



ulm university universität
uulm

Precoding und Spreizen für MIMO-OFDM mit teilweiser Kanalkennntnis

Diplomarbeit

von

Benjamin Baumgärtner

Institut für Informationstechnik
Universität Ulm
August 2007

D/2007/DY/1

Ich versichere, dass ich die vorliegende Diplomarbeit selbständig und ohne unzulässige fremde Hilfe angefertigt und die Satzung der Universität Ulm zur Sicherung guter wissenschaftlicher Praxis beachtet habe.

Ulm, 27.08.2007

(Benjamin Baumgärtner)



Diplomarbeit

Precoding und Spreizen für MIMO-OFDM mit teilweiser Kanalkennntnis

Erläuterungen:

Im Gegensatz zu Single-Input Single-Output (SISO)-Systemen verfügen Multiple-Input Multiple-Output (MIMO)-Systeme über mehrere Empfangs- und Sendeantennen, die Korrelationen aufweisen können. Antennenkorrelationen treten beispielsweise auf, wenn der Abstand der Antennen oder die Cluster-Angle-Ausbreitung um das Antennen-Array klein ist. Stärkere Korrelationen verursachen sowohl größere Interferenzen als auch stärkeres Fading am Empfänger, was die Bitfehlerrate deutlich verschlechtert. Wenn volle oder teilweise Kanalkennntnis am Sender vorhanden ist, kann mit Precoding die Bitfehlerrate am Empfänger reduziert werden. Volle Kanalkennntnis ist jedoch am Sender oft nicht verfügbar, insbesondere wenn sich der Kanal schnell ändert. Antennenkorrelationen hingegen ändern sich langsamer und können somit leichter am Sender zur Verfügung gestellt werden und dementsprechend ausgenutzt werden.

Precoding für nicht-frequenzselektive MIMO-Kanäle mit teilweiser Kanalkennntnis am Sender wurde in vielen Arbeiten untersucht. Die Ziele waren dabei, die Bitfehlerrate zu reduzieren oder die Kanalkapazität zu erhöhen. In dieser Arbeit soll Precoding für OFDM-Übertragungsverfahren über frequenzselektive Kanäle realisiert werden. Precoding auf Basis von Antennenkorrelationskennntnis am Sender um die Bitfehlerrate zu reduzieren wird untersucht, und zwar sowohl für den Fall perfekter als auch nicht perfekter Kennntnisse der Antennenkorrelationen am Sender.

Spreizen ist ein Verfahren, das ohne Kanalkennntnis am Sender die Kanal-Diversität ausnutzt, um die Bitfehlerrate zu verringern. Je nach Spreiz-Verfahren kann die Kanal-Diversität entweder voll oder teilweise ausgenutzt werden. Die Auswirkung des Precodings zusammen mit verschiedenen Spreiz-Verfahren für niedrige, mittlere und stärkere Korrelationsbereiche auf die Bitfehlerrate wird in dieser Arbeit untersucht. Da Interferenzen am Empfänger entstehen, ist Entzerrung für MIMO-OFDM notwendig. Als Detektionsverfahren werden sowohl iterative Entzerrer (Recurrent Neural Network (RNN) Entzerrer und Soft Cholesky Equalizer (SCE)) als auch nicht-iterative Entzerrer (Block Linear Equalizer (BLE) und Block Decision Feedback Equalizer (BDFE)) verglichen. Die Auswirkungen des Precodings, Spreizens und der Interferenzen auf das Leistungsverhalten dieser Entzerrer werden analysiert.

Abgabetermin: 27.08.2007

Bearbeiter: Benjamin Baumgärtner

Betreuer: Dipl.-Ing. Doris Yacoub

Prüfer: Prof. Dr.-Ing. Jürgen Lindner

Katalognr.: D/2007/DY/1

Inhaltsverzeichnis

1	Einleitung	1
2	Digitale Übertragung mit OFDM	3
2.1	Grundlagen vektorwertiger digitale Übertragung	3
2.1.1	Systemmodell	3
2.1.2	Kanalkorrelationen	5
2.1.3	Kanalkapazität	6
2.1.4	Informationen über den Zustand des Kanals	8
2.2	OFDM	9
2.2.1	SISO-OFDM	9
2.2.2	MIMO-OFDM	11
3	Entzerrerkonzepte für Blockübertragungen	13
3.1	Block Linear Equalizer (BLE)	14
3.1.1	Zero Forcing Block Linear Equalizer (ZF-BLE)	14
3.1.2	Minimum Mean Square Error Block Linear Equalizer (MMSE-BLE)	15
3.2	Block Decision Feedback Equalizer (BDFE)	15
3.2.1	Zero Forcing BDFE	16
3.2.2	Minimum Mean Square Error BDFE	18
3.3	Recurrent Neural Network Equalizer (RNN)	18
3.4	Iterative Soft Cholesky Block Decision Feedback Equalizer (SCE)	21
4	Konzepte für codierte Blockübertragungen	25
4.1	Faltungscodierung	25
4.2	Decodierung mit Zuverlässigkeitsinformationen	26
4.2.1	Decodierprinzipien	26
4.2.2	Log-Likelihood-Verhältnisse	27
4.2.3	Symbolweise MAP-Decodierung	28
4.3	Punktierte Faltungscodes	30
4.4	Turbo-Entzerrung	31
4.4.1	Turbo-Entzerrung mit dem RNN-Entzerrer	32
4.4.2	Turbo-Entzerrung mit dem SCE	33
5	Verfahren zur Verbesserung der Leistungsfähigkeit	35
5.1	Spreizverfahren	35
5.1.1	MC-CDM	36
5.1.2	MC-CAFS	37
5.1.3	Simulation	40
5.2	Precoding	41
5.2.1	Nutzung von Kanalkennntnis am Sender	42

5.2.2	Optimierung der paarweisen Fehlerwahrscheinlichkeit (PEP)	42
5.2.3	Precoding für MIMO-OFDM-Übertragungen	43
5.2.4	Einfluss von Precoding auf das Übertragungssystem	45
5.2.5	Simulation	46
5.3	Codierte Übertragung	50
5.3.1	Analyse des Konvergenzverhaltens mit Hilfe von EXIT-Charts	50
5.3.2	Systemmodell	59
5.3.3	Testkanäle	59
5.3.4	Simulationsergebnisse	60
6	Zusammenfassung	75
A	Waterpouring-Prinzip	77
B	Weitere Ergebnisse der codierten Übertragung über Testkanäle	79
	Abbildungsverzeichnis	88
	Tabellenverzeichnis	91
	Abkürzungs- und Symbolverzeichnis	93
	Literaturverzeichnis	97

1 Einleitung

Multiple-Input Multiple-Output (MIMO)-Übertragungssysteme verwenden mehrere Sende- und Empfangsantennen und können wesentlich höhere Kanalkapazitäten erreichen als SISO-Systeme, die jeweils über nur eine Antenne verfügen [16][34]. Dies ist möglich, indem die räumliche Dimension bei der Übertragung ausgenutzt wird. Ihre maximale Kapazität erreichen MIMO-Systeme nur, wenn der Übertragungskanal bereits am Sender vollständig bekannt ist [26]. In der Praxis ist dies jedoch oft nicht möglich. In diesem Fall wurde in der Vergangenheit meist davon ausgegangen, dass überhaupt keine Kanalkennntnis am Sender vorliegt. Doch ebenso dieses Extrem ist in vielen Fällen nicht praxisgerecht. Auch wenn sich der Kanal schnell ändert, kann er statistische Eigenschaften aufweisen, die sich wesentlich langsamer ändern als der Kanal selbst. Dies sind zum Beispiel dessen Mittelwert und die Antennenkorrelationen. Die Kenntnis solcher, über einen längeren Zeitraum als der Kanal selbst gültiger Eigenschaften wird als partielle Kanalkennntnis bezeichnet. Mit ihnen kann der Sender sein Signal zwar nicht optimal an den Kanal anpassen, wie dies bei voller Kanalkennntnis möglich wäre. Die Kenntnis zumindest einiger Eigenschaften des Kanals erlaubt es jedoch, das Signal so gut wie möglich an den Kanal anzupassen und somit beispielsweise die Bitfehlerrate zu senken oder die Kanalkapazität zu erhöhen. Das Anpassen des Signals an den Kanal mit Hilfe von Kanalinformationen wird als Precoding bezeichnet und war bereits Gegenstand zahlreicher Veröffentlichungen [28][36][37].

Antennenkorrelationen treten beispielsweise auf, wenn sich im Funkausbreitungsfeld zu wenig Streu- und Reflexionsquellen befinden, wenn die Antennenelementabstände in Sender oder Empfänger zu klein sind oder wenn eine starke Sichtverbindung besteht. In diesem Fall sind die Kanäle der einzelnen Antennen nicht unabhängig und verhalten sich ähnlich. Dies kann zu erheblichen Verlusten bei der Kanalkapazität führen. Die negativen Auswirkungen von Antennenkorrelationen wurden schon vielfach untersucht [39][43][26].

OFDM ist ein Übertragungsverfahren, das sehr robust gegen Mehrwegeausbreitung ist und in Verbindung mit MIMO zum Aufbau leistungsfähiger Übertragungssysteme verwendet werden kann [23][40]. Die Leistung lässt sich weiter steigern, wenn Spreizverfahren eingesetzt werden, um die Diversität zu erhöhen. Die Leistungsfähigkeit von gespreiztem MIMO-OFDM wurde ebenfalls bereits ausführlich untersucht [41][42][43].

Ziel dieser Arbeit ist es, Precoding auf Basis der Kenntnis der Kanalkorrelationen in einem MIMO-OFDM-Übertragungssystem zu implementieren. Damit soll die Leistung verschiedener Konstellationen von Spreizverfahren und Entzerrern im Empfänger verglichen und untersucht werden, wie sich diese Verfahren in einem gemeinsamen System verhalten. Dabei wird ebenso die Abhängigkeit von der Stärke der Korrelationen, als auch der Einfluss auf eine codierte Übertragung mit einem Turbo-Entzerrer untersucht. Somit soll herausgefunden werden, in welchen Fällen die Verwendung von Precoding auf Basis der Kanalkorrelationen

zu einer Verbesserung der Übertragung führt und in welchen Situationen sein Einsatz nicht sinnvoll ist.

In Kapitel 2 werden die Grundlagen digitaler vektorwertiger Übertragung vorgestellt und wichtige Eigenschaften von MIMO-Kanälen wie Kapazität, Korrelation und Kanalkennntnis erläutert. In der zweiten Hälfte des Kapitels wird das Übertragungsverfahren OFDM, das die Grundlage aller Übertragungen in dieser Arbeit bildet, zunächst für den SISO-Fall eingeführt und auf den MIMO-Fall erweitert. In Kapitel 3 werden verschiedene Entzerrerkonzepte vorgestellt, die im Empfänger zum Einsatz kommen können und im Rahmen dieser Arbeit verglichen werden sollen. Kapitel 4 erweitert das vorgestellte Übertragungssystem um Kanalcodierung. Dazu werden zunächst Grundlagen der Kanalcodierung- und decodierung vorgestellt. Danach werden Decodierung und Entzerrung zu einem Turbo-Entzerrer verbunden. In Kapitel 5 werden Möglichkeiten, die Performance von MIMO-OFDM-Übertragungen zu verbessern, betrachtet und ihre Wirkung mit Simulationen untersucht. Dabei handelt es sich um Spreizverfahren, Precoding und die codierte Übertragung. Der Prozess der Turbo-Entzerrung wird zusätzlich mit Hilfe von EXIT-Charts untersucht. Die Ergebnisse dieser Arbeit sind in Kapitel 6 zusammengefasst.

In dieser Arbeit werden Größen im Zeitbereich durch Kleinbuchstaben und im Frequenzbereich durch Großbuchstaben bezeichnet. Vektoren werden durch einfache, Matrizen durch zweifache Unterstreichung gekennzeichnet.

2 Digitale Übertragung mit OFDM

Dieses Kapitel behandelt die theoretischen Grundlagen der digitalen vektorwertigen Übertragung, die die Basis dieser Arbeit bilden. Nach der Einführung eines allgemeinen Modells zur vektorwertigen digitalen Übertragung werden Antennenkorrelationen in MIMO-Kanälen und die Kapazität von MIMO-Kanälen vorgestellt. Danach wird das Übertragungsverfahren OFDM zunächst für den SISO-Fall eingeführt und auf den MIMO-Fall ausgeweitet.

2.1 Grundlagen vektorwertiger digitale Übertragung

Im Unterschied zur Übertragung in einem SISO-System, bei dem Sende- und Empfangssymbole Skalare sind, werden bei vektorwertiger Übertragung in einem MIMO-System Vektoren gesendet und empfangen. Jedes Vektorelement wird dabei über eine Antenne einer Sendeantennengruppe und einer Empfangsantennengruppe übertragen, die zusammen einen Matrixkanal aufspannen (siehe Abb. 2.1). Für eine ausführliche Einführung in die skalare und vektorwertige digitale Übertragung sei auf [23] verwiesen.

2.1.1 Systemmodell

Der Zusammenhang zwischen einem Empfangsvektor $\underline{g}(t)$ und einem Sendevektor $\underline{s}(t)$ ist gegeben durch [23]

$$\underline{g}(t) = \underline{h}(t) * \underline{s}(t) + \underline{n}(t), \quad (2.1)$$

wobei $\underline{n}(t)$ ein AWGN-Rauschvektor ist und $\underline{h}(t)$ die Kanalmatrix der Form

$$\underline{h}(t) = \begin{bmatrix} h_{11}(t) & \cdots & h_{1n_T}(t) \\ h_{21}(t) & \cdots & h_{2n_T}(t) \\ \vdots & \ddots & \vdots \\ h_{n_R1}(t) & \cdots & h_{n_Rn_T}(t) \end{bmatrix} \quad (2.2)$$

und Größe $n_R \times n_T$, wobei h_{ij} der Kanalstoßantwort zwischen der i -ten Empfangs- und der j -ten Sendeantenne entspricht (siehe Abb. 2.1). Für die h_{ij} werden in dieser Arbeit Rayleigh-Kanäle angenommen. Das bedeutet, dass $h_{ij} = \alpha + j\beta$, wobei α, β unabhängig und normalverteilt sind mit Mittelwert $\mu = 0$ und Varianz $\sigma^2 = 1$. Für die $|h_{ij}|$ ergibt sich dann eine Rayleigh-Verteilung.

Das Blockschaltbild des Systemmodells ist in Abb. 2.2 dargestellt. Die codierten Sende-symbole $\underline{x}(t)$ werden mit einer Matrix $\underline{u}(t)$ multipliziert, die die Elementarsignale auf ihrer Hauptdiagonalen trägt. Nach der Übertragung der Sendevektoren $\underline{s}(t)$ über den Matrixkanal $\underline{h}(t)$ wird der Empfangsvektor $\underline{g}(t)$ mit einem Kanal-Matched-Filter detektiert, dessen Matrix $\underline{v}(t)$ als [23]

$$\underline{v}(t) = \underline{h}(t) * \underline{u}(t) \quad (2.3)$$

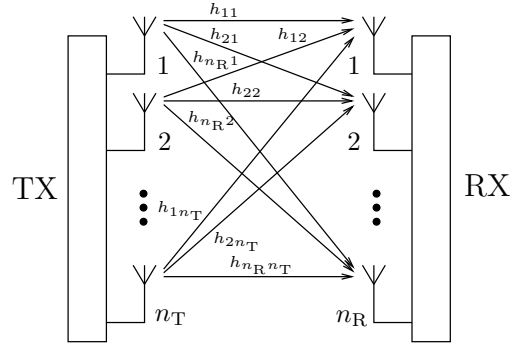


Abbildung 2.1: Modell des MIMO-Kanals: Die n_T Sendeantennen und die n_R Empfangsantennen erzeugen eine Kanalmatrix der Größe $n_R \times n_T$

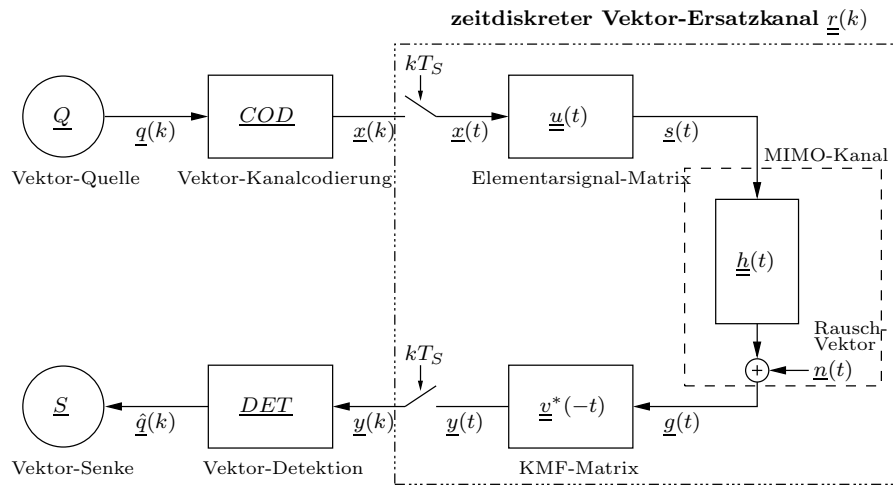


Abbildung 2.2: Systemmodell für eine vektorwertige Übertragung mit linearen Modulationsverfahren

gebildet wird. Dies wird auch als *Maximum Ratio Combining* (MRC) bezeichnet. Wie bei skalaren Übertragungen können bei vektorwertiger Übertragung Elementarsignalmatrix $\underline{u}(t)$, MIMO-Kanal $\underline{h}(t)$ und KMF-Matrix $\underline{v}(t)$ zu einem zeitdiskreten Ersatzkanal $\underline{r}(k)$ zusammengefasst werden (siehe Abb. 2.2), für $\underline{r}(k)$ gilt dann [23]:

$$\underline{r}(k) = \underline{v}^H(-t) * \underline{v}(t)|_{t=kT_s} \quad (2.4)$$

Somit vereinfacht sich der Zusammenhang zwischen Sendesymbolvektor $\underline{x}(k)$ und dem geschätzten Empfangsvektor $\underline{y}(k)$ zu

$$\underline{y}(k) = \underline{r}(k) * \underline{x}(k) + \underline{\tilde{n}}(k), \quad (2.5)$$

wobei $\underline{r}(k)$ eine Matrix mit zeitdiskreten Ersatzkanal-Stoßantworten $r_{ij}(k)$ mit $i, j = 1, \dots, n_T$ und $\underline{\tilde{n}}(k)$ ein Rauschvektor mit der spektralen Leistungsdichte [23]

$$\varphi_{\tilde{n}\tilde{n}}(k) = N_0 \underline{r}(k) \quad (2.6)$$

ist.

2.1.2 Kanalkorrelationen

Reale Kanäle sind nicht Rayleigh-verteilt, wenn beispielsweise der Abstand zwischen Sender und Empfänger sehr groß ist gegenüber dem Elementabstand der Antennenarrays oder wenn im Ausbreitungsfeld nur wenig Streukörper vorhanden sind. In diesen Fällen weisen die h_{ij} Ähnlichkeiten in ihren Eigenschaften auf, was statistisch durch zwei positiv semidefinite und hermitesche Matrizen von Korrelationskoeffizienten beschrieben werden kann: Die Senderkorrelationsmatrix $\underline{\underline{k}}_T$ der Größe $n_T \times n_T$ und die Empfängerkorrelationsmatrix $\underline{\underline{k}}_R$ der Größe $n_R \times n_R$ von der Form

$$\underline{\underline{k}} = \begin{bmatrix} 1 & \rho_{12} & \rho_{13} & \cdots & \rho_{1j} \\ \rho_{21} & 1 & \rho_{23} & \cdots & \rho_{2j} \\ \rho_{31} & \rho_{32} & 1 & \cdots & \rho_{3j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho_{i1} & \rho_{i2} & \rho_{i3} & \cdots & 1 \end{bmatrix} \quad \text{mit } i, j = 1 \dots n_T \text{ bzw. } n_R \quad (2.7)$$

Die Elemente ρ_{ij} entsprechen im Falle der Senderkorrelationsmatrix $\underline{\underline{k}}_T$ der Korrelation zwischen der i -ten und j -ten Sendeantenne, im Falle der Empfängerkorrelationsmatrix $\underline{\underline{k}}_R$ der Korrelation zwischen der i -ten und j -ten Empfangsantenne und sind im Allgemeinen komplexwertig. Es gilt $\rho_{ji} = \rho_{ij}^*$, da die Matrix hermitesch ist. Die Hauptdiagonalelemente $\rho_{i=j}$ beschreiben die Korrelation einer Antenne mit sich selbst und sind identisch 1. Aus einem bekannten Kanal $\underline{\underline{h}}$ werden diese statistischen Größen nach [22]

$$\underline{\underline{k}}_R = \frac{1}{n_T} E \{ \underline{\underline{h}} \underline{\underline{h}}^H \} \quad \underline{\underline{k}}_T = \frac{1}{n_R} E \{ \underline{\underline{h}}^H \underline{\underline{h}} \} \quad (2.8)$$

berechnet, wobei der Erwartungswert über die zeitlich aufeinander abfolgenden Kanalrealisationen gebildet wird.

Korrelationsmodelle

Zur Simulation korrelierter Kanäle existieren verschiedene Modelle zur Bildung von Korrelationsmatrizen $\underline{\underline{k}}_T$ und $\underline{\underline{k}}_R$. Zwei gebräuchliche Modelle sind

- Matrizen mit konstanten Korrelationskoeffizienten und
- Matrizen mit exponentiell abfallenden Korrelationskoeffizienten.

Bei ersteren wird angenommen, dass die Korrelation zwischen beliebigen Antennenpaaren identisch ist und die Elemente ρ_{ij} der Matrix (Gl. 2.7) werden nach der Vorschrift [24]

$$\rho_{ij} = \begin{cases} 1, & \text{wenn } i = j \\ \rho_{\text{const}}, & \text{wenn } i \neq j \end{cases} \quad \forall i, j = 1 \dots n_T \text{ bzw. } n_R \quad (2.9)$$

gebildet. Bei zweiteren wird angenommen, dass die Korrelation zwischen den jeweiligen Antennenpaaren mit wachsender Differenz der Indizes i und j exponentiell abnimmt, weil diese größere Differenz beispielsweise einen größeren Elementabstand in einem linearen Antennenarray widerspiegelt. Die ρ_{ij} werden gebildet nach [24]

$$\rho_{ij} = \begin{cases} \rho^{j-i}, & \text{wenn } i \leq j \\ \rho_{ji}^*, & \text{wenn } i > j \end{cases} \quad \forall i, j = 1 \dots n_T \text{ bzw. } n_R, \quad |\rho| \leq 1 \quad (2.10)$$

wobei ρ der Korrelationskoeffizient zwischen zwei benachbarten Antennenpaaren ($j - i = 1$) ist. Ein korrelierter Kanal $\underline{h}_{\text{corr}}$ kann nun modelliert werden als [26]

$$\underline{h}_{\text{corr}} = \underline{k}_{\text{R}}^{\frac{1}{2}} \underline{h}_w \underline{k}_{\text{T}}^{\frac{1}{2}}, \quad (2.11)$$

mit einem unkorrelierter Kanal \underline{h}_w . Dabei wird angenommen, dass sich die Korrelationen am Sender und am Empfänger nicht gegenseitig beeinflussen, dass die Pfade selbst unkorreliert sind und dass die Antennenkorrelationen unabhängig vom Pfad sind.

2.1.3 Kanalkapazität

Shannon hat in seiner grundlegenden Arbeit der Informationstheorie gezeigt, dass für einen Kanal mit der Kapazität C und eine Quelle mit der Entropie $H(X) \leq C$ eine Codierung existiert, die es ermöglicht, Quellenfolgen $q_x(k)$ mit beliebig kleiner Fehlerwahrscheinlichkeit zur Senke zu übertragen. Das bedeutet, dass auch über einen gestörten Kanal fehlerfreie Übertragung möglich ist, solange die Kanalkapazität nicht überschritten und eine geeignete Codierung verwendet wird. Die Kanalkapazität wird in der Einheit *bit* angegeben, was äquivalent ist zu *bit/s/Hz*. Für einen SISO-AWGR-Kanal ist die Kanalkapazität gegeben als [27]

$$C = \log_2(1 + SNR) \quad (2.12)$$

mit dem mittleren SNR im Empfänger.

MIMO-Kanäle können wesentlich höhere Kanalkapazitäten erreichen als SISO-Kanäle. Eine ausführliche Betrachtung und Herleitung der Kapazität von MIMO-Kanälen ist in [34] [16][18][26] zu finden. Die wesentlichen Punkte sind im Folgenden wiedergegeben.

Bei der Betrachtung der Kanalkapazität eines MIMO-Systems können die zwei Fälle unterschieden werden, ob am Sender Kanalkennntnis vorliegt oder nicht. In beiden Fällen wird perfekte Kanalkennntnis am Empfänger angenommen.

Ohne Kanalkennntnis am Sender

Wenn der Sender keine Kanalkennntnis hat und damit keine Information darüber, welche Ausbreitungswege gut oder schlecht sind, bleibt ihm nur die Möglichkeit, die Komponenten des Sendevektors statistisch unabhängig zu wählen und die Leistung gleichmäßig über alle Sendeantennen zu verteilen, um die maximale Kapazität zu erreichen. Die Kapazität kann berechnet werden als [34]

$$C = \log_2 \det \left(\underline{I}_{n_{\text{R}} \times n_{\text{R}}} + \frac{E_s}{n_{\text{T}} N_0} \underline{h} \underline{h}^H \right) \quad (2.13)$$

$$= \log_2 \det \left(\underline{I}_{n_{\text{R}} \times n_{\text{R}}} + \frac{E_s}{n_{\text{T}} N_0} \Lambda \right) \quad (2.14)$$

$$= \sum_{i=1}^r \log_2 \left(1 + \frac{E_s}{n_{\text{T}} N_0} \lambda_i \right), \quad (2.15)$$

wobei $r = \text{rang}(\underline{h})$, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{n_{\text{R}}})$ ($\lambda_i \geq 0$) und λ_i ($i = 1, 2, \dots, r$) die positiven Eigenwerte von $\underline{h} \underline{h}^H$ sind. Gl. 2.15 beschreibt die Kapazität des MIMO-Kanals als Summe von

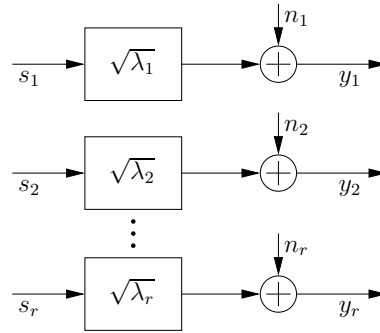


Abbildung 2.3: MIMO-Kanal vom Rang r mit Kanalkennntnis, modale Zerlegung: Es bilden sich r Moden aus, die für parallele Datenübertragung genutzt werden können.

r SISO-Kanälen mit Gewinn λ_i ($i = 1, 2, \dots, r$) und Sendeleistung E_s/n_T , wenn E_s definiert ist als gesamte am Sender zur Verfügung stehende Leistung. Man kann sich somit vorstellen, dass der Einsatz mehrerer Sende- und Empfangsantennen mehrere skalare Übertragungskanäle ausbildet, die als Moden bezeichnet werden. Ohne Kenntnis des Kanals sind diese Moden jedoch nicht explizit bekannt. Es kann somit nicht direkt auf einzelne Moden zugegriffen oder ihnen individuelle Sendeleistung zugeordnet werden [26].

Mit Kanalkennntnis am Sender

Dies ändert sich, wenn der Sender über Kanalkennntnis verfügt. Eine Eigenschaft von Eigenvektoren einer Abbildung ist, dass ihre Richtung durch die Abbildung nicht verändert wird, sondern sie lediglich mit einem Faktor skaliert werden, der dem zugehörigen Eigenwert entspricht [46]. Die Eigenvektoren der Kanalmatrix beschreiben somit Moden des Kanals, über die verzerrungsfrei mit einer Skalierung übertragen werden kann und die als Eigenmoden bezeichnet werden. Jeder Mode kann explizit Sendeleistung zugewiesen werden. Die Kanalkapazität entspricht der Summe der Kapazität von r SISO-Kanälen jeweils mit Verstärkung λ_i , die für parallele Datenübertragung genutzt werden können, wie in Abb. 2.3 dargestellt ist. Dies wird als modale Zerlegung (engl. *modal decomposition*) des Kanals bezeichnet. Die Kanalkapazität berechnet sich dann aus [26]

$$C = \sum_{i=1}^r \log_2 \left(1 + \frac{E_s \gamma_i}{n_T N_0} \lambda_i \right), \quad (2.16)$$

wobei γ_i die Sendeleistung des i -ten Subkanals festlegt und $\sum_{i=1}^r \gamma_i = n_T$ ist. Die Kanalkapazität kann nun durch die Wahl geeigneter γ_i maximiert werden. Eine Lösung für das Optimierungsproblem [26]

$$C = \max_{\sum_{i=1}^r \gamma_i = n_T} \sum_{i=1}^r \log_2 \left(1 + \frac{E_s \gamma_i}{n_T N_0} \lambda_i \right) \quad (2.17)$$

und damit die optimale Energieverteilung γ_i^{opt} für maximale Kapazität liefert der Water-pouring-Algorithmus, der in Anhang A näher erläutert wird. Neben der Maximierung der Kanalkapazität wird dort auch die Minimierung der Bitfehlerrate als Optimierungsziel beschrieben. Die Sendeleistung E_s ist hier auf den gesamten Sender normiert.

Kapazität korrelierter Kanäle

Bei den Betrachtungen der Kanalkapazität in den vorherigen Abschnitten wurde von einem unkorrelierten Kanal ausgegangen. Die in Abschnitt 2.1.2 vorgestellten Antennenkorrelationen können jedoch die Kanalkapazität deutlich vermindern. Ihr Einfluss auf den Kanal kann nach Gl. 2.11 modelliert werden. Durch Einsetzen von Gl. 2.11 in Gl. 2.13 erhält man

$$C = \log_2 \det \left(\underline{I}_{n_R \times n_R} + \frac{E_s}{n_T N_0} \underline{k}_R^{\frac{1}{2}} \underline{h}_w \underline{k}_T \underline{h}_w^H \underline{k}_R^{\frac{H}{2}} \right). \quad (2.18)$$

Unter der Annahme, dass $n_R = n_T = n$ und dass \underline{k}_R und \underline{k}_T vollen Rang haben, lässt sich dies bei hohem SNR vereinfachen zu [26]

$$C \approx \log_2 \det \left(\frac{\rho}{n} \underline{h}_w \underline{h}_w^H \right) + \log_2 \det \left(\underline{k}_R \right) + \log_2 \det \left(\underline{k}_T \right). \quad (2.19)$$

Es lässt sich für \underline{k}_T und für \underline{k}_R analog zeigen [26], dass $\log_2 \det \left(\underline{k}_T \right) \leq 0$ und nur dann Null wird, wenn $\underline{k}_T = \underline{I}$. Dies zeigt, dass Kanalkorrelationen, wenn sie vorhanden sind, stets zu einem Verlust bei der Kanalkapazität führen.

2.1.4 Informationen über den Zustand des Kanals

Bisher wurde die Kapazität von MIMO-Kanälen unter der Annahme betrachtet, dass der Empfänger über perfekte Kanalkennntnis verfügt, während der Sender entweder keine oder ebenfalls perfekte Kanalkennntnis besitzt. Im ersten Fall ist eine gleichmäßige Verteilung der Sendeleistung über die Subkanäle optimal, eine Verbesserung der Kanalkapazität oder Bitfehlerrate ist jedoch nur möglich, wenn auch der Sender über Kanalkennntnis verfügt.

In Kommunikationssystemen kann die Übertragungsfunktion des Kanals geschätzt werden, in dem eine bekannte Symbolfolge gesendet wird. Der Empfänger kann durch Vergleich bestimmen, wie die Symbole durch den Kanal verändert wurden und diese Information bei der Decodierung des Empfangssignals verwenden. Der Sender hingegen kann selbst keine Informationen über den Kanal gewinnen.

In Systemen, die in beide Richtungen übertragen, in denen beide Teilnehmer also sowohl Sender als auch Empfänger sind, kann die Reziprozität ausgenutzt werden. Dies ist die Annahme, dass der Kanal bei Übertragung von Teilnehmer A zu Teilnehmer B identisch ist wie der Kanal bei Übertragung von Teilnehmer B zu Teilnehmer A. Verwendet das System Zeitduplex (engl. *time division duplex*, TDD), so ist die Kanalschätzung, die B in der Rolle des Empfängers vorgenommen hat eventuell schon veraltet, wenn er in die Rolle des Senders wechselt. Verwendet das System Frequenzduplex (engl. *frequency division duplex*, FDD), bezieht sich die Kanalschätzung auf die Empfangsfrequenz, die von der Sendefrequenz verschieden ist. Die Genauigkeit der Kanalschätzung für den Sender nimmt folglich bei frequenzselektiven Kanälen bzw. Kanälen mit kleiner Kohärenzzeit ab. Als zweite Möglichkeit kann der Empfänger die Kanalschätzung auf einem Rückkanal dem Sender mitteilen. Diese Information ist jedoch mit einer Verzögerung behaftet und vergrößert den Overhead des Systems [36].

Während die Schätzung des Kanals selbst mit den genannten Ungenauigkeiten verbunden ist, ändern sich statistische Eigenschaften des Kanals wie beispielsweise Mittelwert und Antennenkorrelationen (siehe Abschnitt 2.1.2) in vielen Systemen nicht oder nur sehr langsam im Vergleich zur Kohärenzzeit des Kanals. Die Kenntnis dieser beiden Eigenschaften,

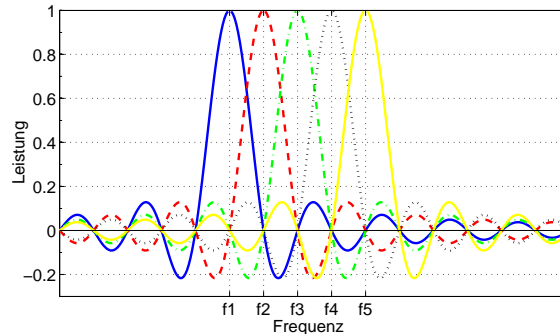


Abbildung 2.4: OFDM-Elementarsignale im Frequenzbereich: Fünf Subträger mit minimal möglichem Abstand $\Delta f = \frac{1}{T}$.

Mittelwert des Kanals und Korrelationsmatrizen \underline{k}_T und \underline{k}_R , wird als partielle Kanalkennntnis am Sender (engl. *Partial Channel State Information at the Transmitter*, CSIT) bezeichnet. Da in dieser Arbeit ein Rayleigh-Kanalmodell verwendet wird, ist der Mittelwert des Kanals im Folgenden a priori $\mu = 0$.

2.2 OFDM

OFDM (engl. *Orthogonal Frequency Division Multiplex*) ist ein Übertragungsverfahren, das nicht nur einen Träger verwendet, sondern die ihm zur Verfügung stehende Bandbreite in eine Vielzahl von Subkanälen unterteilt. Die Träger liegen dabei im Frequenzbereich so eng beisammen (siehe Abb. 2.4), dass sie sich überlappen. Sie sind aufgrund ihrer Orthogonalität zu einander jedoch wieder trennbar. Durch die Zerlegung des im Allgemeinen frequenzselektiven Kanals erhält man Subkanäle, die weit weniger frequenzselektives Fading aufweisen bzw. sogar als nicht-frequenzselektiv (engl. *flat fading*) betrachtet werden können. Parallele Datenübertragung über schmalbandige Subkanäle bedeutet eine längere Symboldauer als bei einer seriellen Übertragung über einen breitbandigen Kanal. Um die Auswirkung von Mehrwegeausbreitung abzuschätzen, betrachtet man die Symboldauer im Verhältnis zur Mehrwegeverbreiterung (engl. *delay spread*), also der Differenz zwischen dem direkten und dem längsten Ausbreitungspfad. OFDM schneidet hierbei aufgrund der längeren Symboldauer günstiger ab als Einträgerverfahren und ist somit resistenter gegen Mehrwegeausbreitung. Durch die dichte Packung der Subträger im Frequenzbereich ist die Bandbreitenausnutzung hoch. Die Realisierung dieses Verfahrens ist mit Hilfe der schnellen Fouriertransformation (engl. *Fast Fourier Transform*, FFT) im Empfänger und ihrer inversen (IFFT) im Sender mit relativ geringem Aufwand möglich. Eine ausführliche Einführung in OFDM ist in [23] zu finden.

2.2.1 SISO-OFDM

Die Elementarsignale für eine OFDM-Übertragung lassen sich im äquivalenten Tiefpassbereich definieren als [23]

$$e_{Ti}(t) = \text{rect}\left(\frac{t}{T}\right) e^{j2\pi f_i t} \quad \forall i = 1, 2, \dots, N. \quad (2.20)$$

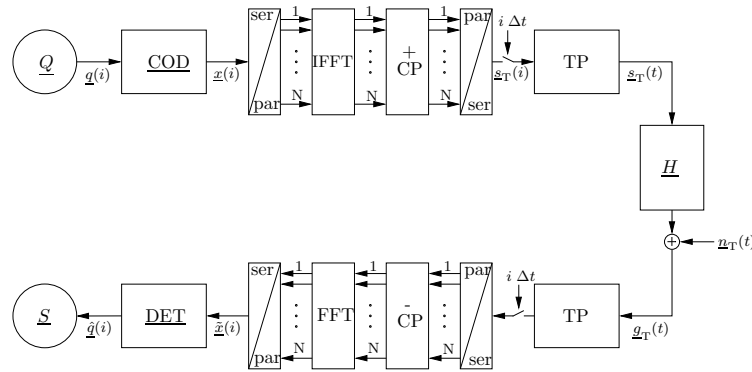


Abbildung 2.5: Übertragungsmodell für SISO-OFDM

Mit

$$f_i = f_0 + i \frac{1}{T} \quad \text{und} \quad T \leq T_s \quad (2.21)$$

ergibt sich ein System orthogonaler Elementarsignale, das im Frequenzbereich gegeneinander verschobenen si-Funktionen entspricht und das verallgemeinerte erste Nyquist-Kriterium erfüllt. Trotzdem kann bei Mehrwegeausbreitung ISI auftreten, welche aber kompensiert werden kann, wenn das zu sendende Elementarsignal periodisch über eine Dauer erweitert wird, die mindestens der Mehrwegeverbreiterung des Kanals entspricht. Diese Schutzzeit wird als zyklische Erweiterung (engl. *cyclic prefix*) bezeichnet und wahrt die Orthogonalität der Subträger. Die verfügbare Bandbreite wird auf N schmalbandige Subträger verteilt, über die N Symbole parallel übertragen werden können. Abb. 2.5 zeigt das Übertragungsmodell für ein SISO-OFDM-System.

Auch dieses Übertragungssystem lässt sich vereinfachen, indem die Elementarsignalmatrix, die DFT bzw. IDFT, der Kanal und der Empfangsfilter zu einem Ersatzkanal $\underline{H}_{\text{SO}}$ zusammengefasst werden. Im Folgenden wird eine zeitdiskrete Stoßantwort des SISO-Kanals mit L Pfaden, genannt engl. *Taps*, der Form $\underline{h} = [h_1, h_2, \dots, h_L]^T$ angenommen. An diese werden Nullen angehängt, bis der Vektor die Länge N erreicht, so dass

$$\underline{h}_{\text{cp}} = [h_1, h_2, \dots, h_L, 0, \dots, 0]^T \quad (2.22)$$

Der SISO-OFDM-Kanal $\underline{H}_{\text{SO}}$ ist dann gegeben durch die Matrix [14]

$$\underline{H}_{\text{SO}} = \text{diag}(\text{DFT}(\underline{h}_{\text{cp}})) \quad (2.23)$$

$$= \begin{bmatrix} H(f_1) & 0 & \cdots & 0 & 0 \\ 0 & H(f_2) & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & H(f_{N-1}) & 0 \\ 0 & 0 & \cdots & 0 & H(f_N) \end{bmatrix}, \quad (2.24)$$

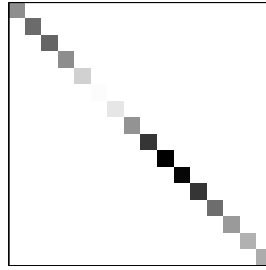


Abbildung 2.6: Ersatzkanalmatrix für SISO-OFDM: $\underline{\underline{R}}_{\text{SO}}$ für $N = 16$ und $L = 4$.

der Größe $N \times N$. Der Ersatzkanal $\underline{\underline{R}}_{\text{SO}}$ lässt sich analog zu Gl. 2.4 berechnen als [14]

$$\underline{\underline{R}}_{\text{SO}} = \underline{\underline{H}}_{\text{SO}}^H \underline{\underline{H}}_{\text{SO}} \quad (2.25)$$

$$= \begin{bmatrix} |H(f_1)|^2 & 0 & \cdots & 0 & 0 \\ 0 & |H(f_2)|^2 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & |H(f_{N-1})|^2 & 0 \\ 0 & 0 & \cdots & 0 & |H(f_N)|^2 \end{bmatrix}_{N \times N} \quad (2.26)$$

Auf der Hauptdiagonalen der Ersatzkanalmatrix $\underline{\underline{R}}_{\text{SO}}$ stehen die Betragsquadrate der einzelnen Subträger, was deren Leistung entspricht. Die Subträger können frequenzselektivem Fading, verursacht durch den Mehrwegekanal, unterworfen sein. Die übrigen Elemente der Matrix sind jedoch gleich Null, was eine Übertragung ohne ISCI bedeutet. Abb. 2.6 zeigt ein Beispiel für $\underline{\underline{R}}_{\text{SO}}$ in einem System mit $N = 16$ Subträgern und $L = 4$ Pfaden, wobei Schwarz für maximale und Weiß für keine Leistung steht. Es ist deutlich zu erkennen, dass einzelne Subträger durch Fading abgeschwächt werden.

2.2.2 MIMO-OFDM

Das OFDM-Prinzip aus Abschnitt 2.2.1 kann vom SISO-Fall auf den MIMO-Fall ausgeweitet und verallgemeinert werden [40][9]. Die beschriebenen Vorteile von OFDM können so mit den Vorteilen von MIMO-Systemen verknüpft werden. Neben der Dimension der Frequenz kann nun auch die räumliche Dimension ausgenutzt werden, um über verschiedene parallele Moden mehr Daten zu übertragen oder die Diversität zu erhöhen. Die genutzte Bandbreite und Sendeleistung bleiben hingegen konstant.

Bei MIMO-OFDM senden alle n_T Sendeantennen über den vollen Satz von N Subträgern zum selben Zeitpunkt, jedoch unabhängig von einander. Dadurch entstehen $n_T N$ Subkanäle. Die MIMO-OFDM-Matrix für die i -te Empfangsantenne kann deshalb gebildet werden, indem mehrere Matrizen $\underline{\underline{H}}_{i,j}$ des SISO-Falls (vgl. Gl. 2.23), die die einzelnen Sendeantennen j repräsentieren, zusammengesetzt werden [40]. Dazu werden n_T Matrizen

$$\underline{\underline{H}}_{i,j} = \text{diag}(\text{DFT}(\underline{h}_{i,j})) \quad (2.27)$$

zusammengefügt in der Form

$$\underline{\underline{H}}_i = \begin{bmatrix} \underline{\underline{H}}_{i1} & \underline{\underline{H}}_{i2} & \underline{\underline{H}}_{i3} & \cdots & \underline{\underline{H}}_{i n_T} \end{bmatrix} \quad (2.28)$$

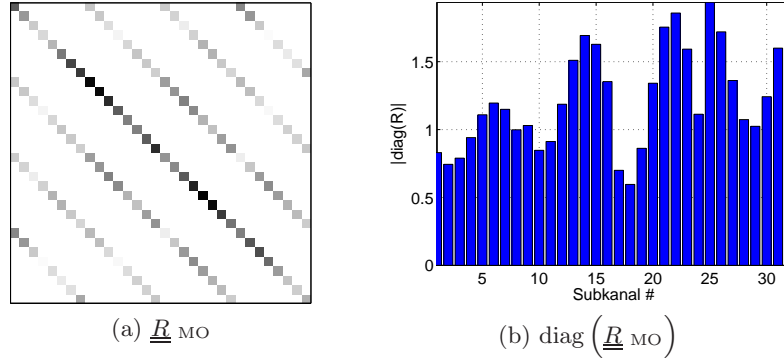


Abbildung 2.7: Ersatzkanalmatrix für MIMO-OFDM. (a): $\underline{\underline{R}}_{\text{MO}}$ für $n_{\text{T}} = 4$, $n_{\text{R}} = 4$, $L = 2$ und $N = 8$ Subträger. (b): Betrag der 32 Diagonalelemente von $\underline{\underline{R}}_{\text{MO}}$.

und bilden eine Matrix der Größe $N \times n_{\text{T}}N$, die den MIMO-OFDM-Kanal für die i -te Empfangsantenne beschreibt. Aus den $\underline{\underline{H}}_i$ lässt sich jeweils eine Ersatzkanalmatrix $\underline{\underline{R}}_{\text{MO},i}$ der Größe $n_{\text{T}}N \times n_{\text{T}}N$ berechnen durch

$$\underline{\underline{R}}_{\text{MO},i} = \underline{\underline{H}}_i^H \underline{\underline{H}}_i. \quad (2.29)$$

Die Ersatzkanalmatrix $\underline{\underline{R}}_{\text{MO}}$ für das gesamte System kann nun einfach gebildet werden, indem die Ersatzkanalmatrizen aller n_{R} Empfangsantennen aufsummiert werden, so dass gilt [40]

$$\underline{\underline{R}}_{\text{MO}} = \frac{1}{n_{\text{R}}} \sum_{i=1}^{n_{\text{R}}} \underline{\underline{R}}_{\text{MO},i} = \frac{1}{n_{\text{R}}} \sum_{i=1}^{n_{\text{R}}} \underline{\underline{H}}_i^H \underline{\underline{H}}_i. \quad (2.30)$$

Durch die Addition sinkt die Wahrscheinlichkeit, dass ein Subträger durch Fading vollständig ausgelöscht wird, drastisch unter der Annahme unkorrelierter Empfangsantennen. $\underline{\underline{R}}_{\text{MO}}$ wird normalisiert, so dass die mittlere Leistung jedes Subträgers $\text{E}\{r_{ii}\} = 1$ ist, also [41]

$$\text{E}\left\{\text{Tr}\left(\underline{\underline{R}}_{\text{MO}}\right)\right\} = \text{E}\left\{\sum_{i=1}^{n_{\text{T}}N} \lambda_i\left(\underline{\underline{R}}_{\text{MO}}\right)\right\} = n_{\text{T}}N, \quad (2.31)$$

wobei λ_i der i -te Eigenwert von $\underline{\underline{R}}_{\text{MO}}$ ist.

Abbildung 2.7a zeigt eine Ersatzkanalmatrix $\underline{\underline{R}}_{\text{MO}}$ für $n_{\text{T}} = 4$, $n_{\text{R}} = 4$, $L = 2$ und $N = 8$. Sie umfasst $n_{\text{T}}N = 32$ Subkanäle und hat somit die Größe 32×32 . Die Subkanäle werden durch die Elemente der Hauptdiagonale repräsentiert, wobei Blöcke von jeweils $N = 8$ Subträgern zu einer Sendeantenne gehören. Auf den Nebendiagonalen sind die Interferenzen (ISCI) zu erkennen, die dadurch verursacht werden, dass alle n_{T} Sendeantennen die selben Subträger im Frequenzbereich verwenden. Bei einem MIMO-OFDM-System ist daher immer eine Entzerrung auf Seite des Empfängers nötig. Abbildung 2.7b zeigt die Beträge der 32 Hauptdiagonalelemente, die die Leistung des jeweiligen Subträgers widerspiegeln. Es ist zu erkennen, dass diese frequenzsektivem Fading unterworfen sind und alle unterschiedliche Leistungen aufweisen.

3 Entzerrerkonzepte für Blockübertragungen

In diesem Kapitel werden vier verschiedene Entzerrer vorgestellt, deren Leistung in Verbindung mit den in Kapitel 5 beschriebenen Spreizverfahren und Precoding in dieser Arbeit untersucht werden soll. Die vier betrachteten Entzerrer unterscheiden sich erheblich in Komplexität, Vorgehensweise und Leistung.

Werden Codesequenzen \underline{x} über einen verzerrenden Kanal übertragen, so ist auf Seiten des Empfängers eine Komponente nötig, die versucht, die Veränderungen an der empfangenen Sequenz \underline{y} so weit wie möglich rückgängig zu machen, um die ursprünglich gesendete Sequenz \underline{x} mit größtmöglicher Sicherheit wiederherzustellen, so dass die entschiedene Sequenz $\hat{\underline{x}} = \underline{x}$. Dazu existieren sehr unterschiedliche Ansätze. Optimale Entzerrer auf Basis der *Maximum Likelihood Sequence Estimation (MLSE)* versuchen, unter allen möglichen Bitsequenzen diejenige zu finden, die mit größter Wahrscheinlichkeit der gesendeten Sequenz entspricht. Der Aufwand für diese Bestimmung wächst mit der Länge der Sequenz und der Größe des Symbolalphabets stark an. Deshalb werden in der Praxis suboptimale Entzerrer verwendet, die die empfangene Sequenz mit einer ähnlich hohen Sicherheit bestimmen können, dafür aber einen geringeren Aufwand benötigen.

Die im Folgenden vorgestellten Entzerrerkonzepte unterscheiden sich in zwei wesentlichen Punkten. Bei Block Linear Equalizer (BLE) und Block Decision Feedback Equalizer (BDFE) handelt es sich um Blockdetektoren, die nur einmal durchlaufen werden und dabei eine harte Entscheidung treffen, während die Entzerrerkonzepte Recurrent Neural Network (RNN) und Iterative Soft Cholesky Block Decision Feedback Equalizer (SCE) ihre Entscheidung in einem iterativen Prozess entwickeln und dabei weiche Entscheidungen verwenden. Im Gegensatz zum BLE verfügen BDFE und SCE über eine Entscheidungsrückführung. Das bedeutet, dass der Entzerrer beim aktuellen Symbol auch die Entscheidung für vorherige Symbole berücksichtigt, zwischen denen bei einem Mehrwegekanal mit ISI/ISCI ein direkter Zusammenhang besteht. Beim RNN ist dies davon abhängig, ob er mit seriellem oder parallelem Updating betrieben wird (siehe Abschnitt 3.3). Einen Überblick über die grundlegenden Unterschiede zwischen den Entzerrerkonzepten gibt Tabelle 3.1.

Konzept	Iterativ	Entscheidungsrückführung
BLE	-	-
BDFE	-	×
RNN	×	×/-
SCE	×	×

Tabelle 3.1: Vergleich grundlegender Eigenschaften der Entzerrer-Konzepte

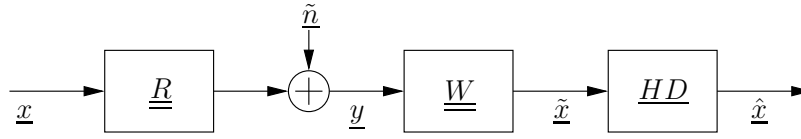


Abbildung 3.1: Blockschaltbild eines BLE: Empfangene Symbole \underline{y} werden mit Hilfe der Filtermatrix \underline{W} geschätzt, über die geschätzten Symbole $\tilde{\underline{x}}$ wird hart entschieden.

3.1 Block Linear Equalizer (BLE)

Der Block Linear Equalizer versucht, mit Hilfe einer Filtermatrix \underline{W} eine geschätzte Sequenz $\tilde{\underline{x}}$ für die gesendete Sequenz \underline{x} aus \underline{y} zu erhalten [11]. Danach entscheidet er jedes Symbol \hat{x}_i separat auf Basis von \tilde{x}_i . Das Blockschaltbild eines BLE ist in Abb. 3.1 dargestellt. Die Komplexität reduziert sich gegenüber den optimalen Verfahren dadurch, dass die Entscheidung nicht für die gesamte Sequenz, sondern für jedes Symbol einzeln getroffen wird. Für die Wahl der Filtermatrix \underline{W} existieren unterschiedliche Optimierungskriterien, von denen zwei, *zero forcing (ZF)* und *minimum mean square error (MMSE)*, vorgestellt werden sollen. Um von einem geschätzten Symbol \tilde{x}_i zu einem entschiedenen Symbol \hat{x}_i zu gelangen, ist eine Entscheidungsfunktion notwendig. Diese Entscheidungsfunktion kann beschrieben werden als [11]

$$\hat{x} = \arg \left[\max_{\check{x} \in \mathcal{A}} P(x = \check{x} | \tilde{x}) \right], \quad (3.1)$$

wobei \mathcal{A} für das Symbolalphabet steht, in dem das entschiedene Symbol enthalten sein muss. Diese Entscheidungsregel maximiert die a-posteriori-Wahrscheinlichkeiten und wird deshalb als Maximum-a-posteriori-Regel (MAP) bezeichnet. Mit dem Satz von Bayes lässt sich dies umformen in [11]

$$\hat{x} = \arg \left[\max_{\check{x} \in \mathcal{A}} p(\tilde{x} | x = \check{x}) P(x = \check{x}) \right]. \quad (3.2)$$

Treten alle Symbole mit gleicher Wahrscheinlichkeit $P(x = \check{x})$ auf, entfällt dieser Wert und man erhält die Maximum-Likelihood-Regel (ML) [11]

$$\hat{x} = \arg \left[\max_{\check{x} \in \mathcal{A}} p(\tilde{x} | x = \check{x}) \right]. \quad (3.3)$$

3.1.1 Zero Forcing Block Linear Equalizer (ZF-BLE)

Der Zero Forcing Block Linear Equalizer (ZF-BLE), der auch als inverser Filter oder de-korrelierender Filter bezeichnet wird, verwendet ein einfaches Prinzip, um den Einfluss des Kanals rückgängig zu machen. Ist die Kanalmatrix \underline{R} regulär, so wird sie invertiert und als Filtermatrix [11]

$$\underline{W} = \underline{R}^{-1} \quad (3.4)$$

verwendet. Eine Übertragung ohne Rauschen vorausgesetzt, entfernt dieser Filter alle durch den Kanal verursachten Interferenzen perfekt. In der Anwendung ist jedoch immer mit einer Rauschkomponente zu rechnen, so dass

$$\begin{aligned} \tilde{\underline{x}} &= \underline{R}^{-1} \underline{y} \\ &= \underline{x} + \underline{R}^{-1} \tilde{\underline{n}} \\ &= \underline{x} + \tilde{\underline{n}}_e. \end{aligned} \quad (3.5)$$

Die Kovarianzmatrix des farbigen Rauschens $\tilde{\mathbf{n}}_e$ ist gegeben durch [11]

$$\underline{\underline{\Phi}}_{\tilde{\mathbf{n}}_e \tilde{\mathbf{n}}_e} = \underline{\underline{R}}^{-1} \quad (3.6)$$

mit den Varianzen des Rauschens für die einzelnen Symbole auf der Hauptdiagonale

$$\sigma_{\tilde{\mathbf{n}}_e, i}^2 = w_{ii} \quad (3.7)$$

Ist die Ersatzkanalmatrix $\underline{\underline{R}}$ schlecht konditioniert, gehen also Eigenwerte gegen Null, so werden die Varianzen des Rauschens $\tilde{\mathbf{n}}_e$ sehr groß im Verhältnis zur Energie des übertragenen Symbols. Dieser Effekt ist der große Nachteil der ZF-Methode und wird als *noise enhancement* bezeichnet [11]. Insbesondere bei MIMO-OFDM ist dieses Problem relevant, da durch Fading viele Nullstellen in der Übertragungsfunktion des Kanals enthalten sein können.

3.1.2 Minimum Mean Square Error Block Linear Equalizer (MMSE-BLE)

Um die Probleme durch das *noise enhancement* zu begrenzen und trotzdem die Interferenzen so weit wie möglich zu reduzieren, muss ein neuer Optimierungsansatz zur Wahl der Filtermatrix $\underline{\underline{W}}$ verwendet werden. Ziel des Minimum Mean Square Error Block Linear Equalizer ist es, den mittleren quadratischen Fehler (engl. *mean square error*, *MSE*) zwischen den übertragenen Symbolen $\underline{\underline{x}}$ und den geschätzten Symbolen

$$\tilde{\mathbf{x}} = \underline{\underline{W}} \mathbf{y} \quad (3.8)$$

zu minimieren, also [11]

$$\underline{\underline{W}} = \arg \left[\min_{\underline{\underline{W}} \in \mathbb{C}^{N \times N}} E_{xy} \left\{ \left\| \underline{\underline{x}} - \underline{\underline{W}} \mathbf{y} \right\|^2 \right\} \right] \quad (3.9)$$

Die Lösung dieses Optimierungsproblem ist der Wiener Filter [27]. Für die Filtermatrix $\underline{\underline{W}}$ muss gelten [11]

$$\underline{\underline{W}} = \left(\underline{\underline{R}} + \underline{\underline{\Phi}}_{\underline{\underline{x}} \underline{\underline{x}}}^{-1} \right)^{-1}. \quad (3.10)$$

Dabei wird die Kovarianzmatrix des Sendesignals angenommen als $\underline{\underline{\Phi}}_{\underline{\underline{x}} \underline{\underline{x}}} = \frac{\sigma_x^2}{\sigma_n^2} \underline{\underline{I}}$. Eingesetzt in Gl. 3.8 ergibt sich

$$\begin{aligned} \tilde{\mathbf{x}} &= \left(\underline{\underline{R}} + \underline{\underline{\Phi}}_{\underline{\underline{x}} \underline{\underline{x}}}^{-1} \right)^{-1} \mathbf{y} \\ &= \left(\underline{\underline{R}} + \underline{\underline{\Phi}}_{\underline{\underline{x}} \underline{\underline{x}}}^{-1} \right)^{-1} \underline{\underline{R}} \underline{\underline{x}} + \left(\underline{\underline{R}} + \underline{\underline{\Phi}}_{\underline{\underline{x}} \underline{\underline{x}}}^{-1} \right)^{-1} \tilde{\mathbf{n}}. \end{aligned} \quad (3.11)$$

Wie man erkennen kann, muss in diesem Fall die Kanalmatrix $\underline{\underline{R}}$ selbst nicht regulär sein, da nicht sie selbst, sondern die Summe mit $\underline{\underline{\Phi}}_{\underline{\underline{x}} \underline{\underline{x}}}^{-1}$ invertiert wird. Für hohes SNR fallen die Lösungen für ZF-BLE und MMSE-BLE zusammen.

3.2 Block Decision Feedback Equalizer (BDFE)

Die linearen Filter aus Abschnitt 3.1 erreichen bei Kanälen mit hoher ISI nur geringe Performance. Bessere Ergebnisse liefert ein nicht-linearer Ansatz, der die Symbole $\tilde{\mathbf{x}}$ nicht nur auf Basis von Statistik, sondern auch unter Zuhilfenahme von zuvor entschiedenen Symbolen $\hat{\mathbf{x}}$

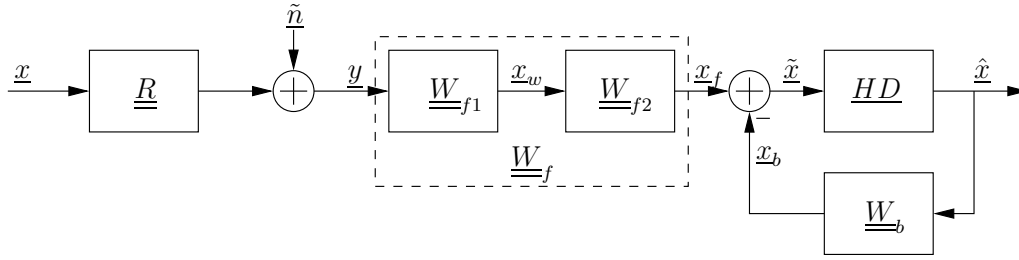


Abbildung 3.2: Blockschaltbild eines BDFE

schätzt. Dabei können analog zu Abschnitt 3.1 ZF- oder MMSE-Kriterium zum Einsatz kommen. Ein Block Decision Feedback Equalizer [11] benötigt, im Unterschied zum BLE, zwei Filter: Ein Vorwärtsfilter \underline{W}_f und ein Rückwärtsfilter \underline{W}_b zur Rückführung der entschiedenen Symbole und zur Entfernung der durch sie verursachten Interferenz. Das Blockschaltbild eines BDFE ist in Abb. 3.2 dargestellt: Empfangene Symbole y werden mit Hilfe der Vorwärtsfiltermatrix \underline{W}_f und den rückgeführten Entscheidungen über die Rückwärtsfiltermatrix \underline{W}_b geschätzt, über die geschätzten Symbole \tilde{x} wird hart entschieden. Wie der BLE entscheidet der BDFE auf Symbolbasis, jedes entschiedene Symbol \hat{x}_i hängt also nur von \tilde{x}_i ab. Ebenso wird angenommen, dass zwischen einzelnen Blöcken keine Interferenz existiert.

3.2.1 Zero Forcing Block Decision Feedback Equalizer (ZF-BDFE)

Block Decision Feedback Equalizer für MIMO-Kanäle können auf Basis der Cholesky-Zerlegung [4] implementiert werden. Die Cholesky-Zerlegung findet für eine hermitesche, positiv definite Kanalmatrix \underline{R} eine eindeutige Lösung für eine Matrix \underline{F} , so dass gilt

$$\underline{R} = \underline{F}^H \underline{F}. \quad (3.12)$$

Die Matrix \underline{F} ist dann eine obere Dreiecksmatrix und wird abgekürzt bezeichnet als

$$\underline{F} = \text{chol}(\underline{R}). \quad (3.13)$$

Mit dieser Darstellung kann der Ersatzkanal \underline{R} zerlegt werden in einen Teil \underline{F} vor der Addition des Rauschens und einen Teil \underline{F}^H danach, siehe Abb. 3.3. Die Vorwärtsfiltermatrix \underline{W}_f besteht aus zwei Komponenten, \underline{W}_{f1} und \underline{W}_{f2} . Die Inverse des Ersatzkanals nach der Addition des Rauschens wird als Vorwärtsfiltermatrix \underline{W}_{f1} gewählt, so dass [11]

$$\underline{W}_{f1} = (\underline{F}^H)^{-1}. \quad (3.14)$$

Für die Symbole x_w nach der Filterung mit \underline{W}_{f1} gilt dann [13]

$$\begin{aligned} x_w &= (\underline{F}^H)^{-1} y \\ &= (\underline{F}^H)^{-1} (\underline{R}x + \tilde{n}) \\ &= \underline{F}x + \underline{n} \\ &= \begin{pmatrix} f_{00} & f_{01} & \cdots & f_{0N-1} \\ & f_{11} & \cdots & f_{1N-1} \\ & & \ddots & \vdots \\ \underline{0} & & & f_{N-1N-1} \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{pmatrix} + \underline{n}. \end{aligned} \quad (3.15)$$

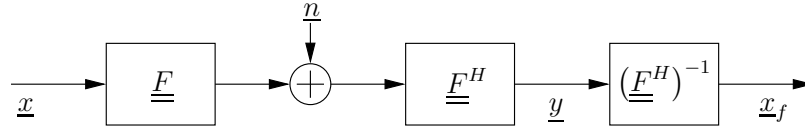


Abbildung 3.3: Blockschaltbild des Cholesky-zerlegten Ersatzkanals mit Vorwärtsfilter: Das in \underline{x}_f enthaltene Rauschen ist weiß.

Dabei steht N für die Länge des zu detektierenden Blocks. Für die nachfolgenden Blöcke des Entzerrers erscheint das vom Kanal hinzugefügte Rauschen weiß. Der Vorwärtsfilter \underline{W}_{f1} erfüllt somit die Funktion eines *pre-whitening filters*. Die Inversion der Matrix \underline{F}^H ist möglich, da \underline{F} positiv definit ist, wenn \underline{R} positiv definit ist. Wie man an der Struktur von Gl. 3.15 erkennen kann, ist auf Grund der Dreiecksstruktur jedes Symbol $x_{w,i}$ dekorreliert von allen anderen Symbolen $x_{w,j}$ mit $j < i$. Die $x_{w,i}$ lassen sich auch darstellen als [13]

$$x_{w,i} = f_{ii} x_i + \sum_{j=i+1}^N f_{ij} x_j + n_i \quad (3.16)$$

Die zweite Komponente \underline{W}_{f2} des Vorwärtsfilters wird dem bereits in Abschnitt 3.1.1 vorgestellten ZF-Ansatz entsprechend gewählt als

$$\underline{W}_{f2} = \text{diag}(\underline{F})^{-1}, \quad (3.17)$$

so dass die gesamte Vorwärtsfiltermatrix [11]

$$\underline{W}_F = \underline{W}_{f2} \underline{W}_{f1} = \text{diag}(\underline{F})^{-1} (\underline{F}^H)^{-1} \quad (3.18)$$

entspricht. Über den Rückwärtsfilter wird die Interferenz, die durch vorherige Symbole verursacht wurde, über die bereits entschieden wurde, entfernt. Interferenz sind in der Ersatzkanalmatrix \underline{R} alle Nicht-Diagonalelemente, so dass eine Matrix, die die Interferenz vollständig entfernt, von der Form [11]

$$\underline{W}_b = \underline{W}_F \underline{R} - \text{diag}(\underline{W}_F \underline{R}) \quad (3.19)$$

ist. Für diesen Entzerrer bedeutet dies, dass der Rückwärtsfilter [11]

$$\underline{W}_b = \text{diag}(\underline{F})^{-1} \underline{F} - \underline{I} \quad (3.20)$$

entspricht. Ihm werden die bereits entschiedenen Symbole \hat{x}_j mit $j = i + 1 \dots N$ zugeführt und die daraus berechnete Interferenz wird von $x_{f,i}$ abgezogen. Wenn mit dem letzten Symbol des Blocks mit der Entzerrung begonnen wird, kann der ZF-BDFE die Interferenz vollständig entfernen und über Symbole entscheiden, die nur durch weißes Rauschen beeinflusst sind. Voraussetzung dafür ist jedoch, dass die Entscheidungen über die vorherigen Symbole korrekt waren. Im Falle einer falschen Entscheidung pflanzt sich dieser Fehler in die Entzerrung aller übrigen Symbole des gleichen Blocks fort (engl. *error propagation*) und verschlechtert somit die Performance. Ebenso wie der ZF-BLE verursacht der ZF-BDFE *noise enhancement* bei einem Kanal mit kleinen Eigenwerten, so dass die Rauschvarianzen erheblich werden im Vergleich zur Energie der Symbole [11].

3.2.2 Minimum Mean Square Error Block Decision Feedback Equalizer (MMSE-BDFE)

Wie auch im Falle des BLE sind ZF-BDFE und MMSE-BDFE eng miteinander verwandt. Sie unterscheiden sich jedoch im Optimierungskriterium zur Wahl des Vorwärtsfilters $\underline{\underline{W}}_F$. Dieser wird so gewählt, dass der mittlere quadratische Fehler zwischen den übertragenen Symbolen \underline{x} und den geschätzten Symbolen $\tilde{\underline{x}}$ minimiert wird. Die Lösung des Optimierungsproblems liefert, analog zu Gl. 3.10, eine Matrix [11]

$$\underline{\underline{\tilde{R}}} = \underline{\underline{R}} + \underline{\underline{\Phi}}_{xx}^{-1}. \quad (3.21)$$

Mit der Cholesky-Zerlegung $\underline{\underline{\tilde{R}}} = \underline{\underline{\tilde{F}}}^H \underline{\underline{\tilde{F}}}$ ergibt sich die Vorwärtsfiltermatrix

$$\underline{\underline{W}}_F = \text{diag}(\underline{\underline{\tilde{F}}})^{-1} (\underline{\underline{\tilde{F}}}^H)^{-1}. \quad (3.22)$$

Der Rückwärtsfilter ist dann entsprechend Gl. 3.20 [11]

$$\underline{\underline{W}}_b = \text{diag}(\underline{\underline{\tilde{F}}})^{-1} \underline{\underline{\tilde{F}}} - \underline{\underline{I}}. \quad (3.23)$$

Da die Matrix $\underline{\underline{\tilde{R}}}$ Cholesky-zerlegt wird, muss auch in diesem Fall nicht die Ersatzkanalmatrix $\underline{\underline{\tilde{R}}}$ selbst, sondern ihre Summe mit $\underline{\underline{\Phi}}_{xx}^{-1}$ positiv definit sein. Während der Einfluss des *noise enhancement* durch den MMSE-Ansatz geringer ist, leidet auch der MMSE-BDFE unter *error propagation*, die sich insbesondere bei hohen Interferenzen bemerkbar macht [11]. Wie beim BLE fallen die Ergebnisse von ZF- und MMSE-BDFE für hohes SNR zusammen.

3.3 Recurrent Neural Network Equalizer (RNN)

Im Vergleich zu den bisher vorgestellten Entzerrern folgt der Recurrent Neural Network Equalizer (RNN) einem völlig anderen Ansatz. Grundlage des Entzerrers bildet ein Hopfield-Netz, ein künstliches neuronales Netz. Neuronale Netze sind vor allem aus dem Bereich der Mustererkennung bekannt und eignen sich zur Lösung von Klassifizierungsproblemen. Entzerrung kann als ein Klassifizierungsproblem aufgefasst werden, dessen Ziel die Auswahl des am wahrscheinlichsten übertragenen Vektors aus einem begrenzten Satz von möglichen Sendevektoren darstellt. Dazu wird der Entzerrer iterativ eine definierte Anzahl von i Durchgängen durchlaufen, in denen die Entscheidung Schritt für Schritt präzisiert wird. Dieser Typ von Entzerrer wurde vorgestellt in [33][32][15] und in [31] erweitert für die Verwendung in einem Turbo-Schema. Eine ausführliche Darstellung neuronaler Netze und ihrer Anwendbarkeit zur Entzerrung ist in [14][30] zu finden. Das hierzu verwendete Hopfield-Netz stellt eine besonders einfache Form eines neuronalen Netzes mit einer Schicht von Neuronen dar, deren Ausgänge jeweils, gewichtet mit einem Faktor, auf die Eingängen aller Neuronen geführt wird. Diese Gewichte, die in allgemeinen neuronalen Netzen meist durch Training gewonnen werden, werden in diesem Fall der Ersatzkanalmatrix $\underline{\underline{R}}$ entnommen [30].

Im verwendeten Übertragungssystem kann jedes Symbol eines Block dargestellt werden als Summe aus Nutzteil, Interferenz und Rauschen, so dass [33]

$$y_i = r_{ii}x_i + \sum_{j=1, j \neq i}^N r_{ij}x_j + \tilde{n}_i. \quad (3.24)$$

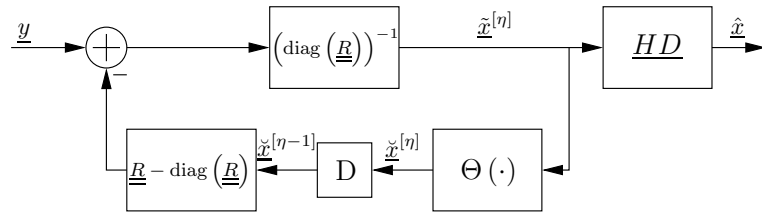


Abbildung 3.4: Blockschaltbild des RNN-Entzerrers

Das Blockschaltbild eines RNN-Entzerrers ist in Abb. 3.4 zu sehen. Zu Beginn jedes Iterationsschrittes η wird zunächst die im vorherigen Iterationsschritt $\eta - 1$ geschätzte Interferenz $(\underline{R} - \text{diag}(\underline{R})) \check{\underline{x}}^{[\eta-1]}$ vom empfangenen Vektor \underline{y} abgezogen. Auf den teilweise entzerrten Vektor $\check{\underline{x}}$ wird dann eine weiche, nicht-lineare und veränderliche Entscheidungsfunktion $\Theta(\cdot)$ angewandt, die den neuen geschätzten Vektor $\check{\underline{x}}^{[\eta]}$ liefert. Die vorherige Multiplikation mit $1/r_{ii}$ dient der Normierung der empfangenen Symbole auf die Verstärkung des Kanals, da sonst die Energie des Vektors $\check{\underline{x}}$ nicht der des gesendeten Vektors \underline{x} entspricht. Das Updating des geschätzten Vektors kann auf zwei verschiedene Arten erfolgen. Bei parallelem Updating wird während des aktuellen Iterationsschrittes η lediglich die geschätzte Interferenz $\check{\underline{x}}^{[\eta-1]}$ des vorherigen Schrittes entfernt. Beim seriellen Updating werden zusätzlich alle bereits geschätzten Symbole $\check{\underline{x}}^{[\eta]}$ des gegenwärtigen Iterationsschrittes berücksichtigt. Somit ergeben sich die aktuellen, entzerrten Symbole bei parallelem Updating aus [33][32]

$$\begin{aligned} \check{x}_i^{[\eta]} &= \frac{y_i}{r_{ii}} - \sum_{j=1, j \neq i}^N \frac{r_{ij}}{r_{ii}} \check{x}_j^{[\eta-1]} \\ &= x_i + \frac{\tilde{n}_i}{r_{ii}} + \sum_{j=1, j \neq i}^N \frac{r_{ij}}{r_{ii}} (x_j - \check{x}_j^{[\eta-1]}). \end{aligned} \quad (3.25)$$

Im Falle des seriellen Updating wird für Symbole, die im aktuellen Schritt schon entzerrt wurden, die aktuelle Schätzung verwendet, für die übrigen Symbole die Schätzung des vorherigen Iterationsschrittes, also [33][32]

$$\begin{aligned} \check{x}_i^{[\eta]} &= \frac{y_i}{r_{ii}} - \sum_{j=1}^{i-1} \frac{r_{ij}}{r_{ii}} \check{x}_j^{[\eta]} - \sum_{j=i+1}^N \frac{r_{ij}}{r_{ii}} \check{x}_j^{[\eta-1]} \\ &= x_i + \frac{\tilde{n}_i}{r_{ii}} + \sum_{j=1}^{i-1} \frac{r_{ij}}{r_{ii}} (x_j - \check{x}_j^{[\eta]}) + \sum_{j=i+1}^N \frac{r_{ij}}{r_{ii}} (x_j - \check{x}_j^{[\eta-1]}). \end{aligned} \quad (3.26)$$

Die geschätzten Symbole enthalten folglich neben dem entzerrten Symbol x_i einen Term mit farbigem Rauschen und einen bzw. zwei Terme mit verbleibender Interferenzleistung (engl. *residual interference*). Im Falle einer optimalen Schätzung wären die jeweils letzten Terme gleich Null, die Entzerrung wäre optimal. Für den ersten Iterationsschritt wird $\check{\underline{x}}^{[0]}$ als Nullvektor initialisiert. Somit wird beim ersten Iterationsschritt keine Interferenz subtrahiert, sondern es findet nur eine weiche Entscheidung statt, wenn der RNN mit parallelem Updating betrieben wird.

Dies muss bei der Wahl einer geeigneten Entscheidungsfunktion berücksichtigt werden. Die Entscheidungsfunktion $\Theta(\cdot)$ muss $\check{\underline{x}}$ aus $\tilde{\underline{x}}$ so berechnen, dass $\check{\underline{x}}$ möglichst \underline{x} entspricht

und keine Fehler zurückgeführt werden. Das Ziel ist also, den mittleren quadratischen Fehler

$$J = \mathbb{E} \left\{ |x - \check{x}|^2 | \tilde{x} \right\} \quad (3.27)$$

zu minimieren und damit die Subtraktion von Rauschen und Interferenz zu optimieren. Unter der Annahme, dass das Modulationsalphabet $\mathcal{A} = \{a_1, \dots, a_M\}$ komplex ist, und die *residual interference* als additives Gaußsches Rauschen mit Varianz σ^2 betrachtet werden kann, zeigt [30], dass die Lösung für das optimale geschätzte Symbol

$$\check{x}_i = \Theta(\tilde{x}_i) = \frac{\sum_{m=1}^M a_m \exp \left(-\frac{|a_m|^2}{2\sigma^2} + \frac{\operatorname{Re}\{a_m\} \operatorname{Re}\{\tilde{x}_i\}}{\sigma^2} + \frac{\operatorname{Im}\{a_m\} \operatorname{Im}\{\tilde{x}_i\}}{\sigma^2} \right)}{\sum_{m=1}^M \exp \left(-\frac{|a_m|^2}{2\sigma^2} + \frac{\operatorname{Re}\{a_m\} \operatorname{Re}\{\tilde{x}_i\}}{\sigma^2} + \frac{\operatorname{Im}\{a_m\} \operatorname{Im}\{\tilde{x}_i\}}{\sigma^2} \right)} \quad (3.28)$$

lautet. Zur Berechnung der weichen Entscheidung muss nun die Varianz des Rauschens bekannt sein. Neben dem farbigen Rauschen des Kanals, dass sich nach [30] als

$$\sigma_{\check{n},i}^2 = \frac{\sigma_n^2}{r_{ii}} \quad (3.29)$$

berechnen lässt, muss außerdem die Leistung der Interferenzen mit $\sigma_{I,i}^2$ berücksichtigt werden. Diese hängen vom Iterationsschritt ab, da die verbleibende Restinterferenz direkt von der Qualität der Schätzung $\check{x}^{[\eta]}$ abhängt. Unter der Annahme, dass die beiden Interferenzterme in Gl. 3.26 statistisch unabhängig sind, ist die Varianz σ^2 in Gl. 3.28 in Iterationsschritt η gegeben durch die Summe aus Rausch- und Interferenzleistung [32]

$$\begin{aligned} \sigma_{\text{ges},i}^2 &= \sigma_{\check{n},i}^2 + \sigma_{I,i}^2 \quad (3.30) \\ &= \frac{\sigma_n^2}{r_{ii}} + \sum_{j=1}^{i-1} \left| \frac{r_{ij}}{r_{ii}} \right|^2 \sigma_{\text{res},j}^2 \quad (3.30) \\ &\quad + \sum_{j=i+1}^N \left| \frac{r_{ij}}{r_{ii}} \right|^2 \sigma_{\text{res},j}^2 \quad (3.30) \end{aligned}$$

Die verbleibende Interferenzleistung $\sigma_{\text{res},i}^2$ ist der Restfehler, der nach der weichen Entscheidung über \tilde{x}_i in \check{x}_i verbleibt. Dieser Restfehler ist das minimierte Maß J aus Gl. 3.27. Die verbleibende Interferenzleistung lässt sich also beschreiben als

$$\sigma_{\text{res},i}^2 = J_{\min} = \mathbb{E} \left\{ |x_i - \check{x}_i|^2 | \tilde{x}_i \right\} \quad (3.31)$$

und kann nach [32] umgeformt werden in

$$\sigma_{\text{res},i}^2 = \mathbb{E} \left\{ |x_i|^2 | \tilde{x}_i \right\} - |\check{x}_i|^2. \quad (3.32)$$

Dieser Ausdruck lässt sich analog zu Gl. 3.28 auswerten als [32]

$$\sigma_{\text{res},i}^2 = \frac{\sum_{m=1}^M |a_m|^2 \exp \left(-\frac{|a_m|^2}{2\sigma^2} + \frac{\operatorname{Re}\{a_m\} \operatorname{Re}\{\tilde{x}_i\}}{\sigma^2} + \frac{\operatorname{Im}\{a_m\} \operatorname{Im}\{\tilde{x}_i\}}{\sigma^2} \right)}{\sum_{m=1}^M \exp \left(-\frac{|a_m|^2}{2\sigma^2} + \frac{\operatorname{Re}\{a_m\} \operatorname{Re}\{\tilde{x}_i\}}{\sigma^2} + \frac{\operatorname{Im}\{a_m\} \operatorname{Im}\{\tilde{x}_i\}}{\sigma^2} \right)}. \quad (3.33)$$

Je nach Empfängerstruktur kann der RNN-Entzerrer in zwei verschiedenen Varianten verwendet werden. Bei einer uncodierten Übertragung oder einer konventionellen Decodierung,

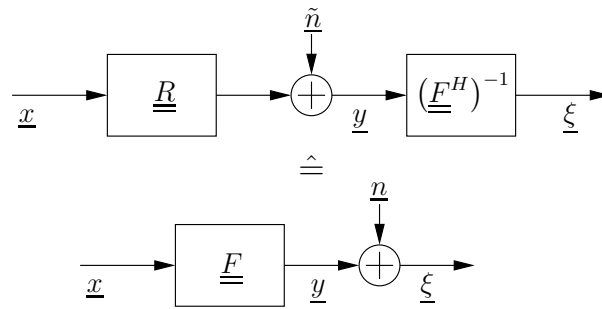


Abbildung 3.5: Blockschaltbild des äquivalenten Cholesky-zerlegten Ersatzkanals

also Entzerrung und nachfolgende Decodierung ohne Rekursion, wird nach Durchlauf der letzten Iteration l über den letzten Schätzvektor $\hat{x}^{[l]}$ hart entschieden und somit \hat{x} bestimmt. Wird der RNN hingegen in einem Turbo-Schema zur gemeinsamen Entzerrung und Decodierung verwendet, wie in Abschnitt 4.4 beschrieben, so werden keine harten Entscheidungen ausgegeben, sondern weiche Entscheidungen in Form von L-Werten (siehe Abschnitt 4.2). Gleichzeitig erhält der Entzerrer einen zweiten Eingang, über den er Zuverlässigkeitsinformationen des Decodierers erhält. Deshalb muss in diesem Fall das Modell des RNN für die Verwendung von L-Werten erweitert werden, was in Abschnitt 4.4.1 beschrieben ist.

Vorteil des RNN-Entzerrers ist die Implementierbarkeit mit geringem Rechenaufwand. Er liefert jedoch nur in Systemen mit begrenzter Interferenz gute Ergebnisse. Er neigt früh zur Bildung eines Fehlerteppichs (engl. *error floor*), wenn die verbleibende Interferenz nicht mehr genauer geschätzt werden kann. Bei hoher Interferenz entstehen bereits im ersten Iterationsschritt, in dem insbesondere bei parallelem Updating noch keine Interferenz subtrahiert werden kann, so viele Fehler, dass auf Grund der Fehlerfortpflanzung dieser initialen falschen Entscheidungen die Bitfehlerrate sehr hoch wird und auch mit erhöhtem E_b/N_0 kaum geringer wird.

3.4 Iterative Soft Cholesky Block Decision Feedback Equalizer (SCE)

Der Iterative Soft Cholesky Block Decision Feedback Equalizer (SCE) wurde in [13] vorgestellt und verbindet die vom RNN bekannten Prinzipien der weichen Entscheidung und der mehrfachen Iterationen mit dem Cholesky-basierten BDFE. Dadurch erreicht der SCE im Vergleich zu einem BDFE eine verbesserte Performance und kann dank weicher Entscheidungen in einem Turbo-Schema (siehe Abschnitt 4.4) eingesetzt werden. Seine Anwendung bei gemeinsamer Entzerrung und Decodierung wurde in [12] beschrieben. In diesem Abschnitt ist die Herleitung aus den genannten Quellen zusammengefasst.

Zunächst wird wie beim in Abschnitt 3.2 vorgestellten *pre-whitening* des BDFE aus dem Ersatzkanal \underline{R} mit Hilfe der Cholesky-Zerlegung $\underline{R} = \underline{F}^H \underline{F}$ ein äquivalenter Ersatzkanal gebildet, siehe Abb. 3.5. Die Verbindung des Ersatzkanals \underline{R} und dem Vorwärtsfilter $(\underline{F}^H)^{-1}$

ergibt den äquivalenten Empfangsvektor $\underline{\xi}$ der Länge N mit [12]

$$\begin{aligned}\underline{\xi} &= (\underline{F}^H)^{-1} (\underline{R} \underline{x} + \underline{\tilde{n}}) \\ &= \underline{F} \underline{x} + \underline{n} \\ &= \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1N} \\ & f_{22} & \cdots & f_{2N} \\ & & \ddots & \vdots \\ \underline{0} & & & f_{NN} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} + \underline{n},\end{aligned}\quad (3.34)$$

vergleiche Gl. 3.15. Eine zweidimensionale Entscheidungsfunktion $\text{dec}(\cdot)$ kann nun die Dreieckseigenschaft der Matrix \underline{F} ausnutzen und, mit dem letzten Symbol beginnend, entscheiden [12]:

$$\begin{aligned}\hat{x}_N &= \text{dec}_{\text{soft}} \left(\frac{1}{f_{NN}} \xi_N \right) \\ \hat{x}_{N-1} &= \text{dec}_{\text{soft}} \left(\frac{1}{f_{N-1N-1}} (\xi_{N-1} - f_{N-1N} \hat{x}_N) \right) \\ &\vdots \\ \hat{x}_1 &= \text{dec}_{\text{soft}} \left(\frac{1}{f_{11}} (\xi_1 - \sum_{i=2}^N f_{1i} \hat{x}_i) \right).\end{aligned}$$

Während diese Entscheidung beim BDFE hart war, wird beim SCE eine weiche Entscheidungsfunktion dec_{soft} verwendet, die in mehreren Iterationsschritten η angewandt wird. Der SCE berechnet den Schätzwert $\tilde{\underline{x}}_l^{[\eta]}$ für ein Symbol \underline{x}_l in der Iteration η als [12]

$$\tilde{\underline{x}}_l^{[\eta]} = \text{dec}_{\text{soft}}(\underline{\xi}^{[\eta,l]}, \underline{\Phi}_{ww}^{[\eta,l]}) \quad \text{mit} \quad \underline{\xi}^{[\eta,l]} = \underline{\xi} - \underline{F}_{\setminus l} \tilde{\underline{x}}^{[\eta,l]} \quad (3.35)$$

und

$$\tilde{\underline{x}}^{[\eta,l]} = \left[\tilde{\underline{x}}_1^{[\eta-l]T}, \dots, \tilde{\underline{x}}_l^{[\eta-l]T}, \tilde{\underline{x}}_{l+1}^{[\eta]T}, \dots, \tilde{\underline{x}}_N^{[\eta]T} \right]^T. \quad (3.36)$$

Die Matrix $\underline{\Phi}_{ww}^{[\eta,l]}$ ist die Kovarianz-Matrix von Rauschen und Interferenz, die Matrix $\underline{F}_{\setminus l}$ wird aus \underline{F} gewonnen, in dem die l -te Spalte zu Null gesetzt wird. Für die erste Iteration existieren noch keine geschätzten Symbole und der Vektor $\tilde{\underline{x}}^{[1,N]}$ wird zu Null initialisiert. Als Endzustand soll erreicht werden, dass sämtliche Interferenz subtrahiert wird und somit $\tilde{\underline{x}}^{[\eta,l]} = \underline{x}$ ist, also [12]

$$\underline{\xi}^{[\eta,l]} = \underline{F}_l \underline{x}_l + \underline{n}. \quad (3.37)$$

Dabei ist \underline{F}_l eine Teilmatrix von \underline{F} , die aus deren l -ter Spalte besteht und damit den Einfluss des l -ten Sendesymbols \underline{x}_l auf den Empfangsvektor $\underline{\xi}$ beschreibt. Das Updating der geschätzten Symbole ist in Abb. 3.6a abgebildet.

Die Entscheidungsfunktion basiert wie im Falle des RNN auf der Grundidee der Minimierung des mittleren quadratischen Fehlers. Um einen SNR-Verlust durch das *pre-whitening* zu vermeiden, wurde sie so erweitert, dass sie nicht nur die Energie der Hauptdiagonalelemente von \underline{F} berücksichtigt, sondern jeweils eine gesamte Spalte dieser Matrix betrachtet. Wird das gesendete Symbol \underline{x}_l mit \underline{F}_l multipliziert, wird dessen Energie über den Empfangsvektor $\underline{\xi}^{[\eta,l]}$

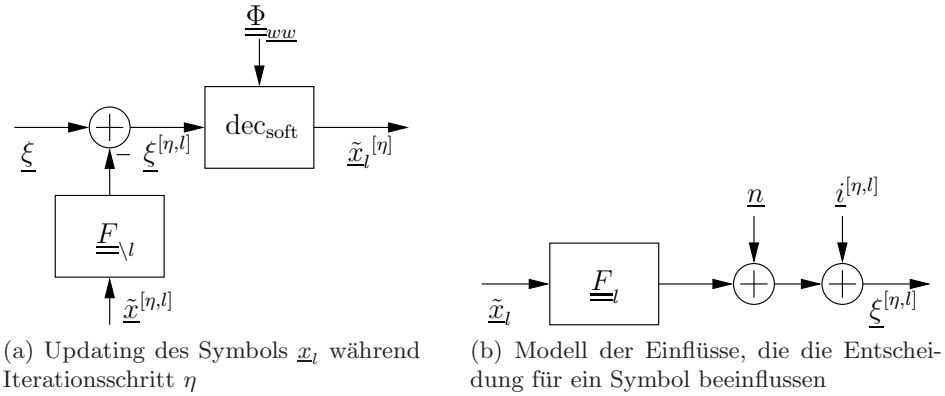


Abbildung 3.6: SCE: Iteratives Updating der geschätzten Symbole (a) und Modell der Entscheidung (b)

verteilt. Zusätzlich addieren sich zwei Störungen, erstens der AWGN-Vektor \underline{n} und zweitens die Interferenz $\underline{i}^{[\eta,l]}$, die durch die anderen Symbole verursacht wird und als [12]

$$\underline{i}^{[\eta,l]} = \underline{F}_{\setminus l}(\underline{x} - \underline{\tilde{x}}^{[\eta,l]}) \quad (3.38)$$

beschrieben werden kann. Somit ergibt sich der Empfangsvektor [12]

$$\underline{\xi}^{[\eta,l]} = \underline{F}_l \underline{x}_l + \underline{n} + \underline{i}^{[\eta,l]}, \quad (3.39)$$

wie in Abb. 3.6b dargestellt. Die Minimierung des mittleren quadratischen Fehlers für $\underline{\tilde{x}}_l^{[\eta]}$ [12]

$$\underline{\tilde{x}}_l^{[\eta]} = \arg \left[\min_{\underline{\tilde{x}}_l} \mathbb{E} \left\{ \left| \underline{\tilde{x}}_l^{[\eta]} - \underline{x}_l \right|^2 \middle| \underline{\xi}^{[\eta,l]} \right\} \right] \quad (3.40)$$

liefert mit dem Symbolalphabet $\mathcal{A} = \{\underline{a}_0, \underline{a}_1, \dots, \underline{a}_{M-1}\}$ die Lösung [12]

$$\underline{\tilde{x}}_l^{[\eta]} = \mathbb{E} \left\{ \underline{x}_l \middle| \underline{\xi}^{[\eta,l]} \right\} = \sum_{m=0}^{M-1} \underline{a}_m P(\underline{x}_l = \underline{a}_m \middle| \underline{\xi}^{[\eta,l]}). \quad (3.41)$$

Mit dem Satz von Bayes und der Annahme, dass alle Sendesymbole mit gleicher Wahrscheinlichkeit auftreten, lässt sich $P(\underline{x}_l = \underline{a}_m \middle| \underline{\xi}^{[\eta,l]})$ ausdrücken als [12]

$$P(\underline{x}_l = \underline{a}_m \middle| \underline{\xi}^{[\eta,l]}) = \frac{p(\underline{\xi}^{[\eta,l]} \middle| \underline{x}_l = \underline{a}_m)}{\sum_{m=0}^{M-1} p(\underline{\xi}^{[\eta,l]} \middle| \underline{x}_l = \underline{a}_m)}. \quad (3.42)$$

Nimmt man an, dass die Störungen, also Rauschen und verbleibende Interferenz, weiß und unkorreliert sind, kann die Wahrscheinlichkeitsdichtefunktion in Gl. 3.42 beschrieben werden als [13]

$$p(\underline{\xi}^{[\eta,l]} \middle| \underline{x}_l = \underline{a}_m) = \prod_{j=0}^{N-1} p(\xi_j^{[\eta,l]} \middle| \underline{x}_l = \underline{a}_m). \quad (3.43)$$

Aus den Gleichungen 3.41 und 3.42 erhält man die weiche Entscheidungsfunktion [13]

$$\begin{aligned}\tilde{\underline{x}}_l^{[\eta]} &= \text{dec}_{\text{soft}}\left(\underline{\xi}^{[\eta,l]}, \underline{\Phi}_{\underline{w}\underline{w}}^{[\eta,l]}\right) \\ &= \sum_{m=0}^{M-1} \underline{a}_m \frac{\prod_{j=0}^{N-1} p(\underline{\xi}^{[\eta,l]} | \underline{x}_l = \underline{a}_m)}{\sum_{k=0}^{M-1} \prod_{j=0}^{N-1} p(\underline{\xi}^{[\eta,l]} | \underline{x}_l = \underline{a}_k)}.\end{aligned}\quad (3.44)$$

Die Kovarianzmatrix $\underline{\Phi}_{\underline{w}\underline{w}}^{[\eta,l]}$ erhält man aus dem Vektor $\underline{w}^{[\eta,l]} = \underline{i}^{[\eta,l]} + \underline{n}$, so dass [12]

$$\underline{\Phi}_{\underline{w}\underline{w}}^{[\eta,l]} = \underline{\Phi}_{\underline{i}\underline{i}}^{[\eta,l]} + \sigma_n^2 \underline{I}.\quad (3.45)$$

Die Kovarianzmatrix der Interferenz kann berechnet werden aus

$$\underline{\Phi}_{\underline{i}\underline{i}}^{[\eta,l]} = \underline{F} \underline{\Phi}_{\underline{x}\underline{x}}^{[\eta,l]} \underline{F}^T,\quad (3.46)$$

wie in [12] gezeigt. Die Kovarianzmatrix des verbleibenden Entscheidungsfehlers $\underline{\Phi}_{\underline{x}\underline{x}}^{[\eta,l]}$ ist unter der Annahme der statistischen Unabhängigkeit der Entscheidungsfehler eine diagonale Blockmatrix [12]

$$\underline{\Phi}_{\underline{x}\underline{x}}^{[\eta,l]} = \underline{\text{diag}}\left(\underline{\Phi}_{\underline{x}_1\underline{x}_1}^{[\eta-1]}, \dots, \underline{\Phi}_{\underline{x}_l\underline{x}_l}^{[\eta-1]}, \underline{\Phi}_{\underline{x}_l\underline{x}_l}^{[\eta]}, \dots, \underline{\Phi}_{\underline{x}_N\underline{x}_N}^{[\eta]}\right)\quad (3.47)$$

mit den Submatrizen

$$\underline{\Phi}_{\underline{x}_l\underline{x}_l}^{[\eta]} = \text{E}\left\{\underline{x}_l \underline{x}_l^T | \underline{\xi}^{[\eta,l]}\right\} - \tilde{\underline{x}}_l^{[\eta]} \tilde{\underline{x}}_l^{[\eta]T},\quad (3.48)$$

die den Entscheidungsfehler für das l -te Symbol beschreiben und berechnet werden können als [12]

$$\text{E}\left\{\underline{x}_l \underline{x}_l^T | \underline{\xi}^{[\eta,l]}\right\} = \sum_{m=0}^{M-1} \underline{a}_m \underline{a}_m^T P(\underline{x}_l = \underline{a}_m | \underline{\xi}^{[\eta,l]}).\quad (3.49)$$

Ebenso wie der RNN kann der SCE entweder allein mit harten Entscheidungen nach der letzten Iteration betrieben werden oder in einem Turbo-Schema zur gemeinsamen Entzerrung und Decodierung eingesetzt werden. Dafür muss das Modell des SCE um die Verwendung von L-Werten erweitert werden. Dies wurde in [12] beschrieben und ist in Abschnitt 4.4.2 dargestellt.

Der SCE ist ein sehr leistungsfähiger Entzerrer, der auch bei Kanälen mit hohen Interferenzen in der Lage ist, diese zu entfernen und gute Bitfehlerraten liefert. Nachteil ist der größere Implementierungsaufwand im Vergleich zum RNN. Insbesondere die aufwändige Entscheidungsfunktion dec_{soft} benötigt viel Rechenzeit bei einer Implementierung in MATLAB, hinzu kommen die häufigen Cholesky-Zerlegungen. Der SCE benötigt jedoch deutlich weniger Iterationen als ein RNN um zu konvergieren.

4 Konzepte für codierte Blockübertragungen

In diesem Kapitel werden die bisher vorgestellten Konzepte für Blockübertragungen um Kanalcodierung erweitert. Nach einer kurzen Vorstellung der Faltungscodierung und -decodierung wird die symbolweise MAP-Decodierung mit Zuverlässigkeitsinformationen, so genannten Log-Likelihood-Verhältnissen, kurz L-Werten, eingeführt. Danach wird die Entzerrung und Decodierung in einem gemeinsamen Schema betrachtet. Durch die Verknüpfung dieser beider Komponenten ergibt sich ein so genannter Turbo-Entzerrer, der die Leistung des Empfängers deutlich steigern kann. Dazu werden beide Komponenten mehrmals iterativ durchlaufen und versorgen sich dabei gegenseitig mit Informationen, die das Decodierergebnis verbessern. Dazu müssen der in Abschnitt 3.3 vorgestellte RNN-Entzerrer und der in Abschnitt 3.4 vorgestellte SCE für diese Anwendung erweitert werden.

4.1 Faltungscodierung

Bei den so genannten Faltungscodes handelt es sich neben den sogenannten Blockcodes um die zweite große Familie von fehlerkorrigierenden Kanalcodes. Faltungscodes entstehen durch eine lineare Abbildung von unendlich langen Informationssequenzen auf unendlich lange Codesequenzen. Diese Abbildung lässt sich durch eine Generatormatrix bzw. Generatorpolynome beschreiben. Realisiert wird ein Faltungscodierer als digitales LTI-System, in der Regel als Verkettung von Schieberegistern. Die Abbildung erzeugt kontinuierlich aus k Quellsymbolen n Codesymbole, was zu einer Coderate $r = k/n$ führt. Weiterer wichtiger Parameter zur Charakterisierung eines Faltungscodes ist die Gedächtnislänge m , die im Faltungscodierer der Zahl der Schieberegister entspricht. Die Vorteile der Faltungscodes liegen zum einen darin, dass ein Maximum-Likelihood-Verfahren zur Decodierung mit realisierbarem Aufwand existiert und zum anderen die Decodierung leicht mit Soft-In- und Soft-Out-Werten implementiert werden kann [1]. An dieser Stelle werden nur die wichtigsten, für diese Arbeit relevanten, Grundlagen genannt. Für eine ausführliche Einführung in die Codierung mit Faltungscodes sei auf [1] verwiesen.

Zur Beschreibung des Faltungscodierers werden in dieser Arbeit Generatorpolynome in oktaler Schreibweise verwendet. Abbildung 4.1 zeigt das Blockschaltbild eines mit Schieberegistern aufgebauten Faltungscodierers mit $r = \frac{1}{2}$, $m = 2$ und Generatorpolynomen $[7, 5]_8$. Die Angabe der Generatorpolynome kann wie folgt verstanden werden: Die Oktalzahlen 7_8 und 5_8 entsprechen 111_2 und 101_2 in binärer Schreibweise. Diese beschreiben die Kopplung der einzelnen verzögerten Stufen D^0 , D^1 und D^2 des Schieberegisters auf die Modulo-2-Addierer-Zweige. In Abbildung 4.1 beschreibt 111_2 den oberen und 101_2 den unteren Addierer-Zweig. Am Ausgang werden die Bits von jedem Addiererzweig abgefragt und zu einer Bitfolge der Länge n zusammengesetzt. Der in Abb. 4.1 dargestellte Codierer nimmt pro Takt ein Informationsbit i entgegen und liefert 2 Codebits c . Dabei ist jedes Codebit nicht nur vom aktuellen Informationsbit, sondern auch von den vorherigen m abhängig.

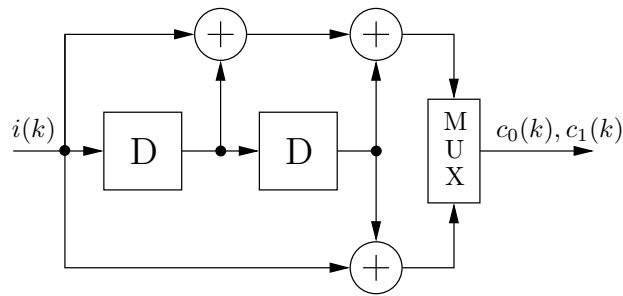


Abbildung 4.1: Blockschaltbild eines Faltungscodierers mit $r = \frac{1}{2}$, $m = 2$ und Generatorpolynomen $[7, 5]_8$

Da die Codesequenzen eines Faltungscodes unendlich lang sind, dies aber nicht den praktischen Anforderungen in einem Übertragungssystem entspricht, gibt es Verfahren, um eine Codesequenz zu beenden. Beim ersten Verfahren, Terminierung, wird die Informationssequenz mit so vielen Nullwerten ergänzt, dass alle Speicherzellen des Codierers mit Null belegt sind, was dem Initialisierungszustand entspricht. Durch diese angehängten Nullbits verringert sich jedoch die Nutzrate des Codes. Ist die Länge der terminierten Informationssequenz jedoch sehr groß im Verhältnis zur Gedächtnislänge des Codes, fällt der Ratenverlust nicht ins Gewicht. Dieses Verfahren wurde auch im Rahmen dieser Arbeit verwendet. Der Ratenverlust wird beim zweiten Verfahren mit dem Namen Truncation vermieden. Bei diesem Verfahren wird die Codesequenz nach einer bestimmten Länge einfach abgeschnitten. Dadurch gehen jedoch Codebits für die Informationsbits am Ende der Sequenz verloren, der Fehlerschutz verschlechtert sich. Bei der dritten Variante, dem so genannten Tail-Biting, wird der Faltungscodierer bereits in dem Zustand gestartet, in dem er nach Verarbeitung von einer bestimmten Zahl von Informationsbits später gestoppt werden soll. Um den Endzustand berechnen zu können, ist die Kenntnis der gesamten Informationssequenz vor Beginn der Codierung des Blocks nötig, es entsteht jedoch kein Ratenverlust. Damit verhält sich der erzeugte Code wie ein Blockcode, es handelt sich um einen so genannten quasi-zyklischen Blockcode [1].

4.2 Decodierung mit Zuverlässigkeitsinformationen

Während sich der vorherige Abschnitt mit der Erzeugung von codierten Sequenzen beschäftigt hat, wird in diesem Abschnitt die Decodierung von Faltungscodes betrachtet. Dazu werden zunächst grundlegende Ansätze zur Decodierung verglichen. Nach einer Einführung in die Log-Likelihood-Verhältnisse, eine Darstellung von Zuverlässigkeitsinformationen, wird die symbolweise Maximum-a-posteriori-Decodierung näher beschrieben, die in dieser Arbeit in Form des BCJR-Algorithmus zum Einsatz kam. Die folgende Beschreibung der Decodierung mit Zuverlässigkeitsinformationen orientiert sich an [29].

4.2.1 Decodierprinzipien

Allen Decodierprinzipien ist gemein, dass sie versuchen, die Wahrscheinlichkeit für ein korrektes Decodierergebnis zu maximieren. Dabei gibt es jedoch grundsätzlich verschiedene Ansätze. Bei der Maximum-Likelihood-Decodierung (ML) wird versucht, ein empfangenes Codewort y als das Codewort \hat{c} zu decodieren, dass mit der größten Wahrscheinlichkeit gesendet wurde,

also [29]

$$\hat{c} = \arg \left[\max_{c \in \mathcal{C}} P(y|c) \right] \quad (4.1)$$

Bei einer Übertragung über einen symmetrischen Binärkanal ist das dasjenige Codewort \hat{c} , das die geringste Hamming-Distanz zu y hat. Dieses Decodierprinzip ist optimal für minimale Codewortfehlerwahrscheinlichkeit. Einen anderen Ansatz verfolgt die symbolweise Maximum-a-posteriori-Decodierung (s/s-MAP). Hierbei wird jeweils nur ein einzelnes Codesymbol c_i des gesamten Codeworts betrachtet. Für dieses wird versucht, das wahrscheinlichste gesendete Symbol zu finden. Alle Symbole c_i des gesamten Codeworts $c = (c_0, c_1, \dots, c_{n-1})$ werden separat betrachtet und entschieden. Wenn $d = (d_0, d_1, \dots, d_{n-1})$ das vom Decodierer entschiedene Codewort ist, so ist die Wahrscheinlichkeit für ein fehlerhaftes Symbol [29]

$$P_{\text{symb}} = P(c_i \neq d_i|y) = 1 - P(c_i = d_i|y), \quad (4.2)$$

was der Bestimmung der Maximum-a-posteriori-Wahrscheinlichkeit der Stelle i entspricht. Somit erhält man, wenn der Empfangsvektor y beobachtet wurde, das am wahrscheinlichsten gesendete Codesymbol c_i aus [29]

$$\hat{c}_i = \arg \left[\max_{s \in \text{GF}(q)} (P(c_i = s|y)) \right]. \quad (4.3)$$

Dieses Decodierprinzip ist folglich optimal für minimale Codesymbolfehlerwahrscheinlichkeit. Da dieses Prinzip die optimale Entscheidung für alle Symbole separat, nicht jedoch für das Codewort als Ganzes trifft, kann sich das Decodierergebnis von dem einer ML-Decodierung unterscheiden. Die Wahrscheinlichkeiten $P(c_i = s|y)$ werden als engl. *symbolwise a posteriori probabilities* (s/s-APP) bezeichnet.

Bei der Verwendung von iterativen Turbo-Schemata ist es notwendig, symbolweise Informationen zwischen den beteiligten Komponenten auszutauschen. Deshalb wird im weiteren Verlauf ein Decodierprinzip benötigt, das nicht auf Codewortbasis, sondern auf Codesymbolbasis arbeitet. Diese Turbo-Schemata basieren darauf, dass die einzelnen Komponenten während des iterativen Prozesses keine harten Entscheidungen (engl. *hard decisions*) über die Symbole treffen, sondern neben der Entscheidung auch Informationen über deren Zuverlässigkeit austauschen (engl. *soft decision*). Diese Zuverlässigkeitsinformationen werden im nächsten Abschnitt betrachtet.

4.2.2 Log-Likelihood-Verhältnisse

Betrachtet man einen binären Code, so hängt die Entscheidung für ein Symbol c_i von den Zuverlässigkeiten $P(c_i = 0|y)$ und $P(c_i = 1|y)$ ab. Ist $P(c_i = 0|y) > P(c_i = 1|y)$, wurde mit größerer Sicherheit eine 0 gesendet, ist $P(c_i = 0|y) < P(c_i = 1|y)$ eine 1. Je größer der Unterschied zwischen den beiden Wahrscheinlichkeiten ist, desto sicherer ist die Entscheidung. Somit ist es sinnvoll, das Verhältnis dieser beiden Wahrscheinlichkeiten bzw. deren Logarithmus zu betrachten. Der Quotient [17]

$$L(c_i) = \log \left(\frac{P(c_i = 0|y)}{P(c_i = 1|y)} \right) \quad (4.4)$$

wird als Log-Likelihood-Verhältnis, kurz L-Wert, bezeichnet. Dabei entspricht das Vorzeichen $\text{sgn}(L(c_i))$ der Entscheidung über das Symbol c_i , während der Betrag $|L(c_i)|$ das Maß der

Zuverlässigkeit dieser Entscheidung widerspiegelt. Für Berechnungen mit Log-Likelihood-Verhältnissen wurde in [17] eine Log-Likelihood-Algebra definiert, auf die an dieser Stelle nicht weiter eingegangen werden soll.

4.2.3 Symbolweise MAP-Decodierung

Um eine s/s-MAP-Decodierung nach Gl. 4.3 für ein Symbol c_i durchführen zu können, wenn ein Codewort $c \in \mathcal{C}$ übertragen und ein Vektor y empfangen wurde, muss zunächst der L-Wert $L(c_i)$ nach Gl. 4.4 bestimmt werden. Das entschiedene Symbol \hat{c}_i erhält man aus [29]

$$\hat{c}_i = \begin{cases} 0 & \text{sgn}(L(c_i)) \geq 0 \\ 1 & \text{sgn}(L(c_i)) < 0 \end{cases}. \quad (4.5)$$

Der L-Wert liefert neben der Entscheidung über \hat{c}_i auch Informationen über die Zuverlässigkeit von \hat{c}_i (engl. *soft output*), die in Abschnitt 4.4 für die iterative Turbo-Entzerrung benötigt werden. Um die bedingten Wahrscheinlichkeiten $P(c_i|y)$ für Gl. 4.4 zu berechnen, werden die Wahrscheinlichkeiten $P(c|y)$ für alle Codewörter c aufsummiert, deren i -tes Element c_i Eins bzw. Null ist. Dazu wird \mathcal{C} in zwei Teilmengen [17]

$$\mathcal{C}_{c_i}^{(0)} = \{c \in \mathcal{C} | c_i = 0\} \quad \text{und} \quad \mathcal{C}_{c_i}^{(1)} = \{c \in \mathcal{C} | c_i = 1\} \quad (4.6)$$

zerlegt, womit

$$P(c_i = 0|y) = \sum_{c \in \mathcal{C}_{c_i}^{(0)}} P(c|y) \quad \text{und} \quad P(c_i = 1|y) = \sum_{c \in \mathcal{C}_{c_i}^{(1)}} P(c|y) \quad (4.7)$$

berechnet werden können. Eingesetzt in Gl. 4.4 erhält man [29]

$$L(c_i) = \log \left(\frac{\sum_{c \in \mathcal{C}_{c_i}^{(0)}} P(c|y)}{\sum_{c \in \mathcal{C}_{c_i}^{(1)}} P(c|y)} \right). \quad (4.8)$$

Mit dem Satz von Bayes kann dies umgeformt werden in [17]

$$L(c_i) = \log \left(\frac{\sum_{c \in \mathcal{C}_{c_i}^{(0)}} p(y|c)P_a(c)}{\sum_{c \in \mathcal{C}_{c_i}^{(1)}} p(y|c)P_a(c)} \right). \quad (4.9)$$

Die a-priori-Wahrscheinlichkeit $P_a(c)$ für die Auswahl von Codewort c entfällt unter der Annahme, dass alle Codewörter mit der selben Wahrscheinlichkeit übertragen werden. Die Wahrscheinlichkeitsdichte $p(y|c)$ kann aus den Wahrscheinlichkeiten $p(y_i|c_i)$ aller Symbole berechnet als

$$p(y|c) = \prod_{m=0}^{n-1} p(y_m|c_m) \quad (4.10)$$

wenn wir einen gedächtnisfreien Kanal mit statistisch unabhängigen Empfangssymbolen y_i , $i = 0, \dots, n-1$, annehmen [29]. Eingesetzt in Gl. 4.9 ergibt sich

$$L(c_i) = \log \left(\frac{\sum_{c \in \mathcal{C}_{c_i}^{(0)}} \prod_{m=0}^{n-1} p(y_m|c_m)}{\sum_{c \in \mathcal{C}_{c_i}^{(1)}} \prod_{m=0}^{n-1} p(y_m|c_m)} \right). \quad (4.11)$$

Die $p(y_i|c_i)$ beschreiben den Zusammenhang zwischen gesendetem und empfangenem Symbol, es handelt sich um die Übergangswahrscheinlichkeitsdichten des Kanals. Mit diesen können nun die L-Werte des Kanals $L_{\text{ch}}(y_i)$ definiert werden als [17]

$$L_{\text{ch}}(y_i) = L(y_i|c_i) = \log \left(\frac{p(y_i|c_i = 0)}{p(y_i|c_i = 1)} \right). \quad (4.12)$$

Mit $p(y_i|c_i = 0) + p(y_i|c_i = 1) = p(y_i)$ lässt sich dies vereinfachen zu [17][29]

$$p(y_i|c_i) = \frac{p(y_i)}{1 + \exp(-L_{\text{ch}}(y_i))} \cdot \exp(-L_{\text{ch}}(y_i)c_i) \quad (4.13)$$

Wird diese Gleichung in Gl. 4.11 eingefügt, so erhält man [17][29]

$$L(c_i) = \log \left(\frac{\sum_{c \in \mathcal{C}_{c_i}^{(0)}} \prod_{m=0}^{n-1} \exp(-L_{\text{ch}}(y_m)c_m)}{\sum_{c \in \mathcal{C}_{c_i}^{(1)}} \prod_{m=0}^{n-1} \exp(-L_{\text{ch}}(y_m)c_m)} \right) \quad (4.14)$$

und kann die L-Werte $L(c_i)$ aus den L-Werten des Kanals $L_{\text{ch}}(y_i)$ berechnen. Diese Gleichung kann umgeschrieben werden in die anschaulichere Form [17][29]

$$L(c_i) = \underbrace{\log \left(\frac{1}{\exp(-L_{\text{ch}}(y_i))} \right)}_{L_{\text{ch}}(y_i)} + \underbrace{\log \left(\frac{\sum_{c \in \mathcal{C}_{c_i}^{(0)}} \prod_{\substack{m=0 \\ m \neq i}}^{n-1} \exp(-L_{\text{ch}}(y_m)c_m)}{\sum_{c \in \mathcal{C}_{c_i}^{(1)}} \prod_{\substack{m=0 \\ m \neq i}}^{n-1} \exp(-L_{\text{ch}}(y_m)c_m)} \right)}_{L_{\text{ext}}(y)} \quad (4.15)$$

Der L-Wert $L(c_i)$ kann folglich in zwei Teile zerlegt werden. Den L-Wert des Kanals $L_{\text{ch}}(y_i)$, der vom empfangenen Symbol y_i abhängt und aus diesem auf c_i schließt, sowie $L_{\text{ext}}(y)$, der von allen anderen empfangenen Symbolen y abhängt, jedoch nicht von y_i . Aufgrund der Kanaldecodierung ist die Information des gesendeten Symbols c_i über mehrere Symbole verteilt und die übrigen empfangenen Symbole y tragen Informationen über y_i , die für die $L_{\text{ch}}(y_i)$ ausgewertet werden. Daher werden $L_{\text{ch}}(y_i)$ auch als intrinsische (lat. *von innen kommend*) L-Werte bezeichnet, während $L_{\text{ext}}(y)$ für extrinsische (lat. *von außen kommend*) L-Werte steht.

Falls weitere Informationen über ein Symbol c_i vorliegen, die nicht aus der Beobachtung von y_i stammen, so wird $L(c_i)$ um eine dritte Komponente erweitert. Ist bekannt, dass ein Symbol c_i mit einer bestimmten Wahrscheinlichkeit $P_a(c_i)$ auftritt, so wird aus dieser a-priori-Information der a-priori-L-Wert [29]

$$L_a(c_i) = \log \frac{P_a(c_i = 0)}{P_a(c_i = 1)} \quad (4.16)$$

gebildet. Dies ist vor allem bei einem Turbo-Entzerrer von Bedeutung, der nicht nur das empfangene Symbol y kennt, sondern ab der zweiten Iteration zusätzlich auch a-priori-Informationen vom Decodierer (siehe Abschnitt 4.4). Die a-priori-L-Werte werden in der Berechnung von $L(c_i)$ berücksichtigt, in dem sie zu $L_{\text{ch}}(y_i)$ addiert werden. Somit verändert sich

der Term für $L_{\text{ext}}(y)$ in Gl. 4.15 in [29]

$$L_{\text{ext}}(y, L_a(c)) = \log \left(\frac{\sum_{c \in \mathcal{C}_{c_i}^{(0)}} \prod_{\substack{m=0 \\ m \neq i}}^{n-1} \exp(-(L_{\text{ch}}(y_m) + L_a(c_i)) c_m)}{\sum_{c \in \mathcal{C}_{c_i}^{(1)}} \prod_{\substack{m=0 \\ m \neq i}}^{n-1} \exp(-(L_{\text{ch}}(y_m) + L_a(c_i)) c_m)} \right). \quad (4.17)$$

Ein gebräuchlicher Algorithmus zur Implementierung einer s/s-MAP-Decodierung für die Trellisdarstellung ist der BCJR-Algorithmus, der nach seinen Erfindern Bahl, Cocke, Jelinek und Raviv benannt ist. Er kann mit L-Werten an Ein- und Ausgang verwendet werden und ist optimal im Sinne von minimaler Symbolfehlerwahrscheinlichkeit. Der BCJR-Algorithmus wurde im Decodierer des im Rahmen dieser Arbeit simulierten Übertragungssystems verwendet, für eine detaillierte Beschreibung des Algorithmus sei jedoch auf [1] verwiesen.

4.3 Punktierte Faltungscodes

Punktierte Faltungscodes werden aus gewöhnlichen Faltungscodes, so genannten Muttercodes, erzeugt, indem periodisch Codebits der Codesequenz \mathcal{C} des Muttercodes nach einem festgelegten Muster \mathcal{S} gestrichen werden. Das Muster \mathcal{S} wird charakterisiert durch seine Periodizität p und die Anzahl $p - w$ der pro Periode gestrichenen (punktierten) Bits. Die Rate des punktierten Codes ist somit [1]

$$r_p = \frac{p}{w} r_m, \quad (4.18)$$

wobei r_m die Rate des Muttercodes ist. Die Rate des punktierten Codes ist somit größer als die des Muttercodes, da einige Codebits gestrichen und nicht übertragen werden. Auf Seite des Empfängers wird die Punktierung rückgängig gemacht, indem an Stelle der punktierten Bits Dummy-Bits eingefügt werden. Wenn ein punktierter Code der Rate $r_p = k_p/n_p$ von einem Muttercode der Rate $r_m = 1/n_m$ abgeleitet wird, kann das Streichmuster $\mathcal{S}(p, w)$ als Punktierungsmatrix $\underline{\underline{P}}$ der Größe $n_m \times k_p$ dargestellt werden, deren Elemente $p_{ij} \in \{0, 1\}$ sind. Dabei bedeutet 0, dass ein Bit entfernt wird und 1, dass das Bit erhalten bleibt.

Es ist vorteilhaft, die Rate eines niederratigen Faltungscodes durch Punktierung zu steigern, anstatt direkt einen höherratigen Code einzusetzen. Die Decodierkomplexität nimmt mit anwachsender Rate eines Faltungscodes zu, während ein punktierter Code mit der Komplexität des Muttercodes decodiert werden kann. Außerdem ist es mit Punktierung möglich, neue höherratige Faltungscodes zu erhalten, die auf bereits bekannten Codes mit guten Distanzeigenschaften beruhen [1].

Die Verwendung von punktierten Faltungscodes für codierte Blockübertragung ist häufig anzutreffen und leicht als zusätzlicher Block am Ausgang des Kanalcodierers zu implementieren. Im Rahmen dieser Arbeit wurden keine Simulationen für punktierte Codes erstellt. Trotzdem wurde das De-Puncturing im Schema des Turbo-Entzerrers (siehe Abschnitt 4.4) berücksichtigt.

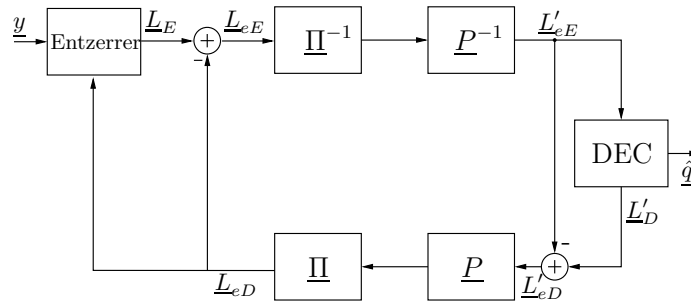


Abbildung 4.2: Blockschaltbild von gemeinsamer Entzerrung und Decodierung

4.4 Turbo-Entzerrung

Verwendet eine Übertragung Kanalcodierung, so wird im einfachsten Fall der Empfänger invers zum Sender aufgebaut. Die empfangenen Symbole werden zunächst entzerrt und dann an den Decodierer weitergereicht. Die Performance dieses Schemas erhöht sich, wenn der Entzerrer den Decodierer nicht nur mit den entschiedenen Codebits versorgt, sondern auch mit Zuverlässigkeitsinformationen (siehe Abschnitt 4.2). Einen großen Fortschritt jedoch stellt die in [10] vorgestellte iterative Empfängerstruktur dar, die in Abbildung 4.2 skizziert ist. Die einzelnen Komponenten, Entzerrer und Decodierer, arbeiten nicht alleine, sondern versorgen sich gegenseitig mit Zuverlässigkeitsinformationen während mehrerer Iterationen. Hierzu wurde das als *Turbo Coding* bekannte iterative Decodierschema für parallel verkettete Faltungscodes abgewandelt. Dahinter steht die Erkenntnis, dass eine codierte Übertragung über einen Kanal frequenzselektiven Kanal mit ISI auch als Verkettung zweier Codes, des Kanalcodes einerseits und der Übertragungsfunktion des Kanals andererseits, betrachtet werden kann. Deshalb wird dieses Empfängerkonzept analog als Turbo-Entzerrer (engl. *Turbo equalizer*) bezeichnet.

Abbildung 4.2 zeigt den Aufbau eines typischen Turbo-Entzerrers, vergleiche [31]. Als Entzerrerkomponente können RNN oder SCE zum Einsatz kommen. Die empfangenen Symbole \underline{y} durchlaufen zunächst den Entzerrer. Dieser liefert am Ausgang die L-Werte \underline{L}_E , die nach dem De-Interleaving $\underline{\Pi}^{-1}$ und dem De-Puncturing \underline{P}^{-1} als \underline{L}'_{eE} am Eingang des Decodierers DEC anliegen, der im Rahmen dieser Arbeit als BCJR-Decodierer realisiert war. Die Zuverlässigkeitswerte am Eingang können vom Decodierer nun als a-priori-Informationen verwendet werden und die Decodierung verbessern. Der Gewinn an Zuverlässigkeitsinformationen am Ausgang, die extrinsische Information $\underline{L}'_{eD} = \underline{L}'_D - \underline{L}'_{eE}$ wird nun wieder zum Entzerrer zurückgeführt, und dient nun diesem als a-priori-Information für den nächsten Iterationsdurchgang. Die Subtraktion verhindert, dass Zuverlässigkeiten früherer eigener Iterationen rückgekoppelt werden. Vorher müssen die L-Werte jedoch durch Puncturing \underline{P} und Interleaving $\underline{\Pi}$ wieder ins passende Format gebracht werden. Um wiederum nur extrinsische Information an den Decodierer weiterzugeben, wird auch hier die Differenz $\underline{L}_{eE} = \underline{L}_E - \underline{L}_{eD}$ gebildet. In der ersten Iteration entfiel dieser Schritt, da \underline{L}_{eD} zu Null initialisiert ist. Nach einer festgelegten Zahl von Iterationen entscheidet der Decodierer DEC hart und liefert die entschiedenen Symbole $\hat{\underline{q}}$ am Ausgang. Damit kann die Verarbeitung des nächsten Blocks der Übertragungssequenz beginnen.

4.4.1 Turbo-Entzerrung mit dem RNN-Entzerrer

Um den in 3.3 vorgestellten RNN-Entzerrer in einem Turbo-Schema verwenden zu können, ist eine Erweiterung nötig, die in [31] beschrieben ist. Wie in Abb. 4.2 zu erkennen ist, besitzt der Entzerrer in diesem Fall zwei Eingänge. Neben dem Empfangsvektor aus dem Kanal erhält er a-priori-Informationen vom Decodierer, um mit deren Hilfe seine Entzerrung in einem weiteren Iterationsschritt zu verbessern. An seinem Ausgang wird nun nicht mehr hart entschieden, sondern er liefert Zuverlässigkeitsinformationen in Form von L-Werten, die dem Decodierer als a-priori-Informationen dienen. Dabei muss berücksichtigt werden, dass der Entzerrer zunächst Zuverlässigkeiten für Symbole benutzt, der Decodierer hingegen Zuverlässigkeitsinformationen für Codebits erwartet. Weiterhin müssen einige Formeln um die Beschränkung auf den Code \mathcal{C} erweitert werden. So lautet das Optimierungskriterium aus Gl. 3.31, das die verbleibende Interferenz darstellt mit der Beschränkung auf \mathcal{C} [31]

$$\sigma_{\text{res},i}^2 = J_{\min} = \mathbb{E} \left\{ |x_i - \check{x}_i|^2 \mid \tilde{x}_i, \mathcal{C} \right\} \quad (4.19)$$

Für das optimale geschätzte Symbol \check{x}_i gilt dann [31]

$$\check{x}_i^{[\eta]} = \Theta \left(\tilde{x}_i^{[\eta]} \right) = \mathbb{E} \left\{ x_i^{[\eta]} \mid \tilde{x}_i^{[\eta]}, \mathcal{C} \right\} = \sum_{m=1}^M a_m P(x_i = a_m \mid \tilde{x}_i^{[\eta]}, \mathcal{C}) \quad (4.20)$$

mit

$$P(x_i = a_m \mid \tilde{x}_i^{[\eta]}, \mathcal{C}) = \frac{p(\tilde{x}_i^{[\eta]} \mid x_i = a_m) P(x_i = a_m \mid \mathcal{C})}{p(\tilde{x}_i^{[\eta]})} \quad (4.21)$$

und

$$p(\tilde{x}_i^{[\eta]}) = \sum_{m=1}^M p(\tilde{x}_i^{[\eta]} \mid x_i = a_m) P(x_i = a_m \mid \mathcal{C}). \quad (4.22)$$

Die statistische Unabhängigkeit von \mathcal{C} und $\tilde{x}_i^{[\eta]}$ muss dabei durch den Interleaver sichergestellt sein. Dann erhält man [31]

$$\Theta \left(\tilde{x}_i^{[\eta]} \right) = \frac{\sum_{m=1}^M a_m p \left(\tilde{x}_i^{[\eta]} \mid x_i = a_m \right) P(x_i = a_m \mid \mathcal{C})}{\sum_{m=1}^M p \left(\tilde{x}_i^{[\eta]} \mid x_i = a_m \right) P(x_i = a_m \mid \mathcal{C})}, \quad (4.23)$$

wobei die Wahrscheinlichkeiten $P(x_i = a_m \mid \mathcal{C})$ aus den vom Decodierer gelieferten L-Werten \underline{L}_{eD} berechnet werden als [31]

$$\begin{aligned} P(x_i = a_m \mid \mathcal{C}) &= \prod_{\nu=1}^{\log_2 M} P(c_{i,\nu} = \mathcal{B}[a_j, \nu] \mid \mathcal{C}) \\ &= \prod_{\nu=1}^{\log_2 M} \frac{\exp((1 - \mathcal{B}[a_j, \nu]) L_{eD}(c_{i,\nu} \mid \mathcal{C}))}{1 + \exp(L_{eD}(c_{i,\nu} \mid \mathcal{C}))}. \end{aligned} \quad (4.24)$$

Dabei bezeichnet \mathcal{B} die Mapping-Regel zwischen Symbolen und Bits. Die L-Werte \underline{L}_E für die Codebits \underline{c} am Ausgang des Entzerrers werden aus $\tilde{\underline{x}}^{[\eta]}$ berechnet als [31]

$$L_E \left(\tilde{\underline{x}}_i^{[\eta]} | c_i \right) = \ln \left(\frac{\sum_{x_i^{\{0\}}} \frac{1}{2\pi\sigma^2} \exp \left(-\frac{|x_i^{\{0\}} - \tilde{x}_i^{[\eta]}|^2}{2\sigma^2} \right)}{\sum_{x_i^{\{1\}}} \frac{1}{2\pi\sigma^2} \exp \left(-\frac{|x_i^{\{1\}} - \tilde{x}_i^{[\eta]}|^2}{2\sigma^2} \right)} \right) \quad (4.25)$$

Dabei kann für die Varianz σ^2 entweder die Rauschvarianz verwendet werden, was eine ideale Entzerrung unterstellt, oder die Varianz aller Störungen $\sigma_{\text{ges},i}^2$, die auch die verbleibende Interferenz $\sigma_{\text{res},i}^2$ nach der letzten Iteration beinhaltet.

4.4.2 Turbo-Entzerrung mit dem SCE

Ebenso wie beim RNN im vorherigen Abschnitt müssen auf beim SCE Erweiterungen vorgenommen werden, um ihn in einem Turbo-Schema einsetzen zu können [12]. Einige Gleichungen müssen um die Beschränkung auf den Code \mathcal{C} erweitert werden. Die in Gl. 3.41 und 3.49 verwendete Wahrscheinlichkeit $P(\underline{x}_l = \underline{a}_m | \underline{\xi}^{[\eta,l]})$ lautet nun unter der Annahme, dass $\underline{\xi}^{[\eta,l]}$ und \mathcal{C} statistisch unabhängig sind, was durch den Interleaver sichergestellt werden muss [12],

$$P(\underline{x}_l = \underline{a}_m | \underline{\xi}^{[\eta,l]}) = \frac{p(\underline{\xi}^{[\eta,l]} | \underline{x}_l = \underline{a}_m, \mathcal{C}) P(\underline{x}_l = \underline{a}_m | \mathcal{C})}{p(\underline{\xi}^{[\eta,l]})} \quad (4.26)$$

Die Wahrscheinlichkeiten $P(\underline{x}_l = \underline{a}_m | \mathcal{C})$ können, vergleiche Gl. 4.24, aus den extrinsischen L-Werten des Decodierers \underline{L}_{eD} berechnet werden aus [12]

$$P(x_i = a_m | \mathcal{C}) = \prod_{\nu=1}^{\log_2 M} \frac{\exp((1 - \mathcal{B}[a_j, \nu]) L_{eD}(c_{i,\nu} | \mathcal{C}))}{1 + \exp(L_{eD}(c_{i,\nu} | \mathcal{C}))}. \quad (4.27)$$

Am Ausgang werden nun bit-basierte Zuverlässigkeitsinformationen $L_E(c_{li} | \underline{y})$ benötigt. Wie in [12] gezeigt, können aus den Symbolwahrscheinlichkeiten des Entzerrers $P(\underline{x}_l | \underline{\xi}^{[\eta,l]})$ die L-Werte für die Codebits $L_E(c_{li} | \underline{y})$ berechnet werden mit

$$L_E(c_{li} | \underline{y}) = L_E(c_{li} | \underline{\xi}^{[\eta,l]}) = \log \frac{\sum_{\underline{a}_m \in \mathcal{A}_i^{\{0\}}} P(\underline{x}_l = \underline{a}_m | \underline{\xi}^{[\eta,l]})}{\sum_{\underline{a}_m \in \mathcal{A}_i^{\{1\}}} P(\underline{x}_l = \underline{a}_m | \underline{\xi}^{[\eta,l]})}, \quad (4.28)$$

wobei $\mathcal{A}_i^{\{0\}}$ die Menge der Modulationssymbole bezeichnet, deren Bitsequenz eine Null an der Stelle i hat, während $\mathcal{A}_i^{\{1\}}$ eine Eins an der Stelle i hat.

5 Verfahren zur Verbesserung der Leistungsfähigkeit

In diesem Kapitel werden Verfahren vorgestellt, die die Leistungsfähigkeit der in Kapitel 2 vorgestellten MIMO-OFDM-Übertragungen verbessern können. Diese Verfahren können in zwei Gruppen eingeteilt werden. Für die Spreizverfahren ist keine Kanalkennntnis nötig. Sie beeinflussen die Leistungsfähigkeit des Systems, indem sie die Diversität erhöhen und so die Wahrscheinlichkeit des Verlustes einzelner Symbole verringern. Für Precoding hingegen ist eine zumindest teilweise Kenntnis des Kanals am Sender nötig. Ist diese vorhanden, können die gesendeten Symbole an die Eigenschaften des Kanals angepasst werden. Dies kann die Beeinflussung der Symbole durch den Kanal verringern und somit die Bitfehlerhäufigkeit verbessern. Des Weiteren werden im dritten Abschnitt dieses Kapitels Ergebnisse für die in Kapitel 4 vorgestellte codierte Übertragung, für die ebenfalls keine Kanalkennntnis nötig ist, gezeigt. Dabei wurden sowohl zeitvariante stochastische Kanäle, als auch zeitinvariante Testkanäle untersucht. Das Konvergenzverhalten von letzteren wurde zusätzlich mit Hilfe von EXIT-Charts analysiert.

5.1 Spreizverfahren

Spreizverfahren (engl. *spreading schemes*) sind lineare Transformationen, die dazu dienen, die Energie von Symbolen über eine oder mehrere Dimensionen zu verteilen und somit die Übertragung robuster gegen Störungen zu machen. Seit langer Zeit sind Spreizverfahren wie *Direct Sequence Spread Spectrum (DSSS)* oder *Frequency Hopping Spread Spectrum (FHSS)* [27] im Einsatz, bei denen ein schmalbandiges Nutzsignal über einen breitbandigen Kanal im Frequenzbereich gespreizt wird. Die MIMO-Technik ermöglicht, den Raum als zusätzliche Dimension beim Spreizen auszunutzen und somit höhere Diversität zu erreichen. Als Diversität bezeichnet man die Verteilung von Information innerhalb einer oder über mehrere physikalische Größen hinweg mit dem Ziel, die Möglichkeit des Verlust der Information zu minimieren. Dies können verschiedene Frequenzen, Ausbreitungswege oder Modulationsalphabete sein.

Im Folgenden werden zwei Mehrträger-Spreizverfahren (engl. *multicarrier spread spectrum schemes*) beschrieben, die es ermöglichen, die Energie der Sendesymbole über mehrere Subträger zu verteilen, wie sie beim in Abschnitt 2.2 vorgestellten OFDM verwendet werden. Diese Spreizverfahren bauen auf SISO- bzw. MIMO-Übertragungssysteme auf und erweitern sie um eine Spreizkomponente am Sender und eine Entspreizkomponente am Empfänger. In einem ungespreizten System sind die Subträger nicht-frequenzselektivem Rayleigh-Fading unterworfen und betroffene Symbole können somit vollständig ausgelöscht werden. Die Korrelation des Fadings der einzelnen Subträger ist abhängig von der Kohärenzbandbreite des Kanals. Subträger, die mehr als die Kohärenzbandbreite im Spektrum von einander entfernt

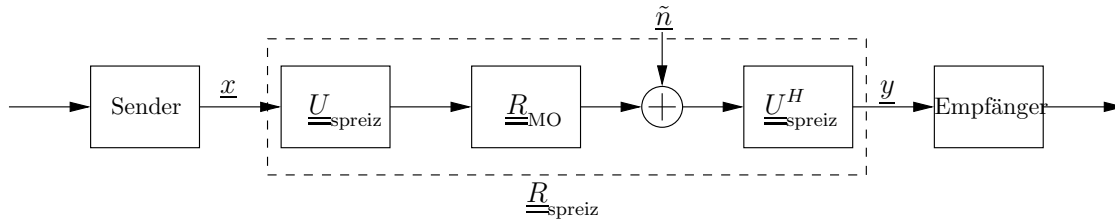


Abbildung 5.1: Der MIMO-OFDM-Ersatzkanal und das Spreizen und Entspreizen lassen sich zu einem gespreizten Ersatzkanal zusammenfassen.

sind, faden unkorreliert. Somit sinkt die Wahrscheinlichkeit, dass mehrere Subträger zeitgleich von tiefem Fading betroffen sind, wenn die Kohärenzbandbreite klein ist im Vergleich zur Bandbreite der Subträger. Wird somit jedes Symbol über eine genügend große Anzahl von Subträgern gespreizt, verschwindet somit auch die Wahrscheinlichkeit der vollständigen Auslöschung eines Symbols. Dadurch erzielt das Spreizen einen Diversitätsgewinn gegenüber einer ungespreizten Übertragung.

Auf OFDM basierende Mehrträger-Spreizverfahren erhöhen durch eine lineare Transformation die Diversität des Symbolalphabets im Signalraum [5]. Der Aufwand für die Detektion ist gegenüber anderen Verfahren (z.B. [2]) günstig, es wird jedoch ein Entzerrer auf Seite des Empfängers benötigt. Ein gutes Spreizverfahren muss neben Orthogonalität und möglichst gleichmäßiger Verteilung der Energie über alle Subträger weitere Kriterien erfüllen. In [5] wurden Kriterien zur Beurteilung von Spreizverfahren entwickelt auf Basis der asymptotischen Bitfehlerhäufigkeit bei hohem SNR.

Das Blockschaltbild eines Spreizschemas für MIMO-OFDM-Übertragungen ist in Abb. 5.1 dargestellt. Zwischen Sender und Kanal wird eine Spreizkomponente in Form einer Spreizmatrix $\underline{U}_{\text{spreiz}}$ hinzugefügt, auf Seite des Empfängers wird die Spreizung im Falle orthonormaler Spreizmatrizen durch $\underline{U}_{\text{spreiz}}^H$ rückgängig gemacht. Beispiele für solche Spreizmatrizen werden in den beiden folgenden Abschnitten vorgestellt. Das gesamte Spreizschema lässt sich zu einem gespreizten Ersatzkanal $\underline{R}_{\text{spreiz}}$ zusammenfassen.

5.1.1 MC-CDM

MC-CDM (engl. *Multicarrier Code Division Multiplex*) ist ein Mehrträger-Spreizverfahren, das auf OFDM aufsetzt [23][9]. Die Sendesymbolvektoren werden vor der Übertragung mit einer Spreizmatrix $\underline{U}_{\text{MC-CDM}}$ multipliziert, wodurch die Informationen über die Vektorkomponenten verteilt werden. Jeder Subträger beinhaltet somit die Energie mehrerer Symbole. Auf Seite des Empfängers wird die KMF-Matrix um die konjugiert Komplexe der Spreizmatrix erweitert. Wählt man $\underline{U}_{\text{MC-CDM}}$ als unitäre Matrix, so gilt

$$\left(\underline{U}_{\text{MC-CDM}}\right)^{-1} = \underline{U}_{\text{MC-CDM}}'$$

Zusammen mit der OFDM-Ersatzkanalmatrix $\underline{R}_{\text{MO}}$ (siehe Abschnitt 2.2.2) kann eine neue Ersatzkanalmatrix $\underline{R}_{\text{MC-CDM}}$ des gespreizten Kanals gebildet werden [23]:

$$\underline{R}_{\text{MC-CDM}} = \underline{U}_{\text{MC-CDM}}^H \underline{R}_{\text{MO}} \underline{U}_{\text{MC-CDM}} \quad (5.1)$$

Durch die Orthonormalität der Spreizmatrix tritt im Falle eines nicht-frequenzselektiven Kanals keine Intersubkanal-Interferenz (engl. *intersubchannel interference*, ISCI) auf. Frequenzselektive Kanäle führen jedoch zu ISCI und machen eine Entzerrung im Empfänger notwendig, die in Abschnitt 3 beschrieben ist.

Die Spreizmatrizen $\underline{\underline{U}}_{\text{MC-CDM}}$ für MIMO-OFDM haben die Form [40][9]

$$\underline{\underline{U}}_{\text{MC-CDM}} = \begin{pmatrix} \underline{\underline{S}} & \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{S}} & \underline{\underline{0}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{S}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{S}} \end{pmatrix} \quad (5.2)$$

wobei $\underline{\underline{S}}$ eine $N \times N$ -Sub-Spreizmatrix und $\underline{\underline{0}}$ die $N \times N$ -Nullmatrix ist. Die Größe N der Sub-Spreizmatrizen wird dabei so gewählt, dass sich maximale Diversität ergibt, was der Anzahl der zur Verfügung stehenden OFDM-Subträger entspricht.

Walsh-Hadamard-Matrizen sind Matrizen, deren Elemente entweder -1 oder 1 sind und deren Zeilen und Spalten jeweils zu einander orthogonal sind [1]. Sie erfüllen somit die Anforderung an Unitarität und werden in dieser Arbeit als Spreizmatrizen $\underline{\underline{S}}$ verwendet, beispielsweise

$$\underline{\underline{S}}_{4 \times 4} = \underline{\underline{WH}}(4) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}, \quad (5.3)$$

wobei $\underline{\underline{WH}}(4)$ eine Walsh-Hadamard-Matrix vierter Ordnung bezeichnet.

Die Energie der Symbole wird innerhalb jeder Antenne gleichmäßig auf alle Subträger verteilt, über Antennengrenzen hinweg findet jedoch kein Spreizen statt. Während die Dimension der Frequenz beim Spreizen ausgenutzt wird, bleibt die räumliche Dimension somit ungenutzt und MC-CDM erreicht nicht volle Diversität, sondern

$$\text{Div} = L n_{\text{R}}. \quad (5.4)$$

Abb. 5.2 zeigt ein Beispiel für $n_{\text{T}} = 4$, $n_{\text{R}} = 4$, $L = 2$ und $N = 8$ Subträger. Die Kanalmatrix $\underline{\underline{R}}_{\text{MO}}$ aus 5.2a wurde nach Gl. 5.1 mit $\underline{\underline{U}}_{\text{MC-CDM}}$ (5.2b, weiße Felder stehen für -1, schwarze für 1 und Grau für 0) gespreizt und es entsteht ein gespreizter Ersatzkanal $\underline{\underline{R}}_{\text{MC-CDM}}$ (5.2c). An der Blockstruktur in (b) ist zu erkennen, dass das Spreizen jeweils nur innerhalb einer Sendeantenne stattfindet.

5.1.2 MC-CAFS

MC-CAFS (engl. *Multicarrier Cyclic Antenna Frequency Spreading*) ist eine Familie von Spreizmatrizen, die sowohl die räumliche Dimension als auch die Dimension der Frequenz ausnutzen und somit höhere Diversität erreichen als das in Abschnitt 5.1.1 vorgestellte MC-CDM. Die Symbole werden über alle Sendeantennen und über einen Satz von Frequenzen, der für jede Sendeantenne unterschiedlich ist, gespreizt. MC-CAFS wurde vorgestellt in [41] und in [44][43] fortgeführt.

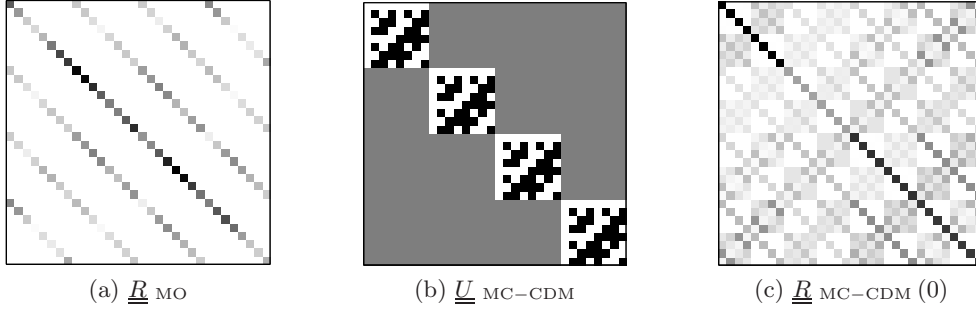


Abbildung 5.2: Beispiel für MC-CDM-Spreading. (a): \underline{R}_{MO} für $n_T = 4$, $n_R = 4$, $L = 2$ und $N = 8$ Subträger, (b): Spreizmatrix \underline{U}_{MC-CDM} , (c): Gespreizter Kanal $\underline{R}_{MC-CDM}(0)$

Die MC-CAFS-Spreizmatrix $\underline{U}_{MC-CAFS}$ ist definiert als [41]

$$\underline{U}_{MC-CAFS} = \begin{pmatrix} s_{11}\underline{I} & s_{12}\underline{I} & \cdots & s_{1(B_{n_T})}\underline{I} \\ \vdots & \vdots & \ddots & \vdots \\ s_{B_1}\underline{I} & s_{B_2}\underline{I} & \cdots & s_{B(B_{n_T})}\underline{I} \\ \hline s_{(B_1+1)1}\underline{I}_{p_1} & s_{(B_1+1)2}\underline{I}_{p_1} & \cdots & s_{(B_1+1)(B_{n_T})}\underline{I}_{p_1} \\ \vdots & \vdots & \ddots & \vdots \\ s_{(B_2)1}\underline{I}_{p_1} & s_{(B_2)2}\underline{I}_{p_1} & \cdots & s_{(B_2)(B_{n_T})}\underline{I}_{p_1} \\ \hline s_{(B_2+1)1}\underline{I}_{p_2} & s_{(B_2+1)2}\underline{I}_{p_2} & \cdots & s_{(B_2+1)(B_{n_T})}\underline{I}_{p_2} \\ \vdots & \vdots & \ddots & \vdots \\ s_{(B_3)1}\underline{I}_{p_2} & s_{(B_3)2}\underline{I}_{p_2} & \cdots & s_{(B_3)(B_{n_T})}\underline{I}_{p_2} \\ \hline \vdots & \vdots & \vdots & \vdots \\ \hline s_{(B_{n_T-1}+1)1}\underline{I}_{p_n} & \cdots & \cdots & s_{(B_{n_T-1}+1)(B_{n_T})}\underline{I}_{p_n} \\ \vdots & \vdots & \ddots & \vdots \\ s_{(B_{n_T})1}\underline{I}_{p_n} & \cdots & \cdots & s_{(B_{n_T})(B_{n_T})}\underline{I}_{p_n} \end{pmatrix} \quad (5.5)$$

Die Blockgröße B bezeichnet die Anzahl der Frequenzen an jeder Sendeantenne, über die jedes Symbol gespreizt wird. $B_m = mB$ und die horizontalen Linien verdeutlichen die Trennung der einzelnen Sendeantennen. Die s_{ij} sind Elemente einer Sub-Spreizmatrix \underline{S} der Größe $(n_TB) \times (n_TB)$, \underline{I} ist eine Einheitsmatrix der Größe $(N/B) \times (N/B)$ und \underline{I}_{p_i} ist ihre i -te Permutation. Da sich aus \underline{I} $(N/B) - 1$ verschiedene Permutationen bilden lassen, $(n_T - 1)$ Permutationen nötig sind und sich keine Permutation wiederholen darf, um die Orthogonalität der Spreizmatrix $\underline{U}_{MC-CAFS}$ nicht zu verlieren, muss gelten [41]

$$\frac{N}{B} \geq n_T. \quad (5.6)$$

Um die Mehrwegeausbreitung des Kanals auszunutzen, muss jedes Symbol über eine genügend große Anzahl von Frequenzen gespreizt werden. Das bedeutet, dass die Blockgröße B mindestens gleich der Anzahl der Ausbreitungspfade L sein muss [41], also

$$B \geq L. \quad (5.7)$$

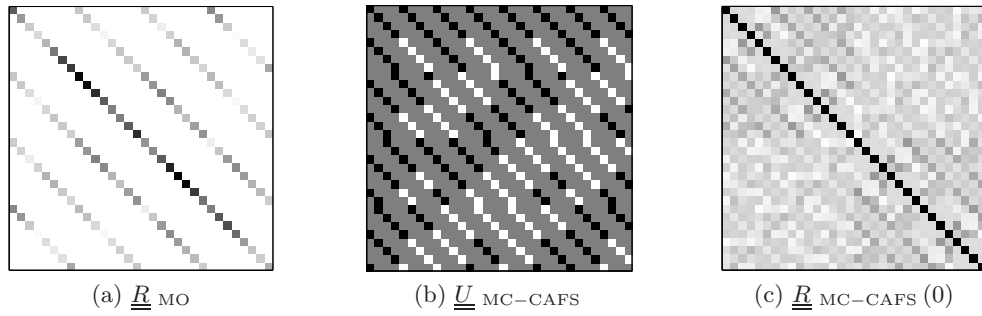


Abbildung 5.3: Beispiel für MC-CAFS-Spreading. (a): \underline{R}_{MO} für $n_T = 4$, $n_R = 4$, $L = 2$ und $N = 8$, (b): Spreizmatrix $\underline{U}_{MC-CAFS}$ ($B = 2$), (c): Gespreizter Kanal $\underline{R}_{MC-CAFS}(0)$

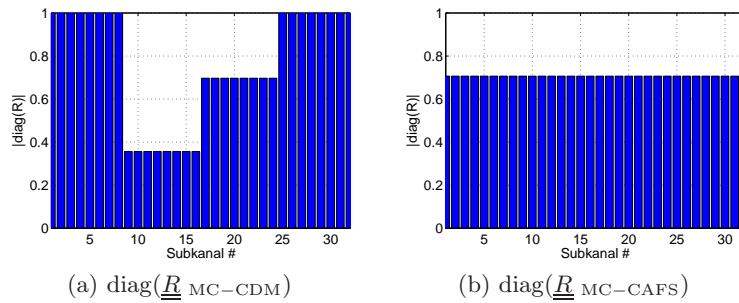


Abbildung 5.4: Beträge der Diagonalelemente von \underline{R}_{MC-CDM} und $\underline{R}_{MC-CAFS}$ für $N = 8$ und $n_T = 4$: Während bei MC-CDM die Energie der Subkanäle nur innerhalb einer Sendeantenne konstant ist, ist sie dies bei MC-CAFS über alle Sendeantennen hinweg.

Aus Gl. 5.6 und Gl. 5.7 folgt die Bedingung für die Blockgröße B

$$L \leq B \leq \frac{N}{n_T} \quad \text{mit} \quad \left(\frac{N}{n_T}\right) \in \mathbb{N}^+ \quad (5.8)$$

Somit gibt es mehrere mögliche Werte für B in einem System mit gegebenem N , n_T und L .

In Abbildung 5.4 wird die Verteilung der Sendeleistung über die Subkanäle verglichen. Von den 32 gezeigten Subkanälen eines gehören jeweils $N = 8$ zu einer der $n_T = 4$ Sendeantennen. Bei MC-CAFS wird die Sendeleistung gleichmäßig über alle Subkanäle verteilt, während bei MC-CDM eine Nivellierung nur jeweils innerhalb einer Sendeantenne stattfindet. Die Verteilung der Sendeleistung in einem ungespreizten System wurde in Abb. 2.7b gezeigt. Durch die Nutzung der räumlichen Dimension der Sendeantennen erhöht sich die Diversität im Falle von MC-CAFS auf

$$\text{Div} = L n_T n_R \quad (5.9)$$

und ist somit größer als im Falle von MC-CDM (vergleiche Gl. 5.4).

Eine Verbesserung der Performance ist möglich, wenn das MC-CAFS-Spreading um Rotationen erweitert wird. In [44] wurden drei verschiedene Rotationstypen vorgestellt, von denen in dieser Arbeit jedoch nur die dort als Typ II bezeichneten Rotationen verwendet werden, die sich als am effizientesten herausgestellt haben. Dabei wird die MC-CAFS-Spreizmatrix

aus Gl. 5.5 mit einer Diagonalmatrix $\underline{\underline{D}}_n$ multipliziert, so dass die neue rotierte Spreizmatrix

$$\underline{\underline{U}}_{\text{MC-CAFS}_{\text{RotII}}} = \underline{\underline{U}}_{\text{MC-CAFS}} \underline{\underline{D}}_n \quad (5.10)$$

entsteht. Die Diagonalmatrix $\underline{\underline{D}}_n$ hat die Elemente

$$d_{ii} = \exp\left(j \frac{2\pi}{M} \frac{(i-1)}{n}\right), \quad (5.11)$$

wobei M die Größe des Symbolalphabetes ist und $n = n_T N$.

5.1.3 Simulation

Systemmodell

Die Performance von gespreizten Systemen mit MC-CDM und MC-CAFS im Vergleich zu ungespreizten Systemen wurde simuliert für $n_T = n_R = 4$ Antennen, $N = 32$ Subträger, $L = 6$ und ein 4PSK-Modulationsalphabet. Die Blockgröße von MC-CAFS wurde zu $B = 8$ gewählt. Dabei wurden zeitvariante stochastische Kanäle mit einer Kohärenzzeit verwendet, die der Symboldauer entsprach. Auf Seite des Empfängers wurde keine Korrelation angenommen. Am Sender wurden Korrelationen mit exponentiell abfallenden Korrelationskoeffizienten (siehe Abschnitt 2.1.2) simuliert, wie auch in allen folgenden Simulationen. Zur Vereinfachung werden deshalb im Weiteren nicht die Korrelationsmatrizen $\underline{\underline{k}}_T$ bzw. $\underline{\underline{k}}_R$ angegeben, sondern nur der Korrelationskoeffizient ρ_{ij} zweier Antennen mit Abstand $i - j = 1$, der als ρ_T bzw. ρ_R bezeichnet wird. Somit steht $\rho_T = 0,3$ für eine Korrelationsmatrix

$$\underline{\underline{k}}_T = \begin{bmatrix} 1 & 0,3 & 0,09 & 0,027 \\ 0,3 & 1 & 0,3 & 0,09 \\ 0,09 & 0,3 & 1 & 0,3 \\ 0,027 & 0,09 & 0,3 & 1 \end{bmatrix}. \quad (5.12)$$

Über die nach Gl. 2.11 gebildeten korrelierten Kanäle wurden 4000 Pakete bestehend aus $Nn_T = 128$ Informationsbits übertragen. Für die Detektion wurden MMSE-BLE, MMSE-BDFE, RNN mit 10 Iterationen oder SCE mit 5 Iterationen verwendet.

Simulationsergebnisse

Abbildung 5.5 zeigt die Bitfehlerraten in Abhängigkeit vom E_b/N_0 am Sender für BLE und BDFE (5.5a) sowie RNN und SCE (5.5b) bei Übertragung über einen Kanal mit schwachen Korrelationen $\rho_T = 0,3$. Es ist zu erkennen, dass sich die Bitfehlerrate durch Spreizen für alle Entzerrer verbessert. Die Bitfehlerkurven werden steiler, je höher die Diversität des Übertragungsschemas ist. Der Gewinn fällt jedoch unterschiedlich aus. Bei den nicht-iterativen Entzerrern ist MC-CAFS etwas besser als MC-CDM, bei den iterativen fällt der Vorsprung jedoch sehr deutlich aus.

Im Vergleich der nicht-iterativen Entzerrer ist zu erkennen, dass der BDFE dem BLE überlegen ist. Nicht nur die Bitfehlerrate des BDFE für das ungespreizte System ist niedriger als die des BLE, sondern auch der Gewinn durch das Spreizen fällt beim BDFE größer aus als beim BLE. Die Interferenz im gespreizten Ersatzkanal ist größer als im ungespreizten Ersatzkanal, was in den Abbildungen 5.2 und 5.3 erkennbar ist. Möglicherweise verliert der

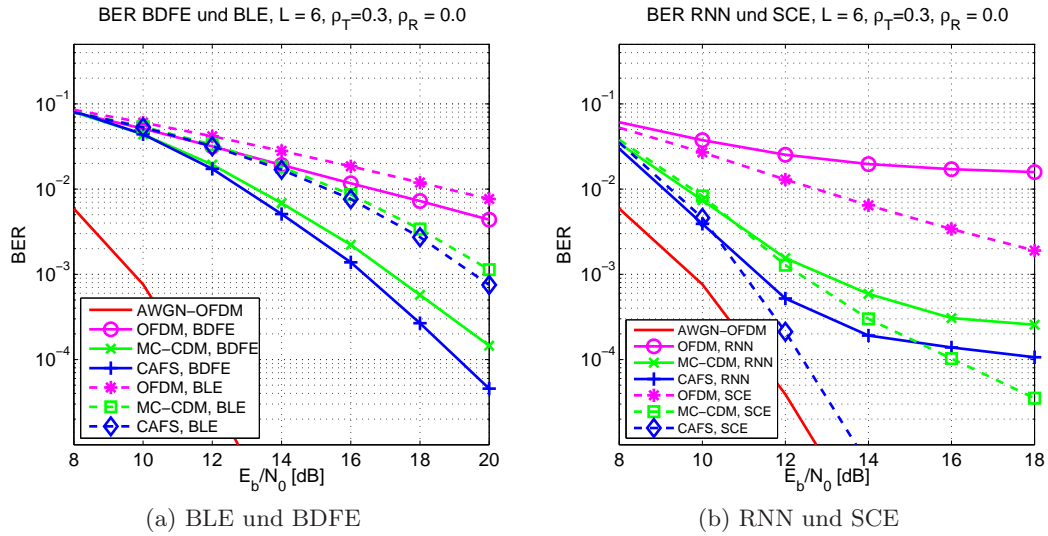


Abbildung 5.5: Bitfehlerrate in Abhängigkeit von E_b/N_0 mit dem Korrelationskoeffizienten $\rho_T = 0.3$ am Sender bei, keine Korrelation am Empfänger, exponentielle Korrelationsmatrix, uncodierte Übertragung für die Entzerrer BLE, BDFE, RNN und SCE.

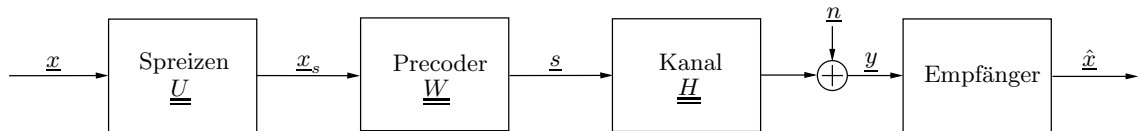


Abbildung 5.6: Precoding im Systemmodell

BLE einen Teil des durch die höhere Diversität bedingten Gewinns durch das stärkere *noise enhancement* in Verbindung mit der höheren Interferenz des gespreizten Kanals, während dieser Verlust beim BDFE kleiner ausfällt.

Auch bei den iterativen Entzerrern RNN und SCE ist zu beobachten, dass die Bitfehlerrate des gespreizten Systems stets besser ist als die des ungespreizten. Während die Performance der beiden Entzerrer bei niedrigem E_b/N_0 annähernd gleich ist, flachen die Bitfehlerkurven des RNN bei steigendem E_b/N_0 deutlich ab und neigen recht schnell zur Bildung eines *error floor*. Dies zeigt, dass der RNN auf Grund der Fehlerfortpflanzung auch bei hohem E_b/N_0 die Interferenzen nicht vollständig entfernen kann. Der SCE hingegen kann den Diversitätsgewinn durch die Spreizschemas voll ausnutzen, da er in der Lage ist, die durch das Spreizen verstärkten Interferenzen zu entfernen.

5.2 Precoding

Lineares Precoding ist ein Verarbeitungsschritt im Sender (siehe Abb. 5.6), der Kanalkennntnis ausnutzt, um den Sendevektor an den Kanal anzupassen [36] und die Übertragung zu optimieren. Dieses Kanalkennntnis kann vollständig oder teilweise sein. In dieser Arbeit wird Precoding bei teilweiser Kanalkennntnis untersucht, die wie in Abschnitt 2.1.4 beschrieben, aus Kenntnis der Antennenkorrelationen am Sender besteht.

5.2.1 Nutzung von Kanalkennntnis am Sender

Es können drei verschiedene Fälle von Kanalkennntnis (CSIT) und daraus resultierende Möglichkeiten zur Anpassung des Signals an den Kanal unterschieden werden. Im ersten Fall liegt keine Kanalkennntnis am Sender vor. Wie in Abschnitt 2.1.3 für die Kanalkapazität besprochen, gibt es ohne Kennntnis des Kanals keine Möglichkeit, das Sendesignal an diesen anzupassen und die Sendeleistung wird gleichmäßig verteilt. Im zweiten Fall verfügt der Sender über volle Kanalkennntnis. Dann können mit Hilfe der Eigenwertzerlegung des Kanals dessen Eigenmoden genutzt werden. Dies wird als *eigen-beamforming* oder modale Zerlegung bezeichnet [20][26]. Über die Eigenmoden des Kanals kann, bedingt durch ihre Eigenschaft als Eigenvektoren, verzerrungsfrei und ohne Interferenz übertragen werden und es kann die maximale Kanalkapazität erreicht werden, wenn die Leistung den einzelnen Moden mit dem in Anhang A beschriebenen Waterpouring-Algorithmus zugeteilt wird [26]. Dieser bewirkt, dass starken Moden mehr und schwachen Moden weniger Leistung zugewiesen wird.

Im dritten Fall, dem Fall der teilweisen Kanalkennntnis, ist die Kanalmatrix am Sender nicht bekannt. Somit kann keine Eigenwertzerlegung des Kanals durchgeführt werden. Dafür verfügt der Sender jedoch über die Korrelationsmatrix \underline{k}_T des Senders. Diese kann nun zur Optimierung des Sendesignals verwendet werden, wie im folgenden Abschnitt beschrieben.

5.2.2 Optimierung der paarweisen Fehlerwahrscheinlichkeit (PEP)

In [28] wurde eine Lösung für die ideale Precoding-Matrix \underline{w} im Sinne von minimaler Bitfehlerwahrscheinlichkeit bei Übertragung mit *Space-Time Block Codes* (STBC) gezeigt. Dabei wurde ein nicht-frequenzselektiver MIMO-Kanal angenommen. Das Optimierungsproblem wurde mit Hilfe der paarweisen Bitfehlerwahrscheinlichkeit (engl. *pairwise error probability*, PEP) zwischen zwei beliebigen Codewörtern beschrieben. Die Fehlermatrix

$$\underline{e} = \underline{x}_1 - \underline{x}_2 \quad (5.13)$$

beschreibt den Fehler, wenn ein Codewort \underline{x}_1 gesendet wurde und sich der Empfänger für Codewort \underline{x}_2 entscheidet. In [28] wird eine obere Schranke für die mittlere paarweise Fehlerwahrscheinlichkeit abgeleitet, die auf die Bedingung für den optimalen Precoder \underline{w}

$$\max_{\underline{w}} J = \det \left(\underline{I} + \frac{1}{\sigma^2} \underline{k}_T^{-\frac{1}{2}} \underline{w} \underline{e} \underline{e}^H \underline{w}^H \underline{k}_T^{\frac{H}{2}} \right) \quad (5.14)$$

mit der Randbedingung für die Sendeleistung

$$\text{Tr}(\underline{w} \underline{w}^H) = p_0 \quad (5.15)$$

führt, wobei p_0 die verfügbare Sendeleistung ist. Nun werden die Eigenwertzerlegung der Fehlermatrix \underline{e}

$$\underline{e} \underline{e}^H = \underline{v} \underline{\varepsilon} \underline{v}^H \quad (5.16)$$

und die Singulärwertzerlegung der Korrelationsmatrix \underline{k}_T des Senders

$$\underline{k}_T = \underline{u} \underline{\varepsilon} \underline{u}^H \quad (5.17)$$

benötigt. Im Falle einer hermiteschen, positiv semidefiniten Matrix, wie es die Korrelationsmatrix \underline{k}_T per Definition ist, ist das Ergebnis der Singulärwertzerlegung und der Eigenwertzerlegung identisch [46], so dass hier ebenso die Eigenwertzerlegung verwendet werden kann.

Die Singulär- bzw. Eigenvektoren in $\underline{\underline{v}}_T$ sind der Größe der Eigenwerte entsprechend aufsteigend sortiert. Mit Hilfe dieser Zerlegungen wird in [28] nun bewiesen, dass die Lösung für das Optimierungsproblem in Gl. 5.14

$$\underline{\underline{w}} = \underline{\underline{v}}_T \underline{\underline{\lambda}}^{\frac{1}{2}} \underline{\underline{v}}_e^H \quad (5.18)$$

lautet. Dabei ist $\underline{\underline{\lambda}}$ eine Diagonalmatrix, die die Leistungsverteilung unter Berücksichtigung von Gl. 5.15 festlegt. Für orthogonale STBC ist $\underline{\underline{e}} \underline{\underline{e}}^H = \beta \underline{\underline{I}}$ mit einem Skalar β und damit ist $\underline{\underline{v}}_e = \underline{\underline{I}}$. In diesem Fall ist reduziert sich die Precoding-Matrix zu

$$\underline{\underline{w}} = \underline{\underline{v}}_T \underline{\underline{\lambda}}^{\frac{1}{2}} \quad (5.19)$$

und der optimale Precoder ist ein statistischer *eigen-beamformer* auf Basis der Korrelationsmatrix $\underline{\underline{k}}_T$ des Senders [28]. In [19] wurde gezeigt, dass das Senden entlang der Eigenvektoren der Korrelationsmatrix auch im Sinne von Maximierung der Kanalkapazität die optimale Lösung ist. Die Sendeleistung wird mit Hilfe des Waterpouring-Algorithmus (siehe Anhang A) auf deren Eigenmoden aufgeteilt. Der Precoder bewirkt eine Dekorrelation des Kanals am Sender. Betrachtet man das Modell des korrelierten Kanals, kann man sich dies folgendermaßen veranschaulichen: Der Einfluss der Korrelationsmatrix $\underline{\underline{k}}_T$ kann verringert werden, wenn man in Richtung ihrer Eigenmoden sendet, da die Matrix die dann übertragenen Vektoren skaliert, aber nicht verzerrt. Somit kann die Übertragung zwar nicht auf den gesamten Kanal optimiert werden, jedoch auf eine Komponente, die eine der wesentlichen negativen Einflussgrößen auf die Übertragung ist (siehe Gl. 2.19).

5.2.3 Precoding für MIMO-OFDM-Übertragungen

Die Lösung für einen optimalen Precoder wurde im vorherigen Abschnitt für Übertragungen mit OSTBC über nicht-frequenzselektive Kanäle vorgestellt und soll nun auf MIMO-OFDM übertragen werden. In [36] wurde gezeigt, dass die Matrix $\underline{\underline{v}}_e$ nicht nur für OSTBC entfallen kann, sondern auch bei Verwendung anderer Übertragungsverfahren wie quasi-orthogonalen STBCs. Bei diesen ergibt sich für $\underline{\underline{v}}_e$ ebenfalls eine skalierte Einheitsmatrix, wenn man die Eigenwertzerlegung der mittleren Fehlermatrix $E \{ \underline{\underline{e}} \underline{\underline{e}}^H \}$ betrachtet. Dies kann auch auf den Fall von MIMO-OFDM erweitert werden, wenn angenommen wird, dass die Kovarianz der gesendeten Symbole

$$\underline{\underline{Q}}_{xx} = \underline{\underline{I}}. \quad (5.20)$$

Unter dieser Annahme sind Precoding und Spreizen unabhängig von einander, solange nur orthogonale Spreizmatrizen verwendet werden. Deshalb wird die Lösung für den optimalen Precoder nicht durch den Spreizcode beeinflusst. Dieser ändert die Kovarianz der gesendeten Symbole nicht.

Wie in Abschnitt 2.2 beschrieben, wird bei Verwendung von OFDM ein breitbandiger, frequenzselektiver Kanal in mehrere schmalbandige Frequenzblöcke zerlegt, die als nicht-frequenzselektiv betrachtet werden können. Somit muss im Falle von OFDM Precoding für mehrere Subträger parallel im Frequenzbereich angewandt werden, wie in [25][36][38][45] gezeigt wurde. Dazu werden *eigen-beamforming* und Leistungszuteilung mit Hilfe des Kronecker-Produkts \otimes für alle Subträger wiederholt, so dass

$$\underline{\underline{V}}_T = \underline{\underline{I}}_N \otimes \underline{\underline{v}}_T \quad \text{und} \quad \underline{\underline{\Lambda}} = \underline{\underline{I}}_N \otimes \underline{\underline{\lambda}}. \quad (5.21)$$

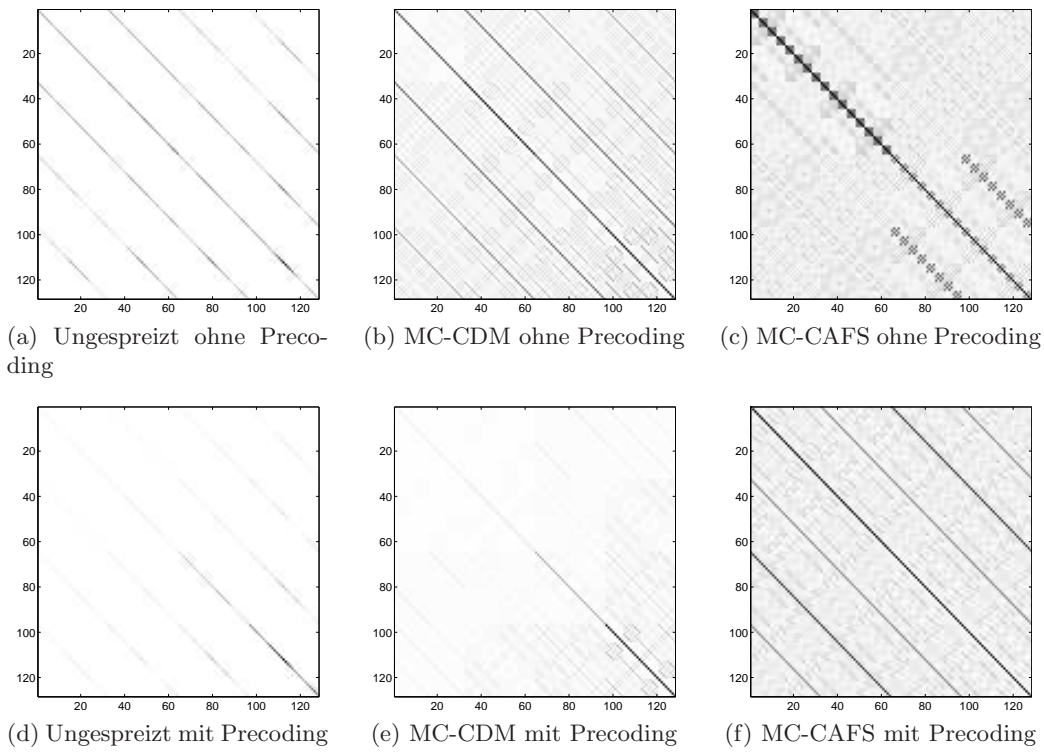


Abbildung 5.7: Ersatzkanalmatrizen \underline{R} mit und ohne Precoding für $n_T = n_R = 4$, $N = 32$, $L = 6$, $\rho_T = 0,7$ und $\rho_R = 0,0$ bei $E_b/N_0 = 12$ dB

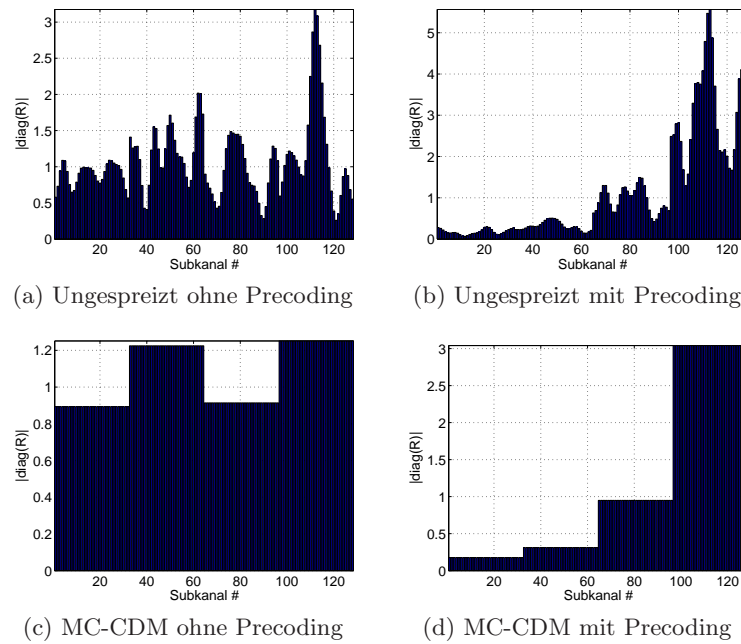


Abbildung 5.8: Beträge der Hauptdiagonalelemente der Ersatzkanalmatrizen \underline{R} aus Abb. 5.7 für ungespreiztes OFDM und MC-CDM mit und ohne Precoding

Die Precoding-Matrix für MIMO-OFDM im Frequenzbereich lautet dann

$$\underline{\underline{W}} = \underline{\underline{V}}_T \underline{\underline{\Lambda}}^{\frac{1}{2}}. \quad (5.22)$$

Das Gesamtsystem mit Spreizen, Precoding und Kanal kann, entsprechend Abb. 5.6, zu einem effektiven Kanal $\underline{\underline{H}}_{\text{eff}}$ zusammengefasst werden, der gegeben ist durch

$$\underline{\underline{H}}_{\text{eff}} = \underline{\underline{H}} \underline{\underline{W}} \underline{\underline{U}} \quad (5.23)$$

Mit dessen Hilfe kann durch Einsetzen von Gl. 5.22 und Gl. 5.23 in Gl. 2.30 ein äquivalenter Ersatzkanal $\underline{\underline{R}}$ gebildet werden:

$$\begin{aligned} \underline{\underline{R}} &= \underline{\underline{U}}^H \underline{\underline{W}}^H \underline{\underline{R}}_{MO} \underline{\underline{W}} \underline{\underline{U}} \\ &= \underline{\underline{U}}^H \underline{\underline{\Lambda}}^{\frac{1}{2}H} \underline{\underline{V}}_T^H \underline{\underline{R}}_{MO} \underline{\underline{V}}_T \underline{\underline{\Lambda}}^{\frac{1}{2}} \underline{\underline{U}} \end{aligned} \quad (5.24)$$

Der Kanal ist normiert, so dass die Leistung der einzelnen Subträger r_{ii} im Mittel $E\{r_{ii}\} = 1$ ist. Für die Ersatzkanalmatrix $\underline{\underline{R}}_{MO}$ bedeutet dies, dass

$$E\left\{\text{Tr}\left(\underline{\underline{R}}_{MO}\right)\right\} = E\left\{\sum_{i=1}^{n_T N} \lambda_i\right\} = n_T N \quad (5.25)$$

ist, wobei λ_i der i -te Eigenwert von $\underline{\underline{R}}_{MO}$ ist.

5.2.4 Einfluss von Precoding auf das Übertragungssystem

Der Einfluss des Precoding auf das Gesamtsystem ist nach Gl. 5.24 in der Ersatzkanalmatrix $\underline{\underline{R}}$ enthalten. Vergleicht man Ersatzkanalmatrizen $\underline{\underline{R}}$ mit und ohne Precoding, so werden einige Unterschiede deutlich. Abbildung 5.7 zeigt Beispiele für drei Matrizen $\underline{\underline{R}}$ ohne Spreizen, mit MC-CDM und mit MC-CAFS, jeweils mit und ohne Precoding. Weiterhin zeigt Abbildung 5.8 die Beträge der Diagonalelemente der ungespreizten bzw. mit MC-CDM gespreizten Kanäle. Vergleicht man Abb. 5.7a und Abb. 5.7d, Abb. 5.7b und Abb. 5.7e sowie Abb. 5.7c und Abb. 5.7f, ist sichtbar, dass Precoding die Interferenzen reduziert. Die Reduzierung ist bei ungespreiztem OFDM und MC-CDM deutlich, bei MC-CAFS hingegen fällt sie im Vergleich sehr gering aus.

Der zweite sichtbare Effekt tritt sowohl bei ungespreiztem OFDM (Abb. 5.7a und 5.7d) als auch bei MC-CDM (Abb. 5.7b und 5.7e) auf. Es ist erkennbar, dass eine Sortierung stattfindet, so dass sich im oberen linken Bereich der Matrix Elemente mit kleinen Beträgen befinden, während im unteren rechten Bereich die Elemente mit größeren Beträgen stehen. Ursache hierfür ist die Multiplikation $\underline{\underline{V}}_T^H \underline{\underline{R}} \underline{\underline{V}}_T$ mit den Eigenvektoren der Korrelationsmatrix. Die Eigenvektoren sind in der Matrix $\underline{\underline{V}}_T$ spaltenweise angeordnet und so sortiert, dass der Eigenvektor zum kleinsten Eigenwert in der ersten, der Eigenvektor zum größten Eigenwert in der letzten Spalte steht. Die Abbildung 5.8 zeigt die Beträge Hauptdiagonalelemente für ungespreiztes OFDM und MC-CDM mit und ohne Precoding im Vergleich. MC-CAFS ist von diesem Effekt nicht betroffen, da hier durch die Spreizmatrix bedingt alle Hauptdiagonalelemente gleich groß sind. Die Verteilung der Hauptdiagonalelemente der Kanalmatrix wiederum hat Einfluss auf die Matched Filter Bound. Die Matched Filter Bound ist eine obere Schranke für die Bitfehlerperformance bei interferenzfreier Übertragung bei perfekter Kanalkennntnis am Empfänger. Sie verschlechtert sich mit steigenden Differenzen zwischen

den Hauptdiagonalelementen. Somit kann Precoding durch die Veränderung der Hauptdiagonalelemente die obere Schranke für die Bitfehlerperformance verschlechtern.

Durch die Leistungszuteilung verändern sich bei allen Spreizverfahren die Eigenwerte, die die Energieverteilung im Kanal widerspiegeln. Moden mit kleinen Eigenwerten erhalten weniger Leistung, Moden mit größeren Eigenwerten mehr. Dadurch werden kleine Eigenwerte kleiner und große Eigenwerte größer. Dies führt zu einer größeren Konditionszahl (engl. *condition number*, CN), die definiert ist als

$$\text{CN} = \frac{\lambda_{\max}}{\lambda_{\min}} \quad \text{mit } \underline{\lambda} = \text{eig}(\underline{R}). \quad (5.26)$$

5.2.5 Simulation

Systemmodell

Für die Simulation des gespreizten Systems mit Precoding wurde das in Abschnitt 5.1.3 beschriebene uncodierte MIMO-OFDM-Übertragungssystem Abb. 5.6 entsprechend um einen Precoding-Block erweitert. Im Unterschied zum gespreizten System wurde hier eine Ersatzkanalmatrix des gespreizten Kanals mit Precoding wie in Gl. 5.24 gebildet. Es wurden $n_T = n_R = 4$ Antennen, $N = 32$ Subträger, $L = 6$ und ein 4PSK-Modulationsalphabet für verschiedene Werte von ρ_T und $\rho_R = 0$ simuliert. Dieses Szenario kann man sich beispielsweise im Uplink einer Mobilfunkverbindung vorstellen, in der am Empfänger genügend Platz ist, um die Empfangsantennen mit großem Abstand aufzustellen, während sich beim mobilen Teilnehmer bedingt durch die Baugröße des Geräts die Antennen eng bei einander befinden und somit korreliert sind. Für die Detektion wurden MMSE-BLE, MMSE-BDFE, RNN mit 10 Iterationen oder SCE mit 5 Iterationen verwendet.

Simulationsergebnisse

Die Abbildung 5.9 zeigt die Simulationsergebnisse für $\rho_T = 0,3$ und $\rho_R = 0,0$ aufgetragen über das E_b/N_0 am Sender für die verschiedenen Entzerrer. Dabei wurden eine ungespreizte Übertragung sowie Übertragungen mit MC-CDM und MC-CAFS jeweils mit und ohne Precoding simuliert. Die Übertragungen ohne Precoding sind identisch mit denen, deren Ergebnisse in Abb. 5.5 dargestellt wurden. Abbildung 5.9a zeigt die Ergebnisse für den BLE. Es ist zu erkennen, dass Precoding bei Verwendung dieses Entzerrers keine Vorteile bringt, sondern sich die Bitfehlerrate in allen Fällen sogar etwas verschlechtert. Ursache hierfür können die höhere *condition number* und die schlechtere Matched Filter Bound sein. Abbildung 5.10b zeigt Die Bitfehlerkurve des BLE für ein fixes $E_b/N_0 = 16$ dB aufgetragen über ρ_T . Es ist zu erkennen, dass die Verschlechterung der Bitfehlerrate für OFDM und MC-CDM mit Precoding mit steigender Korrelation zunimmt, die Kurven für $\rho_T \rightarrow 1$ jedoch wieder zusammenlaufen. Sowohl mit als auch ohne Precoding verschlechtert sich die Bitfehlerrate mit zunehmenden Korrelationen, da die Kanalmatrix singulärer wird.

Die Bitfehlerrate aufgetragen über E_b/N_0 für den BDFE ist in Abbildung 5.9b dargestellt. Es ist zu erkennen, dass Precoding für ungespreiztes OFDM als auch für MC-CDM einen deutlichen Gewinn erzielt. Neben der Reduzierung der Interferenz profitiert der BDFE von der Veränderung der Hauptdiagonalelemente. Wie in Abb. 5.8 gezeigt, bewirkt Precoding, dass die Hauptdiagonalelemente der Kanalmatrix im unteren rechten Bereich groß sind und

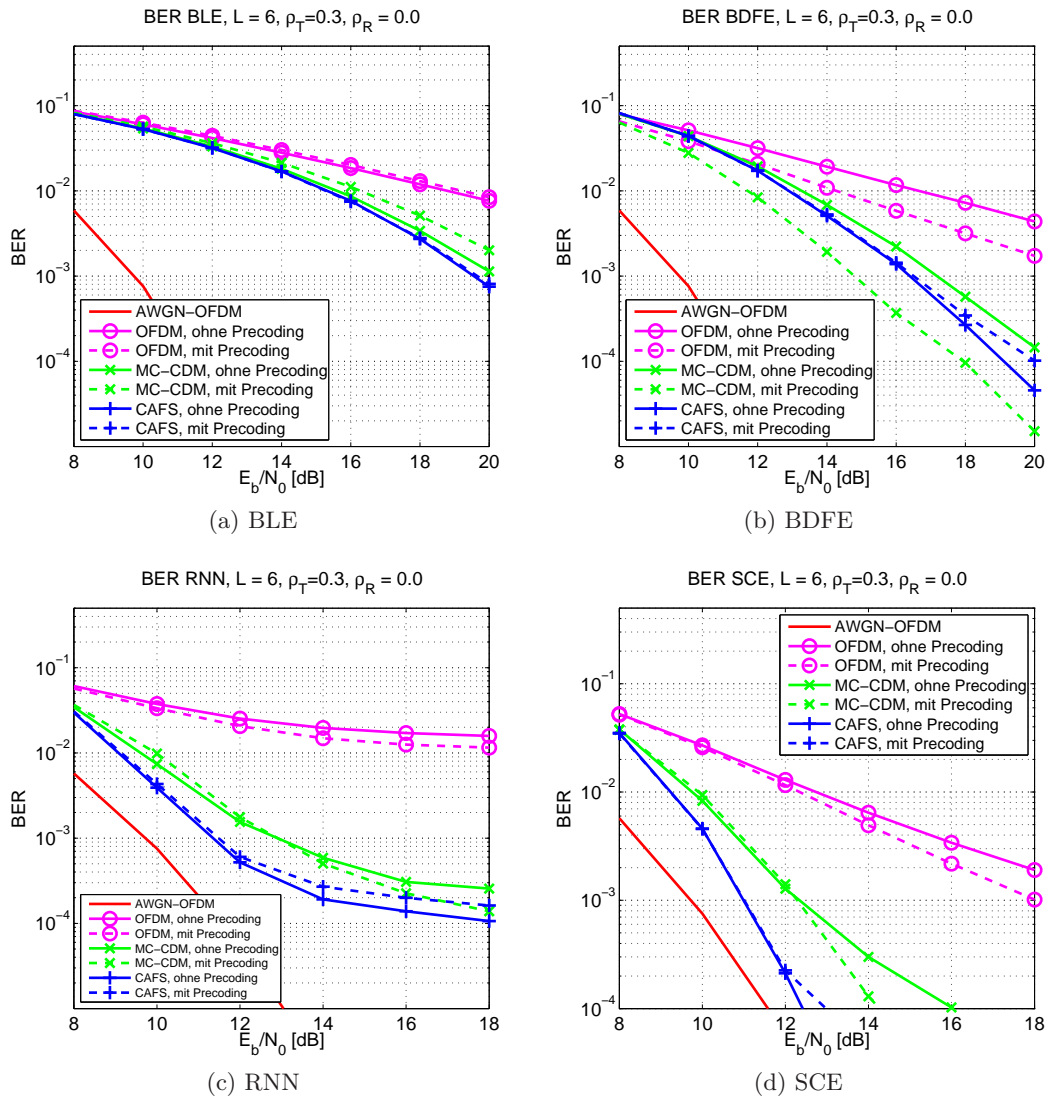


Abbildung 5.9: Bitfehlerrate in Abhängigkeit von E_b/N_0 mit dem Korrelationskoeffizienten $\rho_T = 0.3$ am Sender bei, keine Korrelation am Empfänger, exponentielle Korrelationsmatrix, uncodierte Übertragung für die Entzerrer BLE, BDFE, RNN und SCE.

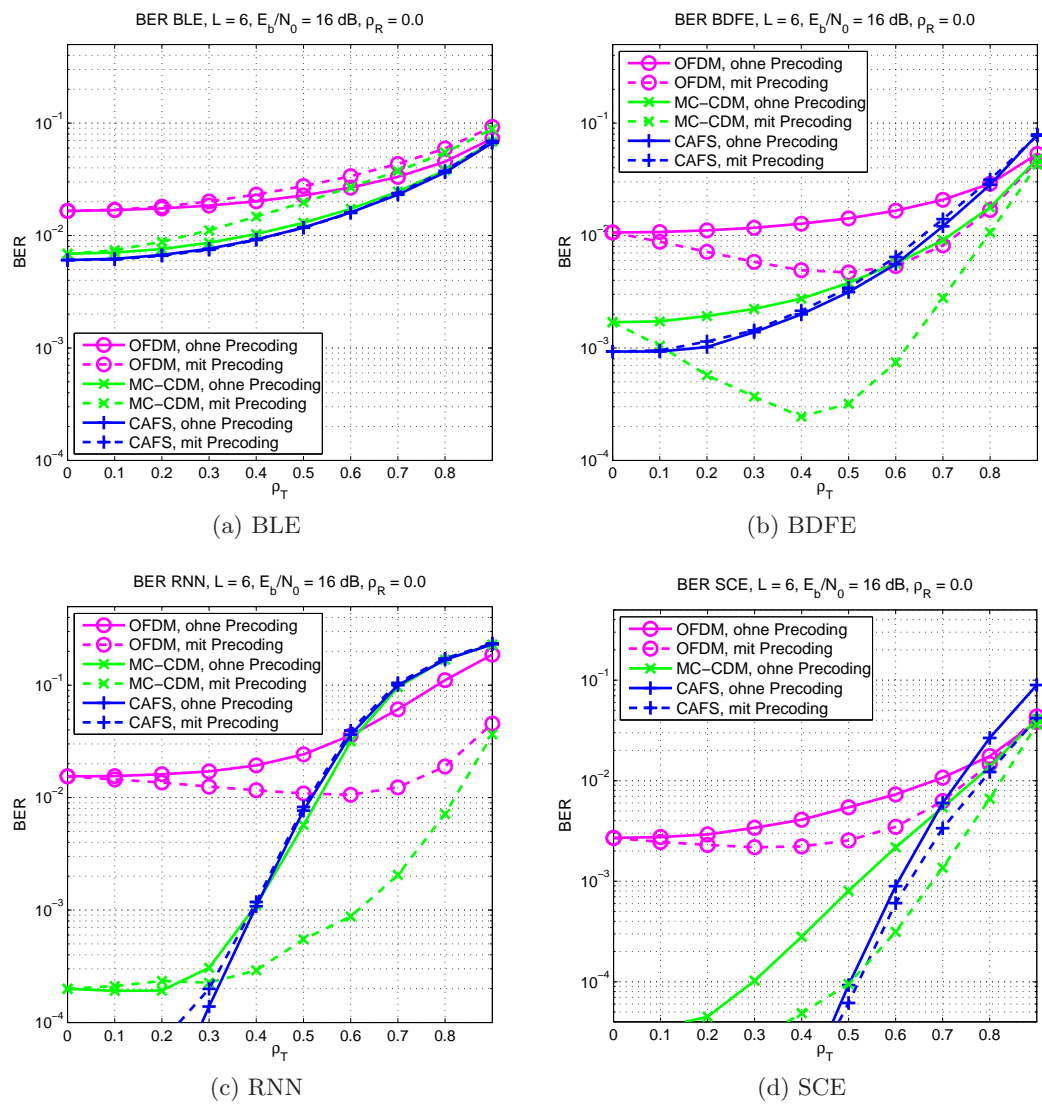


Abbildung 5.10: Bitfehlerrate in Abhängigkeit von den Korrelationen ρ_T am Sender bei $E_b/N_0 = 16$ dB, keine Korrelation am Empfänger, exponentielle Korrelationsmatrix, uncodierte Übertragung für die Entzerrer BLE, BDFE, RNN und SCE.

im oberen linken Bereich klein. Der BDFE beginnt, wie in Abschnitt 3.2 beschrieben, mit der Entzerrung des letzten Hauptdiagonalelementes. Somit werden zuerst die starken Subträger entzerrt und danach die schwächeren. Dies bewirkt eine Reduzierung der Fehlerfortpflanzung, da die Wahrscheinlichkeit einer falschen Entscheidung für die starken Subträger niedriger ist, und verbessert die Entfernung der Interferenz, die durch bereits entschiedene Symbole verursacht wurde. Für MC-CAFS existiert dieser Effekt nicht, da hier keine Veränderung der Hauptdiagonalelemente stattfindet. Dagegen verschlechtert sich mit Precoding die Performance für hohes E_b/N_0 . Abbildung 5.10a zeigt den Verlauf der BER über ρ_T für $E_b/N_0 = 16$ dB. Hier ist zu erkennen, dass der Precoding-Gewinn für ungespreiztes OFDM und MC-CDM bis zu einem Maximum bei ca. $\rho_T = 0,5$ bzw. $\rho_T = 0,4$ ansteigt und danach wieder abfällt. Für MC-CAFS sind die Verläufe mit und ohne Precoding nahezu identisch. Es ist auffällig, dass MC-CAFS für niedrige Korrelationen die beste Bitfehlerrate, für hohe Korrelationen die schlechteste Bitfehlerrate erreicht. Dies hängt vermutlich damit zusammen, dass MC-CAFS einerseits das Spreizverfahren mit der höchsten Diversität ist, jedoch andererseits die höchsten Interferenzen verursacht. Während der Entzerrer geringe bis mittlere Interferenzen kompensieren kann, ist wahrscheinlich mit wachsendem ρ_T die Summe der gesamten, von Spreizen und Antennenkorrelationen verursachten Interferenzen zu groß für eine vollständige Entfernung.

Abbildung 5.9c zeigt die Bitfehlerrate über das E_b/N_0 für den RNN. Auch hier ist eine Verbesserung der Bitfehlerrate für ungespreiztes OFDM und MC-CDM mit Precoding zu erkennen. Ebenso gibt es keinen Precoding-Gewinn für MC-CAFS, wie Abb. 5.10c zeigt, sondern für kleine Korrelationswerte sogar eine Verschlechterung. Der Precoding-Gewinn für ungespreiztes OFDM und MC-CDM ist signifikant von der Stärke der Korrelationen abhängig und nimmt für $\rho_T > 0,3$ deutlich zu. Wie die Simulationen gezeigt haben, reagiert der RNN sehr empfindlich auf Interferenzen. Die Bitfehlerrate erhöht sich für steigende Korrelationen deutlich, insbesondere bei den gespreizten Systemen, die zusätzliche Interferenzen beinhalten. Für ungespreiztes OFDM und insbesondere MC-CDM kann dieser Anstieg mit Precoding deutlich verzögert werden. Die beste Bitfehlerrate für geringe Korrelationen erreicht auch hier MC-CAFS, ebenso wie es für hohe Korrelationen die schlechteste Bitfehlerrate erzielt. Die Bitfehlerkurven für alle Varianten in Abb. 5.9c bilden einen deutlich Fehlerteppich und zeigen, dass der RNN auch für hohes E_b/N_0 die Interferenzen, vermutlich aufgrund der Fehlerfortpflanzung, nicht vollständig entfernen kann.

Der SCE in Abbildung 5.9d zeigt von allen Entzerrern die beste Bitfehlerperformance, verhält sich jedoch ähnlich wie die bereits besprochenen Entzerrer. Man kann erkennen, dass auch in diesem Fall ungespreiztes OFDM und vor allem MC-CDM vom Precoding profitieren. Der Gewinn ist für MC-CDM größer als für ungespreiztes OFDM. Die deutliche Verschlechterung für MC-CAFS mit Precoding für hohes E_b/N_0 ist etwas überraschend. Wie Abbildung 5.10d zeigt, wird für MC-CAFS bei höheren Korrelationen die BER mit Precoding zunehmend besser als ohne Precoding.

Zusammenfassend kann gesagt werden, dass die Verwendung von Precoding nicht mit jedem Entzerrer zu einer Verbesserung der Bitfehlerrate führt, sondern sie im Gegenteil verschlechtern kann. Beim BLE führt Precoding stets zu einem schlechteren Ergebnis. Dieser Entzerrer dürfte jedoch aufgrund seiner im Vergleich zu den anderen Entzerrern schlechten Performance nur dann zum Einsatz kommen, wenn sehr niedrige Komplexität gefordert ist. Bei BDFE, RNN und SCE stellt sich für ungespreiztes OFDM und MC-CDM abhängig

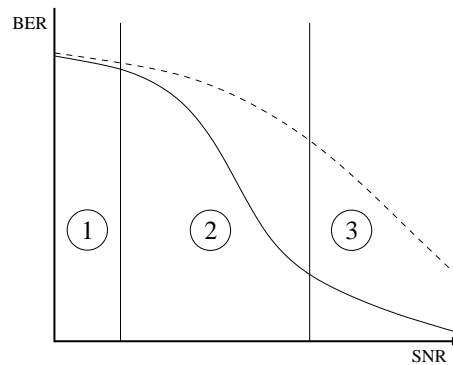


Abbildung 5.11: Turbo-Entzerrer: Die durchgezogene Linie skizziert die Bitfehlerkurve eines Turbo-Entzerrers, die gestrichelte die eines konventionellen Entzerrers. Die Bitfehlerkurven lassen sich in drei Bereiche einteilen: 1. Bereich niedrigen SNRs. 2. Wasserfall-Bereich. 3. Fehlerteppich.

vom Korrelationskoeffizienten ein mitunter deutlicher Precoding-Gewinn ein. Der Precoding-Gewinn sinkt in allen Fällen für sehr hohe Korrelationen und verschwindet bei BDFE und SCE nahezu ganz. In Verbindung mit MC-CAFS ist die Performance mit und ohne Precoding über weite Bereiche identisch. Bei BDFE und RNN ist sie etwas schlechter, beim SCE gewinnt Precoding einen Vorsprung für sehr hohe Korrelationen. Für sehr niedrige Korrelationen liefert MC-CAFS stets die beste Performance, für hohe Korrelationen zeigt MC-CDM sehr gute Ergebnisse. Eine Möglichkeit, die Bitfehlerrate für Precoding zu verbessern, wäre die Veränderung der Leistungszuteilung. Wenn schwächere Moden nicht benutzt werden und die Leistung auf die besseren Moden konzentriert wird, sinkt die Bitfehlerrate. Dies führt jedoch gleichzeitig zu einem Ratenverlust, da über die nicht verwendeten Moden keine Nutzdaten mehr übertragen werden.

5.3 Codierte Übertragung

5.3.1 Analyse des Konvergenzverhaltens mit Hilfe von EXIT-Charts

Die ursprünglich in [35] zur Analyse von verketteten Codes vorgestellten EXIT (engl. *Extrinsic Information Transfer*) Charts können auch zur Untersuchung von Turbo-Entzerrern eingesetzt werden [8][7]. Die Bitfehlerkurve einer codierten Übertragung mit Turbo-Entzerrung kann in drei Bereiche eingeteilt werden, wie in Abb. 5.11 skizziert [35]. Im Bereich von niedrigem SNR bringt das Turbo-Prinzip kaum Vorteile und die Bitfehlerrate liegt nahe bei der eines konventionellen Entzerrers. Im Wasserfall-Bereich (engl. *waterfall region* bzw. *turbo cliff*) hingegen erzielt das Turbo-Prinzip einen deutlichen Gewinn und die Bitfehlerkurve fällt stärker ab als die des konventionellen Entzerrers. Für hohes SNR gelangt man in den Bereich des Fehlerteppichs (engl. *BER floor region*), in dem auch der konventionelle Empfänger gute Performance erreicht und der Gewinn des Turbo-Entzerrers deshalb gering ausfällt. Das EXIT-Chart ist ein Werkzeug zur Untersuchung der *waterfall region*. In den beiden anderen Bereichen liefert es wenig Informationen, da hier teilweise Mechanismen die Bitfehlerperformance dominieren, die das EXIT-Chart nicht abbildet.

Die EXIT-Chart-Analyse zerlegt eine Turbo-Gruppe in ihre Komponenten und betrach-

Der Decodierer erhält als Eingangswerte die Zufallsvariable A_{dec} und liefert am Ausgang die Zufallsvariable E_{dec} und die entschiedenen Informationsbits I , wie in Abb. 5.12 dargestellt. Die meisten Aussagen in diesem Abschnitt gelten sowohl für den Decodierer, als auch für den Entzerrer. Bezieht sich eine Größe nur auf eine Komponente, ist dies durch den Index „dec“ bzw. „eq“ kenntlich gemacht. Trägt die Größe keinen Index, so ist die Aussage für beide Komponenten gültig. Die Zufallsvariable C , die die über den Kanal übertragenen Codewörter abbildet, liegt lediglich am Eingang des Entzerrers an. Bei der Beschreibung dieser Zufallsvariablen helfen Beobachtungen, die [35] durch Simulationen gewonnen hat. Erstens sind die a-priori Werte A und die empfangenen Codewörter C für große Interleavertiefen annähernd unkorreliert. Grund hierfür ist, dass zu einem Zeitpunkt t der Wert von E_t nicht durch A_t und C_t beeinflusst wird, der Interleaver sorgt für eine weitere zeitliche Trennung. Zweitens nähern sich die extrinsischen Ausgangswerte E mit zunehmender Iterationszahl einer Gauß-Verteilung an. Grund hierfür sind zum einen der Gaußsche Kanal und der zentrale Grenzwertsatz zum anderen, da bei der Berechnung der L-Werte viele Summen von Zufallsvariablen gebildet werden, was letztendlich zu einer Normalverteilung führt. Die Ausgangswerte E sind für die nächste Komponente im Turbo-Schema die Eingangswerte A .

Aufgrund dieser Beobachtungen können die Eingangswerte A beschrieben werden als Summe einer normalverteilten, mittelwertfreien Zufallsvariable n_A mit Varianz σ_A^2 und den bekannten, übertragenen Codebits x [35]:

$$A = \mu_A x + n_A \quad (5.30)$$

mit

$$\mu_A = \frac{\sigma_A^2}{2}, \quad (5.31)$$

damit die Verteilung von A der von L-Werten entspricht, die auf einer Normalverteilung beruhen, siehe [35][29]. Die Zufallsvariable X beschreibt die über den Kanal übertragenen Codewörter. Es ist jedoch nötig, sie als modulierte Codebits darzustellen, also ein Mapping

$$\begin{aligned} 0 &\rightarrow 1 \\ 1 &\rightarrow -1 \end{aligned}$$

durchzuführen. Dies ist unabhängig vom Mapping und dem Symbolalphabet auf dem Kanal, da dessen Demapping der Entzerrer bereits vorgenommen hat.

Nun können wir die wechselseitige Information $I_A = I(X; A)$ berechnen, also die wechselseitige Information zwischen den übertragenen Codewörtern und den a-priori L-Werten, die von A näherungsweise beschrieben werden. X ist auf die Werte $\{-1, 1\}$ beschränkt und wir nehmen an, dass

$$p_A(x = -1) = p_A(x = 1) = \frac{1}{2}, \quad (5.32)$$

so dass sich eingesetzt in Gl. 5.29 die wechselseitige Information $I_A = I(X; A)$ berechnen lässt als [29]

$$I_A = \frac{1}{2} \sum_{x \in \{-1, 1\}} \int_{-\infty}^{\infty} p_A(\xi | X = x) \log_2 \left(\frac{p_A(\xi | X = x)}{p_A(\xi)} \right) d\xi. \quad (5.33)$$

Durch Einsetzen von

$$p_A(\xi) = p_A(\xi | X = -1) p_A(X = -1) + p_A(\xi | X = 1) p_A(X = 1) \quad (5.34)$$

und Gl. 5.32 lässt sich Gl. 5.33 umformen zu [35]

$$I_A = \frac{1}{2} \sum_{x \in \{-1,1\}} \int_{-\infty}^{\infty} p_A(\xi|X=x) \log_2 \left(\frac{2 p_A(\xi|X=x)}{p_A(\xi|X=-1) + p_A(\xi|X=1)} \right) d\xi. \quad (5.35)$$

Da für A eine Normalverteilung angenommen wurde, kann die Wahrscheinlichkeitsdichte $p_A(\xi|X=x)$ aus deren Verteilungsfunktion berechnet werden [4]:

$$p_A(\xi|X=x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left(-\frac{(\xi - \mu_A x)^2}{2\sigma_A^2} \right) \quad (5.36)$$

Setzt man Gl. 5.36 und Gl. 5.31 in Gl. 5.35 ein, so erhält man [29]

$$I_A = 1 - \frac{1}{\sqrt{2\pi\sigma^2}} \sum_{x \in \{-1,1\}} \int_{-\infty}^{\infty} \exp \left(-\frac{\left(\xi - \frac{\sigma_A^2}{2} x\right)^2}{2\sigma_A^2} \right) \log_2(1 + \exp(-\xi)) d\xi. \quad (5.37)$$

Wie von [35] vorgeschlagen, definieren wir nun die Funktion

$$J(\sigma) = I_A(\sigma_A = \sigma) \quad (5.38)$$

mit

$$\lim_{\sigma \rightarrow 0^+} J(\sigma) = 0, \quad \lim_{\sigma \rightarrow \infty} J(\sigma) = 1, \quad \sigma > 0 \quad (5.39)$$

Die wechselseitige Information I_E zwischen den Codebits X und den extrinsischen L-Werten L_{ext} wird auf ähnliche Weise berechnet wie die wechselseitige Information I_A zwischen X und den a-priori L-Werten. Analog zu A wird eine Zufallsvariable

$$E = L_{\text{ext}}(y) \quad (5.40)$$

definiert, die die extrinsischen L-Werte am Ausgang des Decodierers modelliert. Im Gegensatz zur Verteilung von A ist uns die Verteilung von E jedoch zunächst nicht bekannt. Die wechselseitige Information $I_E = I(X; E)$ zwischen dem Codewort X und den L-Werten am Ausgang wird analog zu I_A (Gl. 5.35) berechnet als [35]

$$I_E = \frac{1}{2} \sum_{x \in \{-1,1\}} \int_{-\infty}^{\infty} p_E(\xi|X=x) \log_2 \left(\frac{2 p_E(\xi|X=x)}{p_E(\xi|X=-1) + p_E(\xi|X=1)} \right) d\xi. \quad (5.41)$$

Der Ausgang des Decodierers ist abhängig von der Zufallsvariablen A und deren Varianz σ_A^2 , die die L-Werte am Eingang des Decodierers modelliert. Die Funktion $J(\sigma)$ kann nicht in geschlossener Form dargestellt werden, in [6] wurde jedoch gezeigt, dass die Funktion monoton steigend ist. Daher ist sie invertierbar und σ_A kann berechnet werden aus

$$\sigma_A = J^{-1}(I_A). \quad (5.42)$$

Werte für σ_A in Abhängigkeit von I_A lassen sich numerisch berechnen. Da diese Funktion unabhängig ist vom Kanal und dem verwendeten Code, ist es zweckmäßig, einmal eine Wertetabelle mit der benötigten Genauigkeit zu berechnen. Diese kann dann für alle folgenden

I_A	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8	0,9	1,0
σ_A	0,771	1,132	1,443	1,740	2,044	2,369	2,740	3,200	3,878	6,874

Tabelle 5.1: Wertetabelle für Standardabweichung $\sigma_A = f(I_A)$

EXIT-Chart-Berechnungen verwendet werden. Eine solche Tabelle ist in Tab. 5.1 dargestellt.

Durch diesen Zusammenhang wird klar, dass $I_{E,\text{dec}}$ eine Funktion in Abhängigkeit von I_A ist, also

$$I_{E,\text{dec}} = T(I_A). \quad (5.43)$$

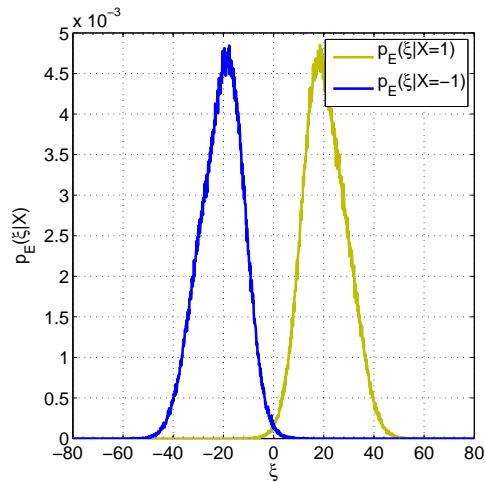
Da in diesem Systemmodell die übertragenen Symbole C nur am Entzerrer verarbeitet werden, nicht jedoch am Decodierer, ist dieser unabhängig vom E_b/N_0 des Kanals. In Systemen, in denen auch der Kanaldecodierer mit Kanalinformationen betrieben wird, besteht für $I_{E,\text{dec}}$ zusätzlich eine Abhängigkeit von E_b/N_0 . Zur Berechnung von $I_{E,\text{dec}}$ wird nun die Wahrscheinlichkeitsdichte $p_E(\xi|X)$ für ein I_A benötigt. Diese erhält man, indem man mittels Monte-Carlo-Simulation Histogramme erzeugt. Dazu wird zunächst die a-priori-Varianz σ_A für I_A berechnet bzw. aus Tabelle 5.1 ausgelesen. Dann wird die Eingangsvariable A_{dec} gemäß Gl. 5.30 erzeugt. Diese Eingangsvariable A_{dec} wird nun als a-priori Information dem Decodierer übergeben, der daraufhin am Ausgang L-Werte liefert, die E_{dec} entsprechen. Da uns das übertragene Codewort X in einer Simulation bekannt ist, können wir die erhaltenen Werte für E_{dec} nun trennen, je nachdem ob -1 oder 1 übertragen wurde, und in zwei separaten Histogrammen für $p_E(\xi|X = -1)$ und $p_E(\xi|X = 1)$ sammeln. Diese Schritte müssen nun wiederholt werden, bis genug Werte für eine aussagekräftige Statistik der Histogramme vorhanden sind. Ein Beispiel für solche Histogramme zeigt Abb. 5.13a, in der die erwartete Normalverteilung für E sichtbar ist.

Mit den so gewonnenen Verteilungsfunktionen $p_E(\xi|X = -1)$ und $p_E(\xi|X = 1)$ kann das Integral in Gl. 5.41 numerisch ausgewertet und I_E hiermit bestimmt werden. Wird dieses Vorgehen für mehrere Punkte $0 \leq I_A \leq 1$ wiederholt, so erhält man einen Verlauf der Funktion $I_{E,\text{dec}} = T(I_A)$. Ein Beispiel hierfür ist in Abb. 5.13b dargestellt.

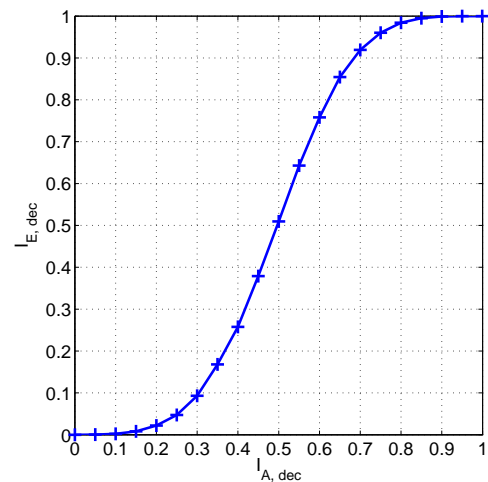
Untersuchung des Entzerrers

Die Untersuchung des Entzerrers verläuft im Wesentlichen entsprechend der des Decodierers. Der wesentliche Unterschied ist in Abb. 5.12 zu erkennen. Der Entzerrer hat im Gegensatz zum Decodierer zwei Eingangsgrößen, die übertragenen Symbole C und die a-priori L-Werte A_{eq} . Die Symbole C werden durch eine reguläre Übertragungssimulation erzeugt, analog einer Bitfehlerkurvenmessung. Dabei können beliebige Operationen wie Spreizcodes, Precoding und Kanalkorrelationen zum Einsatz kommen.

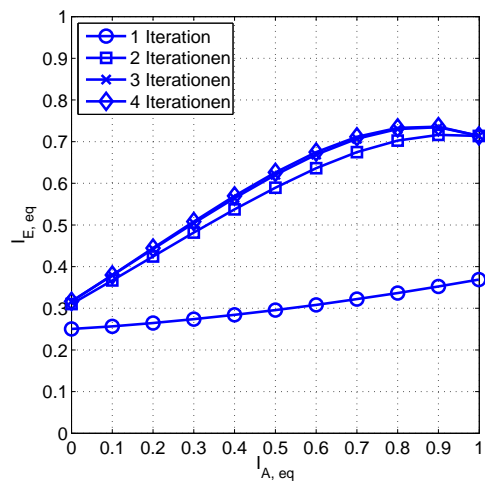
Sowohl der RNN-, als auch der SCE-Entzerrer arbeiten iterativ und verfügen über ein Gedächtnis, wie in Abschnitt 3.3 bzw. 3.4 erläutert. Daraus resultiert ein Problem, dass in [7] ausführlich behandelt wird. Die Gleichungen 3.25, 4.23 und 4.24 zeigen, dass die Symbole $\tilde{x}^{[\eta-1]}$ sowohl von den a-priori-Informationen, als auch von den teilweise entzerrten Symbolen $\tilde{x}^{[\eta-1]}$ abhängig sind. Daher kann die extrinsische Information des Decodierers der aktuellen Iteration nicht von der vorherigen Iterationen und den Schätzungen des Entzerrers getrennt werden. Die Symbole $\tilde{x}^{[\eta-1]}$ wiederum sind abhängig von den Iterationen $\eta - 1$ und $\eta - 2$. Dieses Gedächtnis muss berücksichtigt werden, wenn das Verhalten des Entzerrers untersucht



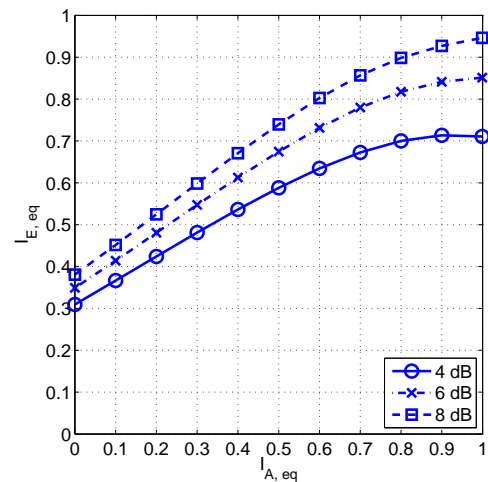
(a) Histogramm von $p_E(\xi|X=1)$ und $p_E(\xi|X=-1)$ für einen Code mit Rate $r = 1/2$, Gedächtnislänge $m = 2$ und Generatorpolynomen $[7, 5]_8$ an der Stelle $I_A = 0,8$



(b) Verlauf von $I_{E,dec} = T(I_{A,dec})$ für einen Code mit Rate $r = 1/2$, Gedächtnislänge $m = 2$ und Generatorpolynomen $[7, 5]_8$



(c) Verlauf von $I_{E,dec} = T(I_{A,dec})$ in Abhängigkeit von der Anzahl der Iterationen für einen RNN-Entzerrer, MC-CAFS, ohne Precoding bei $E_b/N_0 = 4\text{dB}$



(d) Verlauf von $I_{E,eq} = T(I_{A,eq})|_{E_b/N_0}$ für einen RNN-Entzerrer, MC-CAFS, ohne Precoding

Abbildung 5.13: Die aus den Histogrammen berechneten Verläufe werden in ein Koordinatensystem eingetragen.

wird.

Um diesem Effekt Rechnung zu tragen, werden die Entzerrer während der Monte-Carlo-Simulation zur Erzeugung der Histogramme nicht nur einmal, sondern mehrmals durchlaufen, während die a-priori L-Werte konstant gehalten werden [8]. Zu Beginn der ersten Iteration hat der Feedback-Vektor $\underline{\check{x}}$ den Initialisierungswert Null, so dass die a-priori-Informationen erst ab dem zweiten Symbol verwendet werden. Abbildung 5.13c zeigt den Verlauf $I_{E,\text{eq}} = T(I_A) |_{E_b/N_0=4\text{dB}}$ für einen RNN-Entzerrer in Abhängigkeit von der Anzahl von Iterationen, die der RNN durchlaufen hat. Es ist zu erkennen, dass eine größere Zahl an Iterationen als zwei zu keiner wesentlichen Änderung mehr führt. Deshalb wurden die EXIT-Charts im Folgenden stets mit zwei Iterationen berechnet. Da dieses Vorgehen nicht das wirkliche Verhalten des Entzerrers im Turbo-Schema erfasst, beschreibt diese Untersuchung den Entzerrer nur näherungsweise. Während bei einer echten Decodierung sich die a-priori-Information fortlaufend ändert, wird hier das Gedächtnis des Entzerrers mit konstanter a-priori-Information nachgebildet.

Die Berechnung von I_A für den Entzerrer unterscheidet sich nicht von der für den Decodierer. Für die Zufallsvariable A_{eq} , die E_{dec} nach Re-Interleaving und Re-Puncturing entspricht, kann entsprechend den bereits erwähnten Beobachtungen von [35] ebenfalls eine Normalverteilung angenommen werden. Demnach gilt auch hier Gl. 5.37 und es kann die selbe Wertetabelle für $\sigma_A = J^{-1}(I_A)$ verwendet werden wie zuvor. Die Verteilung der empfangenen Symbole C ist jedoch abhängig vom verwendeten Kanal. Unter der Annahme, dass der Ersatzkanal \underline{R} und damit Spreizcode, Precoding und der Kanal selbst während einer Simulationsreihe nicht verändert werden, ist die Verteilung von C nur abhängig von E_b/N_0 . Der Entzerrer berechnet aus den empfangenen Symbolen C und den a-priori L-Werten A_{eq} extrinsische L-Werte E_{eq} . Die Berechnung von $I_{E,\text{eq}} = I(X; E_{\text{eq}})$ erfolgt nach Gl. 5.41 und ist im Unterschied zu Gl. 5.43 eine Funktion in Abhängigkeit von I_A und E_b/N_0 , also

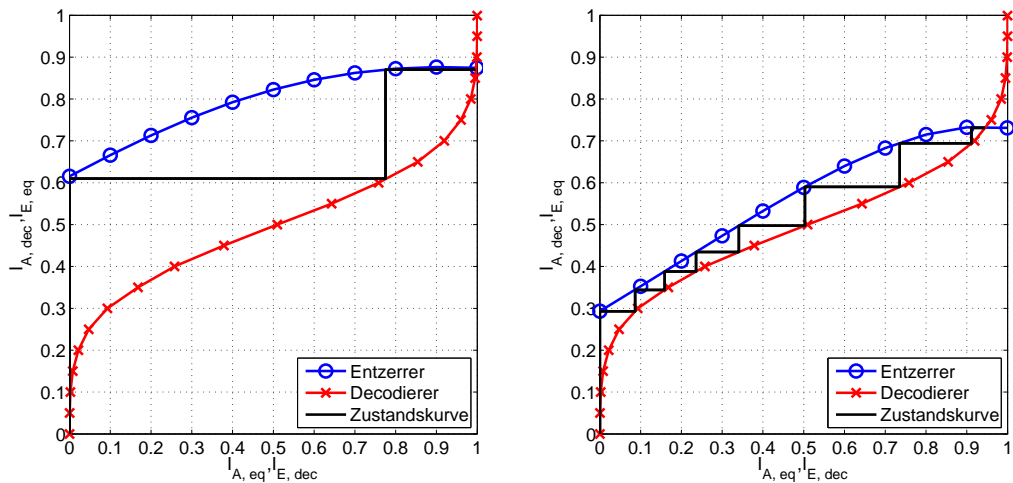
$$I_{E,\text{eq}} = T(I_A, E_b/N_0). \quad (5.44)$$

Ein solcher Verlauf ist beispielhaft für einen RNN-Entzerrer mit festem Kanal, jedoch mit verschiedenen Werten für E_b/N_0 in Abb. 5.13d dargestellt.

Graphische Darstellung des Verlaufs der extrinsischen Information

Mit den berechneten Verläufen $I_{E,\text{eq}} = T(I_A, E_b/N_0)$ und $I_{E,\text{dec}} = T(I_A)$ lässt sich der Verlauf des Turbo-Prozesses graphisch darstellen. Während in Abb. 5.13b und 5.13d die Kurven für beide Komponenten separat gezeichnet wurden, wird nun zu einer gemeinsamen Darstellung übergegangen, die zum Verständnis des gesamten Prozesses beiträgt. Dazu wird zunächst der Verlauf des Entzerrers, also $I_{E,\text{eq}} = T(I_{A,\text{eq}})$, in ein Koordinatensystem eingetragen. Der extrinsische Ausgang der einen Komponente ist gleich dem a-priori-Eingang der anderen Komponente, also $I_{E,\text{dec}} = I_{A,\text{eq}}$ und $I_{E,\text{eq}} = I_{A,\text{dec}}$, so dass auf der x-Achse $I_{A,\text{eq}}$ und $I_{E,\text{dec}}$ aufgetragen werden, auf der y-Achse $I_{E,\text{eq}}$ und $I_{A,\text{dec}}$. Die Funktion $I_{E,\text{dec}} = T(I_{A,\text{dec}})$ wird folglich mit vertauschter x- und y-Achse gezeichnet. Das resultierende Diagramm, das nun beide Komponenten des Turbo-Schemas enthält und den Ausgang der einen als den Eingang der jeweils anderen Komponente darstellt, wird als EXIT-Chart bezeichnet.

Den Verlauf des Decodierprozesses kann man sich im EXIT-Chart mit Hilfe einer Zustandskurve folgendermaßen vorstellen: Zu Beginn der Decodierung eines Blocks liegen kei-



(a) Weit geöffnetes EXIT-Chart: Der Schnittpunkt der Kurven wird innerhalb von vier Iterationsschritten erreicht.

(b) Wenig geöffnetes EXIT-Chart: Obwohl der Decodierprozess konvergiert, sind 16 Iterationsschritte nötig.

Abbildung 5.14: EXIT-Chart-Darstellung mit idealer Zustandskurve: Beispiele für ein weit geöffnetes EXIT-Chart mit schnellem und ein wenig geöffnetes mit langsamem Konvergenzverhalten

nerlei a-priori-Informationen vor. Der Startpunkt ist deshalb der Punkt $(0,0)$. Die erste Bearbeitung eines Blocks von empfangenen Symbolen C findet im Entzerrer statt. Dieser liefert als Ergebnis L-Werte mit $I_{E,eq}$, die Zustandskurve bewegt sich vertikal aufwärts zum Punkt $(0, I_{E,eq})$. Diese L-Werte mit $I_{E,eq}$ dienen nun dem Decodierer als Eingangswerte $I_{A,dec}$ und er berechnet daraus $I_{E,dec}$. Somit bewegt sich die Zustandskurve horizontal seitwärts und erreicht den Punkt $(I_{E,dec}, I_{E,eq})$. Damit ist die erste Iteration abgeschlossen und die Ausgangs-L-Werte des Decodierers dienen wiederum dem Entzerrer als Eingangswerte für die nächste Iteration. Diese Schritte wiederholen sich nun, bis die festgelegte Zahl von Iterationen pro Codewort-Block erreicht ist. Dabei steigt die Zustandskurve weiter treppenförmig an. Zwei Beispiele für Verläufe der Zustandskurve sind in Abb. 5.14 dargestellt. Nach der letzten Iteration entscheidet der Decodierer hart und liefert an seinem zweiten Ausgang die entschiedenen Symbole I . Somit ist für die Bitfehlerperformance ausschlaggebend, welche extrinsische Information $I_{E,dec}$ beim letzten Iterationsschritt erzielt wird, das entspricht dem x-Wert, den die Zustandskurve erreicht. Wird $I_{E,dec} = 1$ erreicht, sinkt die Bitfehlerleistung auf diejenige einer Übertragung über einen reinen AWGN-Kanal. Danach beginnt der vollständige Turbo-Prozess erneut für den nächsten Block von C .

Interpretation der EXIT-Chart-Darstellung

Aus der Beschreibung der Konstruktion der treppenförmigen Zustandskurve wurde bereits eine wichtige Regel deutlich: Die Zustandskurve bewegt sich zwischen den beiden Kurven für Entzerrer und Decodierer. Der Decodierprozess benötigt somit einen offenen Tunnel im EXIT-Chart. Schneiden sich die Kurven des Entzerrers und des Decodierers, kann der Decodierprozess nicht über diesen Punkt hinaus fortschreiten und bleibt stecken. Die Öffnung des EXIT-Charts ist zum Beispiel vom Kanal und dessen E_b/N_0 abhängig. Abbildung 5.13d zeigt, dass das E_b/N_0 die Entzerrer-Kurve nach oben bzw. unten verschieben kann. Bei zu geringem

E_b/N_0 , wenn die Entzerrer-Kurve weit unten liegt im EXIT-Chart, schneiden sich Entzerrer- und Decodiererkurve bereits bei einem kleinen Wert von $I_{E,dec}$ und der Turbo-Entzerrer erreicht eine schlechte Bitfehlerperformance. Weiteren Einfluss haben Kanalkorrelationen, die für den Unterschied zwischen den Abbildungen 5.14a und 5.14b verantwortlich sind. Von der Größe der Öffnung des EXIT-Charts ist die Anzahl der Iterationsschritte abhängig, die bis zur Konvergenz des Turbo-Entzerrers nötig sind. Wird dieser mit einer festen Anzahl von Iterationsschritten betrieben, kann das EXIT-Chart kann somit hilfreich bei der Beantwortung der Frage nach der idealen Anzahl sein. Werden bei einer Übertragung zu wenige Iterationen durchgeführt, so wird der mögliche Turbo-Vorteil nicht voll ausgenutzt. Werden hingegen zu viele Iterationen durchgeführt, wird mehr Rechenzeit als nötig aufgewendet.

Die EXIT-Chart-Darstellung beschreibt, wie bereits erläutert, das Verhalten eines Turbo-Schemas nur näherungsweise. Die auf diesem Wege konstruierte Zustandskurve entspricht nicht dem echten Decodierprozess, sondern stellt eine Vorhersage dar, die auf einigen Vereinfachungen basiert. Sie ist eine Abschätzung für den günstigsten Fall und wird oft von der echten Zustandskurve nicht erreicht. Das bedeutet, dass die echte Zustandskurve nicht immer vollständig zwischen Decodierer- und Entzerrerkurve pendelt, sondern sowohl die einzelnen Schritte als auch der gesamte Decodierprozess früher zum Stehen kommen. Bei der Interpretation der EXIT-Charts muss man sich dessen bewusst sein. Deshalb werden im folgenden Abschnitt die Vereinfachungen und Ungenauigkeiten dieser Darstellung nochmals verdeutlicht.

Für die empfangenen Codewörter C und die L-Werte A wurde eine Normalverteilung angenommen. Auf die Verteilung von C hat neben dem gaußförmigen Rauschen jedoch auch die Charakteristik des Kanals Einfluss, die im Allgemeinen einer anderen Verteilung entsprechen kann. Wie die Beobachtung zeigt, stellt sich bei einem realen Decodierprozess für die L-Werte sehr schnell eine Normalverteilung ein. Allerdings wachsen die Korrelationen zwischen den L-Werten entgegen der Annahme mit steigender Zahl der Iterationen, was aber durch große Interleavertiefen verzögert werden kann. Eine wesentliche Fehlerquelle ist, dass die Wahrscheinlichkeitsdichtefunktionen p_E durch Monte-Carlo-Simulationen mit Hilfe von Histogrammen gewonnen werden. Hier müssen auf jeden Fall genug Simulationsschritte für eine aussagekräftige Statistik durchgeführt werden. Bei der Untersuchung des Entzerrers muss für RNN und SCE ein Gedächtnis berücksichtigt werden. Deshalb wird der Entzerrer mehrmals durchlaufen, während die a-priori-Informationen konstant gehalten werden, was das wirklich Verhalten aber nur näherungsweise beschreibt. Dazu wurde der Entzerrer stets zweimal durchlaufen, was bereits einer guten Näherung entspricht, da das Verhalten schnell konvergiert, wie Abb. 5.13c zeigt, auch wenn erst für eine unendliche Anzahl von Iterationen die obere Schranke für die extrinsische Information des RNN erreicht wird [7]. Die in dieser Arbeit abgebildeten EXIT-Charts zeigen also keine obere Schranke für die Decodierperformance des Turbo-Schemas selbst, sondern eine Näherung davon.

Der wirkliche Verlauf des Decodierprozesses kann auch im EXIT-Chart dargestellt werden. Dazu werden bei einer regulären Bitfehlersimulation sämtliche anfallenden L-Werte aufgezeichnet werden. Nach Ende der Simulation können diese Werte genutzt werden, um die Kurven für Entzerrer und Decodierer zu berechnen und daraus den treppenförmigen Zustandsverlauf zu rekonstruieren. Der Aufwand hierfür ist deutlich höher als beim oben beschriebenen Verfahren, da bei einer Histogramm-Simulation ungefähr Faktor 10 weniger Symbole übertragen werden müssen, um bereits aussagekräftige Ergebnisse zu erhalten. Au-

ßerdem ist die anfallende Datenmenge der L-Werte sehr groß.

Trotz all dieser Nachteile und Ungenauigkeiten kann die EXIT-Chart-Methode ein sehr hilfreiches Werkzeug sein, um einen Einblick in den Decodierprozess zu erhalten. Insbesondere wenn bei einer Übertragung nicht die gewünschte Bitfehlerperformance erreicht wird und zur Optimierung kann das EXIT-Chart wertvolle Hinweise auf Ursachen und Ansatzpunkte liefern.

5.3.2 Systemmodell

Für die Simulation der codierten Übertragung wurde das in den Abschnitten 5.1.3 und 5.2.5 vorgestellte System mit $n_T = n_R = 4$ Antennen, $N = 32$ Subträgern, einem 4PSK-Modulationsalphabet und $L = 4$ auf Senderseite um einen Faltungscodierer (siehe Abschnitt 4.1) erweitert. Im Empfänger wurde ein Turbo-Entzerrer eingesetzt, bestehend aus einem BCJR-Decodierer und einem RNN bzw. SCE mit den in den Abschnitten 4.4.1 und 4.4.2 vorgestellten Modifikationen. Mit dem RNN wurden pro Block 10 Iterationen durchgeführt, mit dem SCE 5 Iterationen. Für die Bitfehlersimulationen wurden 500 Blöcke bestehend aus 10000 Informationsbits übertragen. Als Kanalcode kam, soweit nichts anderes explizit erwähnt wird, ein Code mit der Rate $r = 1/2$, Gedächtnislänge $m = 2$ und den Generatorpolynomen $[7, 5]_8$ zum Einsatz. Die Angabe des E_b/N_0 bezieht sich stets auf den Sender. Bei den Bitfehlerkurven wurde der Ratenverlust in Höhe von $l_r = 10 \log_{10}(r)$ sowie ein Verlust durch das Cyclic Prefix von

$$l_{cp} = 10 \log_{10} \frac{N + L - 1}{N} \quad (5.45)$$

berücksichtigt. Bei den EXIT-Charts wurden diese Korrekturen nicht berücksichtigt, da die Ergebnisse von Entzerrer und Kanalcode separat berechnet und verwendet werden können. Der Ratenverlust abhängig ist vom verwendeten Code, die Ergebnisse für den Code sind jedoch nicht abhängig von E_b/N_0 . Der Kurvenverlauf für den Entzerrer ist abhängig vom E_b/N_0 des Kanals, der Ratenverlust ist aber keine Eigenschaft des Entzerrers. Deshalb ist beim Vergleich der EXIT-Charts und Bitfehlerkurven zu berücksichtigen, dass bei den EXIT-Charts das reine E_b/N_0 des Kanals angegeben ist, während bei den Bitfehlerkurven das korrigierte E_b/N_0 aufgetragen ist. Beispielsweise entspricht ein EXIT-Chart für 4 dB dem Punkt 7,39 dB in der Bitfehlerkurve für $r = 1/2$ und $L = 4$.

5.3.3 Testkanäle

Für die Simulation der Histogramme zur Berechnung der EXIT-Charts sollte der Kanal konstant gehalten werden, da sonst die Statistik über Kanäle mit potentiell unterschiedlichem Verhalten gemittelt wird. Das Verhalten eines Kanals wird nicht nur durch Energie und deren Verteilung bestimmt, sondern auch durch die Interferenzen und deren Verteilung. So können zwei Kanäle, deren Gesamtenergie, also die Summe der Hauptdiagonalelemente der Ersatzkanalmatrix, identisch ist, zu sehr unterschiedlichen Bitfehlerraten führen. Um die Vergleichbarkeit von EXIT-Charts und Bitfehlerkurven sicherzustellen, wurde deshalb ein Satz von Testkanälen gewählt, der sowohl für die Simulation der Bitfehlerkurven, als auch für die Simulation der dazugehörigen EXIT-Charts verwendet wurde. Die Testkanäle wurden zunächst nach Korrelationskoeffizienten unterteilt. So wurden Kanäle mit $[\rho_T = 0,3, \rho_R = 0,3]$, $[\rho_T = 0,7, \rho_R = 0,7]$ und $[\rho_T = 0,7, \rho_R = 0,0]$ gesucht. Für jedes dieser

Wertepaare wurde nun eine große Zahl zufälliger Kanäle erzeugt und deren statistische Eigenschaften untersucht, um repräsentative Kanäle auswählen zu können. Das Ergebnis der statistischen Untersuchung ist in Tabelle 5.2 dargestellt. Dabei wurde als erste Eigenschaft die Konditionszahl (engl. *condition number*, CN) betrachtet, die definiert ist als

$$\text{CN} = \frac{\lambda_{\max}}{\lambda_{\min}} \quad \text{mit } \underline{\lambda} = \text{eig}(\underline{R}). \quad (5.46)$$

Für Eigenwerte nahe Null kann dieser Quotient sehr große Werte annehmen, was die Statistik verzerren kann. Deshalb wurde zusätzlich die Statistik für die Konditionszahlen berechnet, ohne die jeweils kleinsten und größten 10% bzw. 20% Werte zu berücksichtigen, was die Varianz deutlich verringert. Außerdem wurde als Maß für die gesamte Interferenz des Kanals das in [42] vorgestellte

$$\beta_{1/2} = \frac{1}{n_{\text{T}}N} \sum_{i \neq j} |r_{ij}| \geq 0 \quad (5.47)$$

betrachtet, das die Beträge der Nebendiagonalelemente der Kanalmatrix aufsummiert und normiert. Weiterhin wurde die Frobenius-Norm der Differenz der Kanalmatrix und der Einheitsmatrix

$$F_{\text{I}} = \|\underline{R} - \underline{I}\|_{\text{F}} \quad (5.48)$$

sowie der Differenz der Kanalmatrix und einer Matrix mit allen Elementen gleich Eins

$$F_{\text{1}} = \|\underline{R} - \underline{1}\|_{\text{F}} \quad (5.49)$$

betrachtet. Dies ist ein Maß für die Ähnlichkeit der Kanalmatrix zu einem perfekt unkorrelierten Kanal, dargestellt durch die Einheitsmatrix, und einem vollständig korrelierten Kanal, dargestellt durch die Eins-Matrix. Ist die Differenz gleich der Null-Matrix, so ist auch die Frobenius-Norm gleich Null.

Auf diesen Größen basierend wurden nun charakteristische Testkanäle für jedes Wertepaar ausgewählt und mit A,B,C und D, gefolgt vom Korrelationswertepaar, bezeichnet. Die mit A bezeichneten Kanäle weisen eine niedrige Konditionszahl auf, die sich am Minimum von CN ohne äußere 10% orientiert. Dabei entsprechen die übrigen Parameter ungefähr dem Mittelwert. Die mit B bezeichneten Kanäle sind durch eine hohe Konditionszahl charakterisiert, die ungefähr dem Maximum von CN ohne äußere 10% entspricht. Die Konditionszahl der mit C und D beginnenden Kanäle entspricht circa dem Mittelwert von CN ohne äußere 20%. Dafür weisen die C-Kanäle eine niedrige Interferenz von $\beta_{1/2} \approx \mu - 2\sigma$ auf, die D-Kanäle eine hohe Interferenz von $\beta_{1/2} \approx \mu + 2\sigma$. Die Parameter F_{I} und F_{1} sollen auch hier im Mittel liegen. Einen Überblick über die ausgewählten Testkanäle gibt Tabelle 5.3.

5.3.4 Simulationsergebnisse

Simulation des Kanaldecodierers

Das Verhalten des Kanaldecodierers ist, wie in Abschnitt 5.3.1 beschrieben, unabhängig vom Kanal und E_b/N_0 . Wird der Decodierer selbst nicht verändert, ist der Verlauf der wechselseitigen Information nur vom verwendeten Code abhängig. Dazu wurden verschiedene Codes, die sich in Rate, Gedächtnislänge und Generatorpolynomen unterscheiden, untersucht. Entsprechend der in Abschnitt 5.3.1 beschriebenen Vorgehensweise wurden 100 Blöcke der Länge 10000 Bits erzeugt und decodiert, aus den Ergebnissen wurden die Histogramme gewonnen.

Kanäle mit $\rho_T = 0,3$ und $\rho_R = 0,3$					
Eigenschaft	min	max	μ	σ	Median
Condition Number	63	148449460	27829	787893	3513
CN ohne äußere 10%	836	25317	5536	5220	3513
CN ohne äußere 20%	1318	11783	4363	2675	3513
$\beta_{1/2}$	0,91	2,05	1,45	0,13	1,45
F_1	9,0	18,6	12,5	1,1	12,4
F_1	126,6	128,8	127,6	0,26	127,6
Kanäle mit $\rho_T = 0,7$ und $\rho_R = 0,7$					
Eigenschaft	min	max	μ	σ	Median
Condition Number	352	850025370	229815	6106479	24068
CN ohne äußere 10%	5084	182155	38240	37149	24069
CN ohne äußere 20%	8318	82061	29838	18796	24069
$\beta_{1/2}$	1,23	2,74	2,14	0,21	2,15
F_1	11,8	28,6	18,2	2,0	18,1
F_1	126,4	129,0	127,2	0,29	127,1
Kanäle mit $\rho_T = 0,7$ und $\rho_R = 0,0$					
Eigenschaft	min	max	μ	σ	Median
Condition Number	187	1193637679	83141	5453708	8130
CN ohne äußere 10%	1857	57900	12740	12013	8131
CN ohne äußere 20%	2962	27227	10052	6200	8131
$\beta_{1/2}$	1,20	2,55	1,96	0,18	1,96
F_1	10,6	23,7	15,5	1,36	15,4
F_1	126,2	127,8	126,8	0,18	126,7

Tabelle 5.2: Statistische Übersicht über charakteristische Eigenschaften von MIMO-OFDM-Kanälen, Stichprobe von jeweils $n = 10000$ Kanälen

Name	ρ_T	ρ_R	Condition Number	$\beta_{1/2}$	$\max\{ r_{ij} \}$	F_1	F_1
A0303	0,3	0,3	800	1,47	1,04	11,6	128,0
B0303	0,3	0,3	25016	1,43	1,12	12,5	127,3
C0303	0,3	0,3	4302	1,19	1,17	11,6	127,7
D0303	0,3	0,3	4302	1,71	1,72	15,0	127,6
A0707	0,7	0,7	5000	2,12	2,26	17,4	127,1
B0707	0,7	0,7	180191	2,18	2,64	20,0	127,5
C0707	0,7	0,7	29738	1,72	2,15	15,2	127,4
D0707	0,7	0,7	29740	2,57	3,50	25,2	127,6
A0700	0,7	0,0	2000	1,97	1,96	16,6	126,7
B0700	0,7	0,0	60495	1,95	1,22	14,5	126,5
C0700	0,7	0,0	10006	1,59	1,63	13,0	127,0
D0700	0,7	0,0	10007	2,32	1,98	17,8	126,5

Tabelle 5.3: Übersicht über die verwendeten Testkanäle

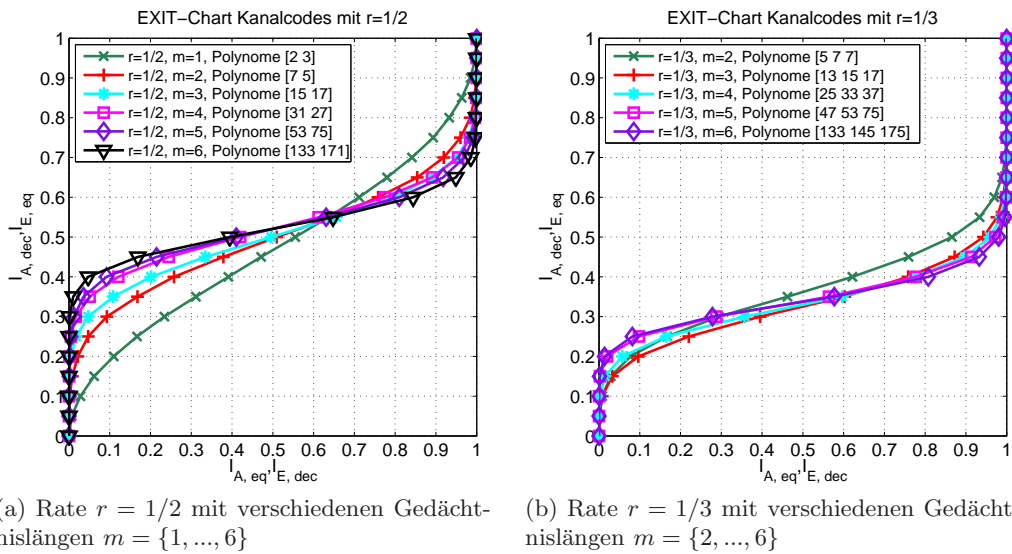


Abbildung 5.15: EXIT-Charts für Codes mit Rate $r = 1/2$ (a) bzw. Rate $r = 1/3$ (b) für verschiedene Gedächtnislängen m

Mit Hilfe der Histogramme wurde der Verlauf der wechselseitigen Information am Ein- und Ausgang berechnet.

Die Ergebnisse für einige exemplarische Codes sind in Abbildung 5.15 dargestellt. In diesen nicht vollständigen EXIT-Charts ist der Verlauf des Decodierers allein ohne Entzerrer abgebildet. Dabei sollte in Erinnerung gerufen werden, dass für den Decoder der Eingang $I_{A,dec}$ auf der y-Achse aufgetragen ist, der Ausgang $I_{E,dec}$ auf der x-Achse. Es ist zunächst zu erkennen, dass die Gedächtnislänge m einen wesentlichen Einfluss auf die Neigung der Kurve hat. Codes mit großer Gedächtnislänge liefern mit steigendem $I_{A,dec}$ zunächst kein höheres $I_{E,dec}$, ab einer bestimmten Schwelle jedoch steigen sie sprunghaft an und $I_{E,dec}$ sehr schnell gegen 1 geht. Damit bilden sie bei Betrachtung des EXIT-Charts ein so genanntes Knie [35]. Codes mit geringer Gedächtnislänge hingegen verlaufen wesentlich linearer. Dieser Effekt ist jedoch nicht nur von der Gedächtnislänge allein abhängig. Es existieren auch Codes, die trotz größerer Gedächtnislänge ein schwächeres Knie ausbilden als Codes mit niedriger Gedächtnislänge. Dies tritt vermutlich bei Codes mit ungünstigen Eigenschaften, z.B. schlechten Distanzeigenschaften, auf, wurde im Rahmen dieser Arbeit aber nicht weiter untersucht.

Weiterhin ist zu erkennen, dass eine Verringerung der Rate zu einer vertikalen Verschiebung nach unten führt. Es ist nachvollziehbar, dass bei niedrigerer Rate, also höherer Redundanz, der sprunghafte Anstieg von $I_{E,dec}$ schon für niedrigere Werte von $I_{A,dec}$ einsetzt. Damit wird das vom Decodierer gebildete Knie flacher.

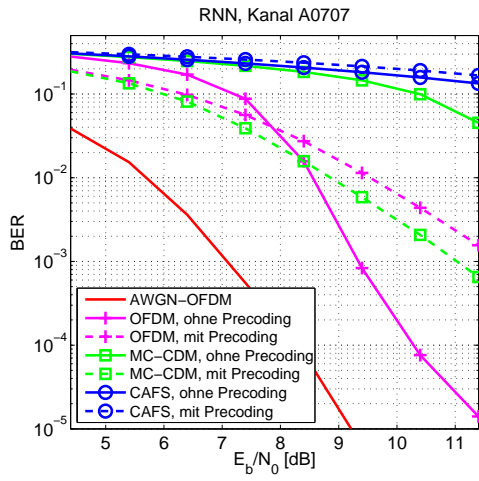
Codierte Übertragung über zeitinvariante Testkanäle

Zur Simulation des gesamten Übertragungssystems wurden nun die in Abschnitt 5.3.3 definierten Testkanäle verwendet. Es wurde für jeden Kanal eine Bitfehlermessung über E_b/N_0 durchgeführt und es wurden EXIT-Charts an verschiedenen Stellen von E_b/N_0 erstellt, jeweils für den RNN und den SCE. Die Ergebnisse für die Testkanäle mit $\rho_T = 0,7$ und $\rho_R = 0,7$

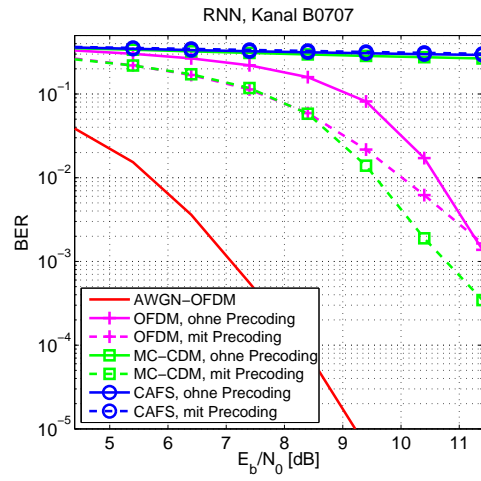
sind in den Abbildungen 5.16 bis 5.19 dargestellt. Die Ergebnisse für die übrigen Testkanäle in Anhang B zu finden. Beim Vergleich der Bitfehlerkurven muss, wie in Abschnitt 5.3.2 beschrieben, beachtet werden, dass bei den EXIT-Charts stets das E_b/N_0 des Kanals ohne Berücksichtigung von Verlusten durch Coderate und Cyclic Prefix angegeben wird. Der verwendete Kanalcode war stets der selbe, so dass in allen EXIT-Charts die gleiche Kanaldecodiererkurve zu sehen ist. Der eigentliche Ausgangspunkt des Turbo-Prozesses im EXIT-Chart ist der Ursprung $(0, 0)$. Zugunsten einfacherer Formulierungen wird im Folgenden jedoch der Punkt $(0, I_{E,eq})$, also das Ergebnis der ersten Iteration durch den Entzerrer, als Startpunkt bezeichnet. Analog wird das Ergebnis der letzten Iteration des Entzerrers Endpunkt genannt.

Zunächst werden die Simulationen des RNN betrachtet. Die Testkanäle A0707 bis D0707 weisen hohe Korrelationen an Sender und Empfänger und somit hohe Interferenzen auf. Die Bitfehlerkurven für den Kanal A0707 in Abb. 5.16a zeigen, dass der Turbo-Entzerrer für MC-CAFS sowie MC-CDM ohne Precoding kaum konvergiert. In den dazugehörigen EXIT-Charts in den Abbildungen 5.16d und 5.16f ist zu erkennen, dass die Öffnung zwischen den Kurven klein ist. Die Startpunkte für MC-CAFS und MC-CDM ohne Precoding liegen, bedingt durch die Interferenzen, zu tief. Zwar ist für diese Kurven eine Öffnung im EXIT-Chart vorhanden, diese scheint aber nicht ausreichend zu sein. Es muss berücksichtigt werden, dass die Kurve für den Entzerrer nur eine Näherung ist und die erste, einfache Iteration in der Bitfehlersimulation ein schlechteres Ergebnis liefern kann als der erste Punkt der EXIT-Chart-Kurve, die mit zwei Iterationen gewonnen wurde. Im EXIT-Chart würde dies bedeuten, dass die wirkliche Zustandskurve in der ersten Iteration nicht bis zur Entzerrer-Linie ansteigt, sondern früher zum Stehen kommt. Dies kann zur Folge haben, dass der Entzerrer in den ersten Iterationsschritten am Knie des Kanaldecodierers stecken bleibt. Die Bitfehlerkurven für MC-CDM und ungespreiztes OFDM mit Precoding konvergieren, verlaufen jedoch recht flach. Im EXIT-Chart ist zu erkennen, dass ihr Startpunkt höher liegt, was auf die reduzierten Interferenzen durch das Precoding zurückzuführen ist. Dafür liegt ihr Endpunkt sehr niedrig, was vermutlich auf die verschlechterte Matched Filter Bound zurückzuführen ist, und die Kurven schneiden die Decodiererkurve früh. Dies wiederum führt zu einem niedrigen Turbo-Gewinn bei der Bitfehlerrate und somit zu einem flacheren Verlauf. Auffällig steil ist der Verlauf der Bitfehlerkurve für ungespreiztes OFDM ohne Precoding. Im EXIT-Chart ist zu sehen, dass ihr Startpunkt in der Mitte zwischen denen der anderen Kurven liegt, da die Interferenz nicht durch Precoding reduziert, aber auch nicht durch Spreizen erhöht wird. Der Startpunkt liegt aber hoch genug, um ein Steckenbleiben der Zustandskurve, wie bei MC-CAFS, zu verhindern. Der Endpunkt hingegen liegt hoch, da die Matched Filter Bound nicht verschlechtert wurde. Die Endpunkte für MC-CAFS mit und ohne Precoding liegen noch höher, jedoch kann der Turbo-Entzerrer hiervon nicht profitieren, da die Entzerrung bereits am Anfang stecken bleibt. Somit erzielt in diesem Szenario überraschenderweise ungespreiztes OFDM ohne Precoding die beste Bitfehlerperformance für steigendes E_b/N_0 .

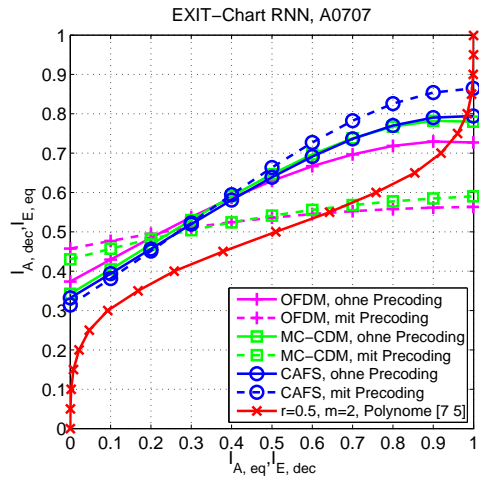
Die Ergebnisse für den Kanal B0707 in Abb. 5.16b zeigen ein ähnliches Bild. MC-CAFS und MC-CDM ohne Precoding konvergieren nicht, was durch den sehr frühen Schnittpunkt in den EXIT-Charts (Abb. 5.16d und 5.16f) bestätigt wird. Bedingt durch die schlechtere *condition number* und die etwas höheren Interferenzen des Kanals im Vergleich zu A0707, liegen alle Startpunkte im EXIT-Chart niedriger. Deshalb verlaufen die übrigen Bitfehlerkurven für niedriges E_b/N_0 zunächst flacher, die Wasserfall-Region mit deutlichem Turbo-Gewinn wird im Vergleich zu Kanal A0707 erst für höheres E_b/N_0 erreicht.



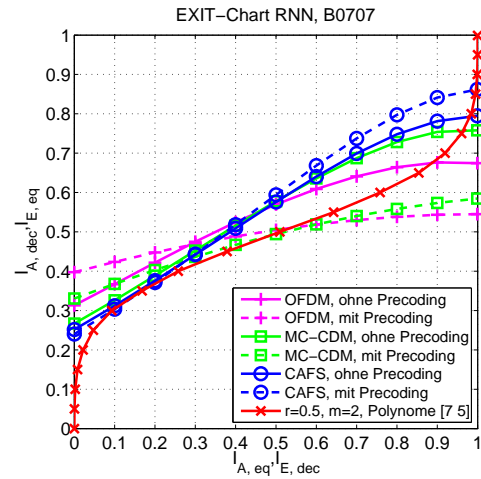
(a) Bitfehlerkurven Kanal A0707



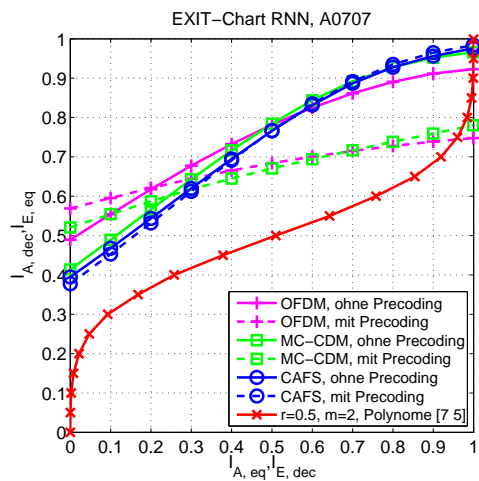
(b) Bitfehlerkurven Kanal B0707



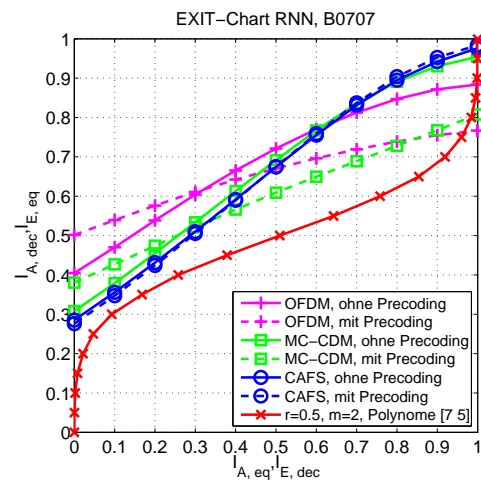
(c) EXIT-Chart für Kanal A0707 bei 4 dB



(d) EXIT-Chart für Kanal B0707 bei 4 dB

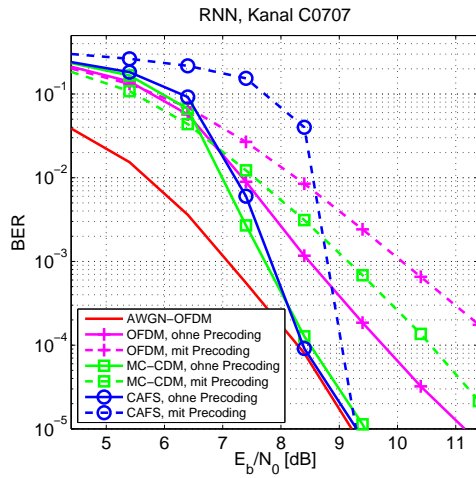


(e) EXIT-Chart für Kanal A0707 bei 8 dB

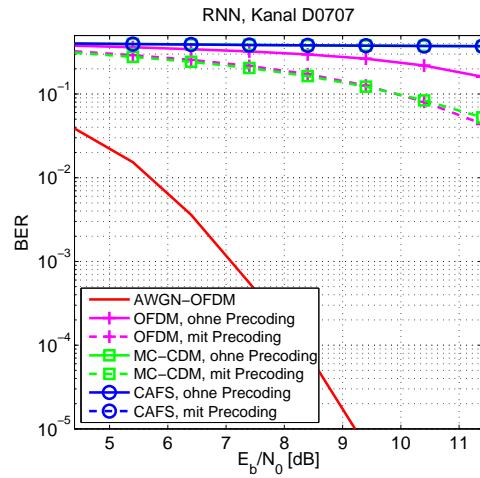


(f) EXIT-Chart für Kanal B0707 bei 8 dB

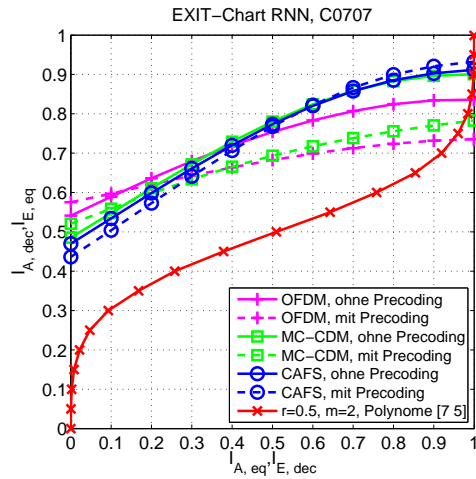
Abbildung 5.16: RNN: Bitfehlerkurven und zugehörige EXIT-Charts an zwei Stellen von E_b/N_0 für die Kanäle A0707 und B0707 und den Kanalcode mit $r = 1/2, m = 2, [7, 5]_8$



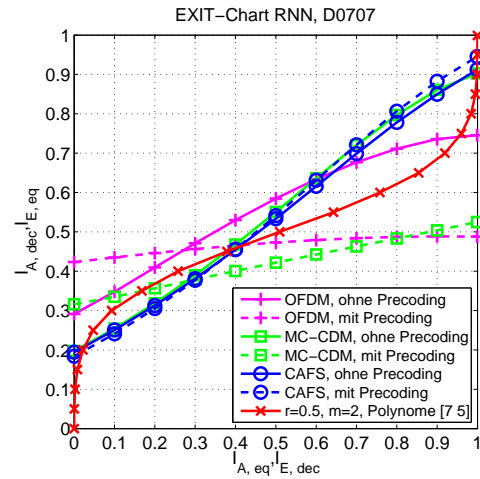
(a) Bitfehlerkurven Kanal C0707



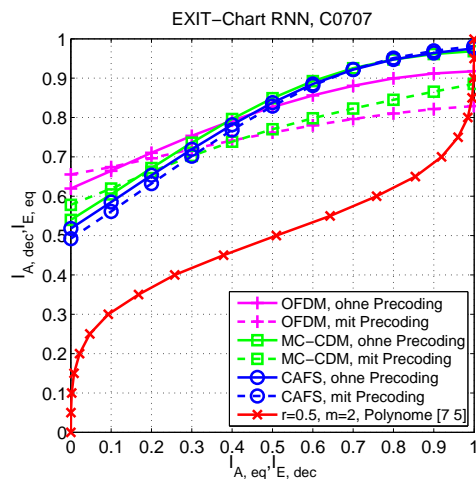
(b) Bitfehlerkurven Kanal D0707



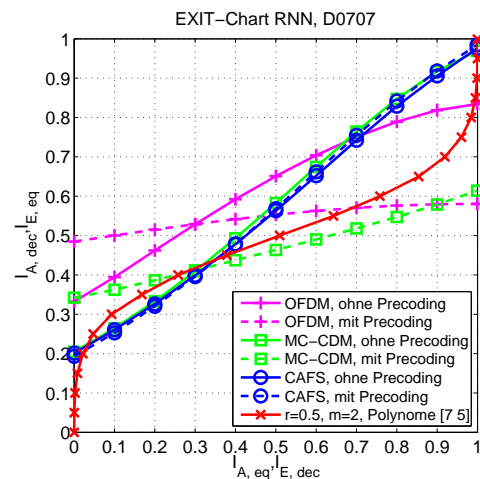
(c) EXIT-Chart für Kanal C0707 bei 6 dB



(d) EXIT-Chart für Kanal D0707 bei 6 dB



(e) EXIT-Chart für Kanal C0707 bei 8 dB



(f) EXIT-Chart für Kanal D0707 bei 8 dB

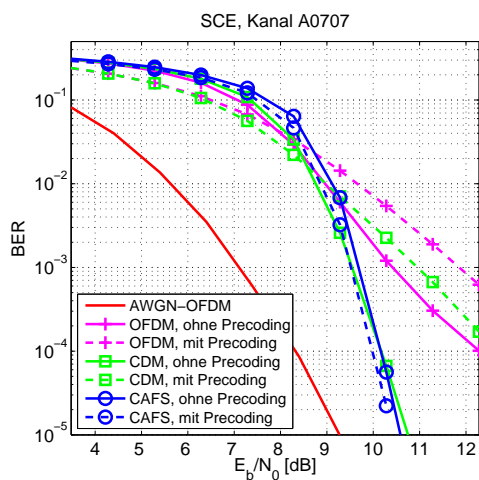
Abbildung 5.17: RNN: Bitfehlerkurven und zugehörige EXIT-Charts an zwei Stellen von E_b/N_0 für die Kanäle C0707 und D0707 und den Kanalcode mit $r = 1/2, m = 2, [7, 5]_8$

Der Kanal C0707 (Abb. 5.17a) zeichnet sich durch geringe Interferenzen aus, was im Vergleich zu den bisherigen Testkanälen zu deutlich besseren Bitfehlerraten führt und die Empfindlichkeit des RNN gegen Interferenzen verdeutlicht. Alle Startpunkte in den EXIT-Charts (Abb. 5.17c und 5.17e) liegen höher und das EXIT-Chart ist im Anfangsbereich für alle Kurven mit Ausnahme von MC-CAFS mit Precoding weit genug geöffnet. Die Bitfehlerperformance wird deshalb durch den Endpunkt bestimmt. MC-CAFS konvergiert hier mit und ohne Precoding, da im EXIT-Chart der Startpunkt für MC-CAFS mit Precoding jedoch niedriger liegt als ohne, setzt der Turbo-Gewinn erst später ein. Der Endpunkt liegt mit Precoding zwar etwas höher, dies macht sich jedoch nicht bemerkbar, da in beiden Fällen $I_{E,dec} \approx 1$ erreicht wird. Somit konvergieren beide Bitfehlerkurven gegen die AWGN-Kurve, ebenso MC-CDM ohne Precoding. Für MC-CDM und ungespreiztes OFDM mit Precoding liegt der Endpunkt aufgrund der verschlechterten MFB niedriger, was zu niedrigerem Turbo-Gewinn und somit flacheren Bitfehlerkurven führt.

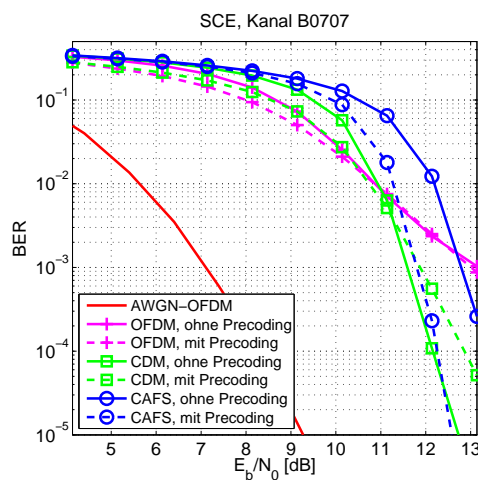
Im Gegensatz zum Kanal C0707 weist der Kanal D0707 hohe Interferenzen auf. Dadurch konvergieren die Bitfehlerkurven in Abb. 5.17b kaum. Die EXIT-Charts (Abb. 5.17d und 5.17f) zeigen nur eine geringe Öffnung, auch wenn Precoding bei ungespreiztem OFDM und MC-CDM den Startpunkt deutlich nach oben verschiebt. Dennoch schneiden die Kurven, aufgrund des niedrigen Endpunkts die Decodiererkurve sehr früh, was zu einer sehr flachen Bitfehlerkurve führt.

Die Ergebnisse für den RNN zeigen, dass dieser Entzerrer empfindlich auf Interferenzen reagiert. Wird die Summe der Interferenzen durch Korrelationen und Spreizen zu groß, konvergiert der Turbo-Entzerrer nicht. In diesem Fall ist der Einsatz eines Spreizschemas mit hoher Diversität wie MC-CAFS nicht effektiv, wie im EXIT-Chart deutlich wird. Auch wenn der Endpunkt im EXIT-Chart hoch liegt und somit einen guten Endwert für $I_{E,dec}$ erlauben würde, ist dies nutzlos, wenn der Decodierprozess bereits im Anfangsstadium stecken bleibt. Deshalb ist es wichtig, dass das EXIT-Chart im Anfangsbereich weit genug geöffnet ist und der Decodierprozess hier nicht behindert wird. Precoding kann die Interferenzen für ungespreiztes OFDM und MC-CDM reduzieren und somit den Startpunkt im EXIT-Chart deutlich verbessern. Dies geschieht allerdings auf Kosten des Endpunktes, der durch die verschlechterte Matched Filter Bound sinkt. Für diesen Entzerrer und stark korrelierte Kanäle kann Precoding helfen, im Bereich von niedrigem E_b/N_0 eine bessere Konvergenz zu erhalten, bis mit wachsendem E_b/N_0 die Öffnung groß genug ist und die Performance durch den Endpunkt der Entzerrer-Kurven bestimmt wird.

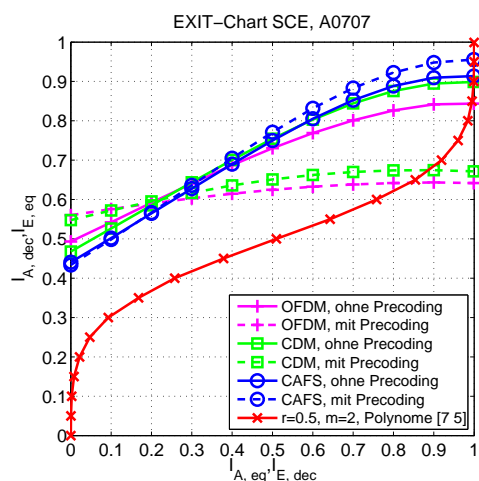
Die Ergebnisse für die Übertragung mit dem SCE sind in den Abbildungen 5.18 und 5.19 dargestellt. Im Unterschied zum RNN sind bei diesem Entzerrer nahezu keine Konvergenzprobleme zu erkennen. Abbildung 5.18a zeigt die Bitfehlerkurven für den Kanal A0707. MC-CAFS konvergiert mit und ohne Precoding und liefert ab einem E_b/N_0 von $\approx 9,5$ dB die beste Bitfehlerrate. Die Startpunkte in den EXIT-Charts liegen für alle Kurven höher als beim RNN für diesen Kanal, was zeigt, dass der SCE die Interferenzen des stark korrelierten Kanals besser entfernen kann. Für niedriges E_b/N_0 , so lange die Performance durch den Startbereich des EXIT-Charts bestimmt wird, liefern ungespreiztes OFDM und MC-CDM die beste Bitfehlerrate, die Bitfehlerkurven verlaufen für steigendes E_b/N_0 jedoch flach, was durch den schlechten Endpunkt im EXIT-Chart erklärbar ist. Sobald die Öffnung des EXIT-Charts groß genug ist, werden sie von MC-CAFS und MC-CDM ohne Precoding überholt, die dann von ihrer höheren Diversität profitieren können und sehr steil verlaufen.



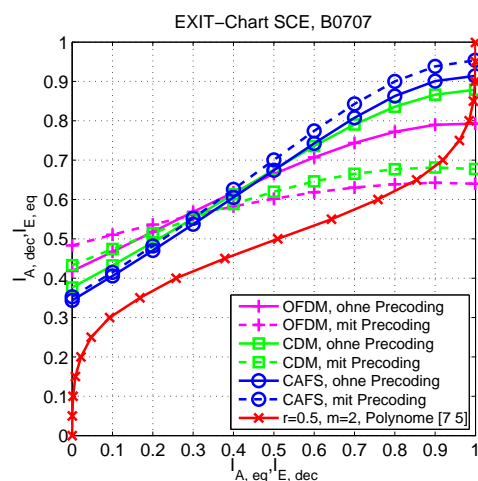
(a) Bitfehlerkurven Kanal A0707



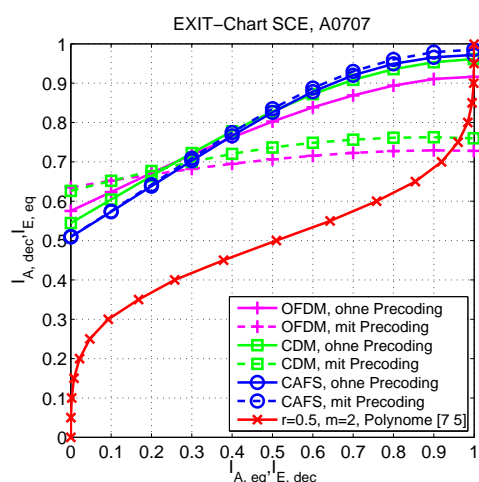
(b) Bitfehlerkurven Kanal B0707



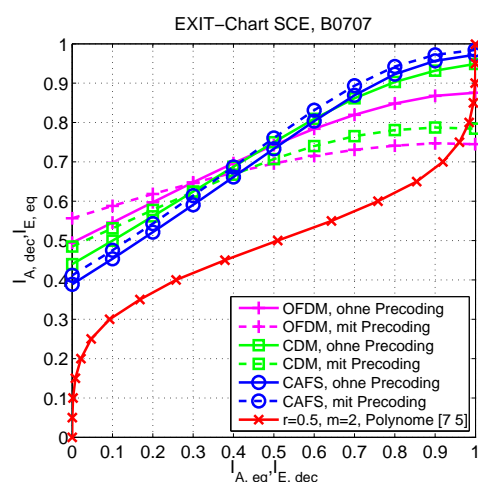
(c) EXIT-Chart für Kanal A0707 bei 4 dB



(d) EXIT-Chart für Kanal B0707 bei 4 dB

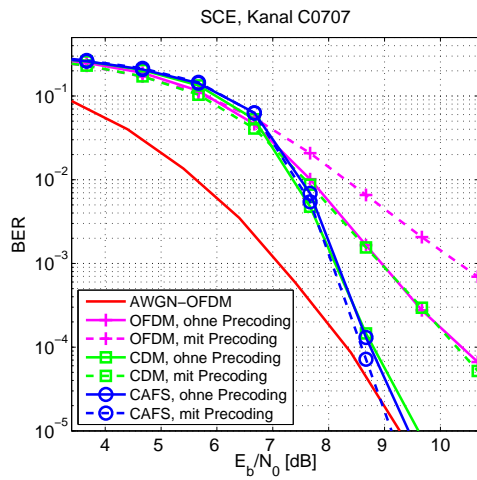


(e) EXIT-Chart für Kanal A0707 bei 6 dB

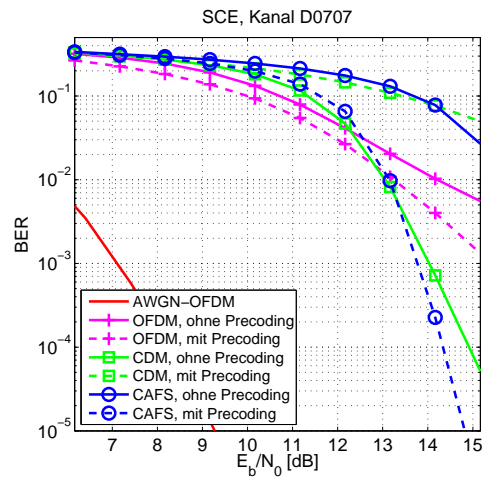


(f) EXIT-Chart für Kanal B0707 bei 6 dB

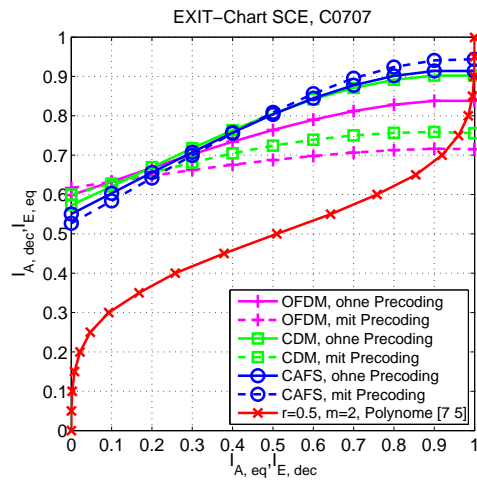
Abbildung 5.18: SCE: Bitfehlerkurven und zugehörige EXIT-Charts an zwei Stellen von E_b/N_0 für die Kanäle A0707 und B0707 und den Kanalcode mit $r = 1/2$, $m = 2$, $[7, 5]_8$



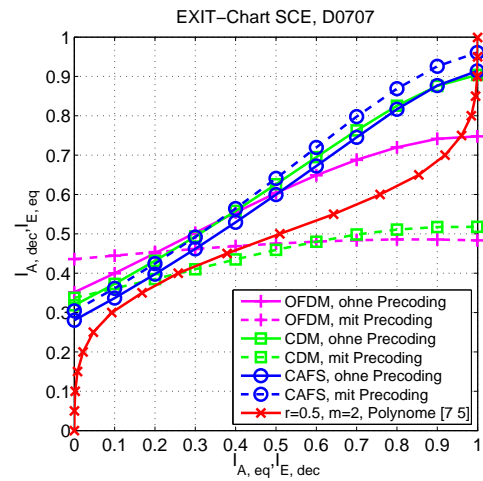
(a) Bitfehlerkurven Kanal C0707



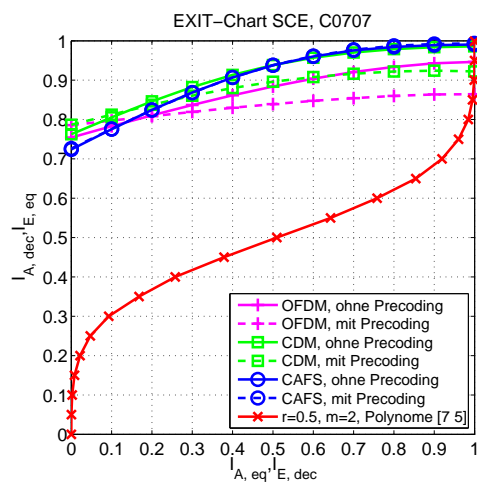
(b) Bitfehlerkurven Kanal D0707



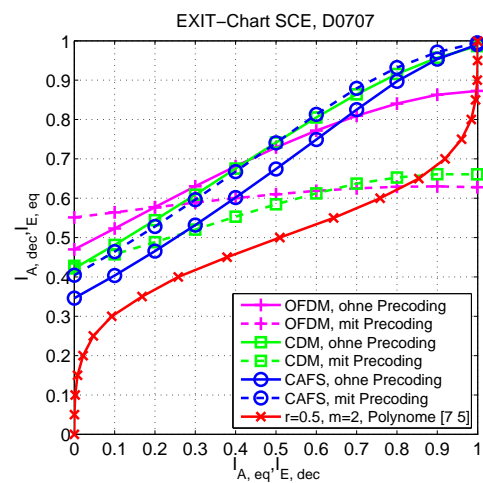
(c) EXIT-Chart für Kanal C0707 bei 4 dB



(d) EXIT-Chart für Kanal D0707 bei 4 dB



(e) EXIT-Chart für Kanal C0707 bei 8 dB



(f) EXIT-Chart für Kanal D0707 bei 8 dB

Abbildung 5.19: SCE: Bitfehlerkurven und zugehörige EXIT-Charts an zwei Stellen von E_b/N_0 für die Kanäle C0707 und D0707 und den Kanalcode mit $r = 1/2$, $m = 2$, $[7, 5]_8$

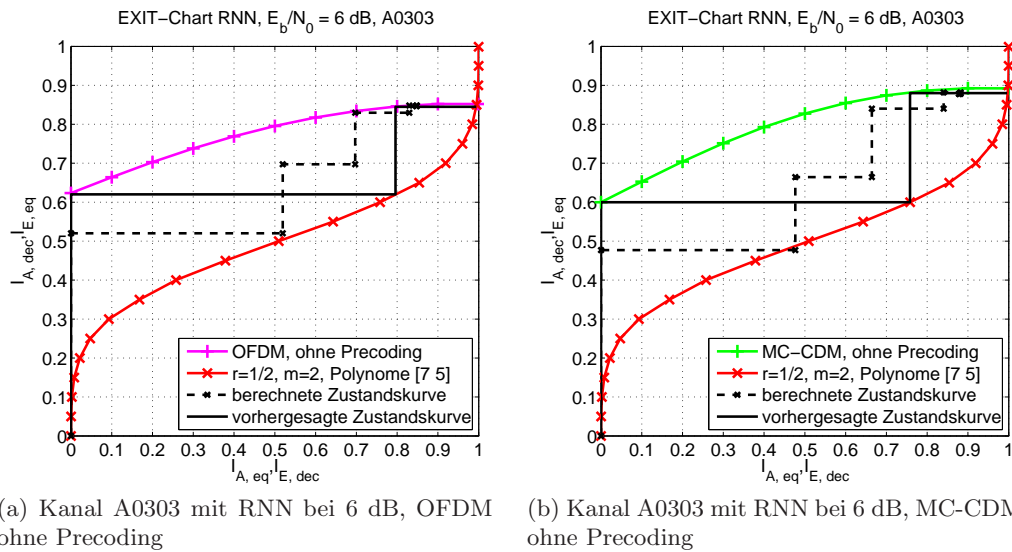
Beim Kanal B0707, dessen Bitfehlerkurven in Abb. 5.18b dargestellt sind, sind die Startpunkte in den EXIT-Charts (Abb. 5.18d und 5.18f), bedingt durch die schlechtere *condition number* und die etwas höheren Interferenzen, nach unten verschoben im Vergleich zu Kanal A0707. Deshalb wird der Punkt, an dem die Bitfehlerrate nicht mehr vom Startbereich, sondern den Endpunkten des EXIT-Charts bestimmt wird, erst für ein höheres E_b/N_0 erreicht.

Kanal C0707 beinhaltet wenig Interferenzen, was an den Bitfehlerkurven in 5.19a und den EXIT-Charts in den Abb. 5.19c und 5.19e nachvollzogen werden kann. Durch die niedrigen Interferenzen bedingt liegen die Startpunkte aller Kurven im EXIT-Chart hoch. Somit wird die Performance durch die Endpunkte bestimmt, die für MC-CAFS und MC-CDM ohne Precoding zu einem Schnittpunkt der Kurven bei $I_{E,dec} \approx 1$ führen. Die entsprechenden Bitfehlerraten nähern sich der AWGN-Kurve an. Die übrigen Kurven verlaufen, ihren niedrigeren Schnittpunkten der Entzerrerkurve mit der Decodiererkurve entsprechend, flacher.

Die starken Interferenzen in Kanal D0707 bewirken, dass die Öffnung der EXIT-Charts in Abb. 5.19d und 5.19f wesentlich kleiner wird. Dementsprechend schlechter sind die Bitfehlerkurven in Abb. 5.19b. Für niedriges E_b/N_0 liefert somit ungespreiztes OFDM durch seinen hohen Startpunkt im EXIT-Chart die beste Bitfehlerrate, während für höheres E_b/N_0 die Öffnung für MC-CAFS mit Precoding und MC-CDM ohne Precoding groß genug wird. Auffällig ist hier, dass die Öffnung für MC-CAFS ohne Precoding erst wesentlich später ausreichend ist.

Die Ergebnisse für den SCE zeigen, dass dieser Entzerrer Interferenzen sehr gut kompensieren kann. Ist das E_b/N_0 groß genug, bietet MC-CAFS eine gute Performance, die mit Precoding stets besser ist als ohne. Für niedriges E_b/N_0 erzielen ungespreiztes OFDM und MC-CDM jeweils mit Precoding die beste Bitfehlerrate, bedingt durch den verbesserten Startpunkt im EXIT-Chart. Somit ist beim SCE der Einsatz von Precoding in jeder Situation vorteilhaft. Die Simulationsergebnisse für die übrigen Testkanäle mit $[\rho_T = 0,3, \rho_R = 0,3]$ und $[\rho_T = 0,7, \rho_R = 0,0]$ sind in Anhang B zu finden.

Abbildung 5.20 zeigt zwei Beispiele für den Verlauf der Zustandskurven, die aus den L-Werten einer Bitfehlersimulation mit dem RNN gewonnen wurden. Zusätzliche ist die vorhergesagte Zustandskurve eingezeichnet, bei der angenommen wird, dass der Turbo-Prozess bei jedem Iterationsschritt den maximal möglichen Weg zwischen den Kurven von Entzerrer und Decodierer zurücklegt. Die aus der Simulation berechneten Zustandskurven zeigen, dass dies nicht der Fall ist, sondern das reale Verhalten schlechter ist. Der Turbo-Prozess kann meist nicht den gesamten Weg bis zum vorhergesagten Schnittpunkt zurücklegen. Dabei muss berücksichtigt werden, dass das EXIT-Chart wie eingangs beschrieben nur eine Abschätzung mit vereinfachenden Annahmen ist und das reale Verhalten davon abweicht. Dies erklärt, warum in manchen Konstellationen der Decodierprozess nicht konvergiert, obwohl im EXIT-Chart eine kleine Öffnung vorhanden ist. Insbesondere der erste Iterationsschritt ist kritisch, da der reale Decodierprozess nicht den berechneten ersten Punkt erreicht, sondern nur einen Teil der Strecke zurücklegt. Somit besteht die Gefahr, dass der Prozess früh im Bereich des Decodierer-Knies beeinträchtigt wird.



(a) Kanal A0303 mit RNN bei 6 dB, OFDM ohne Precoding

(b) Kanal A0303 mit RNN bei 6 dB, MC-CDM ohne Precoding

Abbildung 5.20: EXIT-Charts mit aus Daten der Bitfehlersimulationen berechnetem Verlauf der Zustandskurve

Codierte Übertragung über zeitvariante stochastische Kanäle

Um das Verhalten in einem realen Übertragungssystem besser zu erfassen, wurde bei den folgenden Simulationen stochastisch-zeitvariante Kanäle mit kurzer Kohärenzzeit benutzt. Die Kohärenzzeit entsprach dabei der Dauer eines Codeworts. Die Abbildung 5.21 zeigt die Simulationsergebnisse für Kanäle mit den Korrelationskoeffizientenpaaren $[\rho_T = 0,3, \rho_R = 0,3]$, $[\rho_T = 0,7, \rho_R = 0,7]$ und $[\rho_T = 0,7, \rho_R = 0,0]$ jeweils für RNN und SCE. Die Abbildungen 5.21a und 5.21b zeigen die Ergebnisse für RNN bzw. SCE bei niedrigen Korrelationen. In beiden Fällen bringt MC-CAFS die besten Ergebnisse, Precoding hat beinahe keine Auswirkungen. Beim RNN jedoch flachen die Kurven mit steigendem E_b/N_0 ab, was vermutlich darauf zurückzuführen ist, dass sich aufgrund der durch den Spreizcode erhöhten Interferenzen ein früher *error floor* ausbildet. Dies wird dadurch bestätigt, das MC-CDM ohne Precoding die selbe Tendenz zeigt. Die Bitfehlerrate für ungespreiztes OFDM und MC-CDM ist mit Precoding schlechter, die Reduzierung der Interferenz in diesem interferenzarmen Kanal keinen Vorteil bringt, während sich die verschlechterte Matched Filter Bound bemerkbar macht.

Die Ergebnisse für $[\rho_T = 0,7, \rho_R = 0,7]$ sind in den Abbildungen 5.21c und 5.21d zu sehen und entsprechen weitgehend denen für die bereits gezeigten Testkanäle. MC-CAFS sowie MC-CDM ohne Precoding konvergieren beim RNN aufgrund der hohen Interferenzen nicht, die besten Ergebnisse bringen MC-CDM und ungespreiztes OFDM jeweils mit Precoding. Beim SCE ist der Precoding-Gewinn für MC-CAFS deutlicher als bei den Testkanälen. Für hohes E_b/N_0 liefert MC-CAFS mit Precoding die beste Bitfehlerrate, für niedriges E_b/N_0 ebenfalls ungespreiztes OFDM und MC-CDM mit Precoding.

Die Abbildungen 5.21e und 5.21f zeigen die Resultate für $[\rho_T = 0,7, \rho_R = 0,0]$. Die Ergebnisse für den RNN zeigen, dass sich bei Wegfall der Korrelationen am Empfänger das Konvergenzverhalten für MC-CAFS und MC-CDM ohne Precoding etwas verbessert. Die Kurven verlaufen aber noch immer sehr flach. Die mit Abstand beste Bitfehlerrate liefert MC-CDM mit Precoding. Der SCE hingegen hat keine Konvergenzprobleme. Dementspre-

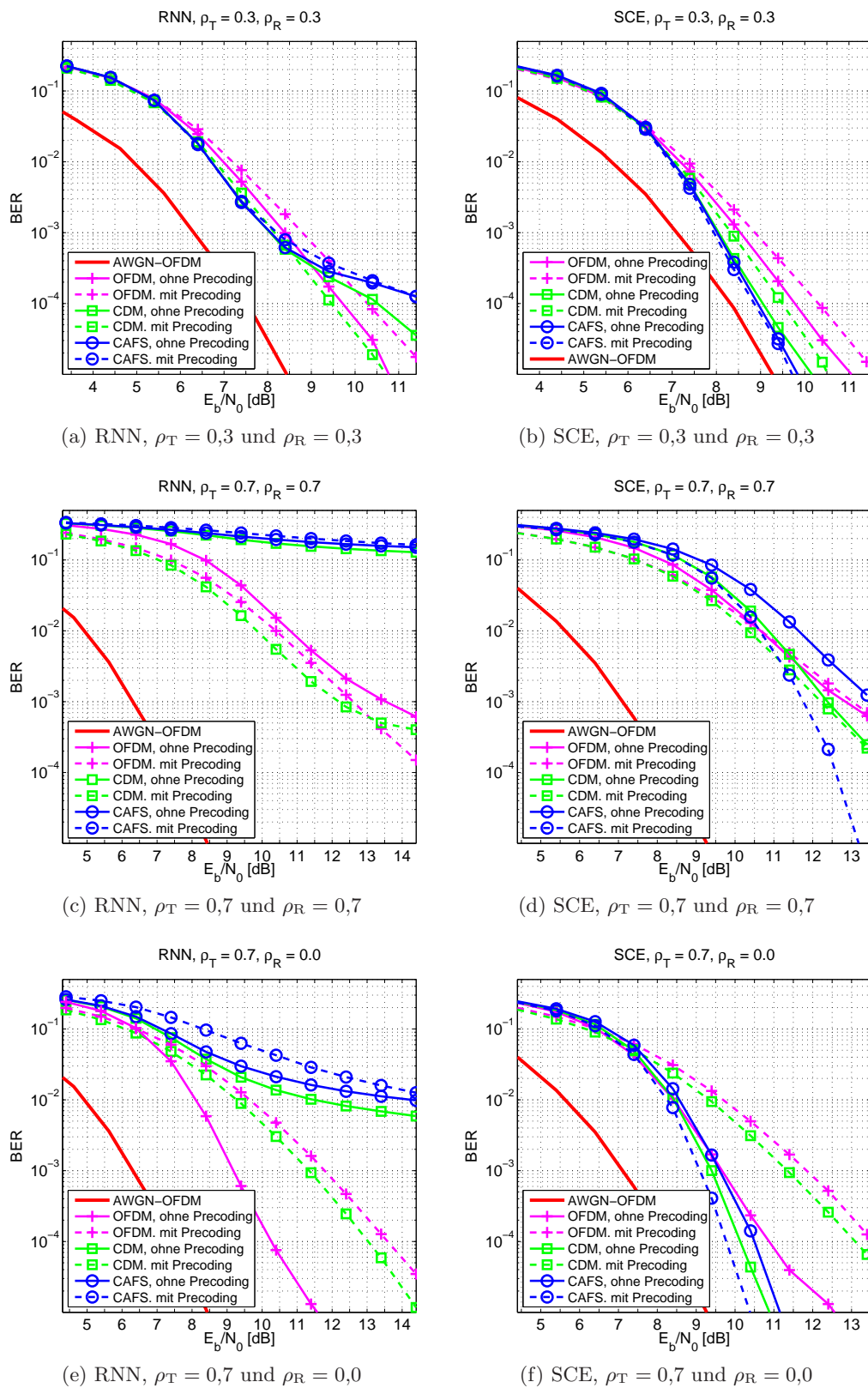
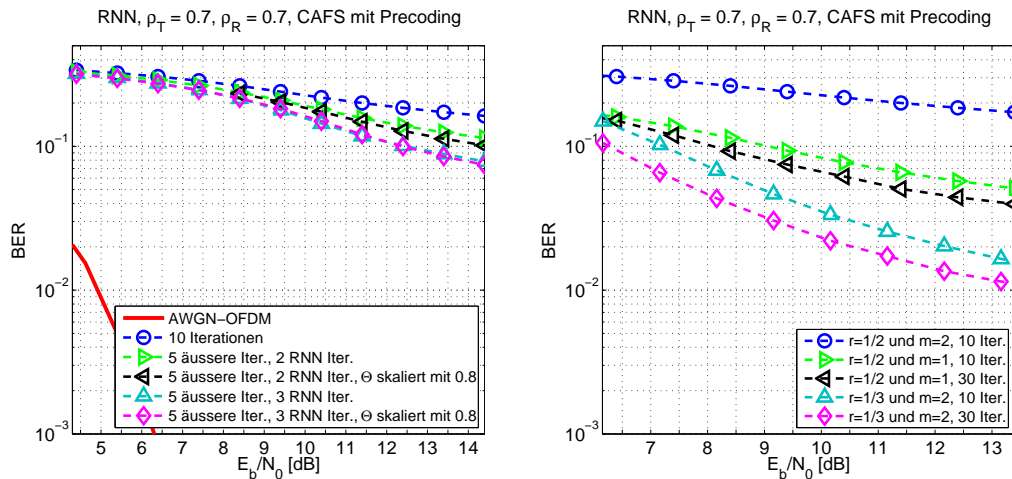


Abbildung 5.21: Zeitvariante stochastische Kanäle: Bitfehlerkurven für RNN und SCE für verschiedene Korrelationskoeffizienten und den Kanalcode mit $r = 1/2$, $m = 2$, $[7, 5]_8$



(a) Kanalcode mit $r = 1/2$, $m = 2$ und veränderte Iterationsschemata

(b) Kanalcode mit $r = 1/2$, $m = 2$ im Vergleich mit $r = 1/3$ oder $m = 1$

Abbildung 5.22: Zeitvariante stochastische Kanäle: Bitfehlerkurven für den RNN und CAFS mit Precoding für verschiedenen Iterationsschemata und Kanalcodes mit niedrigerer Rate r oder Gedächtnislänge m im Vergleich

chend führt MC-CAFS mit Precoding zur besten Bitfehlerrate. MC-CDM und ungespreiztes OFDM mit Precoding sind aufgrund der Matched Filter Bound deutlich abgeschlagen.

In den Simulationsergebnissen wurde deutlich, dass Spreizverfahren mit hoher Diversität wie MC-CAFS ihre Wirkung nur dann entfalten können, wenn sie nicht bereits zu Beginn des Turbo-Prozesses behindert werden, was im EXIT-Chart sichtbar ist. Deshalb wurden einige Ansätze untersucht, das Konvergenzverhalten von MC-CAFS im Anfangsbereich des EXIT-Charts zu verbessern. Dazu wurde MC-CAFS mit Precoding mit dem RNN bei hohen Korrelationen betrachtet. Die Ergebnisse des ersten Ansatzes sind in Abb. 5.22a dargestellt. Ursprünglich sieht das Turbo-Schema vor, dass Entzerrer und Decodierer selbst jeweils einmal abwechselnd durchlaufen werden. Um das Ergebnis der ersten Iteration des Entzerrers zu verbessern, wurde der RNN nun in einer inneren Schleife jeweils zweimal bzw. dreimal durchlaufen, gefolgt von einem Durchlauf des Decodierers. Die gesamte äußere Schleife wurde statt zuvor 10 Mal nun 5 Mal wiederholt. Somit wurde der RNN pro Block insgesamt $2 \cdot 5 = 10$ bzw. $3 \cdot 5 = 15$ Mal durchlaufen, der Decodierer 5 Mal. Als zusätzliche Variante wurde das Ergebnis der Entscheidungsfunktion Θ mit 0,8 skaliert, also die Sicherheit der Entscheidung reduziert. Die Ergebnisse zeigen, dass diese Maßnahmen zu einer geringen Verbesserung der Bitfehlerrate führen. Sie bleibt jedoch auf einem so hohen Niveau, dass dieses Schema trotzdem nicht brauchbar ist. Es hat sich jedoch gezeigt, dass 10 Durchgänge des Entzerrers und 5 Durchgänge des Decodierers ein besseres Ergebnis liefern als jeweils 10 Durchgänge beider Komponenten. Somit bietet sich hier eventuell eine Möglichkeit, Iterationen des Decodierers einzusparen und somit den Rechenaufwand im Empfänger zu reduzieren.

Da der Turbo-Prozess durch das Knie des Decodierers behindert wird, wird im zweiten Ansatz versucht, einen anderen Kanalcode zu verwenden, der ein weniger ausgeprägtes Knie aufweist. Wie die simulierten EXIT-Charts für die verschiedenen Kanalcodes gezeigt haben (siehe Abb. 5.15), hängt die Stärke des Knies primär mit der Gedächtnislänge des Codes zusammen. Deshalb wurde die Übertragung mit einem Code der Gedächtnislänge $m = 1$

simuliert, die Ergebnisse sind in Abbildung 5.22b dargestellt. In einem zweiten Schritt wurde die Zahl der Iterationen erhöht um zu erkennen, ob aufgrund zu niedriger Zahl von Iterationen das Potential des Turbo-Schemas nicht voll ausgeschöpft wurde. Die Simulationen zeigen, dass sich die Bitfehlerrate für niedrigeres m verbessert. Ohne die Erklärung mit Hilfe des EXIT-Charts wäre dieses Resultat überraschend, da man intuitiv für größere m bessere, für kleinere m schlechtere Bitfehlerraten erwarten würde. Mit diesem Kanal und diesem Entzerrer ist jedoch das Gegenteil der Fall. Die von 10 auf 30 erhöhte Zahl der Iterationen bewirkt keine signifikante Verbesserung, verdreifacht jedoch den Rechenaufwand am Empfänger. Eine weitere Möglichkeit, das EXIT-Chart im Anfangsbereich weiter zu öffnen, ist, die Decodierkurve durch Wahl eines Codes mit niedrigerer Rate abzusenken. Dies geht mit einem Verlust in der Datenrate einher, bewirkt aber einen höheren Codierungsgewinn. Für einen Code mit Rate $r = 1/3$ verbessert sich die Bitfehlerrate merklich und die Bitfehlerkurven werden steiler. Eine höhere Anzahl von Iterationen bewirkt eine geringfügige Verbesserung.

6 Zusammenfassung

Ziel dieser Arbeit war, den Einfluss von Precoding auf Basis partieller Kanalkennntnis in einem gespreizten MIMO-OFDM-System zu untersuchen. Dabei wurde das Verhalten bei Verwendung von Spreizcodes unterschiedlicher Diversität und verschiedener Entzerrerkonzepte verglichen. Die Bitfehlerrate wurde für Kanäle mit niedrigen, mittleren und hohen Antennenkorrelationen betrachtet. Nach der Vorstellung des Übertragungsverfahrens MIMO-OFDM in Kapitel 2 wurden in Kapitel 3 verschiedene Entzerrerkonzepte für Blockübertragungen beschrieben. Das Übertragungssystem wurde daraufhin in Kapitel 4 um Kanalcodierung und einen Turbo-Entzerrer erweitert. In Kapitel 5 wurde zunächst Spreizverfahren verschiedener Diversität vorgestellt. Danach wurde Precoding auf Basis partieller Kanalkennntnis für MIMO-OFDM-Systeme mit frequenzselektiven Kanälen beschrieben.

Die Simulationsergebnisse für die uncodierte Übertragung in Abschnitt 5.2.5 zeigen, dass Precoding die Bitfehlerrate meist verbessert. In Verbindung mit dem BLE, der allgemein eine schlechte Performance aufweist, erhöht es sie jedoch. Dabei erreicht MC-CAFS stets die beste Performance. Bei den übrigen Entzerrern hingegen kann Precoding für ungespreiztes OFDM und MC-CDM zu einem erheblichen Gewinn führen, insbesondere beim BDFE. Da mit Precoding die starken Eigenmoden zuerst decodiert werden, verringert sich die Fehlerfortpflanzung. Bei MC-CAFS tritt dieser Effekt nicht auf, weshalb Precoding hier zu nahezu keiner Veränderung führt. Die Ergebnisse für den RNN zeigen eine hohe Abhängigkeit von den Interferenzen im Kanal. Precoding verzögert bei ungespreiztem OFDM und MC-CDM den Anstieg der Bitfehlerrate mit steigenden Korrelationen. Die Resultate für den SCE zeigen die niedrigsten Bitfehlerraten und das robusteste Verhalten bei Interferenzen im Vergleich der Entzerrer. Für alle Spreizvarianten tritt mit Precoding ein Gewinn auf. Zusammenfassend kann für die uncodierte Übertragung gesagt werden, dass die Verwendung von Precoding außer beim BLE sinnvoll ist. Für ungespreiztes OFDM und MC-CDM führt es stets zu einem Gewinn, bei MC-CAFS gibt es zu nahezu keine Veränderung. Bei niedrigen Korrelationen bietet MC-CAFS auch ohne Precoding die beste Bitfehlerrate, während bei höheren Korrelationen MC-CDM mit Precoding sehr gute Ergebnisse liefert.

Die Ergebnisse der codierten Simulationen in Abschnitt 5.3.4 zeigen, dass der Turbo-Prozess mit dem RNN nicht konvergiert, wenn die Summe der Interferenzen zu groß ist. Im EXIT-Chart wurde deutlich, dass der Kurvenverlauf im Bereich der ersten Iterationsschritte darüber entscheidet, ob der Prozess konvergiert oder nicht. Wird der Turbo-Prozess im Anfangsbereich nicht behindert, entscheidet der Schnittpunkt von Entzerrer- und Decodiererkurve, welche Bitfehlerperformance erreicht wird. Precoding vermindert die Interferenzen und verbessert somit die Startpunkte der Entzerrerkurven, was das Konvergenzverhalten von RNN und SCE verbessert. Gleichzeitig kann es die Matched Filter Bound und somit den Endpunkt verschlechtern. Es ist schwer beim RNN vorherzusagen, welches Übertragungsverfahren die besten Ergebnisse liefert, da dies sehr stark von der Summe der Interferenzen und damit auch vom Kanal abhängt. Sind die Interferenzen groß, liefern Spreizverfahren mit

Precoding bessere Bitfehlerraten, für niedrige Interferenzen Spreizcodes höherer Diversität ohne Precoding. Der SCE hingegen kann Interferenzen gut kompensieren und erreicht im Vergleich zum RNN stets die besseren Bitfehlerraten. Sein Verhalten ist meist vorhersagbar. Auffällig ist, dass im Vergleich zur uncodierten Übertragung Precoding bei MC-CAFS zu einem Gewinn führt, der mit steigenden Korrelationen deutlich größer wird. Somit erreicht beim SCE MC-CAFS mit Precoding stets die beste Bitfehlerrate, wenn die *waterfall region* erreicht wird. Für ungespreiztes OFDM und MC-CDM bringt Precoding einen Gewinn bei niedrigem E_b/N_0 , der sich mit steigendem E_b/N_0 in einen Verlust wandelt.

Anhang A

Waterpouring-Prinzip

In Abschnitt 5.2 wurde ein Precoder beschrieben, der die Kanalkapazität und die paarweise Bitfehlerwahrscheinlichkeit optimiert. Das *Waterpouring-* oder auch *Waterfilling-Prinzip* ist eine Lösung für das Teilproblem der optimalen Leistungsverteilung auf die einzelnen Moden eines *eigen-beamformers*. Es wird sowohl beim *eigen-beamforming* auf Basis der Kanalmatrix selbst, als auch auf Basis der Korrelationsmatrix angewendet [18][26][28]. Der Waterpouring-Algorithmus ist allgemein eine Lösung für das Problem, eine begrenzte Leistung auf parallele Gaußkanäle zu verteilen, so dass die maximale Kanalkapazität erreicht wird [21][6]. Mathematisch handelt es sich um ein konvexes Optimierungsproblem [3].

Der Waterpouring-Algorithmus teilt einem besseren Kanal, was im Falle eines MIMO-Systems einer Eigenmode mit einem größeren Eigenwert entspricht, mehr Leistung zu als einem schlechteren Kanal mit kleinerem Eigenwert. Der Name Waterpouring- oder Waterfilling-Algorithmus stammt von der bildlichen Vorstellung in Abbildung A.1. Die Kehrwerte der Eigenwerte bilden den unterschiedlich hohen Boden eines Wasserbeckens. Das Becken wird bis zur Höhe μ mit Wasser gefüllt, was mit der zur Verfügung stehenden Sendeleistung korrespondiert. Die Wasserhöhe zwischen dem Boden und der Wasserlinie entspricht der Leistung, die der jeweiligen Eigenmode zugeteilt wird. Ragt der Boden einer Eigenmode aus dem Wasser heraus, so wird sie nicht benutzt und erhält keine Leistung zugeteilt.

Angenommen, der betrachtete MIMO-Kanal hat den Rang r . Die optimale zugeteilte Leistung für Eigenwert λ_i entspricht dann [26]

$$\gamma_i = \mu - \frac{N_0}{E_s \lambda_i} > 0 \quad \text{mit } i = 1, 2, \dots, r \quad (\text{A.1})$$

wobei E_s die Energie ist, die pro Sendeantenne zur Verfügung steht, und

$$\sum_{i=1}^r \gamma_i = n_T. \quad (\text{A.2})$$

Um die γ_i zu berechnen, wird wie folgt vorgegangen. Zunächst werden die Eigenwerte λ_i absteigend nach Größe sortiert. Dann wird der Iterationszähler $p = 1$ gesetzt und die Konstante μ wird berechnet als [26]

$$\mu = \frac{n_T}{(r - p + 1)} \left[1 + \frac{N_0}{E_s n_T} \sum_{i=1}^{r-p+1} \lambda_i \right] \quad (\text{A.3})$$

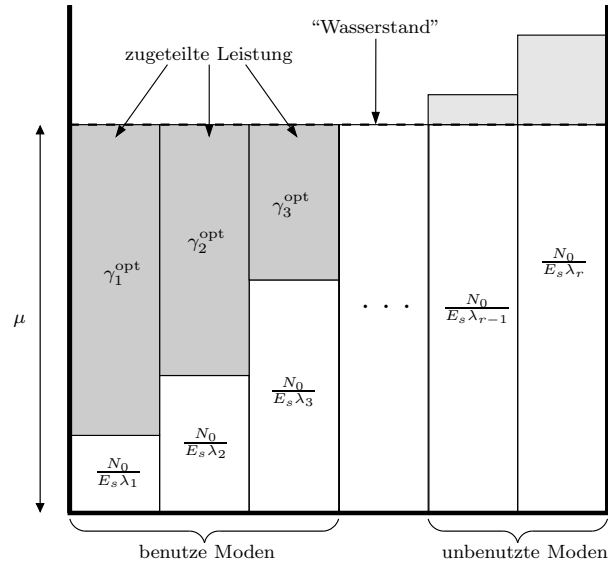


Abbildung A.1: Prinzip des Waterpouring-Algorithmus: γ_i^{opt} ist die optimal zugeteilte Leistung für Eigenmode i

Nun können die γ_i berechnet werden mit

$$\gamma_i = \mu - \frac{N_0}{E_s \lambda_i} \quad \text{mit} \quad i = 1, 2, \dots, r - p + 1. \quad (\text{A.4})$$

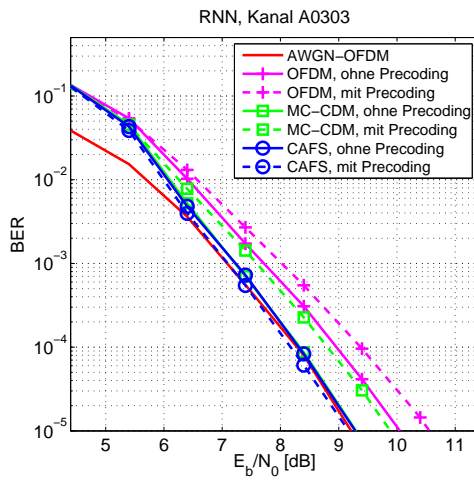
Ist die Leistung, die der Mode mit dem kleinsten Eigenwert zugewiesen wurde, negativ, also $\gamma_{r-p+1} < 0$, so wird diese Mode nicht verwendet und $\gamma_{r-p+1} = 0$ gesetzt. Daraufhin wird p um Eins erhöht und der Algorithmus erneut durchlaufen. Die optimalen γ_i sind gefunden, wenn keiner Mode eine negative Leistung zugeteilt wurde [26].

Anhang B

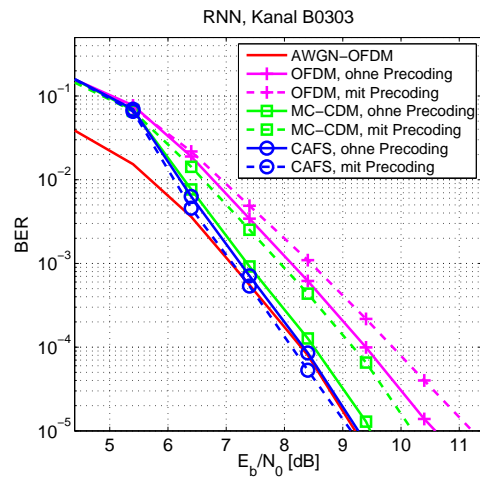
Weitere Ergebnisse der codierten Übertragung über Testkanäle

Dieser Anhang enthält die Simulationsergebnisse der codierten Übertragung über die übrigen Testkanäle. Die Ergebnisse des RNN für die Testkanäle mit $[\rho_T = 0,3, \rho_R = 0,3]$ sind in den Abbildungen B.1 und B.2 dargestellt, die Ergebnisse für die Kanäle mit $[\rho_T = 0,7, \rho_R = 0,0]$ in den Abbildungen B.3 und B.4. Die Simulationsergebnisse für den SCE für $[\rho_T = 0,3, \rho_R = 0,3]$ sind in den Abbildungen B.5 und B.6, für $[\rho_T = 0,7, \rho_R = 0,0]$ in den Abbildungen B.7 und B.8 zu finden.

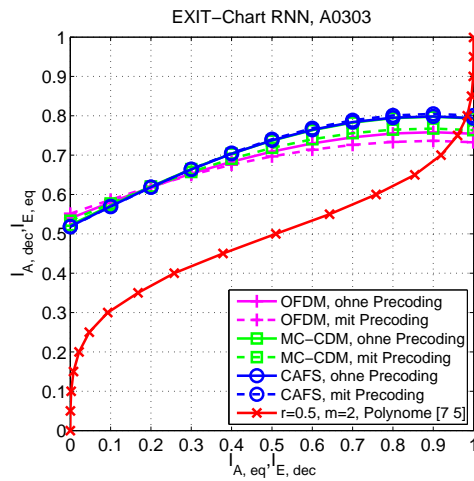
Die Simulationen zeigen ähnlich Resultate wie die in Abschnitt 5.3.4 vorgestellten. Auf eine Besonderheit sollte jedoch explizit aufmerksam gemacht werden. Einige Simulationen zeigen für MC-CAFS mit Precoding eine bessere Performance als AWGN. Dies ist besonders auffällig in den Abbildungen B.3b und B.4a sowie B.7b und B.8a. Dies ist ein Effekt der Leistungszuteilung. In den Abbildungen ist das E_b/N_0 am Sender aufgetragen. Ähnlich wie bei konventionellem Beamforming bewirkt die Konzentration der Leistung auf einige Moden, dass die Bitfehlerkurve nicht nur gegen die AWGN-Performance konvergiert, sondern diese übertroffen wird. Die Multiplikation mit der nicht-orthogonalen Leistungszuteilungsmatrix verändert die Eigenwerte des Kanals. Dabei erhöht sich die Summe der Diagonalelemente und wird größer als die ursprüngliche Normierung $\sum_{i=1}^{Nn_T} r_{ii} = Nn_T$.



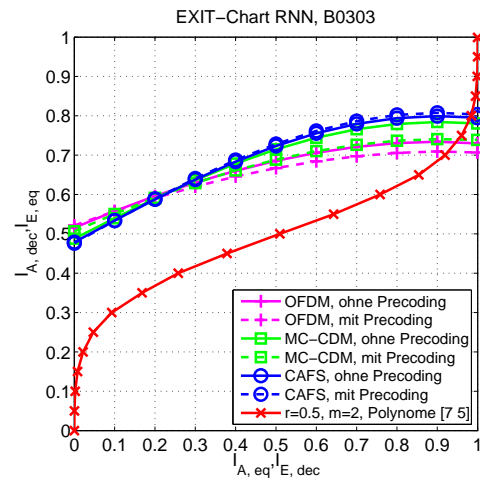
(a) Bitfehlerkurve Kanal A0303



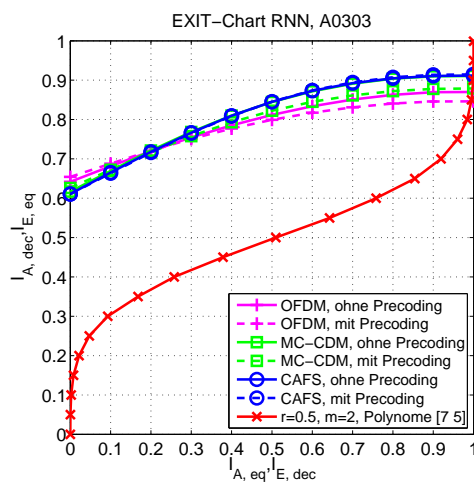
(b) Bitfehlerkurve Kanal B0303



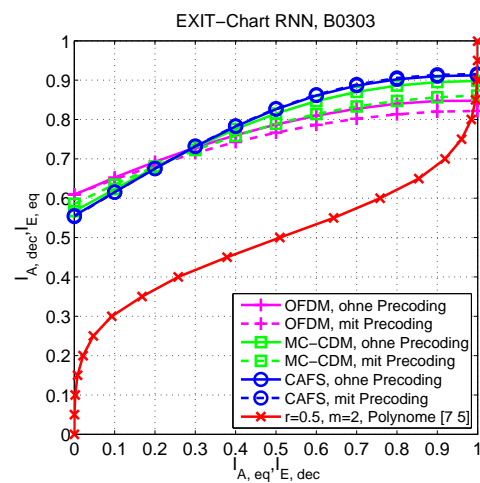
(c) EXIT-Chart für Kanal A0303 bei 7 dB



(d) EXIT-Chart für Kanal B0303 bei 7 dB



(e) EXIT-Chart für Kanal A0303 bei 9 dB



(f) EXIT-Chart für Kanal B0303 bei 9 dB

Abbildung B.1: RNN: Bitfehlerkurven und zugehörige EXIT-Charts an zwei Stellen von E_b/N_0 für die Kanäle A0303 und B0303 und den Kanalcode mit $r = 1/2, m = 2, [7, 5]_8$

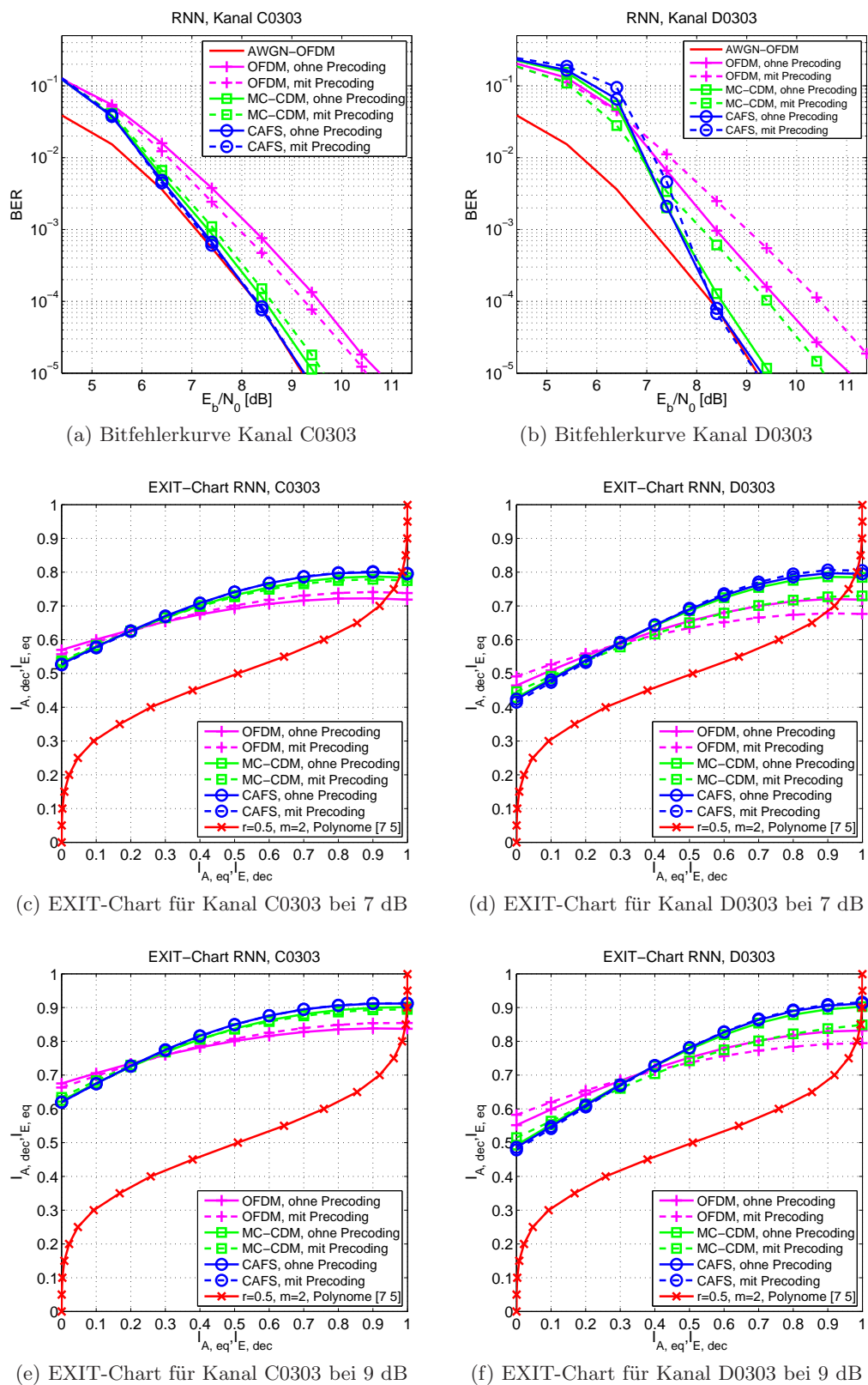
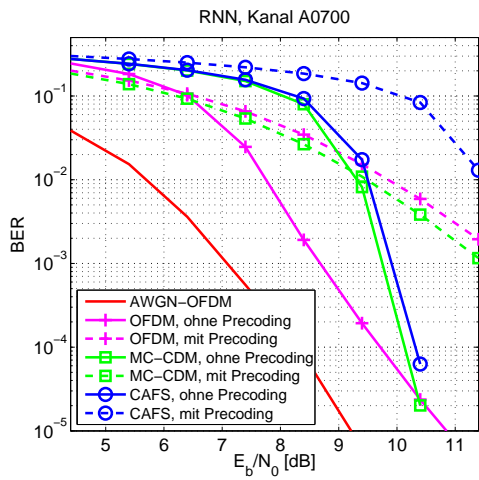
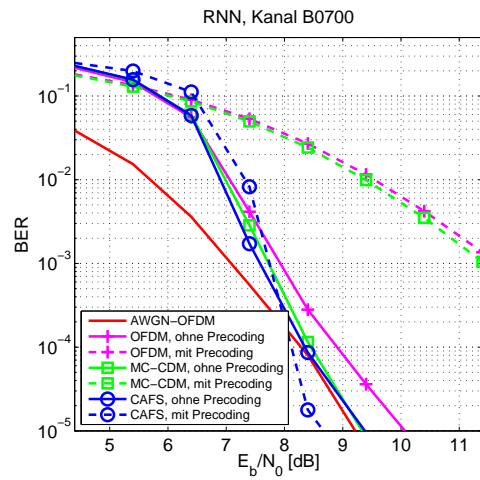


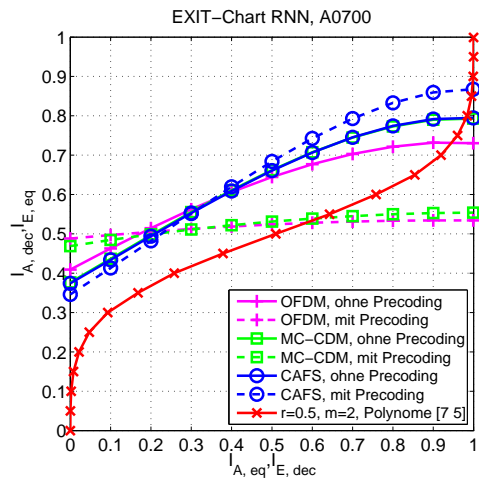
Abbildung B.2: RNN: Bitfehlerkurven und zugehörige EXIT-Charts an zwei Stellen von E_b/N_0 für die Kanäle C0303 und D0303 und den Kanalcode mit $r = 1/2$, $m = 2$, $[7, 5]_8$



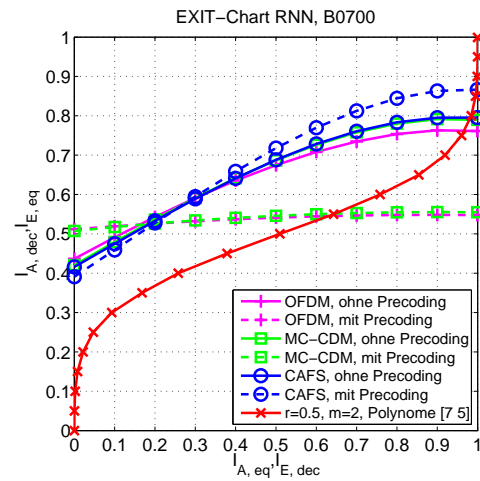
(a) Bitfehlerkurve Kanal A0700



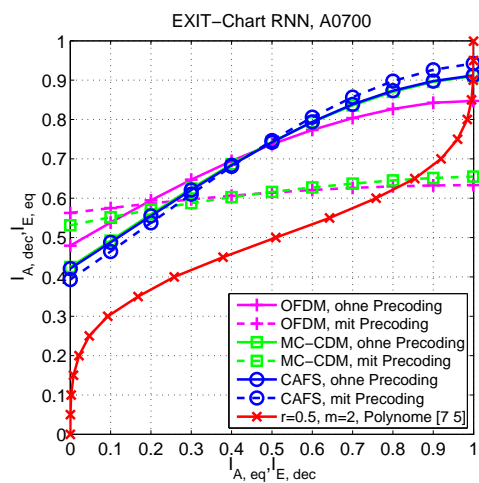
(b) Bitfehlerkurve Kanal B0700



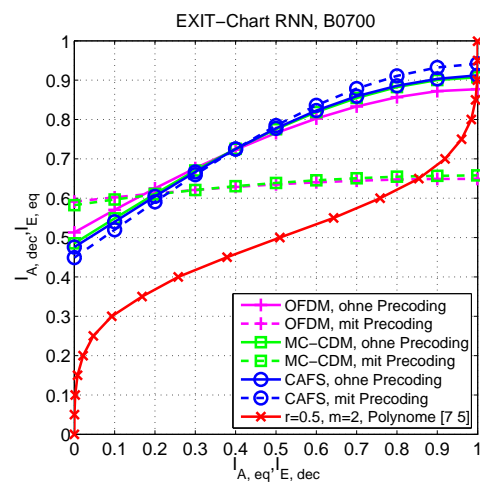
(c) EXIT-Chart für Kanal A0700 bei 7 dB



(d) EXIT-Chart für Kanal B0700 bei 7 dB



(e) EXIT-Chart für Kanal A0700 bei 9 dB



(f) EXIT-Chart für Kanal B0700 bei 9 dB

Abbildung B.3: RNN: Bitfehlerkurven und zugehörige EXIT-Charts an zwei Stellen von E_b/N_0 für die Kanäle A0700 und B0700 und den Kanalcode mit $r = 1/2, m = 2, [7, 5]_8$

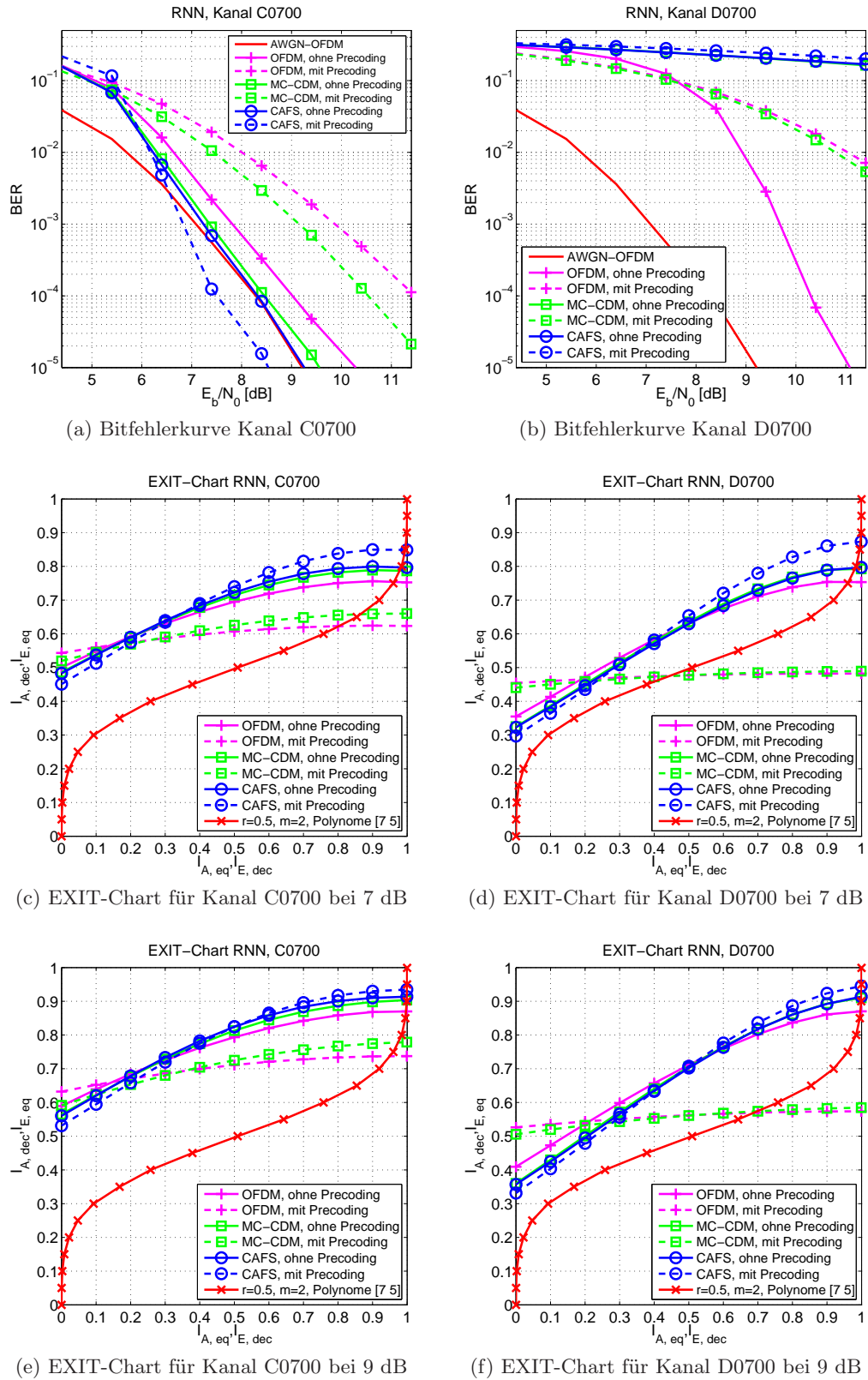
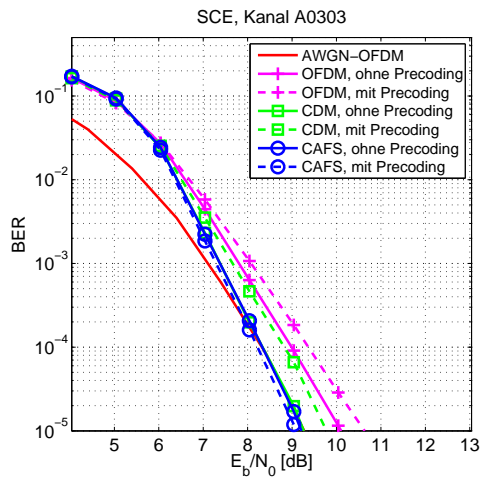
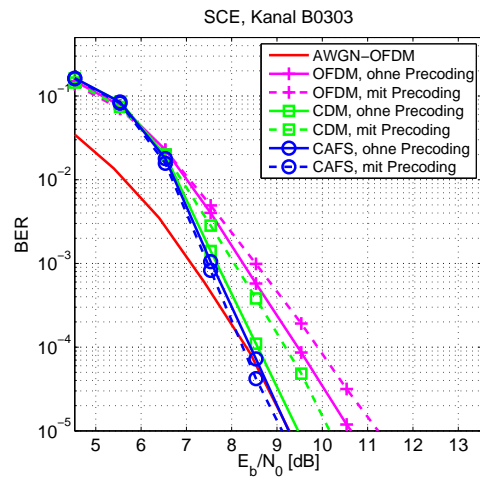


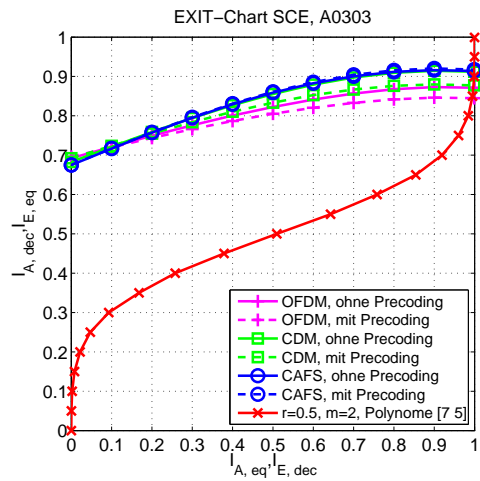
Abbildung B.4: RNN: Bitfehlerkurven und zugehörige EXIT-Charts an zwei Stellen von E_b/N_0 für die Kanäle C0700 und D0700 und den Kanalcode mit $r = 1/2$, $m = 2$, $[7, 5]_8$



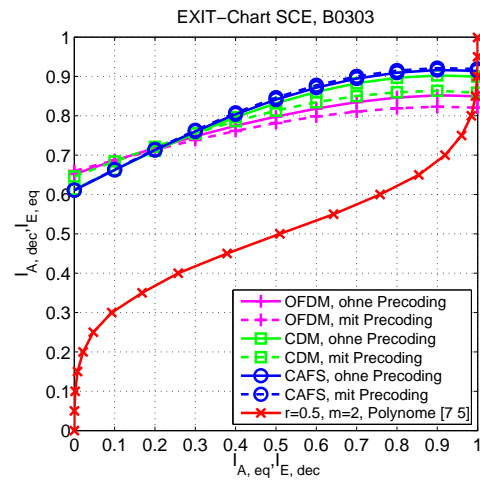
(a) Bitfehlerkurve Kanal A0303



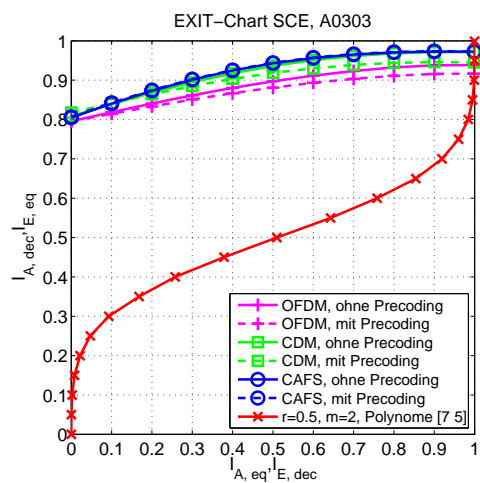
(b) Bitfehlerkurve Kanal B0303



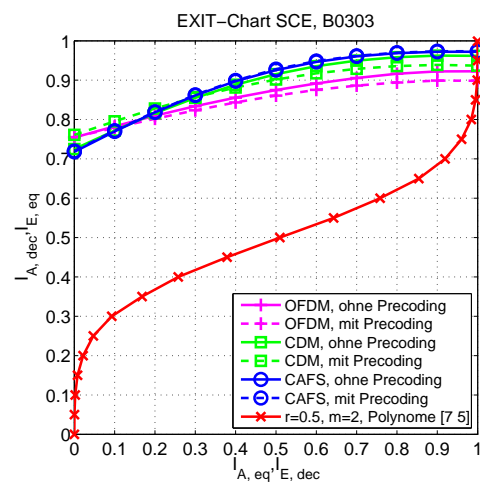
(c) EXIT-Chart für Kanal A0303 bei 7 dB



(d) EXIT-Chart für Kanal B0303 bei 7 dB

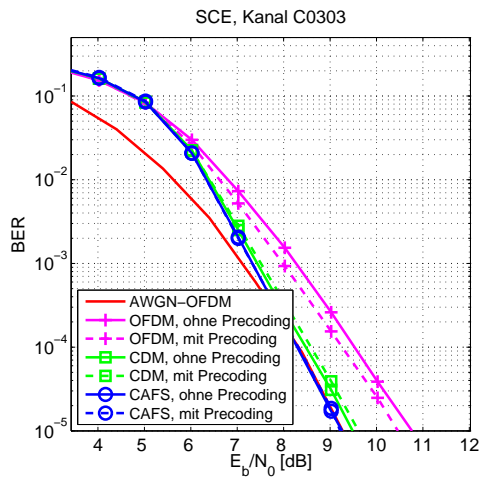


(e) EXIT-Chart für Kanal A0303 bei 9 dB

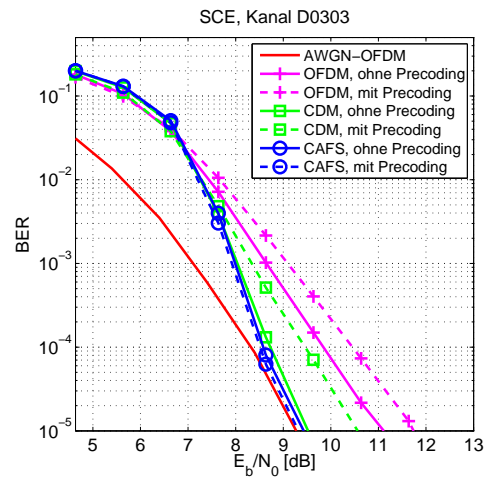


(f) EXIT-Chart für Kanal B0303 bei 9 dB

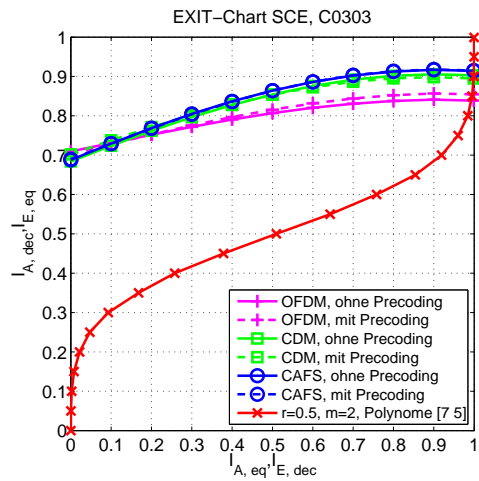
Abbildung B.5: SCE: Bitfehlerkurven und zugehörige EXIT-Charts an zwei Stellen von E_b/N_0 für die Kanäle A0303 und B0303 und den Kanalcode mit $r = 1/2$, $m = 2$, $[7, 5]_8$



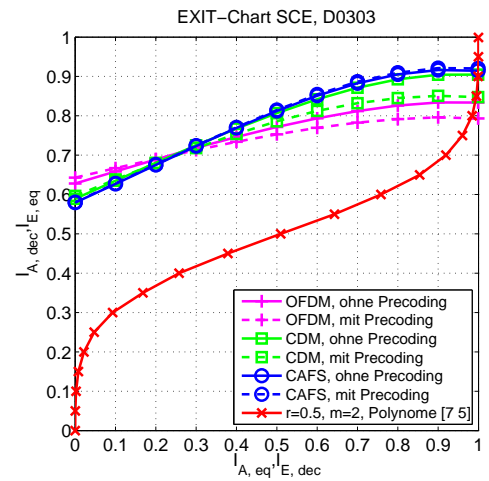
(a) Bitfehlerkurve Kanal C0303



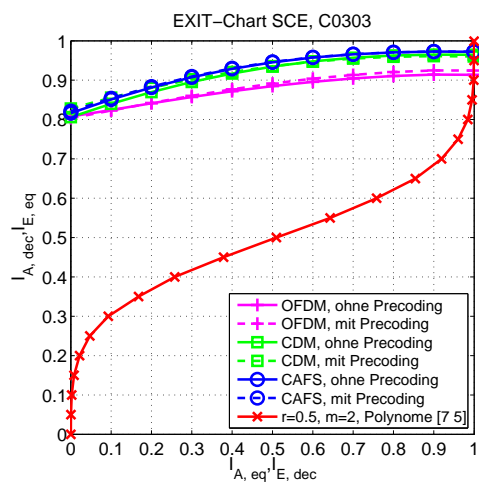
(b) Bitfehlerkurve Kanal D0303



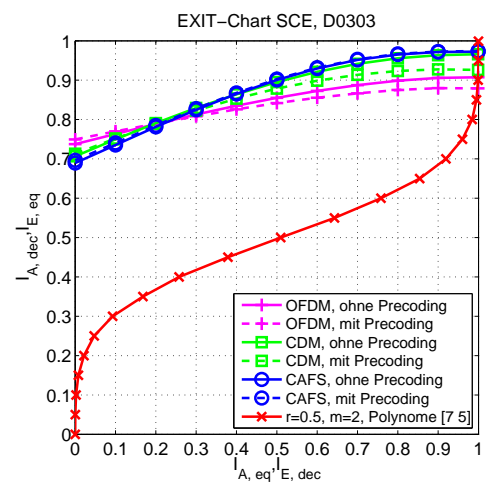
(c) EXIT-Chart für Kanal C0303 bei 7 dB



(d) EXIT-Chart für Kanal D0303 bei 7 dB



(e) EXIT-Chart für Kanal C0303 bei 9 dB



(f) EXIT-Chart für Kanal D0303 bei 9 dB

Abbildung B.6: SCE: Bitfehlerkurven und zugehörige EXIT-Charts an zwei Stellen von E_b/N_0 für die Kanäle C0303 und D0303 und den Kanalcode mit $r = 1/2$, $m = 2$, $[7, 5]_8$

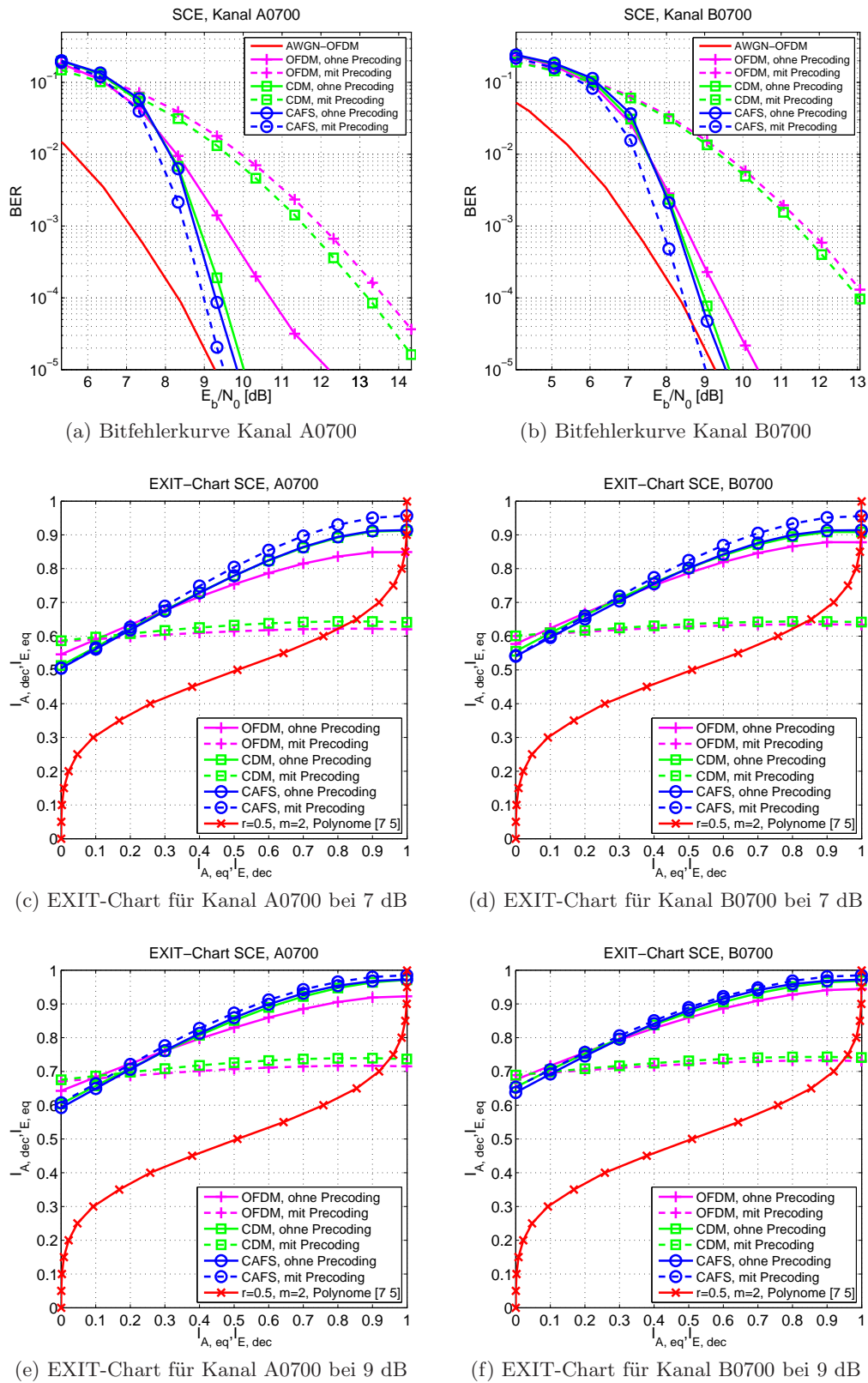


Abbildung B.7: SCE: Bitfehlerkurven und zugehörige EXIT-Charts an zwei Stellen von E_b/N_0 für die Kanäle A0700 und B0700 und den Kanalcode mit $r = 1/2$, $m = 2$, $[7, 5]_8$

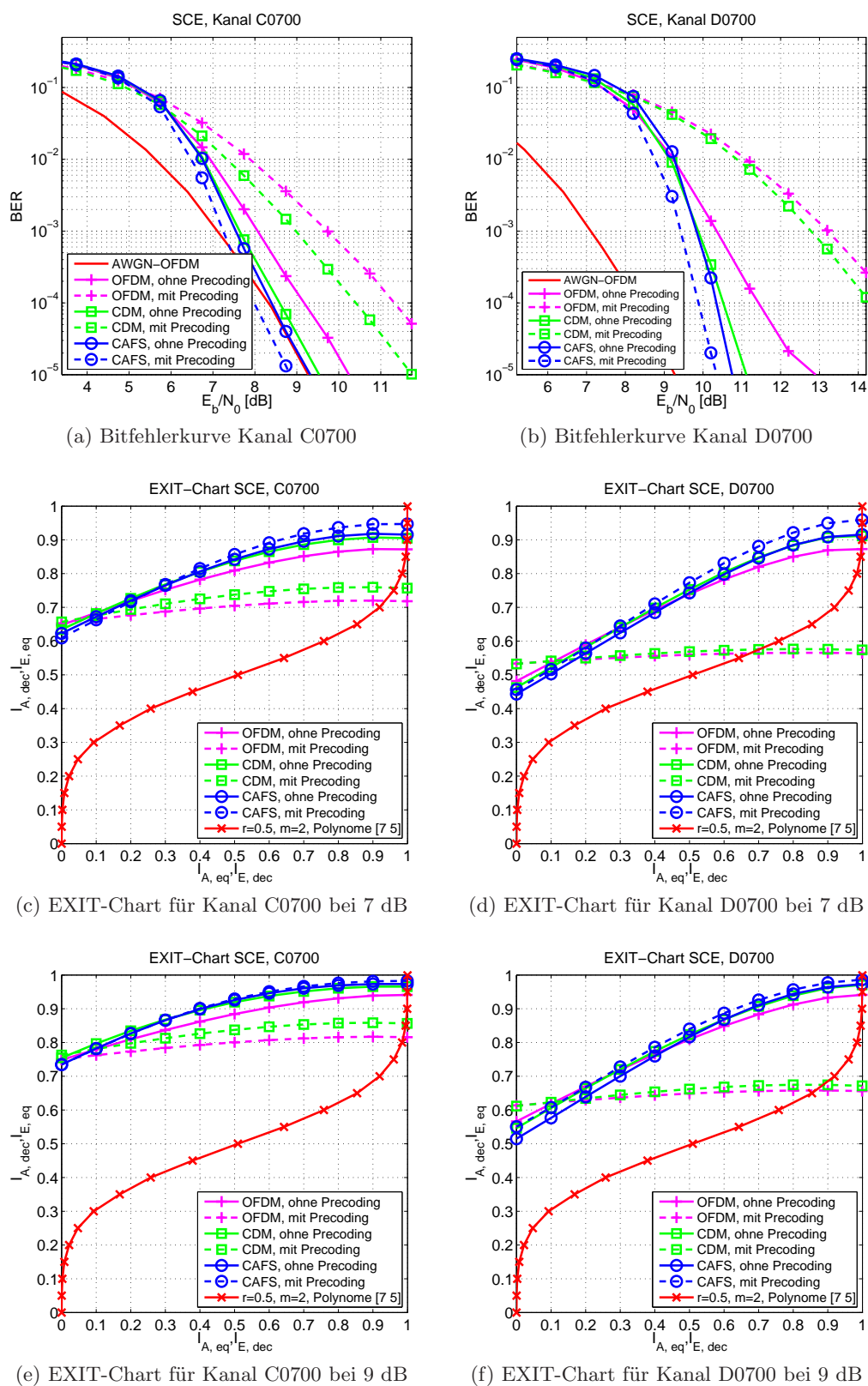


Abbildung B.8: SCE: Bitfehlerkurven und zugehörige EXIT-Charts an zwei Stellen von E_b/N_0 für die Kanäle C0700 und D0700 und den Kanalcode mit $r = 1/2$, $m = 2$, $[7, 5]_8$

Abbildungsverzeichnis

2.1	Modell des MIMO-Kanals	4
2.2	Systemmodell für vektorwertige Übertragung mit linearen Modulationsverfahren	4
2.3	MIMO-Kanal, modale Zerlegung	7
2.4	OFDM-Elementarsignale im Frequenzbereich	9
2.5	Übertragungsmodell für SISO-OFDM	10
2.6	Ersatzkanalmatrix für SISO-OFDM	11
2.7	Ersatzkanalmatrix für MIMO-OFDM	12
3.1	Blockschaltbild des Block Linear Equalizers	14
3.2	Blockschaltbild des Block Decision Feedback Equalizers	16
3.3	Blockschaltbild des zerlegten Ersatzkanals mit Vorwärtsfilter	17
3.4	Blockschaltbild des RNN-Entzerrers	19
3.5	Blockschaltbild des äquivalenten Cholesky-zerlegten Ersatzkanals	21
3.6	SCE: Updating der geschätzten Symbole und Modell der Entscheidung	23
4.1	Blockschaltbild eines Faltungscodierers	26
4.2	Blockschaltbild von gemeinsamer Entzerrung und Decodierung	31
5.1	Der gespreizte Ersatzkanal	36
5.2	MC-CDM-Spreading: Beispiel für \underline{R}_{MO} , \underline{U}_{MC-CDM} und $\underline{R}_{MC-CDM}(0)$	38
5.3	MC-CAFS-Spreading: Beispiel für \underline{R}_{MO} , $\underline{U}_{MC-CAFS}$ und $\underline{R}_{MC-CAFS}(0)$	39
5.4	Beträge der Diagonalelemente der Kanalmatrix für MC-CAFS und MC-CDM	39
5.5	Bitfehlerrate in Abhängigkeit von E_b/N_0 für gespreizte uncodierte Übertragung	41
5.6	Precoding im Systemmodell	41
5.7	Ersatzkanalmatrizen mit und ohne Precoding	44
5.8	Beträge der Hauptdiagonalelemente der Kanalmatrizen	44
5.9	Bitfehlerrate in Abhängigkeit von E_b/N_0 für uncod. Übertragung mit Precoding	47
5.10	Bitfehlerrate in Abhängigkeit von den Korrelationen ρ_T	48
5.11	Turbo-Entzerrer: Bereiche der Bitfehlerkurven	50
5.12	Bei EXIT-Chart-Analyse werden die Komponenten des Turbo-Entzerrers und deren Ein- und Ausgänge getrennt untersucht.	51
5.13	Die aus den Histogrammen berechneten Verläufe werden in ein Koordinatensystem eingetragen.	55
5.14	Beispiel für die EXIT-Chart-Darstellung mit idealer Zustandskurve	57
5.15	EXIT-Charts für Codes mit Rate $r = 1/2$ (a) bzw. Rate $r = 1/3$ (b) für verschiedene Gedächtnislängen m	62
5.16	RNN: Bitfehlerkurven und EXIT-Charts für die Kanäle A0707 und B0707	64
5.17	RNN: Bitfehlerkurven und EXIT-Charts für die Kanäle C0707 und D0707	65

5.18	SCE: Bitfehlerkurven und EXIT-Charts für die Kanäle A0707 und B0707 . .	67
5.19	SCE: Bitfehlerkurven und EXIT-Charts für die Kanäle C0707 und D0707 . .	68
5.20	EXIT-Charts mit aus Daten der Bitfehlersimulationen berechnetem Verlauf der Zustandskurve	70
5.21	Zeitvariante stochastische Kanäle: Bitfehlerkurven für RNN und SCE	71
5.22	Zeitvariante stochastische Kanäle: Bitfehlerkurven für verschiedenen Iterati- onsschemata und Kanalcodes	72
A.1	Prinzip des Waterpouring-Algorithmus	78
B.1	RNN: Bitfehlerkurven und EXIT-Charts für die Kanäle A0303 und B0303 . .	80
B.2	RNN: Bitfehlerkurven und EXIT-Charts für die Kanäle C0303 und D0303 . .	81
B.3	RNN: Bitfehlerkurven und EXIT-Charts für die Kanäle A0700 und B0700 . .	82
B.4	RNN: Bitfehlerkurven und EXIT-Charts für die Kanäle C0700 und D0700 . .	83
B.5	SCE: Bitfehlerkurven und EXIT-Charts für die Kanäle A0303 und B0303 . .	84
B.6	SCE: Bitfehlerkurven und EXIT-Charts für die Kanäle C0303 und D0303 . .	85
B.7	SCE: Bitfehlerkurven und EXIT-Charts für die Kanäle A0700 und B0700 . .	86
B.8	SCE: Bitfehlerkurven und EXIT-Charts für die Kanäle C0700 und D0700 . .	87

Tabellenverzeichnis

3.1	Vergleich grundlegender Eigenschaften der Entzerrer-Konzepte	13
5.1	Wertetabelle für Standardabweichung $\sigma_A = f(I_A)$	54
5.2	Statistische Übersicht über charakteristische Eigenschaften von MIMO-OFDM-Kanälen, Stichprobe von jeweils $n = 10000$ Kanälen	61
5.3	Übersicht über die verwendeten Testkanäle	61

Abkürzungs- und Symbolverzeichnis

γ_i	Leistungszuteilungskoeffizient zu Eigenmode i
λ_i	i -ter Eigenwert einer Matrix
$\underline{\Phi}$	Kovarianzmatrix
ρ	Korrelationskoeffizient zweier benachbarter Antennen bei exponentiellem Modell
$\Theta(\cdot)$	Weiche Entscheidungsfunktion beim RNN
$\underline{\xi}$	Äquivalenter Empfangsvektor nach <i>pre-whitening</i> beim SCE
\otimes	Kronecker-Produkt
$\underline{0}_{m \times n}$	Nullmatrix der Grösse $m \times n$
$\underline{1}_{m \times n}$	Matrix, deren Elemente alle gleich eins sind der Grösse $m \times n$
$\underline{I}_{m \times n}$	Einheitsmatrix der Grösse $m \times n$
$\underline{\mathcal{A}}$	Modulationsalphabet
\mathcal{B}	Mappingfunktion zwischen Bits und Symbolen
\mathcal{C}	Kanalcode
$\text{chol}\{\cdot\}$	Cholesky-Zerlegung einer Matrix
D	Speicherregister bzw. Verzögerung
$\text{diag}\{\cdot\}$	Diagonalelemente einer Matrix
E_b	Energie pro Bit
E_s	Energie pro Symbol
η	Iterationsindex
$E\{\cdot\}$	Erwartungswert
$\text{eig}\{\cdot\}$	Eigenwerte einer Matrix
$\underline{\underline{F}}$	Cholesky-Zerlegung der Ersatzkanalmatrix
$(\cdot)^H$	konjugiert komplex transponiert
\underline{h}	Kanalmatrix im Zeitbereich
\underline{h}_w	unkorrelierte Kanalmatrix für Kronecker-Korrelationsmodell
I_A	Wechselseitige Information am Eingang des Entzerrers/Decodierers
I_E	Wechselseitige Information am Ausgang des Entzerrers/Decodierers
\underline{k}_R	Antennenkorrelationsmatrix am Empfänger
\underline{k}_T	Antennenkorrelationsmatrix am Sender
L	Anzahl der Pfade in einer Kanalstosantwort
\underline{L}	Log-Likelihood-Verhältnis, L-Wert
\underline{L}'_D	L-Werte am Ausgang des Decodierers
\underline{L}'_{eD}	Differenz der L-Werte am Ausgang und Eingang des Decodierers
\underline{L}'_{eE}	L-Werte am Eingang des Decodierers
\underline{L}_E	L-Werte am Ausgang des Entzerrers
\underline{L}_a	a-priori-L-Wert
\underline{L}_{ch}	L-Wert des Kanals

$\underline{L}_{\text{ext}}$	extrinsischer L-Wert
\underline{L}_{eD}	L-Werte am Eingang des Entzerrers
\underline{L}_{eE}	Differenz der L-Werte am Ausgang und Eingang des Entzerrers
M	Große des Modulationsalphabets
m	Gedächtnislänge eines Kanalcodes
N	Anzahl der OFDM-Subträger
N_0	Rauschleistungsdichte
$\underline{\tilde{n}}$	farbiger Rauschvektor
\underline{n}	AWGN-Rauschvektor
n_R	Anzahl der Empfangsantennen
n_T	Anzahl der Sendeantennen
P	Wahrscheinlichkeit
\underline{P}	Punktierungsmatrix (engl. <i>puncturing matrix</i>)
$\underline{\Pi}$	Interleaver
$p(\cdot)$	Wahrscheinlichkeitsdichtefunktion
$\underline{R}_{\text{MO}}$	MIMO-OFDM-Ersatzkanalmatrix
$\underline{R}_{\text{SO}}$	SISO-OFDM-Ersatzkanalmatrix
\underline{r}	Ersatzkanalmatrix im Zeitbereich
r	Coderate
$\text{sgn}\{\cdot\}$	Signum- bzw. Vorzeichenfunktion
$\text{Tr}\{\cdot\}$	Spur einer Matrix
\underline{U}	Spreizmatrix
$\underline{\underline{WH}}(n)$	Walsh-Hadamard-Matrix der Ordnung n
\underline{W}	Filtermatrix
\check{x}	weich entschiedenes Symbol
\hat{x}	Entschiedenenes Symbol
\tilde{x}	gefiltertes Symbol
\underline{x}	Matrix im Zeitbereich
$\underline{\underline{x}}$	Vektor im Zeitbereich
\underline{y}	Empfangsvektor nach dem Matched Filter
APP	engl. <i>a-posteriori probability</i>
AWGN	engl. <i>Additive White Gaussian Noise</i>
BCJR	Algorithmus zur Kanaldecodierung nach Bahl, Cocke, Jelinek und Raviv
BDFE	engl. <i>Block Decision Feedback Equalizer</i>
BER	Bitfehlerrate (engl. <i>Additive White Gaussian Noise</i>)
BLE	engl. <i>Block Linear Equalizer</i>
COD	Kanalcodierung
CP	engl. <i>cyclic prefix</i>
CSIT	Informationen über den Kanalzustand am Sender (engl. <i>Channel State Information at the Transmitter</i>)
DET	Detektion: Entzerrung/Kanaldecodierung
DFT	Diskrete Fouriertransformation (engl. <i>Discrete Fourier Transform</i>)
EXIT	engl. <i>Extrinsic Information Transfer</i>
FDD	Frequenzduplex (engl. <i>Frequency Division Duplex</i>)
FFT	schnelle Fouriertransformation (engl. <i>Fast Fourier Transform</i>)
HD	Harte Entscheidung (engl. <i>hard decision</i>)
IDFT	Inverse diskrete Fouriertransformation (engl. <i>Inverse Discrete Fourier Transform</i>)

IFFT	Inverse schnelle Fouriertransformation (engl. <i>Inverse Fast Fourier Transform</i>)
ISCI	Intersubkanal-Interferenz (engl. <i>Intersubchannel Interference</i>)
MAP	engl. <i>Decodierprinzip Maximum a posteriori</i>
MC-CAFS	engl. <i>Multicarrier Cyclic Antenna Frequency Spreading</i>
MC-CDM	engl. <i>Multicarrier Code Division Multiplex</i>
MIMO	engl. <i>Multiple Input Multiple Output</i>
ML	engl. <i>Maximum Likelihood</i>
MMSE	engl. <i>Minimum Mean Square Error</i>
MRC	engl. <i>Maximum Ratio Combining</i>
OFDM	engl. <i>Orthogonal Frequency Division Multiplexing</i>
PEP	Paarweise Bitfehlerwahrscheinlichkeit (engl. <i>Pairwise Bit Error Probability</i>)
PSK	engl. <i>Phase Shift Keying</i>
RNN	engl. <i>Recurrent Neural Network</i>
s/s-MAP	engl. <i>Symbolweise Maximum-a-posteriori-Decodierung</i>
SCE	engl. <i>Iterative Soft Cholesky Block Decision Feedback Equalizer</i>
SNR	Signal-/Rauschleistungsverhältnis (engl. <i>Signal to Noise Ratio</i>)
STBC	engl. <i>Space-Time Block Code</i>
SVD	Singulärwertzerlegung (engl. <i>Singular Value Decomposition</i>)
TDD	Zeitduplex (engl. <i>Time Division Duplex</i>)
ZF	engl. <i>Zero Forcing</i>

Literaturverzeichnis

- [1] BOSSERT, M.: *Kanalcodierung*. Teubner Verlag, Stuttgart, 2. Aufl., 1998.
- [2] BOUTROS, J. und E. VITERBO: *Signal space diversity: a power- and bandwidth-efficient diversity technique for the Rayleigh fading channel*. IEEE Trans. on Information Theory, 44(4):1453–1467, Juli 1998.
- [3] BOYD, S. und L. VANDENBERGHE: *Convex Optimization*. Cambridge University Press, Cambridge, 1. Aufl., 2004.
- [4] BRONSTEIN, I. N., K. A. SEMENDJAJEW, G. MUSIOL und H. MÜHLIG: *Taschenbuch der Mathematik*. Verlag Harri Deutsch, Frankfurt am Main, 5. Aufl., 2001.
- [5] BURY, A., J. EGLE und J. LINDNER: *Diversity Comparison of Spreading Transforms for Multicarrier Spread Spectrum Transmission*. IEEE Trans. on Communications, 51(5):774–781, Mai 2003.
- [6] COVER, T. M. und J. A. THOMAS: *Elements of Information Theory*. Wiley, New York, 1991.
- [7] DANGL, M. A., C. SGRAJA, W. G. TEICH und J. LINDNER: *Convergence Behavior of Iterative Equalization and Decoding Schemes with Memory*. In: *IEEE Global Communications Conference*, S. 1710–1715, San Francisco, USA, Dezember 2003.
- [8] DANGL, M. A., W. G. TEICH und J. LINDNER: *Joint Iterative Equalization, Demapping and Decoding with a Soft Interference Canceler*. In: *7th International Symposium on Communication Theory and Applications (ISCTA '03)*, S. 36–41, Ambleside, UK, Juli 2003.
- [9] DANGL, M. A., D. YACOU, U. MARXMEIER, W. G. TEICH und J. LINDNER: *Performance of Joint Detection Techniques for Coded MIMO-OFDM and MIMO-MC-CDM*. In: *Proc. COST 273 Workshop*, S. 17/1–17/6, Paris, Frankreich, Mai 2003.
- [10] DOUILLARD, C., M. JÉZÉQUEL, C. BERROU, A. PICART, P. DIDIER und A. GLAVIEUX: *Iterative Correction of Intersymbol Interference: Turbo-Equalization*. European Trans. on Telecommunications, 6(5):507–511, September 1995.
- [11] EGLE, J.: *Detection methods for power and bandwidth efficient single carrier block transmissions*. Fortschritt-Berichte. VDI-Verlag, Düsseldorf, 2004. Dissertation.
- [12] EGLE, J. und J. LINDNER: *Iterative Joint Equalization and Decoding based on Soft Cholesky Equalization for General Complex Valued Modulation Symbols*. In: *6th Int. Symposium on DSP for Communication Systems (DSPCS '02)*, S. 163–170, Sydney-Manly, Januar 2002.

- [13] EGGLE, J., C. SGRAJA und J. LINDNER: *Iterative Soft Cholesky Block Decision Feedback Equalizer - a Promising Approach to Combat Interference*. In: *Proc. IEEE VTC*, Bd. 3, S. 1604–1608, Rhodos, Griechenland, Mai 2001.
- [14] ENGELHART, A.: *Vector Detection Techniques with Moderate Complexity*. Fortschritt-Berichte. VDI-Verlag, Düsseldorf, 2003. Dissertation.
- [15] ENGELHART, A., W. G. TEICH, J. LINDNER, G. JENEY, S. IMRE und L. PAP: *A Survey of Multiuser/Multisubchannel Detection Schemes Based on Recurrent Neural Networks*. *Wireless Communications and Mobile Computing*, 2(3):269–284, Februar 2002.
- [16] FOSCHINI, G. J. und M. J. GANS: *On Limits of Wireless Communications in a Fading Environment when Using Multiple Antennas*. *Wireless Personal Communications*, 6(3):311–335, März 1998.
- [17] HAGENAUER, J., E. OFFER und L. PAPKE: *Iterative decoding of binary block and convolutional codes*. *IEEE Trans. on Information Theory*, 42(2):429–445, März 1996.
- [18] HOLTER, B.: *On the capacity of the MIMO channel : A tutorial introduction*. In: *Proc. IEEE Norwegian Symposium on Signal Processing*, S. 167–172, Trondheim, Oktober 2001.
- [19] JAFAR, S. A., S. VISHWANATH und A. GOLDSMITH: *Channel capacity and beamforming for multiple transmit and receive antennas with covariance feedback*. In: *Proc. IEEE ICC*, Bd. 7, S. 2266–2270, Helsinki, Finnland, 2001.
- [20] JAFARKHANI, H.: *Space-Time Coding: Theory and Practice*. Cambridge University Press, Cambridge, 1. Aufl., 2005.
- [21] JOHANNESSON, R.: *Informationstheorie. Grundlage der (Tele-)Kommunikation*. Addison Wesley Verlag, 1992.
- [22] KYRITSI, P., D. C. COX, R. A. VALENZUELA und P. W. WOLNIANSKY: *Correlation Analysis Based on MIMO Channel Measurements in an Indoor Environment*. *IEEE Journal on Selected Areas in Communications*, 21(5):713–720, 2003.
- [23] LINDNER, J.: *Informationsübertragung. Grundlagen der Kommunikationstechnik*. Springer-Verlag, Berlin, Heidelberg, 1. Aufl., 2005.
- [24] LOYKA, S. L.: *Channel capacity of MIMO architecture using the exponential correlation matrix*. *IEEE Communications Letters*, 5(9):369–371, September 2001.
- [25] PALOMAR, D. P., J. M. CIOFFI und M. A. LAGUNAS: *Joint Tx-Rx Beamforming Design for Multicarrier MIMO Channels: A Unified Framework for Convex Optimization*. *IEEE Transactions on Signal Processing*, 51(9):2381–2401, September 2003.
- [26] PAULRAJ, A., R. NABAR und D. GORE: *Introduction to Space-Time Wireless Communications*. Cambridge University Press, Cambridge, 1. Aufl., 2003.
- [27] PROAKIS, J. G.: *Digital Communications*. McGraw-Hill, New York, 4. Aufl., 2001.
- [28] SAMPATH, H. und A. PAULRAJ: *Linear precoding for space-time coded systems with known fading correlations*. *IEEE Communications Letters*, 6:239–241, Juni 2002.

- [29] SCHMIDT, G.: *Advanced Channel Coding. Lecture Notes.*, 2007. Institut für Telekommunikationstechnik und Angewandte Informationstheorie, Universität Ulm. Erster Entwurf.
- [30] SGRAJA, C.: *Kombinierte Entzerrung und Decodierung bei Paketübertragungsverfahren mit allgemeinen komplexwertigen Modulationsverfahren.* Diplomarbeit, Universität Ulm, Abteilung Informationstechnik, Dezember 1998.
- [31] SGRAJA, C., A. ENGELHART, W. G. TEICH und J. LINDNER: *Combined equalization and decoding for general BFDm packet transmission schemes.* In: *1st International OFDM-Workshop*, Hamburg, September 1999.
- [32] SGRAJA, C., A. ENGELHART, W. G. TEICH und J. LINDNER: *Equalization with recurrent neural networks for complex-valued modulation schemes.* In: *3. Workshop Kommunikationstechnik*, S. 7–12, Schloss Reisingburg, Juli 1999.
- [33] TEICH, W. G. und M. SEIDL: *Code Division Multiple Access Communications: Multiuser Detection based on a Recurrent Neural Network Structure.* In: *Proc. IEEE ISSSTA*, Bd. 3, S. 979–984, Mainz, September 1996.
- [34] TELATAR, I. E.: *Capacity of Multi-antenna Gaussian Channels.* *European Transactions on Telecommunications*, 10(6):585–595, November 1999.
- [35] TEN BRINK, S.: *Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes.* *IEEE Trans. on Communications*, 49(10):1727–1737, Oktober 2001.
- [36] VU, M. und A. PAULRAJ: *MIMO Wireless Precoding.* Angenommen von *IEEE Signal Processing Magazine*. voraussichtlich September 2007.
- [37] VU, M. und A. PAULRAJ: *Optimal Linear Precoders for MIMO Wireless Correlated Channels With Nonzero Mean in Space-Time Coded Systems.* *IEEE Trans. on Signal Processing*, 54(6):2318–2332, Juni 2006.
- [38] XIA, P., S. ZHOU und G. B. GIANNAKIS: *Adaptive MIMO-OFDM Based on partial Channel State Information.* *IEEE Trans. on Signal Processing*, 52(1):202–213, Januar 2004.
- [39] YACOUB, D., C. SCHNEIDER, S. WARZUEGEL, W. TEICH, R. THOMÄ und J. LINDNER: *Capacity of Measured MIMO Channels in Dependence of Array Element Spacing and Distance between Antennas.* *IEEE Workshop on Smart Antennas 2006*, Reisingburg, März 2006.
- [40] YACOUB, D. Y., M. A. DANGL, W. G. TEICH und J. LINDNER: *Comparison of Iterative Detection Schemes for MIMO Systems with Spreading Based on OFDM.* In: *4th International Workshop on Multi-Carrier Spread Spectrum (MC-SS)*, Oberpfaffenhofen, September 2003.
- [41] YACOUB, D. Y., W. G. TEICH und J. LINDNER: *MC-Cyclic Antenna Frequency Spread: A Novel Space-Frequency Spreading for MIMO-OFDM.* In: *8th International Symposium on Communication Theory and Applications (ISCTA '05)*, S. 264–269, Ambleside, UK, Juli 2005.

- [42] YACOUB, D. Y., W. G. TEICH und J. LINDNER: *MC-Cyclic Antenna Frequency Spread: Effect of Interference on the Performance of a Spread MIMO-OFDM System*. In: *10th International OFDM-Workshop 2006*, S. 323–327, Hamburg, August 2005.
- [43] YACOUB, D. Y., W. G. TEICH und J. LINDNER: *Effect of Antenna Correlations on Interference and Performance of a Spread MIMO-OFDM System (MC-CAFS)*. In: *11th International OFDM-Workshop 2006*, S. 111–115, Hamburg, August 2006.
- [44] YACOUB, D. Y., W. G. TEICH und J. LINDNER: *Rotated MC-Cyclic Antenna Frequency Spread: Effect of Rotations in Correlated MIMO-OFDM Systems*. In: *7th IEEE International Workshop on Signal Processing Advances for Wireless Communications (SPAWC 2006)*, Cannes, Frankreich, Juli 2006.
- [45] YOON, E., J. HANSEN und A. PAULRAJ: *Space-Frequency Precoding for an OFDM Based System Exploiting Spatial and Path Correlation*. In: *Proc. IEEE GLOBECOM*, S. 436–440, Dezember 2004.
- [46] ZURMÜHL, R.: *Matrizen und ihre technischen Anwendungen*. Springer-Verlag, Berlin, Heidelberg, 4. Aufl., 1964.