (a/b)(1 + \nu_{fl}) . \quad (6.6)$$

Zur Untersuchung des BMA ist es zunächst nötig, die Rekursionsvorschriften in einer komprimierten Form wiederzugeben. Diese ergibt sich unter der Voraussetzung, daß die Diskrepanz d_i grundsätzlich ungleich Null sei und mit Unterdrückung der

ersten Änderung des Fehlerstellenpolynoms, die sich in Abschnitt 5.3.2 ohnehin als überflüssig erwies, zu:

$$j = 0, 1, 2, \dots \quad d_j = S_j + \sum_{k=1}^{[j/2]} \Gamma_{j-1,k} S_{j-k} \quad (6.7)$$

$$j = 1, 3, 5, \dots \quad \Gamma_{j,i} = \Gamma_{j-1,i} - \frac{d_j}{d_{j-1}} \Gamma_{j-2,i-1} \quad (6.8)$$

$$j = 2, 4, 6, \dots \quad \Gamma_{j,i} = \Gamma_{j-1,i} - \frac{d_j}{d_{j-2}} \Gamma_{j-3,i-2} . \quad (6.9)$$

Bezeichnet man mit \tilde{d}_j respektive $\tilde{\Gamma}_{j,i}$ die mit Rechenungenauigkeiten behafteten Größen und die entsprechenden Rechenfehler mit δ_j bzw. $\varsigma_{j,i}$, so folgt

$$\tilde{d}_j = d_j + \delta_j \quad (6.10)$$

$$\tilde{\Gamma}_{j,i} = \Gamma_{j,i} + \varsigma_{j,i} . \quad (6.11)$$

Die Rekursionen für die fehlerbehafteten Größen können dann wie folgt aufgeschrieben werden:

$$\tilde{d}_j = S_j + \sum_{k=1}^{[j/2]} \tilde{\Gamma}_{j-1,k} S_{j-k} + \zeta_j \quad (6.12)$$

$$\tilde{\Gamma}_{j,i} = \tilde{\Gamma}_{j-1,i} - \frac{\tilde{d}_j}{\tilde{d}_{j-1}} \tilde{\Gamma}_{j-2,i-1} + \xi_{j,i} , \quad j = 2n - 1 \quad (6.13)$$

$$\tilde{\Gamma}_{j,i} = \tilde{\Gamma}_{j-1,i} - \frac{\tilde{d}_j}{\tilde{d}_{j-2}} \tilde{\Gamma}_{j-3,i-2} + \xi_{j,i} , \quad j = 2n , \quad n = 1, 2, \dots , \quad (6.14)$$

wobei ζ_j und $\xi_{j,i}$ die jeweils hinzukommenden Rechenungenauigkeiten kennzeichnen.

Hieraus folgen sofort Beziehungen für die Fehlerfortpflanzung

$$\delta_j = \sum_{k=1}^{[j/2]} \varsigma_{j-1,k} S_{j-k} + \zeta_j \quad (6.15)$$

$$\varsigma_{j,i} = \varsigma_{j-1,i} - \frac{d_j}{d_{j-1}} \varsigma_{j-2,i-1} - \left(\frac{\delta_j}{d_{j-1}} - \frac{d_j}{d_{j-1}^2} \delta_{j-1} \right) \Gamma_{j-2,i-1} + \xi_{j,i} \quad (6.16)$$

$$\varsigma_{j,i} = \varsigma_{j-1,i} - \frac{d_j}{d_{j-2}} \varsigma_{j-3,i-2} - \left(\frac{\delta_j}{d_{j-2}} - \frac{d_j}{d_{j-2}^2} \delta_{j-2} \right) \Gamma_{j-3,i-2} + \xi_{j,i} , \quad (6.17)$$

wobei wieder in Gleichung 6.16: $j = 2n - 1$ und in 6.17: $j = 2n$ ist.

Man gelangt zu den letzten beiden Beziehungen über eine Taylor-Entwicklung, wenn man Fehlerterme zweiter und höherer Ordnung vernachlässigt.¹

Bestimmende Parameter sind sicherlich d_{j-1} in Gleichung 6.16 bzw. d_{j-2} in Gleichung 6.17 im Verhältnis zu d_j , $\Gamma_{j-2,i-1}$ bzw. d_j , $\Gamma_{j-3,i-2}$. Dies bedeutet, je geringer der Betrag der Diskrepanz vor der letzten Längenänderung, um so stärker die Fortpflanzung von Rechenfehlern. Es sei darauf hingewiesen, daß diese Diskrepanz ebenfalls bestimmend für die Konditionierung des jeweiligen Sub-Systems ist (siehe Gleichung 5.58). Oft wird der Rechenfehler im Lösungsvektor wieder auf die rechte Seite umgerechnet. Da dies hier jedoch keine neuen Erkenntnisse liefert, werden entsprechende Rekursionen bei Bezug auf die rechte Seite an dieser Stelle nicht wiedergegeben.

Es werden nun im folgenden einige Simulationsergebnisse behandelt, die den Einfluß der Zahlendarstellung auf den Anteil der korrigierbaren (auffindbaren) Fehlermuster aufzeigen. Dabei wurde vorausgesetzt, daß kein Hintergrundrauschen vorhanden ist, um die Auswirkung von Rechenungenauigkeiten unabhängig von anderen Effekten untersuchen zu können. Überlegungen zur Konditionierung waren ja bereits Inhalt des vorigen Kapitels. Die Simulationen gehen von Gleitkommaarithmetik aus, und zeigen Abhängigkeiten von Mantissenlänge und darstellbarem Zahlenbereich. Der darstellbare Zahlenbereich wird dabei wie folgt festgelegt:

$$10^{-i} \leq X \leq 10^i. \quad (6.18)$$

Jeder Punkt in den beiden nachstehenden Diagrammen kennzeichnet die Mittelung aus 100 Einzelsimulationen (100 verschiedene Codeworte mit jeweils verschiedenen Fehlermustern). Es wurde der 'worst case' mit $e = \lfloor M/2 \rfloor$ zugrundegelegt.

Aus den beiden Darstellungen ist zu ersehen, daß mit zunehmender Codewortlänge und Verringerung der Rate $R = M/N$ die Anforderungen an die Rechengenauigkeit und an die Größe des darstellbaren Zahlenbereichs steigen. Aus Bild 6.2 folgt, daß der erforderliche Zahlenbereich bei allen vier gezeigten Beispielen nicht sehr groß ist, was dafür spricht, daß die Verwendung von Gleitkommaarithmetik nicht zwingend ist. Hingegen ist die benötigte Mantissenstellenzahl gemäß Bild 6.1 bei Codewortlängen über 40 nicht zu unterschätzen. Dennoch dürfte sogar im Falle $N = 100$, $R = 0,8$ eine Festkommadarstellung mit 32 Bit bei *geeigneter Normierung* noch genügen. Genaue Aussagen sind jedoch nur bei gegebener Implementierung möglich. (Dabei sind dann z.B. Fragestellungen von Bedeutung, ob Abschneiden oder Runden günstigere Ergebnisse liefert und Ähnliches.)

¹ $\frac{d_j + \delta_j}{d_{j-1} + \delta_{j-1}} = (d_j + \delta_j) \frac{1}{d_{j-1}} \frac{1}{1 + \frac{\delta_{j-1}}{d_{j-1}}} \approx (d_j + \delta_j) \frac{1}{d_{j-1}} \left(1 - \frac{\delta_{j-1}}{d_{j-1}}\right) \approx \frac{d_j}{d_{j-1}} + \frac{\delta_j}{d_{j-1}} - \frac{d_j}{d_{j-1}^2} \delta_{j-1}$

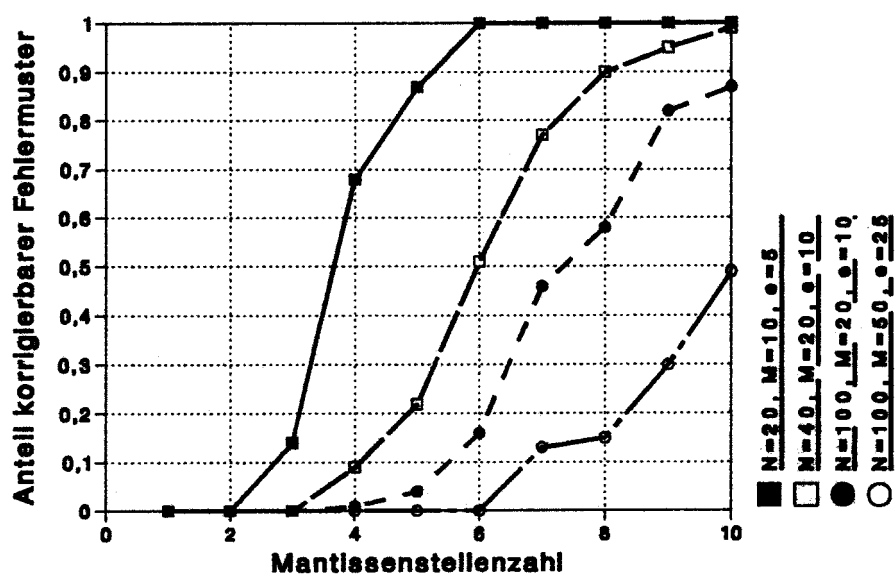


Abbildung 6.1: Der Anteil korrigierbarer Fehlermuster in Abhängigkeit der Mantissenlänge

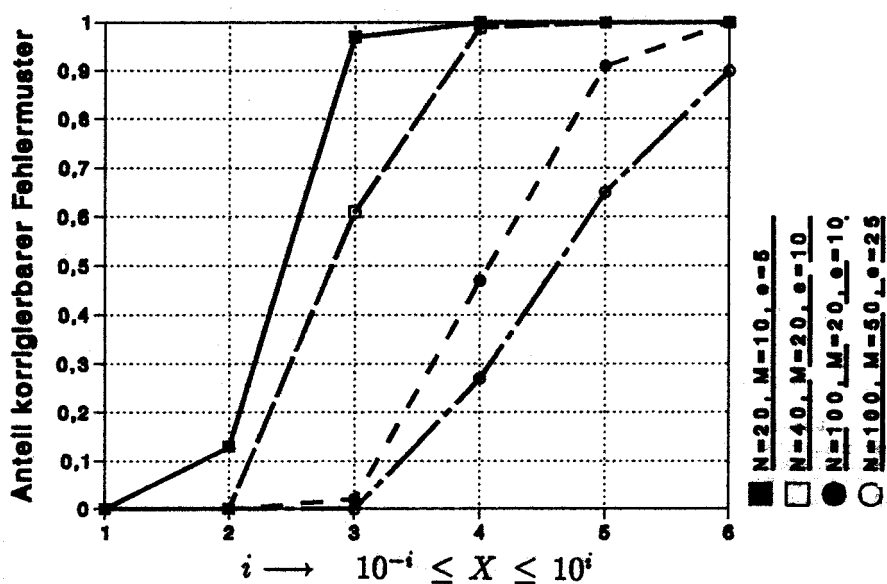


Abbildung 6.2: Der Anteil korrigierbarer Fehlermuster in Abhängigkeit des darstellbaren Zahlenbereichs

Zusammenfassend seien die Einflußfaktoren wie folgt gekennzeichnet: je größer die in die Decodierung eingehenden Gleichungssysteme bzw. Matrizen (DFT-Matrix und Toeplitz-Matrix), um so größer sind die Anforderungen an die Arithmetik. Die Länge der DFT entspricht der Codewortlänge und die Anzahl der Prüfstellen spiegelt sich in der Größe des Toeplitz-Schlüsselgleichungssystems wieder.

Nach diesem kurzen Kapitel, das sich wiederum mit dem BMA beschäftigte, schließt sich nun ein Kapitel an, das noch etwas auf zwei weitere Algorithmen für Toeplitz-Systeme eingeht.

Kapitel 7

Zwei weitere Möglichkeiten zur Lösung von Toeplitz-Systemen

In diesem Kapitel werden ergänzend zwei weitere Algorithmen zur Lösung des Schlüsselgleichungssystems angegeben. Zum einen wird der bereits erwähnte Levinson-Durbin-Algorithmus zur Lösung von Toeplitz-Systemen erläutert und eine *Erweiterung* vorgestellt, die es ermöglicht, neben der Lösung eines Toeplitz-Systems, auch die Inverse der Koeffizientenmatrix mit geringem Aufwand zu bestimmen. Zum anderen wird ein interessanter Zusammenhang zwischen Toeplitz-Problem und dem in der Echokompensation verwendeten LMS-Algorithmus (**L**east **M**ean **S**quare, auch Gradientenverfahren genannt) aufgezeigt.

7.1 Eine Erweiterung des Levinson-Durbin-Algorithmus

Der **Levinson-Durbin-Algorithmus**, im folgenden mit LDA bezeichnet, wird bevorzugt im Bereich der digitalen Signalverarbeitung, wie z.B. der linearen Prädiktion, eingesetzt.

Er ist allerdings nicht universell zur Lösung jeglicher Toeplitz-Gleichungssysteme geeignet. Eine Tatsache, die leider in der Literatur oft unerwähnt bleibt. Der Algorithmus versagt, wenn Untermatrizen singulär sind.

Es ist auch keineswegs richtig, wie in [5] behauptet wird, daß der LDA nur für (zur Hauptdiagonalen) symmetrische oder Hermitesche Toeplitz-Systeme anwendbar sei. Es existiert lediglich eine Spezialisierung hierfür. (z.B. in [45]).

Zunächst wird im folgenden der LDA in seiner ursprünglichen Form für nicht zwangsläufig symmetrische Toeplitz-Matrizen erläutert. Hieran schließt sich die Darstellung der vom Verfasser vorgeschlagenen Erweiterung zur Inversion einer Toeplitz-Matrix an.

Der LDA wurde 1947, kurz nach den Arbeiten von Wiener, in [36] als Verfahren zur Lösung von Problemen der diskreten ‘Least-Squares’-Schätzung vorgestellt. Später wurde der Algorithmus von Durbin ‘wiederentdeckt’ und spezialisiert ([20]).

Die diskrete ‘Least-Squares’-Prädiktion führt auf ein Gleichungssystem der Form

$$(1, A_{m,1}, \dots, A_{m,m}) \underbrace{\begin{pmatrix} R_0 & R_1 & \cdot & \cdot & R_m \\ R_{-1} & R_0 & \ddots & & \cdot \\ \cdot & \ddots & \ddots & \ddots & \cdot \\ \cdot & & \ddots & \ddots & R_1 \\ R_{-m} & \cdot & \cdot & R_{-1} & R_0 \end{pmatrix}}_{=:\underline{R}_m} = (R_m^l, 0, \dots, 0), \quad (7.1)$$

welches unter dem Namen *Yule-Walker-Gleichungen*¹ bekannt geworden ist (siehe hierzu [45], [34] oder [51]). R_m^l bezeichnet den mittleren quadratischen Fehler. Die Koeffizientenmatrix weist ebenso, wie die Syndrommatrix, eine Toeplitz-Struktur auf. Zusätzlich handelt es sich um eine Hermitesche Matrix. Diese Eigenschaft wird jedoch im folgenden nicht verwendet.

Im Gegensatz zum BMA erweitert der LDA rekursiv die betrachteten Sub-Matrizen in Hauptdiagonalenrichtung. Der BMA startet bei dem Koeffizienten links unten – rechts oben wäre genauso möglich gewesen –, der LDA beginnt hingegen mit dem Gleichungssystem

$$(1) (R_0) = (R_0^l) = (R_0^r). \quad (7.2)$$

Ebenso wie beim BMA ergänzt man naheliegenderweise die Lösungsvektoren mit Nullen, wenn das nächst größere Sub-System untersucht wird. Hiermit sei schon angedeutet, daß hier nicht nur ein, sondern zwei Lösungsvektoren rekursiv variiert werden, was die Parallelverarbeitung von zwei Gleichungssystemen bedeutet.

Erweitert man Gleichungssystem 7.1 durch rechtsseitiges Anfügen einer Null, so erhält man

$$(1, A_{m,1}, \dots, A_{m,m}, 0) (\underline{R}_{m+1}) = (R_m^l, 0, \dots, 0, \alpha_m). \quad (7.3)$$

Im Falle $\alpha_m = R_{m+1} + \sum_{i=1}^m A_{m,i} R_{m-i+1} = 0$ hat man bereits die Lösung des erweiterten Systems gefunden.

¹Im Falle mehrkanaliger Prädiktion bezeichnen die R_i Autokorrelationsmatrizen.

Sollte diese ‘Diskrepanz’ ungleich Null sein, so verwendet man ein Hilfs-Gleichungssystem in der Form

$$(0, B_{m,m}, \dots, B_{m,1}, 1) (\underline{R}_{m+1}) = (\beta_m, 0, \dots, 0, R_m^r), \quad (7.4)$$

das ebenso rekursiv ermittelt wird, wie das System mit dem zuvor verwendeten Lösungsvektor \vec{A} .

Durch gewichtete Linearkombination der beiden Lösungsvektoren eliminiert man die Diskrepanzen α_m und β_m . Man erhält durch diese gewichtete Summation

$$(1, A_{m,1} + K_m^\alpha B_{m,m}, \dots, K_m^\alpha) (\underline{R}) = (R_m^l + K_m^\alpha \beta_m, 0, \dots, 0, \alpha_m + K_m^\alpha R_m^r). \quad (7.5)$$

Wählt man

$$K_m^\alpha = -\alpha_m (R_m^r)^{-1}, \quad (7.6)$$

so ergibt

$$(1, A_{m+1,1}, \dots, A_{m+1,m+1}) = (1, A_{m,1}, \dots, A_{m,m}, 0) + K_m^\alpha (0, B_{m,m}, \dots, B_{m,1}, 1) \quad (7.7)$$

die Lösung des erweiterten Gleichungssystems.

Für den zweiten Lösungsvektor folgt mit

$$K_m^\beta = -\beta_m (R_m^l)^{-1} \quad (7.8)$$

$$(B_{m+1,m+1}, \dots, B_{m+1,1}, 1) = (0, B_{m,m}, \dots, B_{m,1}, 1) + K_m^\beta (1, A_{m,1}, \dots, A_{m,m}, 0). \quad (7.9)$$

R_m^l und R_m^r gehorchen dann den Rekursionen

$$R_{m+1}^l = R_m^l - \alpha_m \beta_m (R_m^r)^{-1} \quad (7.10)$$

$$R_{m+1}^r = R_m^r - \alpha_m \beta_m (R_m^l)^{-1}. \quad (7.11)$$

Zusammen mit der Initialisierung des LDA in Gleichung 7.2 folgt hieraus sofort

$$R_m^l = R_m^r. \quad (7.12)$$

Nachdem der LDA in seinen Grundzügen (in Anlehnung an [34]) erläutert ist, wird nun im folgenden der Zusammenhang zu Schlüsselgleichungssystem und Fehlerstellenpolynom erörtert.

Das Fehlerstellenpolynom wurde in Definition 3.1 wie folgt eingeführt:

$$\Gamma(x) = \prod_{i \in \mathbb{F}} (x - z^{-i}) = \Gamma_0 + \Gamma_1 x + \dots + \Gamma_{e-1} + 1x^e. \quad (7.13)$$

Auf den niedrigsten Koeffizienten normiert, ließe sich $\Gamma(x)$ auch als

$$\Gamma(x) = \prod_{i \in \mathbb{F}} (1 - xz^i) = 1 + \Gamma_1 x + \dots + \Gamma_{e-1} + \Gamma_e x^e \quad (7.14)$$

definieren. Es wird hier absichtlich keine neue Bezeichnung eingeführt, da für das Fehlerstellenpolynom nur die Nullstellen, nicht die Normierung, von Bedeutung sind.

Der Schlüsselgleichung

$$\sum_{i=0}^e \Gamma_i \cdot S_{j-i} = 0 \quad j = e, \dots, M-1 \quad (7.15)$$

$$\iff S_j = \sum_{i=1}^e \Gamma_i \cdot S_{j-i}. \quad (7.16)$$

entspricht unter der Annahme ungeradzahlig Prüfstellenzahl $M = 2E + 1$ und für $e = E$ das Gleichungssystem

$$(1, C_1, C_2, \dots, C_E) \begin{pmatrix} S_E & S_{E+1} & \cdot & \cdot & S_{2E} \\ S_{E-1} & S_E & \ddots & & \cdot \\ \cdot & \ddots & \ddots & \ddots & \cdot \\ \cdot & & \ddots & \ddots & S_{E+1} \\ S_0 & \cdot & \cdot & S_{E-1} & S_E \end{pmatrix} = (0, 0, \dots, 0), \quad (7.17)$$

welches in der Struktur den Yule-Walker-Gleichungen 7.1 entspricht.

Es wird im folgenden vorausgesetzt, alle betrachteten *Untermatrizen seien nicht-singulär*. Dies bedeutet insbesondere auch, daß die Komponente S_E ungleich Null ist.

Die Initialisierung des Algorithmus erfolgt mit

$$(1) \cdot (S_E) = (R_0^l) \implies R_0^l = S_E \quad (7.18)$$

bzw.

$$(1) \cdot (S_E) = (R_0^r) \implies R_0^r = R_0^l = S_E, \quad (7.19)$$

wobei ohnehin $R_m^l = R_m^r$ gilt.

Bei nichtverschwindenden Diskrepanzen verlängert der LDA nun sukzessive die Lösungsvektoren, was einem Inkrement des Grades des Fehlerstellenpolynoms entspricht. In dieser Hinsicht besteht eine gewisse Ähnlichkeit zum BMA.

Interessant ist, daß der LDA bei ungeradzahliger Prüfstellenzahl beide oben angegebenen Darstellungsformen des Fehlerstellenpolynoms liefert. Es gelten die Entsprechungen:

$$(1, A_{E,1}, \dots, A_{E,E-1}, A_{E,E}) \longleftrightarrow \Gamma(x) \text{ gemäß 7.14} \quad (7.20)$$

$$(B_{E,E}, B_{E,E-1}, \dots, B_{E,1}, 1) \longleftrightarrow \Gamma(x) \text{ gemäß 7.13.} \quad (7.21)$$

Bislang wurde eine ungeradzahlige Syndromstellenzahl vorausgesetzt, was jedoch nicht notwendig ist, wie sogleich gezeigt wird.

Bei geradzahliger Syndromstellenzahl (S_0, \dots, S_{2E-1}) ist das Element S_{2E} unbekannt. Dieses geht jedoch in die Berechnung von α_{E-1} ein, was dazu führt, daß die $A_{E,i}$ nicht definiert sind. Zur Ermittlung der Koeffizienten $B_{E,i}$ wird $K_{E-1}^\beta = -\beta_{E-1} (R_{E-1}^l)^{-1}$ verwendet. Sowohl für β_{E-1} , als auch für R_{E-1}^l wird S_{2E} nicht benötigt. Somit liefert lediglich der Vektor der $B_{E,i}$ ein in diesem Fall gültiges Ergebnis.

Man hätte natürlich das Gleichungssystem 7.17 auch wie folgt darstellen können:

$$(1, C_1, C_2, \dots, C_E) \begin{pmatrix} S_{E-1} & S_E & \cdot & \cdot & S_{2E-1} \\ S_{E-2} & S_{E-1} & \ddots & & \cdot \\ \cdot & \ddots & \ddots & \ddots & \cdot \\ \cdot & & \ddots & \ddots & S_E \\ S_{-1} & \cdot & \cdot & S_{E-2} & S_{E-1} \end{pmatrix} = (0, 0, \dots, 0). \quad (7.22)$$

Dann wäre S_{-1} das unbekannte Syndromelement, welches dann weder in α_{E-1} noch in R_{E-1}^r eingeht. Mit $K_{E-1}^\alpha = -\alpha_{E-1} (R_{E-1}^r)^{-1}$ ergibt daher der Vektor der $A_{E,i}$ das korrekte Ergebnis.

Der Fall singulärer Untermatrizen

Singuläre Untermatrizen haben im LDA zur Folge, daß die beiden zugehörigen Lösungsvektoren linear abhängig sind und die rechte Seite gänzlich verschwindet, wodurch der Algorithmus abbricht.

Das Fortführen des LDA im Fall singulärer Submatrizen ist in Einzelfällen durch Neustart des Algorithmus an der ersten Komponente der letzten Spalte der singulären Untermatrix und eventuell zusätzlich an der ersten Komponente der letzten

Zeile dieser Submatrix unter Ausnutzung des singulären (homogenen) Subsystems möglich. Eine in jedem Fall anwendbare Vorgehensweise ist jedoch bislang nicht bekannt. Auf eine Wiedergabe der angesprochenen Spezialfälle soll hier verzichtet werden. Einer Singularität bei der Initialisierung, d.h. falls S_E bzw. S_{E-1} gleich Null sein sollten, ist einfach durch Wahl des jeweils anderen Gleichungssystems – 7.17 oder 7.22 – zu begegnen.

Bei analogen Problemen ist die Schwierigkeit der Behandlung von Systemen, deren Koeffizientenmatrix singuläre Untermatrizen enthält, mit dem LDA etwas weniger kritisch zu bewerten, da Determinanten dort bestenfalls näherungsweise, aber nie exakt verschwinden (die Wahrscheinlichkeit hierfür geht gegen Null). Es ergeben sich ‘lediglich’ numerische Probleme, die unter Umständen aber auch nicht tolerierbar sind. Über endlichen Zahlkörpern ist die Möglichkeit der Singularität per Prinzip gegeben, wodurch sich die Anwendung des LDA von selbst verbietet.

Erweiterter LMS-Algorithmus

Die Inversion von Toeplitz-Matrizen ist insbesondere in der Theorie diskreter Zufallsprozesse von Interesse. Als Beispiel einer Anwendung sei hier nur eine Veröffentlichung zur Parameterschätzung genannt [11], ohne jedoch hierauf näher einzugehen.

Bevor eine Erweiterung des LMS-Algorithmus vorgestellt wird, die ebenso, wie z.B. der Trench-Algorithmus ([56]), die Berechnung der Inversen einer Toeplitz-Matrix erlaubt, soll noch der Begriff der *Persymmetrie* gemäß [5] eingeführt werden.

Definition 7.1 Mit ‘persymmetrisch’ kennzeichnet man eine Matrix mit einer Symmetrie bezüglich der Nebendiagonalen ($a_{i,j} = a_{n+1-j,n+1-i}$ bei einer quadratischen $n \times n$ -Matrix).

Toeplitz-Matrizen gehören zu dieser Gruppe von Matrizen. Von besonderem Interesse ist hier der folgende Satz, der ebenfalls, jedoch mit einem fehlerhaften Beweis, in [5] angegeben wird:

Satz 7.1 Die Inverse \underline{A}^{-1} einer persymmetrischen Matrix \underline{A} ist wiederum persymmetrisch.

Beweis: Sei \underline{J} eine Permutationsmatrix derselben Größe wie \underline{A} mit Einsen in der Nebendiagonalen. Mit $\underline{J}^2 = \underline{1}$ folgt $\underline{J}^{-1} = \underline{J}$. Matrizen sind genau dann (und nur dann) persymmetrisch wenn gilt:

$$\underline{J}\underline{A}\underline{J} = \underline{A}^T.$$

Man bedenke dabei, daß ein linksseitiges Multiplizieren mit \underline{J} ein Spiegeln an einer horizontalen Mittelachse und ein entsprechendes rechtsseitiges Multiplizieren ein Spiegeln an einer vertikalen Mittelachse bedeutet. Eine Inversion obiger Beziehung liefert

$$\underline{J} \underline{A}^{-1} \underline{J} = (\underline{A}^T)^{-1} = (\underline{A}^{-1})^T,$$

womit bewiesen ist, daß \underline{A}^{-1} ebenfalls persymmetrisch ist.

Wie schon oben angegeben, basiert auch der LDA auf sukzessivem Erweitern der Lösungsvektoren mit Nullen bei Betrachtung der nächst größeren Sub-Toeplitz-Matrix. Die Auswirkung des Anfügens von Nullen soll im folgenden noch einmal klar herausgestellt werden.

Erweitern durch Anfügen einer Null rechts:

Die rechte Seite des Gleichungssystems bleibt erhalten und eine neue Stelle wird rechts hinzugefügt – symbolisch dargestellt:

$$(\longleftarrow \text{ vorherige rechte Seite } \longrightarrow, \text{ neue Stelle}).$$

Erweitern durch Anfügen einer Null links:

Die rechte Seite des Gleichungssystems bleibt erhalten und eine neue Stelle wird links hinzugefügt – symbolisch dargestellt:

$$(\text{neue Stelle}, \longleftarrow \text{ vorherige rechte Seite } \longrightarrow).$$

Dies bedeutet, daß die vorherige rechte Seite nun entweder links- oder rechtsbündig erscheint. Zusammen mit den Vektoren \vec{A} oder \vec{B} des ursprünglichen LDA, mit welchen man durch Linearkombination die neu auftretende Komponente zum Verschwinden bringen kann, ist es möglich, eine rechte Seite zu erzeugen, die nur eine Komponente ungleich Null besitzt. Normierung des Gleichungssystems auf diese Komponente liefert jeweils eine Zeile der inversen Matrix. Eine schematische Darstellung ist in Bild 7.1 angegeben.

Links und rechts findet sich der herkömmliche LDA (senkrechte Pfeile). Rechtsorientierte Pfeile bezeichnen das Erweitern mit Nullen links (\mathcal{L}), linksorientierte Pfeile das Erweitern mit Nullen rechts (\mathcal{R}). Die jeweils neu hinzukommenden Komponenten werden durch Linearkombination mit den Lösungsvektoren am rechten und linken Rand (herkömmlicher LDA) auf gleicher Ebene eliminiert.

Zur besseren Veranschaulichung wird nachfolgend ein Beispiel zur Inversion einer 4×4 -Matrix mit Komponenten aus $GF(11)$ angegeben. (Aus Gründen der An-

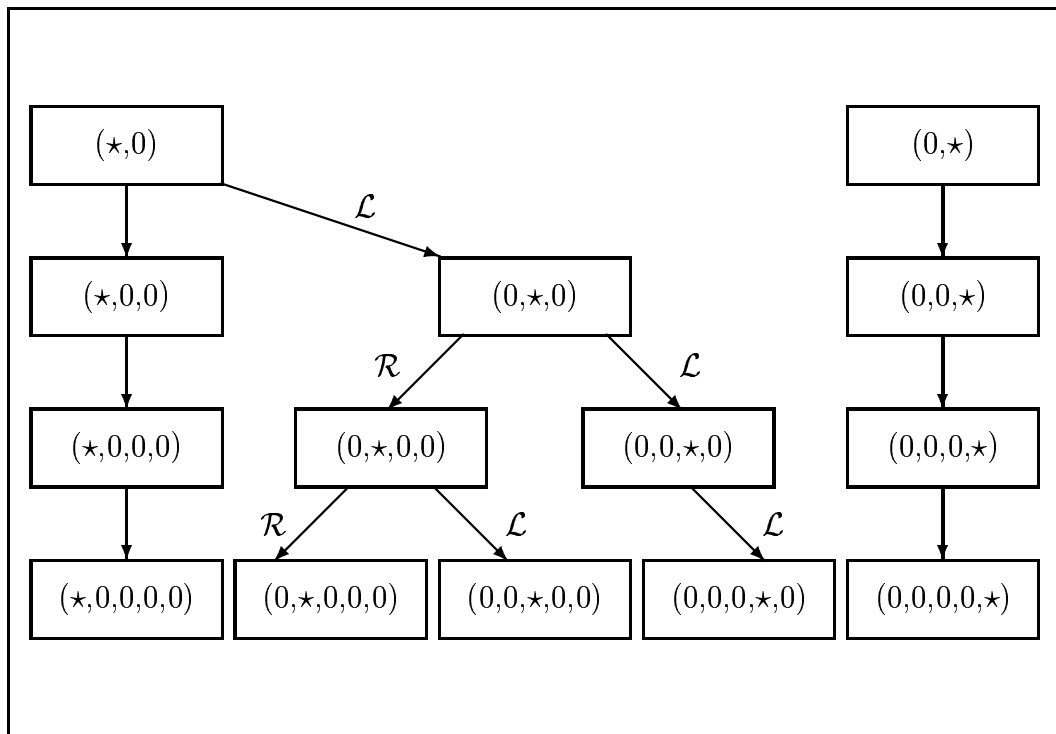


Abbildung 7.1: Schematische Veranschaulichung des erweiterten Levinson-Durbin-Algorithmus – angegeben sind die rekursiv auftretenden rechten Seiten

schaulichkeit wurde ein endlicher Zahlenkörper gewählt. Dabei muß natürlich darauf geachtet werden, daß Untermatrizen nicht singulär sind.)

Beispiel

Gesucht sei die Inverse \underline{A}^{-1} zu

$$\underline{A} = \begin{pmatrix} 10 & 0 & 4 & 0 \\ 2 & 10 & 0 & 4 \\ 9 & 2 & 10 & 0 \\ 5 & 9 & 2 & 10 \end{pmatrix}$$

$$\overline{(1) (10) = (10)}$$

$$\begin{array}{c|c} (1, 0) \begin{pmatrix} 10 & 0 \\ 2 & 10 \end{pmatrix} = (10, 0) & (0, 1) \begin{pmatrix} 10 & 0 \\ 2 & 10 \end{pmatrix} = (2, 10) \\ (1, 0) - \frac{0}{10}(0, 1) = (1, 0) & (0, 1) - \frac{2}{10}(1, 0) = (2, 1) \\ \hline (1, 0) \begin{pmatrix} 10 & 0 \\ 2 & 10 \end{pmatrix} = (10, 0) & (2, 1) \begin{pmatrix} 10 & 0 \\ 2 & 10 \end{pmatrix} = (0, 10) \end{array}$$

$$\begin{array}{c|c} (1, 0, 0) \begin{pmatrix} 10 & 0 & 4 \\ 2 & 10 & 0 \\ 9 & 2 & 10 \end{pmatrix} = (10, 0, 4) & (0, 1, 0) \begin{pmatrix} 10 & 0 & 4 \\ 2 & 10 & 0 \\ 9 & 2 & 10 \end{pmatrix} = (2, 10, 0) \\ (1, 0, 0) - \frac{4}{10}(0, 2, 1) = (1, 8, 4) & (0, 1, 0) - \frac{2}{7}(1, 8, 4) = (6, 5, 2) \\ \hline (1, 8, 4) \begin{pmatrix} 10 & 0 & 4 \\ 2 & 10 & 0 \\ 9 & 2 & 10 \end{pmatrix} = (7, 0, 0) & (6, 5, 2) \begin{pmatrix} 10 & 0 & 4 \\ 2 & 10 & 0 \\ 9 & 2 & 10 \end{pmatrix} = (0, 10, 0) \end{array}$$

$$\begin{array}{c|c|c} (1, 8, 4, 0)\underline{A} = (7, 0, 0, 10) & (6, 5, 2, 0)\underline{A} = (0, 10, 0, 9) & \\ (1, 8, 4, 0) - \frac{10}{7}(0, 2, 2, 1) = (1, 2, 9, 8) & (6, 5, 2, 0) - \frac{9}{3}(4, 1, 7, 1) = (5, 2, 3, 8) & \\ \hline (1, 2, 9, 8)\underline{A} = (3, 0, 0, 0) & (5, 2, 3, 8)\underline{A} = (0, 10, 0, 0) & \end{array}$$

$$\Rightarrow \underline{A}^{-1} = \begin{pmatrix} 1/3 & 2/3 & 9/3 & 8/3 \\ 5/10 & 2/10 & 3/10 & 8/10 \\ 7/10 & 9/10 & 2/10 & 3/10 \\ 4/3 & 1/3 & 7/3 & 1/3 \end{pmatrix} = \begin{pmatrix} 4 & 8 & 3 & 10 \\ 6 & 9 & 8 & 3 \\ 4 & 2 & 9 & 8 \\ 5 & 4 & 6 & 4 \end{pmatrix}$$

$$\begin{array}{c|c}
(0, 2, 1) \begin{pmatrix} 10 & 0 & 4 \\ 2 & 10 & 0 \\ 9 & 2 & 10 \end{pmatrix} = (2, 0, 10) \\
(0, 2, 1) - \frac{2}{10}(1, 0, 0) = (2, 2, 1) \\
\hline
(2, 2, 1) \begin{pmatrix} 10 & 0 & 4 \\ 2 & 10 & 0 \\ 9 & 2 & 10 \end{pmatrix} = (0, 0, 7) \\
\hline
\end{array}$$

$$\begin{array}{c|c|c}
(0, 6, 5, 2)\underline{A} = (1, 0, 10, 0) & (0, 2, 2, 1)\underline{A} = (5, 0, 0, 7) & \\
(0, 6, 5, 2) - \frac{1}{3}(1, 2, 9, 8) = (7, 9, 2, 3) & (0, 2, 2, 1) - \frac{5}{7}(1, 8, 4, 0) = (4, 1, 7, 1) & \\
\hline
(7, 9, 2, 3)\underline{A} = (0, 0, 10, 0) & (4, 1, 7, 1)\underline{A} = (0, 0, 0, 3) & \\
\hline
\end{array}$$

Die Inverse besitzt, entsprechend obigem Satz, die Eigenschaft der Persymmetrie.

Der baumartig erweiterte LDA liefert auf einfache Weise die Inverse einer Toeplitz-Matrix. Es werden hierbei ebenfalls die Inversen aller betrachteten Sub-Matrizen ermittelt.

Die Behandlung ‘direkter’ rekursiver Toeplitz-Algorithmen wird mit diesem Abschnitt abgeschlossen. Es sei an dieser Stelle noch bemerkt, daß diese Arbeit trotz der

Betrachtung der wesentlichen Toeplitz-Algorithmen (BMA, Euklid², LDA) natürlich nicht zum Ziel hat, einen Überblick über alle bekannten Verfahren zu bieten, da dies den hier gegebenen Rahmen sprengen würde. Stattdessen wird im Folgekapitel noch auf einen interessanten Zusammenhang zu einem aus der Echokompensation bekannten Verfahren hingewiesen (ohne jedoch für die Decodierung von praktischer Relevanz zu sein).

7.2 Toeplitz-Systeme und Least-Mean-Square-Algorithmus

In der Echokompensation (u.a.) wird oft ein iteratives Verfahren, der sogenannte LMS-Algorithmus (**L**east **M**ean **S**quare), zum Abgleich eines Transversalfilters verwendet (siehe z.B. [43] und [64]). Hier soll die prinzipielle Möglichkeit aufgezeigt werden, den Algorithmus ebenfalls zur *iterativen Lösung des Schlüsselgleichungssystems* einzusetzen.

Das Schlüsselgleichungssystem ist wieder bestimmt durch

$$S_j + \sum_{i=1}^e \Gamma_i \cdot S_{j-i} = 0 \quad j = e, \dots, M-1, \quad (7.23)$$

wobei die zu ermittelnden Komponenten Γ_i die Koeffizienten des Fehlerstellenpolynoms darstellen.

Ein Transversalfilter mit den Koeffizienten $(-\Gamma_i)$ liefert bei der Eingangsfolge S_{j-e}, \dots, S_{j-1} den Ausgang

$$Y_j = - \sum_{i=1}^e \Gamma_i \cdot S_{j-i}. \quad (7.24)$$

Gemäß der Schlüsselgleichung sollte sich dabei $Y_j = S_j$ ergeben. Die Diskrepanz $\delta_j = S_j - Y_j$ könnte dann im Gradientenverfahren zum Nachführen der Koeffizienten dienen:

$$\vec{\Gamma}_{j+1} = \vec{\Gamma}_j + \alpha \cdot \delta_j \cdot \frac{\vec{S}_j^*}{\|\vec{S}_j\|^2}. \quad (7.25)$$

Das Gradientenverfahren stellt hiermit ein iteratives Verfahren zur Lösung eines Toeplitz-Systems dar. Man wiederholt dabei zyklisch den verwendeten Ausschnitt

²Der Euklidische Algorithmus wurde in dieser Arbeit nicht direkt mit Toeplitz-Systemen in Verbindung gebracht. Hierzu sei auf [5], S. 379 ff. verwiesen.

aus den bekannten Syndromelementen. Den LMS-Algorithmus könnte man damit in die Reihe der sogenannten Relaxationsverfahren (siehe z.B. [55]) einordnen, wobei α dann als Relaxationsparameter zu betrachten wäre ($0 < \alpha < 1$).

Für den praktischen Einsatz zur Decodierung ist das iterative Verfahren jedoch wegen der zu großen Anzahl der nötigen Iterationsschritte (besonders mit zunehmender Impulsfehler-Anzahl e) nicht von Interesse. Dennoch sollten iterative Lösungsmöglichkeiten nicht gänzlich übergangen werden.

Kapitel 8

Tiefpaß-Filterung versus Decodierung – – ein Qualitätsvergleich

Dieses Kapitel, das, sieht man von der Zusammenfassung einmal ab, das Schlußkapitel dieser Arbeit bildet, stellt einen Vergleich zwischen analoger Decodierung und simplem Nullsetzen der Syndromkomponenten, was einer Tiefpaßfilterung entspricht, her. Es wird hierbei Bezug auf eine Arbeit von Helstrom 1964 ([26]) genommen, die als Ergebnis folgende Aussage liefert:

Codierung von K Informationsstellen in N Stellen eines Codewortes zur Übertragung über einen Kanal, dessen Störungen normalverteilt, ‘weiß’ und statistisch unabhängig mit gleicher Standardabweichung für die einzelnen Codesymbole sind, bringt keinerlei Vorteile bezüglich des resultierenden Störabstands gegenüber der direkten Übermittlung der K Informationssymbole. Hierbei wurde vorausgesetzt, daß die Informationsstellen einem bandbegrenzten Signal entstammen. Dieses Ergebnis ist nicht abhängig von statistischen Abhängigkeiten in den Informationssymbolen.

Es sei jedoch betont, daß diese Aussage nur unter obigen Randbedingungen korrekt ist. Im Falle von impulsförmigen Störungen (mit eventuell zusätzlichen statistischen Abhängigkeiten), trifft sie keineswegs zu. Diese Stellungnahme findet sich auch bereits in [68], ohne sich jedoch auf konkrete Untersuchungen zu stützen.

Hier sollen Simulationsergebnisse angegeben werden, die sowohl Helstroms Aussage bestätigen, als auch deren Einschränkung dokumentieren.

8.1 Simulationsergebnisse

Am Beispiel eines analogen RS-Codes mit den Parametern $N = 20$, $K = 10$ werden im folgenden die Eigenschaften von ‘Tiefpaß-Filterung’ und Decodierung verglichen. Es wird hierzu die Kanalcharakteristik zwischen reinem Gauss’schem Rauschen und reinen Impulsstörungen variiert. Die Vorgehensweise im einzelnen:

- ‘Tiefpaß-Filterung’ bedeutet schlichtes Nullsetzen des Syndromanteils im Frequenzbereich.
- Die Decodierung erfolgt mit dem BMA zur Ermittlung der Fehlerstellen und der ‘*Least-Squares-Approximation*’ gemäß Kapitel 3.4, Gleichung 3.29 zur Bestimmung der Fehlerwerte.

Ein ‘Least-Squares’-Verfahren wurde gewählt, um *alle* Syndromstellen bei der Fehlerwertbestimmung zu nutzen.

Die folgenden Diagramme zeigen die Wurzel aus dem quadratischen Abstand zwischen den sich ergebenden korrigierten bzw. ‘gefilterten’ Vektoren und den fehlerfreien Vektoren (Euklidische Distanz d_e ; normiert wurde auf die Euklidische Distanz zwischen fehlerbehaftetem und fehlerfreiem Wort d_{e0}) in Abhängigkeit des Verhältnisses von Impulsfehler-Amplitude zu Standardabweichung $|B|/\sigma$ des Gauss’schen Rauschens. (Die Kurvenzüge basieren auf 21 äquidistanten Stützwerten, die wiederum ein Mittel aus der Untersuchung von 100 unterschiedlichen Fehlervektoren darstellen.) Es wird dabei vorausgesetzt, alle Impulsfehler besäßen gleiche Amplitude und zufällige, gleichverteilte Phase.

Im Falle der ‘Tiefpaß-Filterung’ folgt d_{eTP} bei einem Code der Rate $1/2$ sofort zu $d_{eTP} = 1/\sqrt{2}$, da durch Nullsetzen der Syndromkomponenten die Hälfte der Störenergie eliminiert wird.

In jedem der drei hier wiedergegebenen Diagramme finden sich vier verschiedene Kurven, deren Bezeichnungen wie folgt zu verstehen sind:

- | | |
|-------------------------|--|
| TP : | Nullsetzen der Syndromkomponenten, |
| BMA u. LSQ : | BMA zusammen mit Least-Squares-Approximation, |
| BMA , LSQ u. TP : | BMA zusammen mit Least-Squares-Approximation und anschließend TP , |
| TP od. BMA/LSQ : | Auswahl nach Kriterium im nächsten Abschnitt. |

Aus einem Vergleich der Kurven ‘ TP ’ und ‘ BMA u. LSQ ’ bestätigt sich zunächst die Aussage von Helstrom. Die Fehlerkorrektur ist bei reinem normalverteiltem Rauschen sogar deutlich ungünstiger (ca. um einen Faktor 2) als die reine ‘Tiefpaß-

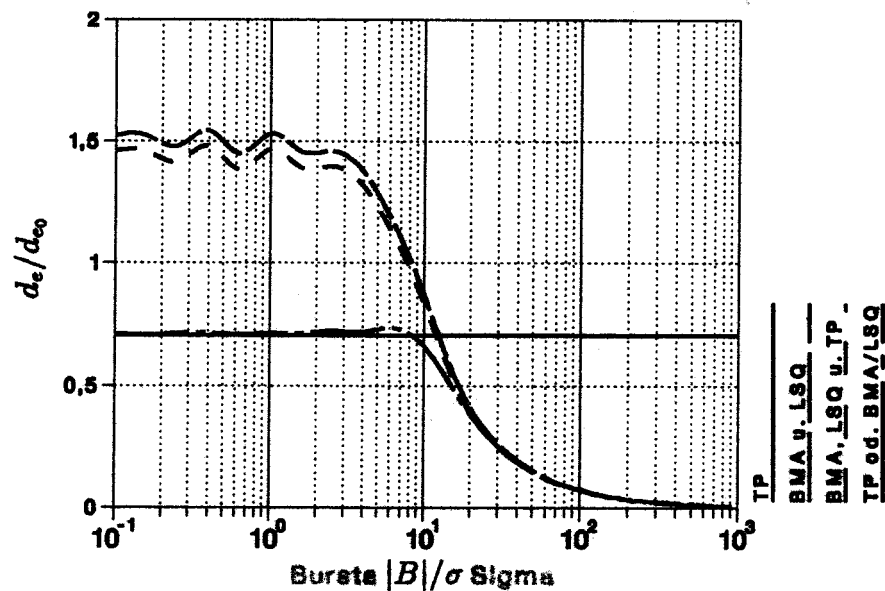


Abbildung 8.2: Die normierte Euklidische Distanz d_e/d_{e_0} bei 'Tiefpaß-Filterung' oder Korrektur (Parameter $N=20$, $K=10$, $e=3$) in Abhängigkeit von $|B|/\sigma$

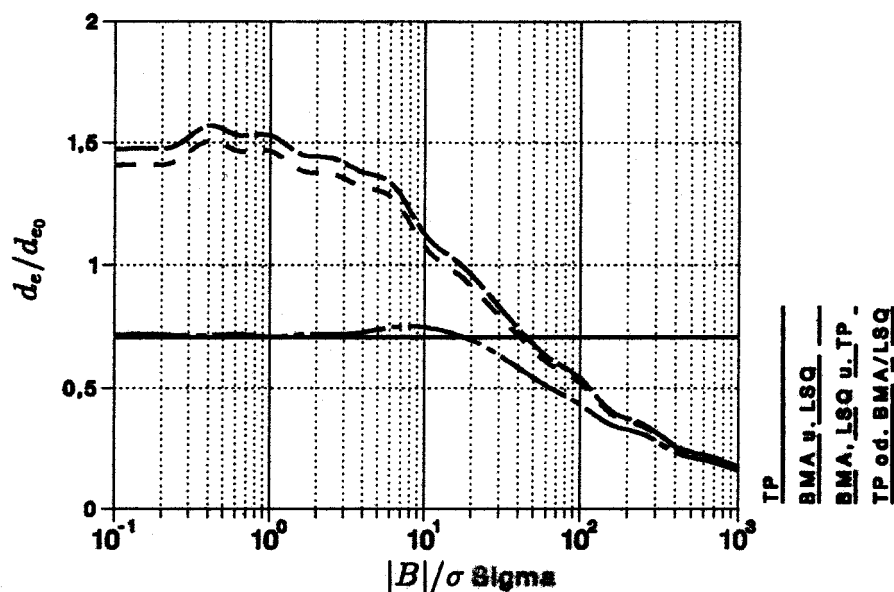


Abbildung 8.3: Die normierte Euklidische Distanz d_e/d_{e_0} bei 'Tiefpaß-Filterung' oder Korrektur (Parameter $N=20$, $K=10$, $e=5$) in Abhängigkeit von $|B|/\sigma$

8.2 Ein Auswahlkriterium

Im vorigen Abschnitt wurde bereits auf den Unterschied zwischen den Kurven ‘*BMA* u. *LSQ*’ und ‘*BMA*, *LSQ* u. *TP*’ hingewiesen. Es fällt auf, daß dieser, ohnehin schon gering, sich mit steigendem $|B|/\sigma$ noch reduziert.

Trägt man die ‘Energie’ im Syndrom nach Least-Squares-Korrektur bezogen auf die ursprüngliche ‘Energie’ ohne Korrektur ($E_{syn\ ko}/E_{syn\ ur}$) über dem Verhältnis $|B|/\sigma$ auf (Bild 8.4), so erkennt man, daß die Kurven einen prinzipiell ähnlichen Verlauf, verglichen mit den Kurven für Korrektur im vorigen Abschnitt, aufweisen. Sie liefern direkt Aussagen über die Kanalcharakteristik, d.h. ob es sich um überwiegend impulsförmige Störungen oder Rauschen handelt.

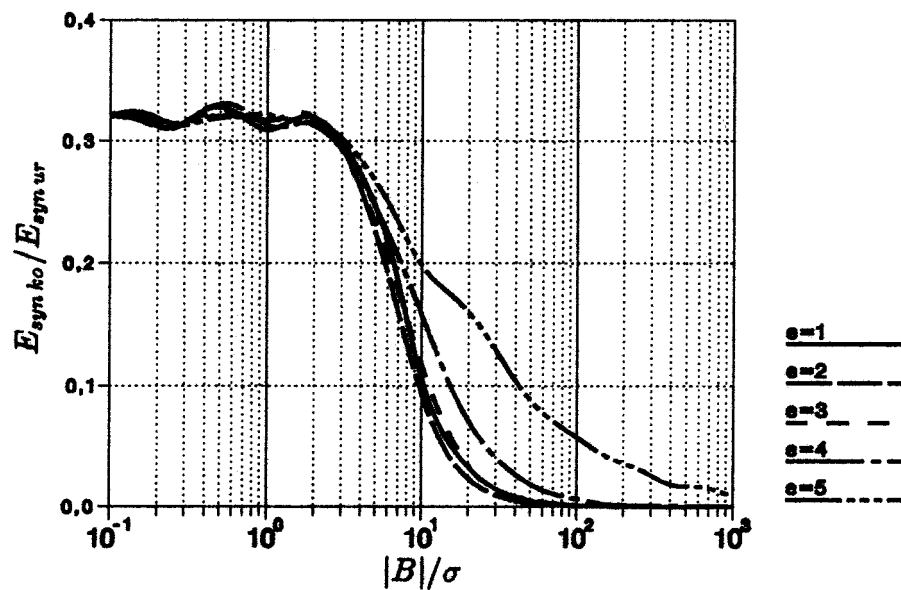


Abbildung 8.4: ‘Energie’ im Syndrom nach Least-Squares-Korrektur bezogen auf die ursprüngliche ‘Energie’ ohne Korrektur ($E_{syn\ ko}/E_{syn\ ur}$) in Abhängigkeit von $|B|/\sigma$

Bei einer Schwellenwahl für $E_{syn\ ko}/E_{syn\ ur}$, beispielsweise hier 0,085, ergibt sich auf einfache Weise ein Auswahlkriterium, ob ‘Tiefpaßfilterung’ oder Korrektur zu bevorzugen ist. Hieraus folgen die im vorigen Abschnitt noch nicht erläuterten Kurven ‘*TP* od. *BMA/LSQ*’, die aufzeigen, daß es auf diese Weise möglich ist, die Vorteile beider Alternativen zu nutzen.

Kapitel 9

Zusammenfassung

Zunächst stellt die vorliegende Arbeit eine Verbindung zwischen sowohl verschiedenen Disziplinen der Mathematik als auch der Ingenieurwissenschaften her. Es werden Gemeinsamkeiten von Algebraischer Codierungstheorie, Approximationstheorie und Numerik und ebenso die Beziehungen zur digitalen Signalverarbeitung, wie z.B. der linearen Prädiktion, aufgezeigt.

Das Bindeglied wird gebildet durch die Konstruktion von Reed-Solomon-Codes oder BCH-Codes (**B**ose, **C**haudhuri, **H**ocquenghem) über komplexen Zahlen. Initiiert wurden die Überlegungen hierzu von Marshall 1981 in [40] und Wolf 1983 in [68], wo sich erste Ansätze zur Einfehlerkorrektur finden. Von Wolf wurde ebenfalls der Begriff ‘*analoge Codes*’ eingeführt.

In dieser Arbeit werden nun die Eigenschaften dieser analogen Codes erstmals genauer untersucht. Besondere Aufmerksamkeit verdienen hierbei die Zusammenhänge zu anderen Arbeitsgebieten; so ist mehr als ein Kapitel den Verbindungen zwischen Codierung und Interpolation bzw. Approximation gewidmet.

Die besonderen Eigenschaften analoger Codes sind:

- beliebig wählbare Codewortlänge
- kein notwendigerweise wertdiskretes, endliches Alphabet
- keine spezielle Algebra und damit
- keine spezielle Hardware zur Codierung und Decodierung
- höhere Kanalkapazität durch Nutzung des analogen Ausgangswertes

- höhere Korrekturfähigkeit ($d - 2$ statt $\lfloor (d - 1)/2 \rfloor$)¹
- Unterscheidbarkeit von Fehlern kleiner und großer Amplitude, d.h. ‘Hintergrundrauschen’ ist bis zu einem gewissen Maß bei der Korrektur der ‘großen’ Fehler tolerierbar und es erfolgt eine Korrektur auf das Maß der ‘kleinen’ Fehler, die unkorrigiert bleiben
- Möglichkeit der Rahmensynchronisation bei Wahl eines diskreten Alphabetes für die Informationsstellen, da die sich ergebenden Prüfstellen teils nicht in diesem Alphabet liegen
- herkömmliche Codier- und Decodierverfahren für Codes über endlichen Zahlkörpern sind nahezu direkt auf den komplexen Fall übertragbar und
- neue Algorithmen aus anderen Disziplinen sind ebenfalls anwendbar
- die Decodierung analoger Codes liefert sogar Aussagen über die Störcharakteristik des Kanals.

Die Betrachtung analoger Codes mit ihren besonderen Eigenschaften bilden ein ‘Gerüst’, in das sich eine Reihe wichtiger Ergebnisse einfügen, die ebenso für andere Anwendungsgebiete von Interesse sind. Diese seien an dieser Stelle noch einmal deutlich herausgestellt.

Zunächst ist dabei ein neues Verfahren zur Syndromdefinition zu nennen, das aus einer Betrachtung des Newton-Schemas zur Polynominterpolation folgt. In dieser Arbeit wird es für einen Beweis der Korrekturfähigkeit analoger Codes eingesetzt und besonders auf die Möglichkeit hingewiesen, mit der Methode direkt fehlerfreie Bereiche der Länge $K + 1$ im Codewort², ohne die Anwendung von Decodieralgorithmen, zu erkennen. Besonders möchte ich darauf hinweisen, daß die Überlegungen hierzu in der Arbeit meines Kollegen H. Schneider weitergeführt werden und sich diese neue Syndromfestlegung dort als sehr vielversprechend erweist.

Ein zentrales Ergebnis der Arbeit bildet sicherlich die neue Beschreibung des wohl wichtigsten Decodieralgorithmus, des Berlekamp-Massey-Algorithmus. Diese bietet einerseits besondere didaktische Vorteile gegenüber Masseys Darstellung, andererseits sind wichtige Eigenschaften des Verfahrens aus dieser Beschreibung auf einfachere Weise zu gewinnen. Hierzu zählt die Erkenntnis, daß der Algorithmus unter der

¹ d : Hamming-Distanz

² K : Anzahl der Informationsstellen

Voraussetzung nicht singulärer Untermatrizen (was man bei analogen Codes voraussetzen kann) eine Zerlegung der Koeffizientenmatrix des sogenannten Schlüsselgleichungssystems in Form einer *triangular 'square root'-factorization* $\underline{S}_L^{-1} = \underline{\Gamma}_L \underline{D}_L^{-1} \underline{\Gamma}_L^T$ bzw. $\underline{S}_L = (\underline{\Gamma}_L^T)^{-1} \underline{D}_L \underline{\Gamma}_L^{-1}$ liefert. $\underline{\Gamma}_L$ kennzeichnet eine Dreiecksmatrix, bestehend aus den rekursiv ermittelten Koeffizienten des Fehlerstellenpolynoms, und \underline{D}_L eine Diagonalmatrix aus den zugehörigen 'Diskrepanzen'. Die Faktorisierung erweist sich insbesondere bei den angestellten Konditionsbetrachtungen als günstig, die der Untersuchung des Einflusses von 'Hintergrundrauschen' dienen.

Ebenfalls sei die vorgestellte Erweiterung des Levinson-Durbin-Algorithmus hervorgehoben. Hierdurch kann das Verfahren neben der Lösung von Toeplitz-Systemen mit spezieller rechter Seite ebenfalls mit recht geringem Rechenaufwand zur Inversion von Toeplitz-Matrizen eingesetzt werden. Dies ist vor allem für Aufgabenstellungen aus dem Bereich der Theorie diskreter Zufallsprozesse von besonderem Interesse (Beispiel: Parameterschätzverfahren).

Die Überlegungen zu Berlekamp-Massey- und Levinson-Durbin-Algorithmus sind keineswegs auf Aspekte der Decodierung von analogen Codes beschränkt, vielmehr beziehen sie sich auf jegliche Problemstellungen, bei denen Toeplitz-Systeme eine Rolle spielen, wie z.B. auch 'Scattering' (Streuprobleme) und lineare Prädiktion.

Die hervorgehobenen Punkte bilden Teile einer genauen Untersuchung von RS-Codes über komplexen Zahlen. Der Ablauf dieser Untersuchung wird im folgenden noch einmal stichpunktartig skizziert.

Nach einer kurzen Einleitung stellt das *zweite Kapitel* zunächst Grundlagen zusammen und definiert analoge RS- und BCH-Codes. Die oben schon aufgeführten Eigenschaften, die man von analogen Codes erwartet, werden dort vorgegeben. Diese hier genannten Besonderheiten bilden gleichzeitig den Gegenstand der Erörterungen der Folgekapitel. Kurz werden dort auch Berechnungen der Kanalkapazitäten bei diskretem und analogem Ausgang angegeben. Diese liefern besonders interessante Ergebnisse hinsichtlich der in dem analogen Ausgangssignal enthaltenen Kanalzustandsinformation. Im letzten Kapitel wird dann sogar ein Verfahren vorgestellt, das aus dem Decodierergebnis Rückschlüsse auf den Kanalzustand, bzw. die Störcharakteristik zuläßt.

Im *dritten Kapitel* werden Parallelen zwischen Interpolation, Approximation und Decodierung aufgezeigt. Folgende Interpolationsverfahren sind hier von besonderem Interesse: Pronys exponentielle Interpolation, Lagrange-Interpolation, Least-Squares-Approximation, Approximation mit Kettenbrüchen und das Newton-Verfahren, das jedoch im Folgekapitel genauer erläutert wird.

Dieses *vierte Kapitel* geht auf die höhere Korrekturfähigkeit analoger Codes ein und definiert über die Newton-Interpolation das schon angesprochene ‘neue’ Syndrom. Die Entwicklung von Algorithmen auf der Basis dieses Syndroms zur besseren Ausnutzung der Korrekturfähigkeit bildet sicherlich eine interessante Aufgabenstellung zukünftiger Forschungsarbeit.

Der Hauptteil wird durch das *fünfte Kapitel* gebildet. Es wird dort der Einfluß der Fehleramplitude untersucht, d.h. die Auswirkung von ‘Hintergrundrauschen’ auf die Korrektur von Fehlern ‘größerer’ Amplitude. Dies geschieht mittels Konditionsbetrachtungen und unter Zuhilfenahme der in diesem Kapitel vorgestellten neuen Beschreibung des Berlekamp-Massey-Algorithmus.

Nachdem im fünften Kapitel Rechengenauigkeitseffekte ausgespart werden, widmet sich das *sechste Kapitel* entsprechenden Untersuchungen, welche insbesondere in bezug auf die praktische Verwendbarkeit analoger Codes unerlässlich sind.

Hierauf folgen im *siebten Kapitel* die Erläuterung des Levinson-Durbin-Algorithmus, mit der schon angesprochenen Erweiterung und einer iterativen Lösungsmöglichkeit der Schlüsselgleichung.

Das *achte Kapitel* zeigt abschließend auf, in welchen Fällen der Einsatz analoger Codierung sinnvoll ist und stellt Vergleiche zu einfacher Tiefpaß-Filterung des gestörten Signals an. Hier wird ebenfalls, wie bereits erwähnt, eine Möglichkeit aufgezeigt, die Störcharakteristik des Kanals mit der analogen Decodierung zu ermitteln.

Besonders betont sei noch einmal, daß diese Arbeit sich auch in der Aufgabe versteht, zwischen verschiedenen Disziplinen zu vermitteln, wenngleich dies in zunehmendem Maße durch die Fülle von Veröffentlichungen erschwert wird.

Anhang

A.1 – zu Kapitel 2.3 –

In oben genanntem Kapitel wurde die Eigenschaft analoger Codes angesprochen, bei Wahl einer systematischen Codierung, d.h. einer Codierung, bei der die Informationssymbole unverändert im Codewort wiederzufinden sind, und einer diskreten, endlichen Symbolmenge zur Bildung der Informationssymbole, Prüfsymbole zu generieren, die nicht grundsätzlich Elemente der gewählten Symbolmenge sind. Es wurde darauf hingewiesen, daß diese Eigenschaft z.B. zur Synchronisation genutzt werden könnte.

Hier sollen nun für einige einfache Beispiele, unter Vorgabe von *m-PSK-Punkten* in der komplexen Ebene als Symbolvorrat der Informationsstellen, die möglichen Prüfsymbole aufgetragen werden. Der Darstellung in Bild A.1 liegt eine systematische Codierung zugrunde, die sich direkt aus Gleichung 2.3 ergibt, denn aus

$$c(x) = i(x) \cdot g(x)$$

folgt sofort

$$c_0 + c_1x + \dots + c_{N-K-1}x^{N-K-1} = - \left(c_{N-K}x^{N-K} + \dots + c_{N-1}x^{N-1} \right) \bmod g(x).$$

Diese Beziehung läßt sich durch ein Schieberegister realisieren, wie es z.B. in [6] wiedergegeben wird.

In der Abbildung sind die Informationssymbole durch Kreise, die Prüfsymbole durch Kreuze gekennzeichnet.

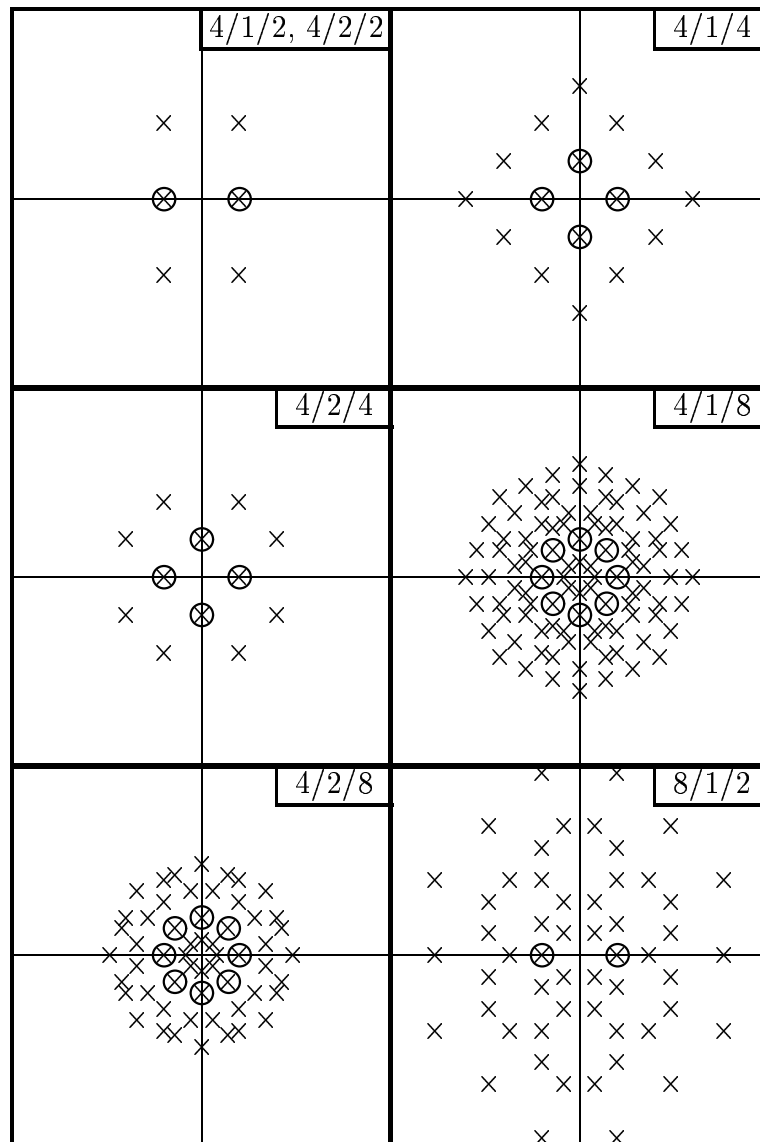


Abbildung A.1: Systematische Codierung bei Vorgabe von m -PSK-Punkten für die Informationsstellen $(N/M/m)$

A.2 – zu Kapitel 2.4 –

II — Zwei Störzustände a , $P(a)$ bekannt, jedoch nicht a selbst —

DMC:

$$C_{DMC} = \max_{P(x_j)} \sum_{j=0}^{Q_j-1} \sum_{i=0}^{Q_i-1} P(x_j) P(y_i|x_j) \text{ld} \frac{P(y_i|x_j)}{P(y_i)}$$

$$P(y_i|x_j) = \sum_{k=0}^{Q_k-1} P(a_k) P(y_i|x_j, a_k)$$

$$P(y_i) = \sum_{j=0}^{Q_j-1} P(x_j) P(y_i|x_j) = \sum_{j=0}^{Q_j-1} \sum_{k=0}^{Q_k-1} P(x_j) P(a_k) P(y_i|x_j, a_k)$$

$$C_{DMC} = \max_{P(x_j)} \sum_{j=0}^{Q_j-1} \sum_{i=0}^{Q_i-1} P(x_j) \sum_{k=0}^{Q_k-1} P(a_k) P(y_i|x_j, a_k) \cdot$$

$$\cdot \text{ld} \frac{\sum_{k'=0}^{Q_k-1} P(a_{k'}) P(y_i|x_j, a_{k'})}{\sum_{j=0}^{Q_j-1} P(x_j) \sum_{k'=0}^{Q_k-1} P(a_{k'}) P(y_i|x_j, a_{k'})}$$

$$P(x_j) = \frac{1}{2} \implies P(y_i) = \frac{1}{2}$$

$$C_{DMC} = \frac{1}{2} \sum_{j=0}^{Q_j-1} \sum_{i=0}^{Q_i-1} \sum_{k=0}^{Q_k-1} P(a_k) P(y_i|x_j, a_k) \text{ld} \left(2 \cdot \sum_{k'=0}^{Q_k-1} P(a_{k'}) P(y_i|x_j, a_{k'}) \right)$$

$$= \frac{1}{2} \sum_{k=0}^{Q_k-1} P(a_k) \sum_{j=0}^{Q_j-1} \sum_{i=0}^{Q_i-1} P(y_i|x_j, a_k) \text{ld} \left(2 \cdot \sum_{k'=0}^{Q_k-1} P(a_{k'}) P(y_i|x_j, a_{k'}) \right)$$

$$= 1 + b \cdot [P_{eb} \text{ld} (bP_{eb} + (1-b)P_{eg}) + (1-P_{eb}) \text{ld} (b(1-P_{eb}) + (1-b)(1-P_{eg}))] +$$

$$+ (1-b) \cdot [P_{eg} \text{ld} (bP_{eb} + (1-b)P_{eg}) + (1-P_{eg}) \text{ld} (b(1-P_{eb}) + (1-b)(1-P_{eg}))]$$

Mit $u_m := bP_{eb} + (1-b)P_{eg} \Rightarrow C_{DMC} = 1 + u_m \text{ld} u_m + (1-u_m) \text{ld} (1-u_m)$

AWGN:

$$C_{AWGN} = \max_{P(x_j)} \sum_{j=0}^{Q_j-1} \int_{-\infty}^{\infty} P(x_j) p(y|x_j) \text{ld} \frac{p(y|x_j)}{p(y)} dy$$

$$p(y|x_j) = \sum_{k=0}^{Q_k-1} P(a_k) p(y|x_j, a_k)$$

$$p(y) = \sum_{j=0}^{Q_j-1} P(x_j) p(y|x_j) = \sum_{j=0}^{Q_j-1} \sum_{k=0}^{Q_k-1} P(x_j) P(a_k) p(y|x_j, a_k)$$

$$P(x_j) = \frac{1}{2}$$

$$p(y) = \frac{1}{2} \cdot b \cdot [p(y|+1, b) + p(y|-1, b)] + \frac{1}{2} \cdot (1-b) \cdot [p(y|+1, g) + p(y|-1, g)]$$

$$p(y|\pm 1) = b p(y|\pm 1, b) + (1-b) p(y|\pm 1, g)$$

$$\begin{aligned} C_{AWGN} &= \frac{1}{2} \int_{-\infty}^{\infty} p(y|+1) \text{ld} \frac{p(y|+1)}{p(y)} dy + \frac{1}{2} \int_{-\infty}^{\infty} p(y|-1) \text{ld} \frac{p(y|-1)}{p(y)} dy \\ &= \int_0^{\infty} p(y|+1) \text{ld} \frac{p(y|+1)}{p(y)} dy + \int_0^{\infty} p(y|-1) \text{ld} \frac{p(y|-1)}{p(y)} dy \end{aligned}$$

III — Zwei Stöorzustände a , $P(a)$ und a selbst bekannt —**DMC:**

$$C_{DMC} = \max_{P(x_j)} \sum_{k=0}^{Q_k-1} \sum_{j=0}^{Q_j-1} \sum_{i=0}^{Q_i-1} P(x_j) P(a_k, y_i|x_j) \text{ld} \frac{P(a_k, y_i|x_j)}{\sum_{m=0}^{Q_j-1} P(x_m) P(a_k, y_i|x_m)}$$

$$= \max_{P(x_j)} \sum_{k=0}^{Q_k-1} \sum_{j=0}^{Q_j-1} \sum_{i=0}^{Q_i-1} P(x_j) \underbrace{P(a_k|x_j)}_{P(a_k)} P(y_i|x_j, a_k) \text{ld} \frac{P(a_k|x_j) P(y_i|x_j, a_k)}{\sum_{m=0}^{Q_j-1} P(x_m) P(a_k|x_m) P(y_i|x_m, a_k)}$$

$$= \max_{P(x_j)} \sum_{k=0}^{Q_k-1} P(a_k) \sum_{j=0}^{Q_j-1} \sum_{i=0}^{Q_i-1} P(x_j) P(y_i|x_j, a_k) \text{ld} \frac{P(y_i|x_j, a_k)}{\sum_{m=0}^{Q_j-1} P(x_m) P(y_i|x_m, a_k)} \quad (*)$$

$$P(x_j) = \frac{1}{2} \implies P(y_i) = \frac{1}{2}$$

$$\implies C_{DMC} = \frac{1}{2} \sum_{k=0}^{Q_k-1} P(a_k) \sum_{j=0}^{Q_j-1} \sum_{i=0}^{Q_i-1} P(y_i|x_j, a_k) \text{ld} (2 \cdot P(y_i|x_j, a_k))$$

$$\begin{aligned}
&= 1 + \frac{1}{2} \sum_{k=0}^{Q_k-1} P(a_k) \sum_{j=0}^{Q_j-1} \sum_{i=0}^{Q_i-1} P(y_i|x_j, a_k) \text{ld}(P(y_i|x_j, a_k)) \\
&= 1 + b \cdot [P_{eb} \text{ld} P_{eb} + (1 - P_{eb}) \text{ld}(1 - P_{eb})] + (1 - b) \cdot [P_{eg} \text{ld} P_{eg} + (1 - P_{eg}) \text{ld}(1 - P_{eg})] \\
&= 1 + b \cdot H(y|x, b) + (1 - b) \cdot H(y|x, g)
\end{aligned}$$

AWGN: $(*) \implies$

$$\begin{aligned}
C_{AWGN} &= \max_{P(x_j)} \sum_{k=0}^{Q_k-1} P(a_k) \sum_{j=0}^{Q_j-1} \int_{-\infty}^{\infty} P(x_j) p(y|x_j, a_k) \text{ld} \frac{p(y|x_j, a_k)}{\sum_{m=0}^{Q_j-1} P(x_m) p(y|x_m, a_k)} dy \\
P(x_j) &= \frac{1}{2}
\end{aligned}$$

$$\begin{aligned}
\implies C_{AWGN} &= \frac{1}{2} b \int_{-\infty}^{\infty} \left[p(y|+1, b) \text{ld} \frac{2p(y|+1, b)}{p(y|+1, b) + p(y|-1, b)} + \right. \\
&\quad \left. + p(y|-1, b) \text{ld} \frac{2p(y|-1, b)}{p(y|+1, b) + p(y|-1, b)} \right] dy + \\
&\quad + \frac{1}{2} (1 - b) \int_{-\infty}^{\infty} \left[p(y|+1, g) \text{ld} \frac{2p(y|+1, g)}{p(y|+1, g) + p(y|-1, g)} + \right. \\
&\quad \left. + p(y|-1, g) \text{ld} \frac{2p(y|-1, g)}{p(y|+1, g) + p(y|-1, g)} \right] dy \\
&= 1 + \frac{b}{2} \int_{-\infty}^{\infty} \left[p(y|+1, b) \text{ld} \frac{p(y|+1, b)}{p(y|+1, b) + p(y|-1, b)} + \right. \\
&\quad \left. + p(y|-1, b) \text{ld} \frac{p(y|-1, b)}{p(y|+1, b) + p(y|-1, b)} \right] dy + \\
&\quad + \frac{(1 - b)}{2} \int_{-\infty}^{\infty} \left[p(y|+1, g) \text{ld} \frac{p(y|+1, g)}{p(y|+1, g) + p(y|-1, g)} + \right. \\
&\quad \left. + p(y|-1, g) \text{ld} \frac{p(y|-1, g)}{p(y|+1, g) + p(y|-1, g)} \right] dy
\end{aligned}$$

IV — Zwei Störzustände a , $P(a)$ bekannt, mit Kanalzustandsschätzung \hat{a} , Schätzfehlerwahrscheinlichkeit $P(\hat{a}|a)$ bekannt —

DMC:

$$C_{DMC} = \max_{P(x_j)} \sum_{k=0}^{Q_k-1} P(\hat{a}_k) \sum_{i=0}^{Q_i-1} \sum_{j=0}^{Q_j-1} P(x_j) P(y_i|x_j, \hat{a}_k) \text{ld} \frac{P(y_i|x_j, \hat{a}_k)}{\sum_{m=0}^{Q_j-1} P(x_m) P(y_i|x_m, \hat{a}_k)}$$

mit

$$\begin{aligned} P(\hat{a}_k) &= \sum_{l=0}^{Q_k-1} P(a_l, \hat{a}_k) = \sum_{l=0}^{Q_k-1} P(a_l) P(\hat{a}_k|a_l) \\ P(y_i|x_j, \hat{a}_k) &= \sum_{l=0}^{Q_k-1} P(a_l) \frac{P(\hat{a}_k|a_l)}{P(\hat{a}_k)} P(y_i|x_j, a_l) \quad (\text{siehe Fußnote}^1) \end{aligned}$$

$$\begin{aligned} P(\hat{a} = G) &= b\epsilon_1 + (1-b)(1-\epsilon_2) \\ P(\hat{a} = B) &= b(1-\epsilon_1) + (1-b)\epsilon_2 \end{aligned}$$

$$\begin{aligned} P(y_i|x_j, \hat{a} = G) &= \frac{b\epsilon_1 P(y_i|x_j, a = B) + (1-b)(1-\epsilon_2) P(y_i|x_j, a = G)}{b\epsilon_1 + (1-b)(1-\epsilon_2)} \\ P(y_i|x_j, \hat{a} = B) &= \frac{b(1-\epsilon_1) P(y_i|x_j, a = B) + (1-b)\epsilon_2 P(y_i|x_j, a = G)}{b(1-\epsilon_1) + (1-b)\epsilon_2} \end{aligned}$$

$$P(x_j) = \frac{1}{2} \implies \sum_{j=0}^{Q_j-1} P(x_j) P(y_i|x_j, \hat{a}_k) = P(y_i|\hat{a}_k) = \frac{1}{2}$$

$$C_{DMC} = 1 + \sum_{k=0}^{Q_k-1} \sum_{i=0}^{Q_i-1} \sum_{j=0}^{Q_j-1} P(\hat{a}_k) \underbrace{P(x_j)}_{1/2} P(y_i|x_j, \hat{a}_k) \text{ld} (P(y_i|x_j, \hat{a}_k))$$

$$P(\hat{a}_k) \cdot P(y_i|x_j, \hat{a}_k) = \sum_{l=0}^{Q_k} P(a_l) P(\hat{a}_k|a_l) P(y_i|x_j, a_l) \quad (\text{Siehe Fußnote}^3)$$

$$\implies \begin{cases} P(\hat{a}_k = G) \cdot P(y_i|x_j, \hat{a}_k = G) = b\epsilon_1 P(y_i|x_j, a_l = B) + \\ \quad + (1-b)(1-\epsilon_2) P(y_i|x_j, a_l = G) \\ P(\hat{a}_k = B) \cdot P(y_i|x_j, \hat{a}_k = B) = b(1-\epsilon_1) P(y_i|x_j, a_l = B) + \\ \quad + (1-b)\epsilon_2 P(y_i|x_j, a_l = G) \end{cases}$$

³folgt aus: $P(\hat{a}_k) P(y_i|\hat{a}_k) = P(\hat{a}_k, y_i) = \sum_{l=0}^{Q_k-1} P(a_l, \hat{a}_k, y_i) = \sum_{l=0}^{Q_k-1} P(a_l) P(\hat{a}_k, y_i|a_l) = \sum_{l=0}^{Q_k-1} P(a_l) P(\hat{a}_k|a_l) \underbrace{P(y_i|a_l, \hat{a}_k)}_{P(y_i|a_l)}$

$$\begin{aligned}
C_{DMC} = & 1 + [b\epsilon_1 P_{eb} + (1-b)(1-\epsilon_2)P_{eg}] \cdot \text{ld} \frac{b\epsilon_1 P_{eb} + (1-b)(1-\epsilon_2)P_{eg}}{b\epsilon_1 + (1-b)(1-\epsilon_2)} + \\
& + [b\epsilon_1(1-P_{eb}) + (1-b)(1-\epsilon_2)(1-P_{eg})] \cdot \\
& \cdot \text{ld} \frac{b\epsilon_1(1-P_{eb}) + (1-b)(1-\epsilon_2)(1-P_{eg})}{b\epsilon_1 + (1-b)(1-\epsilon_2)} + \\
& + [b(1-\epsilon_1)P_{eb} + (1-b)\epsilon_2 P_{eg}] \cdot \text{ld} \frac{b(1-\epsilon_1)P_{eb} + (1-b)\epsilon_2 P_{eg}}{b(1-\epsilon_1) + (1-b)\epsilon_2} + \\
& + [b(1-\epsilon_1)(1-P_{eb}) + (1-b)\epsilon_2(1-P_{eg})] \cdot \\
& \cdot \text{ld} \frac{b(1-\epsilon_1)(1-P_{eb}) + (1-b)\epsilon_2(1-P_{eg})}{b(1-\epsilon_1) + (1-b)\epsilon_2}
\end{aligned}$$

AWGN:

$$C_{AWGN} = \max_{P(x_j)} \sum_{k=0}^{Q_k-1} P(\hat{a}_k) \int_{-\infty}^{\infty} \sum_{j=0}^{Q_j-1} P(x_j) p(y|x_j, \hat{a}_k) \text{ld} \frac{p(y|x_j, \hat{a}_k)}{\sum_{m=0}^{Q_j-1} P(x_m) p(y|x_m, \hat{a}_k)} dy$$

Analog zur Herleitung für den diskreten Fall erhält man

$$C_{AWGN} = 1 + \frac{1}{2} \sum_{k=0}^{Q_k-1} \int_{-\infty}^{\infty} \sum_{j=0}^{Q_j-1} P(\hat{a}_k) p(y|x_j, \hat{a}_k) \text{ld} \frac{p(y|x_j, \hat{a}_k)}{\sum_{m=0}^{Q_j-1} p(y|x_m, \hat{a}_k)} dy.$$

A.3 – zu Kapitel 5.1.2 –

Im folgenden soll die Komplexität der mathematischen Betrachtung des Einflusses normalverteilter Hintergrundrauschens bei der Decodierung aufgezeigt werden.

Vergleichsweise einfach zu beschreiben ist die Bildung des Syndroms über die DFT. Die Frequenzbereichskomponenten ergeben sich als Summe von normalverteilten Zustandsvariablen mit gleicher Standardabweichung. Nach dem *Additionssatz der Normalverteilung* folgt sofort

$$\sigma_S = \sigma_R = \sigma_r \cdot \sqrt{N}.$$

Problematisch ist hingegen die Beschreibung der Lösung des Toeplitz-Systems (Schlüsselgleichung).

Betrachtet man ein Gleichungssystem, dessen Koeffizientenmatrix und rechte Seite normalverteilte Zufallsvariablen enthält und nimmt zusätzlich an, diese seien mittelwertfrei und alle statistisch unabhängig, so läßt sich mit etwas mathematischem Aufwand noch eine geschlossene Lösung für die Verteilungsdichte des Ergebnisvektors angeben.

Es sei die Lösung eines allgemeinen Gleichungssystems

$$\underline{\vec{z}} = \underline{\mathbf{Y}}^{-1} \underline{\vec{x}}$$

betrachtet, wobei $\underline{\mathbf{Y}}$ die Koeffizientenmatrix, $\underline{\vec{x}}$ die rechte Seite und $\underline{\vec{z}}$ den Lösungsvektor darstellt.

Zunächst wird der Einfachheit halber angenommen, die Komponenten seien jeweils reell normalverteilt. Die sich ergebende Änderung für den komplexen Fall wird anschließend angegeben.

Zur Bestimmung der Dichte wird folgender Ansatz gemacht:

$$\begin{aligned} f_{\underline{\vec{z}}}(\underline{\vec{z}}) &= \int_{\underline{\mathbf{Y}}} f_{\underline{\vec{z}}}(\underline{\vec{z}}|\underline{\mathbf{Y}} = \underline{\mathbf{Y}}) \cdot f_{\underline{\mathbf{Y}}}(\underline{\mathbf{Y}}) d\underline{\mathbf{Y}} \\ f_{\underline{\vec{z}}}(\underline{\vec{z}}|\underline{\mathbf{Y}} = \underline{\mathbf{Y}}) &= \frac{f_{\underline{\vec{x}}}(\underline{\mathbf{Y}}\underline{\vec{z}})}{|\det \underline{\mathbf{Y}}^{-1}|} = |\det \underline{\mathbf{Y}}| \cdot f_{\underline{\vec{x}}}(\underline{\mathbf{Y}}\underline{\vec{z}}) \\ f_{\underline{\vec{z}}}(\underline{\vec{z}}) &= \int_{\underline{\mathbf{Y}}} f_{\underline{\vec{x}}}(\underline{\mathbf{Y}}\underline{\vec{z}}) f_{\underline{\mathbf{Y}}}(\underline{\mathbf{Y}}) |\det \underline{\mathbf{Y}}| d\underline{\mathbf{Y}} \end{aligned}$$

$$\begin{aligned}
f_{\vec{x}}(\vec{x}) &= \frac{1}{(\sqrt{2\pi}\sigma)^N} e^{-\frac{1}{2\sigma^2}\vec{x}^T\vec{x}} \\
f_{\underline{Y}}(\underline{Y}) &= \frac{1}{(\sqrt{2\pi}\sigma)^{N^2}} e^{-\frac{1}{2\sigma^2}\sum_{i,k} Y_{i,k}^2} = \frac{1}{(\sqrt{2\pi}\sigma)^{N^2}} e^{-\frac{1}{2\sigma^2}\text{sp}(\underline{Y}^T\underline{Y})} \\
f_{\vec{z}}(\vec{z}) &= \int_{\underline{Y}} \frac{1}{(\sqrt{2\pi}\sigma)^{N^2+N}} e^{-\frac{1}{2\sigma^2}\text{sp}(\underline{Y}^T\underline{Y}) - \frac{1}{2\sigma^2}\text{sp}(\vec{z}^T\underline{Y}^T\underline{Y}\vec{z})} |\det \underline{Y}| d\underline{Y} \\
&= \frac{1}{(\sqrt{2\pi}\sigma)^{N^2+N}} \int_{\underline{Y}} e^{-\frac{1}{2\sigma^2}\text{sp}(\underline{Y}(\underline{1} + \vec{z}\vec{z}^T)\underline{Y}^T)} |\det \underline{Y}| d\underline{Y},
\end{aligned}$$

da $\text{sp}(\underline{Y}^T\underline{Y}) = \text{sp}(\underline{Y}\underline{Y}^T)$.

$\underline{B} := \underline{1} + \vec{z}\vec{z}^T$ ist symmetrisch und positiv definit. Es ist daher eine Aufspaltung in zwei positiv definite Matrizen $\underline{B} = \underline{A} \cdot \underline{A}^T$ eindeutig möglich.

Die Substitution $\underline{U} = \underline{Y} \cdot \underline{A}$ liefert

$$\begin{aligned}
\int \dots d\underline{Y} &= \int \dots |\det \underline{A}|^{-N} d\underline{U} \\
\Rightarrow f_{\vec{z}}(\vec{z}) &= \frac{1}{(\sqrt{2\pi}\sigma)^{N^2+N}} \int_{\underline{U}} e^{-\frac{1}{2\sigma^2}\text{sp}(\underline{U}\underline{U}^T)} |\det \underline{A}|^{-(N+1)} |\det \underline{U}| d\underline{U}
\end{aligned}$$

$$\det \underline{A} = \sqrt{\det(\underline{1} + \vec{z}\vec{z}^T)}$$

$$\det(\underline{1} + \vec{z}\vec{z}^T) = 1 + \text{sp}(\vec{z}\vec{z}^T) = 1 + \sum z_i^2 = 1 + \|\vec{z}\|^2$$

$$\Rightarrow f_{\vec{z}}(\vec{z}) = \text{konst.} \cdot \frac{1}{(1 + \|\vec{z}\|^2)^{\frac{N+1}{2}}}$$

Das Ergebnis lässt sich einfach auf den komplexen Fall übertragen, indem man $2N$ statt N und statt der Quadrate in den Gauss-Dichten Betragsquadrate schreibt.

Bislang wurde Mittelwertfreiheit und statistische Unabhängigkeit vorausgesetzt. Entfällt die Bedingung der Mittelwertfreiheit, so kann keine geschlossene Lösung wie oben angegeben werden. Die Dichten der Matrix und rechten Seite lauten dann (hier werden Zeilenvektoren verwendet):

$$\begin{aligned}
f_{\vec{x}}(\vec{x}) &= \frac{1}{(\sqrt{2\pi}\sigma)^N} e^{-\frac{1}{2\sigma^2}(\vec{x}-\vec{x}_0)(\vec{x}-\vec{x}_0)^T} \\
f_{\underline{Y}}(\underline{Y}) &= \frac{1}{(\sqrt{2\pi}\sigma)^{N^2}} e^{-\frac{1}{2\sigma^2}\sum_{i,k} (Y_{i,k} - Y_{0i,k})^2}.
\end{aligned}$$

Langwierige Umformungen, auf deren Wiedergabe hier verzichtet werden soll, da sie sich ohnehin an die erste Darstellung anlehnen, liefern das Ergebnis

$$f_{\vec{z}}(\vec{z}) = \text{konst.} \cdot e^{\frac{1}{2\sigma^2} \text{SP}((\vec{x}_0^T \vec{z} + \underline{Y}_0^T) \underline{B}^{-1} (\vec{z}^T \vec{x}_0 + \underline{Y}_0))} \cdot \frac{1}{|\det \underline{A}|^{N+1}} \cdot \int_{\underline{U}} \det(\underline{U} + \underline{A}^{-1}(\vec{z}^T \vec{x}_0 + \underline{Y}_0)) e^{-\frac{1}{2\sigma^2} \text{SP}(\underline{U}^T \underline{U})} d\underline{U},$$

wobei $\underline{B} = \underline{1} + \vec{z}^T \vec{z}$ und $\det \underline{A} = \sqrt{1 + \|\vec{z}\|^2}$.

Der Fall komplexer Zufallsvariablen ergibt sich wieder wie oben angedeutet.

Besonders aufwendig gestaltet sich das Problem, wenn man versucht, die im Gleichungssystem gegebenen Abhängigkeiten einzuarbeiten. Hierzu sei das Schlüsselgleichungssystem noch einmal aufgeschrieben:

$$(\Gamma_1, \Gamma_2, \dots, \Gamma_e) \begin{pmatrix} S_{e-1} & & S_{2e} \\ & \ddots & \\ S_0 & & S_{e-1} \end{pmatrix} = (S_e, \dots, S_{2e-1}).$$

Die Abhängigkeiten lassen sich durch Einfügen einer ‘normierten’ Covarianzmatrix \underline{C} wie folgt berücksichtigen:

$$f_{\vec{z}}(\vec{z}) = \int_{\underline{Y}} \frac{|\det \underline{Y}|}{(\sqrt{2\pi}\sigma)^{N^2+N}} e^{-\frac{1}{2\sigma^2} ((\underline{K} - \underline{K}_0) \underline{C}^{-1} (\underline{K} - \underline{K}_0)^T)} d\underline{Y}$$

mit

$$\begin{aligned} \underline{K} &= (Y_{11}, \dots, Y_{1N}, Y_{21}, \dots, Y_{2N}, \dots, Y_{N1}, \dots, Y_{NN}, \vec{x}) \\ \underline{K}_0 &= (Y_{011}, \dots, Y_{01N}, Y_{021}, \dots, Y_{02N}, \dots, Y_{0N1}, \dots, Y_{0NN}, \vec{x}_0) \end{aligned}$$

Betrachtet man die obige Toeplitz-Matrix und die entsprechende rechte Seite, so folgt eine Covarianzmatrix der Form

$$\underline{C} = \left(\begin{array}{cccccccc|cccc} 1 & \leftarrow & N & \rightarrow & 1 & \leftarrow & N & \rightarrow & 1 & \cdots & 0 & 0 & \cdots & 0 & 0 \\ \uparrow & 1 & & 0 & & 1 & & 0 & & & 1 & 0 & \cdots & 0 & 0 \\ N & & 1 & & & & 1 & & & & \uparrow & 1 & & & \\ \downarrow & 0 & & 1 & & & & 1 & & & N & & \ddots & & \\ 1 & & & & \ddots & & & & \ddots & & \downarrow & & & 1 & \\ & \ddots & & & & \ddots & & & & \ddots & 1 & & & & 0 \\ & & \ddots & & & & \ddots & & & & 1 & & & & \\ \hline 0 & 1 & \leftarrow & N & \rightarrow & 1 & & & & & 1 & & & & \\ & & & 1 & & & 1 & & & & & 1 & & & \\ & & & & \ddots & & & \ddots & & & & & \ddots & & \\ & & & & & 1 & & & 1 & & & & & \ddots & \\ & & & & & & 0 & & & 1 & & & & & 1 \end{array} \right) .$$

Als Resume der hier angestellten Betrachtungen läßt sich sagen, daß selbst im mathematisch vergleichsweise leicht faßbaren Fall des normalverteilten Rauschens, es aus Aufwandsgründen praktisch nicht mehr möglich ist, die Lösung der Schlüsselgleichung durch Dichtefunktionen zu beschreiben.

A.4 – zu Kapitel 5.3 –


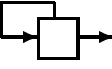

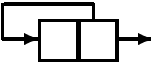
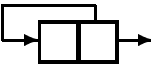
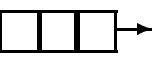
| N | L | $C(D)$ | Schiebereg. | x | $B(D)$ | b | s_N | d |
|-----|-----|-----------|---|-----|-----------|-----|-------|-----|
| 0 | 0 | 1 |  | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | $1 + D$ |  | 1 | 1 | 1 | 0 | 1 |
| 2 | 1 | 1 |  | 2 | 1 | 1 | 1 | 1 |
| 3 | 2 | $1 + D^2$ |  | 1 | 1 | 1 | 0 | 0 |
| 4 | 2 | $1 + D^2$ |  | 2 | 1 | 1 | 0 | 1 |
| 5 | 3 | 1 |  | 1 | $1 + D^2$ | 1 | | |

Abbildung A.2: Masseys Beispiel der Schieberegister-Synthese bei Vorgabe einer Binärsequenz $s_0, s_1, s_2, s_3, s_4 = 1, 0, 1, 0, 0$

Ergänzend wird im folgenden das Originalbeispiel aus Masseys Veröffentlichung [41] (Bild A.2) mit der in Kapitel 5.3 vorgestellten neuen Beschreibung des BMA veranschaulicht. Dies insbesondere, um aufzuzeigen, inwieweit Längenänderungen gemäß Masseys Darstellung hier ebenfalls erkennbar sind.

Neue Matrixbeschreibung:

$$(1, C_1, C_2, C_3) \begin{pmatrix} 0 & 0 & ? & ? \\ 1 & 0 & 0 & ? \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} \stackrel{!}{=} (0, 0, 0, 0)$$

$$(1) \cdot (1) = (1)$$

$$(1, 0) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = (0, 1)$$

$$(1, 0, 0) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = (1, 0, 0)$$

$$(1, 0, 0) - \frac{1}{1}(0, 0, 1) = (1, 0, 1)$$

$$(1, 0, 1) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} = (0, 0, 1)$$

$$(1, 0, 1, 0) \begin{pmatrix} 0 & 0 & ? & ? \\ 1 & 0 & 0 & ? \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} = (0, 1, ?, ?)$$

$$(1, 0, 1, 0) - \frac{1}{1}(0, 0, 1, 0) = (1, 0, 0, 0)$$

Scan including artwork:

$$\begin{aligned}
 (1, C_1, C_2, C_3) \begin{pmatrix} 0 & 0 & ? & ? \\ 1 & 0 & 0 & ? \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} &\stackrel{!}{=} (0, 0, 0, 0) \\
 (1) \cdot (1) &= (1) \\
 (1, 0) \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} &= (0, 1) \\
 (1, 0, 0) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} &= (1, 0, 0) \\
 (1, 0, 0) - \frac{1}{1}(0, 0, 1) &= (1, 0, 1) \\
 (1, 0, 1) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} &= (0, 0, 1) \\
 (1, 0, 1, 0) \begin{pmatrix} 0 & 0 & ? & ? \\ 1 & 0 & 0 & ? \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \end{pmatrix} &= (0, 1, ?, ?) \\
 (1, 0, 1, 0) - \frac{1}{1}(0, 0, 1, 0) &= (1, 0, 0, 0)
 \end{aligned}$$

Verzeichnis der verwendeten Abkürzungen und Formelzeichen

| | |
|------------------------------|---|
| $*$ | konjugiert komplex |
| $\langle a_i, b_i \rangle_i$ | Skalarprodukt von Vektoren \vec{a} und \vec{b} , deren Komponenten mit i indiziert sind |
| $\ \vec{x}\ _\infty$ | Maximumnorm |
| $\ \underline{X}\ _\infty$ | Norm der maximalen Zeilenbetragssumme |
| $\underline{1}$ | Einheitsmatrix |
| a | Kanal- bzw. Störzustände |
| \hat{a} | geschätzter Kanalzustand |
| a_E | Anzahl der Ausfälle (Erasures) |
| <i>AWGN</i> | Additive White Gaussian Noise |
| BMA | Berlekamp-Massey-Algorithmus |
| \mathbb{B}_i | Körper |
| \mathcal{C} | Code |
| c_i | Elemente eines Codewortes ($\vec{c} \in \mathcal{C}$) im ‘Zeitbereich’ |
| C_i | Elemente eines Codewortes im ‘Frequenzbereich’ |
| \vec{c} | Vektor der Länge N mit den Komponenten c_i |
| \vec{C} | Vektor der Länge N mit den Komponenten C_i |
| $c(x)$ | Polynom vom Grad $\leq N - 1$ mit den Koeffizienten c_i |
| $C(x)$ | Polynom vom Grad $\leq N - 1$ mit den Koeffizienten C_i |
| \mathbb{C} | komplexe Zahlen |

| | |
|-------------------------|--|
| C | Kanalkapazität |
| $\Gamma(x)$ | Fehlerstellenpolynom (Frequenzbereich) |
| Γ_i | Koeffizienten des Fehlerstellenpolynoms (Frequenzbereich) |
| $\vec{\Gamma}$ | Vektor aus den Koeffizienten des Fehlerstellenpolynoms (Frequenzbereich) |
| $\gamma(x)$ | In den Zeitbereich transformiertes Fehlerstellenpolynom |
| γ_i | Koeffizienten des Fehlerstellenpolynoms im Zeitbereich |
| $\vec{\gamma}$ | Vektor aus den Koeffizienten des Fehlerstellenpolynoms im Zeitbereich |
| $\bar{\Gamma}(x)$ | Nichtfehlerstellenpolynom |
| $\tilde{\Gamma}_{j,i}$ | Mit Rechenungenauigkeiten behafteter i -ter Koeffizient des Fehlerstellenpolynoms der j -ten Rekursion |
| DFT | diskrete Fourier-Transformation |
| DFT^{-1} | inverse DFT |
| d_i | Diskrepanz im BMA |
| \tilde{d}_i | Mit Rechenungenauigkeiten behaftete Diskrepanz |
| $d_h(\vec{a}, \vec{b})$ | Hamming-Distanz |
| d_h | Mindest-(Hamming)-distanz |
| $d = d_{a,b}$ | ‘analoge’ Hamming-Distanz |
| d_e | Euklid’sche Distanz |
| DMC | Discrete Memoryless Channel |
| δ_{ji} | Kronecker-Delta |
| $\delta_j(\xi)$ | neu definiertes Kronecker-Delta |
| Δ | Abweichung, Differenz |
| $\Delta^k x(i)$ | Operatorschreibweise für eine ‘Divided Difference’ der Newton-Interpolation |
| ID | Menge |
| e | Fehleranzahl |
| E | maximal korrigierbare Fehlerzahl |
| E_b | Energie pro Bit |
| $\operatorname{erf}(x)$ | Error Function $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt$ |
| IE | Indexmenge der Ausfallstellen |
| f_i | Fehlerwerte im Zeitbereich an der Stelle i |
| F_i | Fehlerwerte im Frequenzbereich an der Stelle i |
| \vec{f} | Vektor der Länge N mit den Komponenten f_i |
| \vec{F} | Vektor der Länge N mit den Komponenten F_i |

| | |
|---------------------------|---|
| $f(x)$ | Polynom vom Grad $\leq N - 1$ mit den Koeffizienten f_i |
| $F(x)$ | Polynom vom Grad $\leq N - 1$ mit den Koeffizienten F_i |
| $f[x_i, \dots, x_k]$ | ‘Divided Difference’ der Newton-Interpolation |
| $F_{\mathbf{u}}(u)$ | Verteilungsfunktion |
| $f_{\mathbf{u}}(u)$ | Verteilungsdichte |
| \underline{IF} | Indexmenge der Fehlerstellen |
| \overline{IF} | Indexmenge der fehlerfreien Stellen |
| $GF(q^s)$ | Galoisfeld mit q^s Elementen |
| $g(x)$ | Generatorpolynom |
| $h(x)$ | Prüfpolynom |
| \underline{H} | Prüfmatrix |
| $i(x)$ | Informationspolynom |
| j | imaginäre Einheit $j^2 = -1$ |
| \underline{J} | Permutationsmatrix |
| K | Anzahl der Informationsstellen |
| $\kappa\{\underline{X}\}$ | Konditionszahl |
| ld | Logarithmus zur Basis 2 |
| L_i | Schieberegisterlänge |
| LDA | Levinson-Durbin-Algorithmus |
| LMS | Least-Mean-Square |
| LSQ | Least Squares |
| $\Lambda(x)$ | Ausfallstellenpolynom (Frequenzbereich) |
| Λ_i | Koeffizienten des Ausfallstellenpolynoms (Frequenzbereich) |
| $\vec{\Lambda}$ | Vektor aus den Koeffizienten des Ausfallstellenpolynoms (Frequenzbereich) |
| M | Anzahl der Prüfstellen |
| MDS | Maximum Distance Separable |
| mod | modulo |
| N | Codewortlänge |
| N_0 | einseitige Rauschleistungsdichte |

| | |
|---|---|
| p | Wahrscheinlichkeitsdichte |
| P | Wahrscheinlichkeit |
| P_e | Bitfehlerwahrscheinlichkeit |
| m -PSK | Phase-Shift-Keying mit m Phasenzuständen |
| $R = K/N$ | Coderate |
| r_i | Empfangssymbol an der Stelle i im Zeitbereich |
| R_i | Empfangssymbol an der Stelle i im Frequenzbereich |
| \vec{r} | Vektor der Länge N mit den Komponenten r_i |
| \vec{R} | Vektor der Länge N mit den Komponenten R_i |
| $r(x)$ | Polynom vom Grad $\leq N - 1$ mit den Koeffizienten r_i |
| $R(x)$ | Polynom vom Grad $\leq N - 1$ mit den Koeffizienten R_i |
| \mathbb{R} | Reelle Zahlen |
| $\rho_{\vec{x}} = \frac{\ \Delta \vec{x}\ _{\infty}}{\ \vec{x}_i\ _{\infty}}$ | Relativer Fehler |
| S_i | Syndromstelle (im Frequenzbereich) |
| σ | Standardabweichung |
| sp | Spur einer Matrix |
| T | transponiert |
| TP | Tiefpaß |
| z | primitives Element der multiplikativen Ordnung N |
| $\underline{Z}_{i \times k}$ | DFT -Matrix mit i Zeilen und k Spalten |
| $\underline{Z}_{i \times k}^{-1}$ | Inverse DFT -Matrix mit i Zeilen und k Spalten |
| \mathbb{Z} | Ganze Zahlen |

Literaturverzeichnis

- [1] **Aitken, A.C.**, *Determinanten und Matrizen*, Bibliographisches Institut, Mannheim, Wien, Zürich, 1969.
- [2] **Berlekamp, E.R.**, *Algebraic Coding Theory*, McGraw-Hill, St. Louis, San Francisco, Toronto, London, Sydney, 1968.
- [3] **Berg, L.**, *Lineare Gleichungssysteme mit Bandstruktur*, Carl Hanser Verlag, München, Wien, 1986.
- [4] **Björck, Å., Dahlquist, G.**, *Numerische Methoden*, Oldenbourg Verlag, München, Wien, 1979.
- [5] **Blahut, R.E.**, *Fast Algorithms for Digital Signal Processing*, 2. Aufl., Addison-Wesley, Reading, Menlo Park, Wokingham, Amsterdam, 1987.
- [6] **Blahut, R.E.**, *Theory and Practice of Error Control Codes*, Addison-Wesley, Reading, Menlo Park, Amsterdam, 1983.
- [7] **Blahut, R.E.**, "Transform Techniques for Error Control Codes", *IBM J. Res. Develop.*, Vol. 23, No. 3, S. 299-315, Mai 1979.
- [8] **Bossert, M.**, *Ein Verfahren zur Decodierung von binären linearen Blockcodes über die halbe Mindestdistanz ohne und mit Kanalzustandsinformation*, VDI Verlag, Düsseldorf, 1987.
- [9] **Bossert, M.**, "On Decoding Generalized Concatenated Codes over the Euclidean Space with Binary Linear Outer Codes", *DFG-Abschlußbericht*, 1988.
- [10] **Brent, R.P., Gustavson, F.G., Yun, D.Y.Y.**, "Fast Solution of Toeplitz Systems of Equations and Computation of Padé Approximants", *Journal of Algorithms*, Vol. 1, S. 259-295, 1980.

- [11] **Bresler, Y., Macovski, A.**, "Exact Maximum Likelihood Parameter Estimation of Superimposed Exponential Signals in Noise", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 4, S. 1081-1089, 1986.
- [12] **Cheney, E.W.**, *Approximation Theory*, McGraw-Hill, New York, Toronto, London, 1966.
- [13] **Cheng, U.**, "On the Continued Fraction and Berlekamp's Algorithm", *IEEE Trans. on Information Theory*, Vol. IT-30, No. 3, S. 541-544, Mai 1984.
- [14] **Clark, G.C.C., Cain, J.B.**, *Error-Correction Coding for Digital Communications*, 2. Aufl., Plenum Press, New York, London, 1982.
- [15] **Collatz, L., Meinardus, G.**, *Numerische Methoden der Approximationstheorie*, Birkhäuser Verlag, Basel, Stuttgart, 1972.
- [16] **Cybenko, G.**, "The numerical stability of the Levinson-Durbin Algorithm for Toeplitz systems of equations", *SIAM J. Sci. Stat. Comput.*, Vol. 1, No. 3, S. 303-319, September 1980.
- [17] **Dornstetter, J.L.**, "On the Equivalence Between Berlekamp's and Euclid's Algorithm", *IEEE Trans. on Information Theory*, Vol. IT-33, No. 3, S. 428-431, Mai 1987.
- [18] **Dorsch, B.**, "Vorwärtskorrektur bei zeitvarianten Störungen", *Frequenz*, S. 96-106, 35(1981)3/4.
- [19] **Dorsch, B.**, "Some Coding/Decoding Aspects with Chinese Remainders", *International Workshop*, Varna, Bulgarien, 18-24 Sept. 1988.
- [20] **Durbin, J.**, "The Fitting of Time-Series Models", *Rev. Intern. Statist. Inst.*, Vol. 28, S. 233-244, 1960.
- [21] **Golub, G.H., van Loan, C.F.**, *Matrix Computations*, North Oxford Academic Publishing, Oxford, 1983.
- [22] **Gorenstein, D., Zierler, N.**, "A Class of Error-Correcting Codes in p^m Symbols", *J. SIAM*, Vol. 9, No. 3, S. 207-214, Juni 1961.
- [23] **Hagenauer, J.**, "Zur Kanalkapazität bei Nachrichtenkanälen mit Fading und gebündelten Fehlern", *AEÜ*, Vol. 34, No. 6, S. 229-237, Juni 1980.
- [24] **Hamming, R.W.**, *Numerical Methods for Scientists and Engineers*, McGraw-Hill, Tokyo, Düsseldorf, London, 1973. (neuaufgelegt durch Dover)

- [25] **Hamming, R.W.**, *Introduction to Applied Numerical Analysis*, McGraw-Hill, New York, Düsseldorf, London, 1971.
- [26] **Helstrom, C.W.**, "Topics in the Transmission of Continuous Information", *Westinghouse Research Laboratories*, Research Report 64-8C3-522-R1, Proprietary Class 3, 27. Aug. 1964.
- [27] **Henkel, W.**, "Multiple Error Correction with Analog Codes", *6th International Conference on Applied Algebra, Algebraic Algorithms and Error Correcting Codes (AAECC-6)*, Rom, 4-8 Juli 1988.
- [28] **Henkel, W.**, "Another Description of the Berlekamp-Massey-Algorithm", *IEEE Proc. Part I*, Annahme Dez. 1988, voraussichtl. Erscheinen Frühj. 1989.
- [29] **Henkel, W.**, "An Extended Levinson-Durbin-Algorithm for the Inversion of Toeplitz Matrices", *Electronics Letters (?)*, eingereicht Jan. 1989.
- [30] **Henkel, W.**, "An Extended Berlekamp-Massey-Algorithm for the Inversion of Toeplitz Matrices⁴", *7th International Conference on Applied Algebra, Algebraic Algorithms and Error Correcting Codes (AAECC-7)*, Toulouse, 26-30 Juni 1989, eingereicht Feb. 1989.
- [31] **Hildebrandt, F.B.**, *Introduction to numerical analysis*, McGraw-Hill, New York, Toronto, London, 1956.
- [32] **Hsueh, C.T., Lee, J.Y., Wang, J.F.**, "Using Lagrange's Interpolation Formula to Construct Reed-Solomon Codes over $GF(p)$ ", *Electronics Letters*, Vol. 24, No. 3, S. 174-175, 4. Feb. 1988.
- [33] **Janssen, A.J.E.M., Veldhuis, R.N.J., Vries, L.B.**, "Adaptive Interpolation of Discrete-Time Signals That Can Be Modeled as Autoregressive Processes", *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-34, No. 2, S. 317-330, Apr. 1986.
- [34] **Kailath, T.**, "A View of Three Decades of Linear Filtering Theory", *IEEE Trans. on Information Theory*, Vol. IT-20, No. 2, S. 146-181, März 1974.
- [35] **Kaltofen, E., Lakshman, Y.**, "Improved Sparse Multivariate Polynomial Interpolation Algorithms", *Rensselaer Polytechnic Institute, Technical Report*, Rep. No. 88-17, Juli 1988.

⁴Der Aufsatz beschreibt Ergebnisse, die sich nach Fertigstellung dieser Ausarbeitung aus der neuen Beschreibung des Berlekamp-Massey-Algorithmus ergaben

- [36] **Levinson, N.**, "The Wiener RMS (Root Mean Square) Error Criterion in Filter Design and Prediction", *J. of Mathematics and Physics*, Vol. 25, No. 4, S. 261-278, Jan. 1947.
Abdruck in:
Wiener, N., *Extrapolation, Interpolation and Smoothing of Stationary Time Series, with Engineering Applications*, Technology Press and Wiley, New York, 1949, S. 129 ff..
- [37] **van Lint, J.H.**, *Introduction to Coding Theory*, Springer-Verlag, Berlin, Heidelberg, New York, 1982.
- [38] **Maekawa, Y., Sakaniwa, K.**, "An Extension of DFT Code and the Evaluation of its Performance", *Int. Symp. on Information Theory*, Brighton, England, 24-28 Juni 1985.
- [39] **Mandelbaum, D.M.**, "Construction of Error Correcting Codes by Interpolation", *IEEE Trans. on Information Theory*, Vol. IT-25, No. 1, S. 27-35, Jan. 1979.
- [40] **Marshall, T.G.**, "Real number transform and convolutional codes", in *Proc. 24th Midwest Symp. Circuits Syst.*, S. Karne, Ed., Albuquerque, NM, June 29-30, 1981.
- [41] **Massey, J.L.**, "Shift-Register Synthesis and BCH Decoding", *IEEE Trans. on Information Theory*, Vol. IT-15, No. 1, S. 122-127, Jan. 1969.
- [42] **Mills, W.H.**, "Continued Fractions and Linear Recurrences", *Mathematics of Computation*, Vol. 29, No. 129, S. 173-180, Jan. 1975.
- [43] **Nagumo, J.-I., Noda, A.**, "A Learning Method for System Identification", *IEEE Trans. on Automatic Control*, Vol. AC-12, No. 3, S. 282-287, Juni 1967.
- [44] **Oetken, G.**, *Ein Beitrag zur Interpolation mit digitalen Filtern*, Ausgewählte Arbeiten über Nachrichtensysteme, hrsg. von H.W. Schüssler, Univ. Erlangen-Nürnberg, Nr. 33, 1978.
- [45] **Papoulis, A.**, *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, Singapore, 1984.
- [46] **Peterson, W.W., Weldon, E.J.**, *Error-Correcting Codes*, MIT Press, Cambridge, London, 1972.
- [47] **Piret, P.**, "Bounds for Codes over the Unit Circle", *IEEE Trans. on Information Theory*, Vol. IT-32, No. 6, S. 760-767, Nov. 1986.

- [48] **Proakis, J.G.**, *Digital Communications*, 2. Aufl., McGraw-Hill, Singapore, Hamburg, London, 1985.
- [49] **de Prony, R.**, "Assai experimentale et analytique", *Ecole Polytech.* (Paris), Vol. 1, S. 24-76, Dec. 1795.
- [50] **Rabiner, L.R., Schafer, R.W.**, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, 1978.
- [51] **Robinson, E.A.**, *Multichannel Time-Series Analysis with Digital Computer Programs*, Holden-Day, San Francisco, 1967.
- [52] **Schüßler, H.W.**, *Digitale Systeme zur Signalverarbeitung*, Springer-Verlag, Berlin, Heidelberg, New York, 1973.
- [53] **Stewart, G.W.**, *Introduction to Matrix Computations*, Academic Press, New York, London, 1973.
- [54] **Tewksbury, S.K., Hallock, R.W.**, "Oversampled, Linear Predictive and Noise-Shaping Coders of Order $N > 1$ ", *IEEE Trans. on Circuits and Systems*, Vol. CAS-25, No. 7, S. 436-447, Juli 1978.
- [55] **Törnig, W.**, *Numerische Mathematik für Ingenieure und Physiker*, Bände 1 und 2, Springer-Verlag, Berlin, Heidelberg, New York, 1979.
- [56] **Trench, W.F.**, "An Algorithm for the Inversion of Finite Toeplitz Matrices", *J. SIAM*, Vol. 12, No. 3, S. 512-522, 1964.
- [57] **Tröndle, K.**, "Signalkorrektur, eine Methode zur Beseitigung von Bündelstörungen bei analogen und digitalen Signalübertragungen", *NTG Fachtagung Codierung*, Berlin, 1978.
- [58] **Ungerböck, G.**, "Channel Coding with Multilevel/Phase Signals", *IEEE Trans. on Information Theory*, Vol. IT-28, No. 1, S. 55-67, Jan. 1982.
- [59] **Ungerböck, G.**, "Trellis-Coded Modulation with Redundant Signal Sets", *IEEE Communications Magazine*, Vol. 25, No. 2, S. 5-21, Feb. 1987.
- [60] **Veldhuis, R.N.J.**, "A Method for the Restoration of Burst Errors in Speech Signals", *EUSIPCO*, Den Haag, 2-5 September 1986, S. 403-406.
- [61] **Walsh, J.L.**, *Interpolation and Approximation by Rational Functions*, American Mathematical Society, Rhode Island, 1960.

- [62] **Watson, G.A.**, *Approximation Theory and Numerical Methods*, John Wiley & Sons, Chichester, New York, Brisbane, Toronto, 1980.
- [63] **Welch, L.R., Scholz, R.A.**, "Continued Fractions and Berlekamp's Algorithm", *IEEE Trans. on Information Theory*, Vol. IT-25, No. 1, S. 19-27, Jan. 1979.
- [64] **Widrow, B., McCool, J., Ball, M.**, "The Complex LMS Algorithm", *Proc. of the IEEE*, S. 719-720, Apr. 1975.
- [65] **MacWilliams, F.J., Sloane, N.J.A.**, *The Theory of Error-Correcting Codes*, 4. Aufl., North-Holland, Amsterdam, New York, Oxford, Tokyo, 1977.
- [66] **Wolf, J.K.**, "Analog Codes", *IEEE Int. Conf. on Comm. (ICC '83)*, Boston, MA, USA, 19-22 Juni 1983, Vol. 1, S. 310-12.
- [67] **Wolf, J.K.**, "Decoding of Bose-Chaudhuri-Hocquenghem Codes and Prony's Method for Curve Fitting", *IEEE Trans. on Information Theory*, S. 608, Okt. 1967.
- [68] **Wolf, J.K.**, "Redundancy, the Discrete Fourier Transform, and Impulse Noise Cancellation", *IEEE Trans. on Communications*, Vol. COM-31, No. 3, S. 458-461, März 1983.
- [69] **Zinoviev, V.A., Zyablow, V.V., Portnoy, S.L.**, "Concatenated Methods for Construction and Decoding of Codes in Euclidean Space", *Preprint USSR Academy of Science, Institute for Problems of Information Transmission*, 1987.
- [70] **Zohar, S.**, "Toeplitz Matrix Inversion: The Algorithm of W.F. Trench", *J. of the Association for Computing Machinery*, Vol. 16, No. 4, S. 592-601, Okt. 1969.