

Systementwurf mit Zuverlässigkeitsinformation für mobile Festplatten mit hoher Datendichte

Dissertation
Stefan Schmermbeck

Genehmigte Dissertation zur
Erlangung des akademischen Grades eines Doktors an der
Fakultät für Elektrotechnik und Informationstechnik
Universität Dortmund

Lehrstuhl für Datenverarbeitungssysteme

Dortmund, 14. Juli 2004

1. Gutachter:	Prof. Dr. Uwe Schwiegelshohn
2. Gutachter:	Prof. Dr. Norbert Wehn
Tag der Einreichung:	30. April 2004
Tag der mündlichen Prüfung:	14. Juli 2004

Danksagung

An dieser Stelle möchte ich allen Personen danken, die mich bei der Entstehung dieser Dissertation unterstützt haben.

Insbesondere möchte ich meinem Doktorvater, Prof. Dr. Uwe Schwiigelshohn, für die wissenschaftliche Anleitung und für viele stimulierende Diskussionen meinen großen Dank aussprechen. Ferner bedanke ich mich bei Prof. Dr. Norbert Wehn, dem Co-Referenten meiner Dissertation, für seine Unterstützung.

Die hilfsbereite und inspirierende Atmosphäre am Lehrstuhl für Datenverarbeitungssysteme hat ebenfalls zum Gelingen dieser Arbeit beigetragen. Daher möchte ich allen Mitarbeitern meinen großen Dank aussprechen. Herausheben möchte ich hierbei die sehr erfolgreiche Zusammenarbeit mit Ingo Dahm, Edwin Naroska und Guido Stromberg.

Darüber hinaus möchte ich Yu-hong Chen, Jörg Platte, Tim Schleginski und Yanqi Zu für ihre wertvollen Beiträge danken, die sie als studentische Hilfskräfte oder im Rahmen von Studien- und Diplomarbeiten geleistet haben.

Ein großer Dank geht ebenfalls an Martin Hassner und Greg Silvus für interessante Diskussionen und Anregungen.

Abschließend möchte ich an dieser Stelle meine Familie und meine Freunde hervorheben, deren Unterstützung und Geduld eine wertvolle Hilfe bei der Erstellung dieser Arbeit waren. Vielen Dank!

Kurzfassung

Der zunehmende Einsatz von Festplatten in mobilen Geräten erfordert neue Entwurfsmethoden. Zu den Zielen bei der Entwicklung einer Festplatte für den stationären Betrieb, wie Signalzuverlässigkeit, Speicherkapazität, Datenübertragungsrate und Kosten, kommt mit der Verlustleistung der eingesetzten Schaltungen ein weiterer Parameter hinzu. In dieser Arbeit wird ein Konzept für den Entwurf der signalverarbeitenden Komponenten einer mobilen Festplatte, dem so genannten Lesekanal, präsentiert. Da die Entwicklungsziele im Konflikt zueinander stehen, wird ein systemübergreifender Ansatz gewählt, bei dem die Abhängigkeiten zwischen den unterschiedlichen Zielen berücksichtigt werden.

Um wachsenden Anforderungen an die Signalzuverlässigkeit aufgrund immens steigender Datendichten gerecht zu werden, wird bei den eingesetzten Verfahren die so genannte Zuverlässigkeitinformation verwendet. Diese während der Detektion berechnete Zusatzinformation ist ein Maß für die Fehlerwahrscheinlichkeit der gelesenen Daten. Mit ihrer Hilfe kann die Leistung der eingesetzten Algorithmen verbessert und dadurch die Fehlerrate deutlich abgesenkt werden. Der Nachteil dieses Verfahrens liegt jedoch in der hohen Komplexität der eingesetzten Algorithmen, die darüber hinaus an die Besonderheiten eines magnetischen Kanals angepasst werden müssen.

In dieser Arbeit werden verschiedene Varianten zur Integration von Zuverlässigkeitinformation in einen magnetischen Kanal verfolgt, bei denen die Komplexität der eingesetzten Algorithmen und Schaltungen so gering wie möglich gehalten werden soll. Zu diesem Zweck werden bekannte, verlustleistungsarme Verfahren zum Teil um die Fähigkeit zur Ausgabe oder Weiterverarbeitung von Zuverlässigkeitinformation erweitert oder existierende Verfahren für die Verwendung auf dem magnetischen Kanal optimiert.

Mit einem systemübergreifenden Ansatz wird ein Lesekanal entworfen, bei dem die Komplexitätserhöhung aufgrund der Modifikationen an anderer Stelle durch Vereinfachungen wieder kompensiert wird. Neben dem leistungsfähigen Konzept der Signalraumdetektion, das um die Ausgabe von Zuverlässigkeitinformation erweitert wird, kommen die bei Datenspeichern weit verbreiteten, fehlerkorrigierenden Reed-Solomon Codes zum Einsatz. Ein zusätzlicher, innerer Parity-Check Code dient sowohl als Modulationscode, als auch – in Verbindung mit der Zuverlässigkeitinformation des Detektors – zur Verbesserung der Fehlerkorrektur. Der vorgestellte Kanal erreicht durch diese Maßnahmen eine deutlich niedrigere Fehlerrate. Des Weiteren wird die Eignung für zukünftige mobile Festplatten gezeigt, da der Lesekanal unempfindlich auf die bei hohen Datendichten verstärkt auftretenden Störgrößen, wie zum Beispiel Transitionsrauschen, reagiert.

Inhaltsverzeichnis

Danksagung	iii
Kurzfassung	v
1 Einleitung	1
1.1 Motivation	1
1.2 Ziele der Arbeit	2
1.3 Weiterer Aufbau	4
2 Magnetische Aufzeichnung auf Festplatten	5
2.1 Überblick	5
2.1.1 Aufbau einer Festplatte	5
2.1.2 Komponenten des magnetischen Kanals	6
2.2 Schreibprozess	7
2.2.1 Aufbau und Arbeitsweise des Schreibkopfes	8
2.2.2 Beeinträchtigungen des Schreibprozesses	8
2.3 Leseprozess	10
2.3.1 Aufbau und Arbeitsweise des Lesekopfes	10
2.3.2 Charakterisierung des Lesesignals	11
2.3.3 Beeinträchtigungen des Leseprozesses	13
2.4 Definition des Kanalmodells	14
2.4.1 Auswahl der modellierten Rauschkomponenten	14
2.4.2 Kanalmodell	15
2.5 Zusammenfassung	16

3 Entwurfskriterien	17
3.1 Anforderungen an den magnetischen Kanal	17
3.2 Leistungsarmer Entwurf bei hohen Datendichten	21
3.2.1 Motivation	21
3.2.2 Vorgehensweise	22
3.2.3 Berechnung der Zuverlässigkeitsinformation	23
3.2.4 Lokalisierung von Detektionsfehlern mit Soft-Information	24
3.2.5 Entwicklung eines Soft-In/Soft-Out Modulationsdecoders	25
3.2.6 Auswahl eines Fehlerschutzcodes	25
3.3 Zusammenfassung	27
4 Detektionsverfahren	29
4.1 Detektion auf dem magnetischen Kanal	30
4.2 Partial Response Maximum Likelihood (PRML)	32
4.2.1 Partial Response-Entzerrung	33
4.2.2 Maximum Likelihood Detektion	36
4.3 Signalraumdetektion	41
4.3.1 Fixed Delay Tree Search (FDTS) Detektoren	41
4.3.2 Signalraumdetektoren (SSD)	43
4.4 Soft-Output Signalraumdetektion	45
4.5 Low-Complexity Soft-Output Signalraumdetektoren	46
4.5.1 Berechnung der Zuverlässigkeitsinformation für AWGN	47
4.5.2 Berechnung der Zuverlässigkeitsinformation für farbiges Rauschen	50
4.5.3 Betrachtung konkaver Entscheidungsregionen	51
4.5.4 Berücksichtigung von Fehlerfortpflanzung	53
4.5.5 Detektor-Architektur	54
4.5.6 Adaptiver Signalraumdetektor	56
5 Modulationscodes	59
5.1 Modulationscodes auf dem magnetischen Kanal	60
5.2 Modulationscodes auf Soft-Decoding Kanälen	61
5.2.1 Integration eines Modulationscodes	62

5.2.2	Arbeitsweise eines MTR-Decoders	63
5.2.3	Definition Boolescher Operationen mit Log-Likelihood-Werten . . .	63
5.2.4	Berechnung von Soft-Outputs für Boolesche Operationen	64
5.2.5	Konstruktion des Soft-In/Soft-Out Decoders	69
5.3	Verwendung von Parity-Checks als Modulationscode	70
6	Fehlerkorrigierende Codes	73
6.1	Eigenschaften und Definitionen	74
6.2	Decodierprinzipien	74
6.3	Kriterien zur Auswahl eines geeigneten Codes	76
6.4	Reed-Solomon Codes	77
6.4.1	Erzeugung und Decodierung von Reed-Solomon Codes	78
6.4.2	Erasur-Decodierung von Reed-Solomon Codes	81
6.4.3	Soft-Decodierung mit Hilfe eines inneren Parity-Check Codes . . .	82
6.5	Turbo Codes	85
6.5.1	Das Turbo Prinzip	85
6.5.2	Beschreibung der Encoder und Decoder	88
6.5.3	Eigenschaften der iterativen Decodierung	91
6.5.4	Turbo Codes auf dem magnetischen Kanal	92
6.6	Low-Density Parity-Check (LDPC) Codes	93
6.6.1	Erzeugung und Decodierung von LDPC Codes	93
6.6.2	Implementierungsaspekte	99
6.6.3	LDPC Codes auf dem magnetischen Kanal	99
7	Evaluation verschiedener Lesekanäle	101
7.1	Simulationsumgebung	101
7.2	Betrachtete Komponenten	102
7.3	Untersuchung der Fehlermuster verschiedener Detektoren	103
7.4	Lesekanal mit Reed-Solomon Codes und Hard-Decoding	106
7.4.1	Unbeschränkte Kanäle	106
7.4.2	Modulationscodierte Kanäle	107
7.5	Lesekanäle mit Zuverlässigkeitsinformation	109

7.5.1	Unbeschränkte Kanäle	109
7.5.2	Modulationscodierte Kanäle	115
7.5.3	Systemweite Optimierung	124
7.5.4	Diskussion und Zusammenfassung	126
8	Zusammenfassung und Ausblick	129

Kapitel 1

Einleitung

1.1 Motivation

Moderne Computersysteme erfordern die Möglichkeit zur Speicherung großer Datenmengen. Der rapide Fortschritt der Technologie – zum Beispiel in der Geschwindigkeit moderner Mikroprozessoren – bedingt ebenso einen stark wachsenden Bedarf an Speicherplatz. Die zunehmende Verbreitung des Internet lässt diesen zusätzlich stark ansteigen.

Zur Speicherung großer Datenmengen haben sich magnetische Speicher in den letzten Jahrzehnten als eine effiziente und preisgünstige Lösung erwiesen. Der große Fortschritt in Bezug auf Technologie und Signalverarbeitung hat über viele Jahre einen Anstieg der Speicherkapazität von 60% per annum ermöglicht, der in den vergangenen Jahren sogar auf 100% p.a. gesteigert werden konnte [Gro]. Auch zukünftig ist mit einem weiteren Anstieg des Speicherplatzbedarfs zu rechnen, so dass weitere Innovationen bei der Entwicklung von Festplatten notwendig sind.

Aufgrund des festgelegten Formfaktors kann eine Steigerung der Speicherkapazität nicht durch den Einsatz größerer Magnetscheiben oder durch eine größere Anzahl von Magnetscheiben pro Festplatte erreicht werden. Stattdessen wurde der enorme Fortschritt durch eine permanente Erhöhung der Datendichte erzielt. Die Steigerung der Datendichte ist oftmals auf den Einsatz neuer Technologien zurückzuführen. So konnte beispielsweise durch die Verwendung von magnetoresistiven (MR) Leseköpfen ein deutlich höheres Signal-Rauschleistungs-Verhältnis (SNR) erreicht werden [Tho75]. Zudem hat der dabei erforderliche Einsatz getrennter Schreib- bzw. Leseköpfe eine gezielte Optimierung des Schreib- und Lesevorgangs ermöglicht.

Auch die Signalverarbeitung hat wesentlich zu der Erzielung heutiger Datendichten beigetragen. So gilt zum Beispiel die Einführung von so genannten Partial Response Maximum Likelihood (PRML) Detektoren in den frühen 1990er Jahren als ein Meilenstein bei der Erzielung höherer Datendichten, weil erstmals eine starke Überlagerung (Intersymbol-Interferenz, ISI) der einzelnen Lesepulse zugelassen werden konnte [CDH⁺92].

Neben der Steigerung der Speicherkapazität sind beim Entwurf von Festplatten weitere

Entwicklungsziele zu berücksichtigen. So soll – neben einer möglichst hohen Speicherdichte – auch eine möglichst hohe Datenübertragungsrate erreicht werden. Modernste Lesekanäle erreichen bei der kontinuierlichen Datenrate bereits Spitzenwerte von 1,6 Gbit/s. Des Weiteren ist beim magnetischen Kanal ein extrem hoher Quality-of-Service (QoS) erforderlich. Im Gegensatz zum Mobilfunkkanal, bei dem geringe Störungen des Sprachsignals hinnehmbar sind, ist auf dem magnetischen Kanal eine fehlerfreie Datenübertragung unabdingbar. Bei Festplatten werden im Allgemeinen Bit-Fehlerraten (engl. Bit Error Rate, BER) von $10^{-13} - 10^{-15}$ als notwendig betrachtet [MD95b].

In den vergangenen Jahren haben sich die Anforderungen an den Festplatten-Lesekanal durch einen zunehmenden Einsatz in mobilen Geräten – zum Beispiel in Laptops, aber auch in Digitalkameras und PDAs – stark verändert. Der mobile Einsatz erfordert nun auch eine Optimierung in Bezug auf die Verlustleistung. In aktuellen Modellen wird rund 45% der Leistung in der Elektronik verbraucht [DSS00], so dass hier gezielte Optimierungen durch den Einsatz von Low-Power Implementierungen sinnvoll sind.

Auch bei mobilen Festplatten sollen alle genannten Entwurfskriterien erfüllt sein, insbesondere die Zuverlässigkeit der ausgelesenen Daten muss zwingend erhalten bleiben. In Bezug auf Speicherkapazität und Datenrate sind hingegen Kompromisse denkbar. Dennoch erfordert die auch bei mobilen Festplatten steigende Datendichte Innovationen bei der Signalverarbeitung. Die einfachste Möglichkeit zur Erhöhung der Zuverlässigkeit ist die Verwendung von fehlerkorrigierenden Codes mit einem höheren Redundanzanteil. Allerdings wird durch diese Maßnahme gleichzeitig die Schaltungskomplexität in der Decoderschaltung erhöht und die Nutzdatenrate verkleinert.

1.2 Ziele der Arbeit

In dieser Arbeit soll ein Weg gezeigt werden, bei dem die Fehlerrate des Kanals bei einem gleich bleibenden oder sogar reduzierten Redundanzanteil abgesenkt wird. Zu diesem Zweck wird eine zusätzliche Information im Datenstrom ausgenutzt: Die so genannte Zuverlässigkeitsinformation (im Folgenden auch: Soft-Information oder Soft-Output). Diese ist ein Maß für die Fehlerwahrscheinlichkeit jedes einzelnen Bits. Geeignete fehlerkorrigierende Codes können diese Zusatzinformation ausnutzen und auf diese Weise bei gleicher Redundanz eine niedrigere Fehlerrate erzielen. Der mögliche Gewinn liegt im Bereich von 1,5 bis 3 dB [CHIW98]. Der Nachteil eines Lesekanals unter der Verwendung von Zuverlässigkeitsinformation (im Folgenden auch: Soft-Decoding Kanal) ist allerdings eine erhöhte Komplexität bei allen signalverarbeitenden Komponenten.

Die einzelnen Schritte für die Realisierung eines Soft-Decoding Kanals werden im Folgenden dargestellt: Zunächst muss die Zuverlässigkeitsinformation während des Detektionsprozesses bestimmt werden. Das dabei zu Grunde liegende Prinzip ist die Betrachtung von Alternativen zur getroffenen Detektionsentscheidung. Zu diesem Zweck wird geprüft, wie groß die Wahrscheinlichkeit einer gegenteiligen Detektorausgabe im Vergleich zur wirklichen Ausgabe ist.

Das eigentliche Ziel bei der Verwendung von Zuverlässigkeitsinformation ist die Verbesserung der Fehlerkorrekturereinheit. Nicht jeder fehlerkorrigierende Code ist zu einer

Verarbeitung dieser Zusatzinformation in der Lage. Aus diesem Grund müssen bestehende Verfahren entweder angepasst oder durch besser geeignete Alternativen ersetzt werden. Viele der in der Literatur präsentierten Soft-Decoding Verfahren verwenden allerdings sehr aufwändige, auf probabilistischen Verfahren basierende Decodieralgorithmen. Somit wird die Suche nach effizienten Hardware-Implementierungen zu einer großen Herausforderung beim Entwurf eines Soft-Decoding Lesekanals.

Der magnetische Kanal unterscheidet sich stark von anderen Kommunikationskanälen: Die festgeschriebene Sektorgröße beschränkt die eingesetzten Fehlerkorrektur in der maximal möglichen Codewortlänge. Ebenso sind bei den heutigen Datenübertragungsraten und Datendichten nur Verfahren mit einem sehr geringen Redundanzanteil einsetzbar, die sich darüber hinaus auch sehr schnell decodieren lassen müssen.

Viele der in der Literatur vorgestellten Fehlerkorrektur-Verfahren wurden ursprünglich nicht für den magnetischen Kanal entworfen: Sie erreichen ihre hervorragende Decodierleistung nur bei sehr großen Codelängen und beim Einsatz eines sehr hohen Redundanzanteils. Daher müssen sie zunächst für den magnetischen Kanal angepasst werden und danach mit den üblicherweise bei Festplatten eingesetzten Verfahren verglichen werden.

Außerdem wird auf dem magnetischen Kanal ein Modulationscode benötigt, der ein Minimum an Timing-Information für die Taktrückgewinnung garantiert. Der Modulationsdecoder, der sich zwischen Detektor und Fehlerschutzdecoder befindet, muss also zur Weiterverarbeitung der Zuverlässigkeitsinformation in der Lage sein. Dies erfordert auch hier eine Modifikation am Lesekanal.

In dieser Arbeit werden verschiedene Verfahren zur Integration von Soft-Decoding in einen magnetischen Kanal untersucht. Das Ziel sind Lösungen, bei denen die Berechnung und Weiterverarbeitung der Zuverlässigkeitsinformation unter möglichst geringem Schaltungsaufwand bzw. mit möglichst geringer algorithmischer Komplexität realisierbar sind. Die Optimierung erfolgt systemübergreifend, weil dadurch die Komplexität und die Leistungsaufnahme gegenüber einer separaten Optimierung jeder einzelnen Komponente weiter reduziert werden kann. Dabei wird auch eine Erhöhung der Komplexität einzelner Komponenten in Kauf genommen, sofern die Komplexität des Gesamtsystems reduziert wird.

Als erstes werden verschiedene Detektoren untersucht, die Soft-Outputs generieren können. Neben den in der Praxis häufig verwendeten Partial Response Maximum Likelihood (PRML) Detektoren [WT99] wird das leistungsfähige Konzept der Signalraumdetektion [Pat91, SHS00] betrachtet. Dieses wird in dieser Arbeit zur effizienten Ausgabe von Zuverlässigkeitsinformation modifiziert.

Danach wird ein systemübergreifendes Konzept vorgestellt, das einen Lesekanal mit Reed-Solomon Codes zur Verarbeitung von Zuverlässigkeitsinformation modifiziert: Ein in den Datenstrom eingefügter Parity-Check Code dient dabei sowohl als einfacher Modulationscode, als auch – gemeinsam mit den Soft-Outputs des Detektors – zur Identifikation von Fehlerpositionen im Datenstrom [SSHS03]. Die Kenntnis über Fehlerpositionen kann im Reed-Solomon Decoder im Idealfall zu einer Verdopplung der Korrekturfähigkeit führen.

Des Weiteren werden Kanäle untersucht, bei denen an Stelle des Parity-Check Codes ein

herkömmlicher Modulationscode eingesetzt wird. Zu diesem Zweck wird in dieser Arbeit ein allgemeines Konzept zur Weiterverarbeitung von Zuverlässigkeitsinformation im Modulationsdecoder vorgestellt. Das Konzept wird am Beispiel eines Maximum Transition Run (MTR) Codes evaluiert, kann allerdings ebenso auf andere Modulationsverfahren angewandt werden.

Dieser so genannte Soft-In/Soft-Out Modulationsdecoder wird dann mit fehlerkorrigierenden Verfahren kombiniert, die eine direkte Weiterverarbeitung der Soft-Outputs ermöglichen. An dieser Stelle kommen sowohl Low-Density Parity-Check (LDPC) Codes [Gal62], als auch Turbo Codes [BGT93] zum Einsatz, die jeweils an die Gegebenheiten des magnetischen Kanals angepasst werden. Im Gegensatz zum ersten Ansatz, bei dem die Zuverlässigkeitsinformation nur zur Fehlermarkierung dient, erfolgt hier auch im Fehlerschutzdecoder eine direkte Verarbeitung der Soft-Outputs. Es wird untersucht, ob der höhere Aufwand zu einer niedrigeren Fehlerrate führt oder ob die notwendige Anpassung der Verfahren sich nachteilig auf die Gesamtzuverlässigkeit auswirkt.

Beide Alternativen zur Konstruktion eines Soft-Decoding Kanals werden dann in Bezug auf ihre Komplexität und ihre Zuverlässigkeit gegenüber gestellt. Bei der Optimierung der Lesekanäle erfolgt keine isolierte Betrachtung der Einzelkomponenten, sondern immer eine gleichzeitige Berücksichtigung von Detektor, Modulationscode und Fehlerschutzcode. Dieses systemübergreifende Konzept ist notwendig, da je nach eingesetztem Verfahren oder Kanal andere Fehlermuster entstehen und somit beispielsweise nach dem Austausch des Detektors ein anderer fehlerkorrigierender Code zu den besten Ergebnissen führen kann.

1.3 Weiterer Aufbau

Im weiteren Verlauf der Arbeit erfolgt als erstes in Kap. 2 eine Beschreibung des magnetischen Kanals sowie eine Erläuterung der beim Lesen und Schreiben auftretenden Störungen. Daraus wird dann das dieser Arbeit zu Grunde liegende Kanalmodell hergeleitet. In Kap. 3 werden die beiden Konzepte für die Integration von Zuverlässigkeitsinformation in den Festplattenlesekanal beschrieben. Die dabei verwendeten Komponenten werden in den Kapiteln 4, 5 und 6 beschrieben. In Kap. 7 erfolgt schließlich die Evaluation verschiedener Lesekanäle. Die Arbeit schließt mit einer Zusammenfassung der wichtigsten Ergebnisse in Kap. 8.

Kapitel 2

Magnetische Aufzeichnung auf Festplatten

Magnetische Datenspeicher sind die am weitesten verbreitete Form zur Speicherung großer Datenmengen. Die erste Festplatte – genannt RAMAC, Modell 350 – entwickelte die Firma IBM bereits im Jahre 1957 [Ste81]. Die Platte hatte eine aus heutiger Sicht geringe Gesamtkapazität von 5 MByte, die durch die Speicherung auf insgesamt 50 Magnetscheiben mit einem Durchmesser von je 24 Zoll erreicht wurde.

Seitdem konnte die Kapazität von Festplatten enorm gesteigert werden. Dies ist in erster Linie auf eine Verbesserung der eingesetzten Technologie, zum Beispiel durch Dünnschicht-Medien oder magnetoresistive (MR) Leseköpfe, aber auch durch eine verbesserte Signalverarbeitung erreicht worden. Dennoch erfordert der zunehmende Speicherbedarf auch weiterhin ständige Innovationen auf dem Gebiet der Festplatten-Technologie.

In dieser Arbeit wird der Schwerpunkt auf die Signalverarbeitung bei der Speicherung bzw. beim Auslesen der Daten gelegt. Zum Verständnis der eingesetzten Komponenten und Algorithmen ist zunächst aber ein kurzer Blick auf den Aufbau und die Komponenten einer Festplatte notwendig. Nach einer kurzen Einleitung werden hierbei vor allem die Charakteristik und die Ursachen von Detektionsfehlern erläutert. Anschließend werden die für diese Arbeit relevanten Störgrößen herausgearbeitet und daraus ein Kanalmodell aufgestellt.

2.1 Überblick

2.1.1 Aufbau einer Festplatte

Die Datenspeicherung auf einer Festplatte erfolgt auf einer starren, rotierenden Scheibe aus einer Aluminium-Magnesium-Legierung oder aus Glas, die mit einer dünnen, magnetisierbaren Schicht versehen ist. Die Information wird auf konzentrischen Spuren in typischerweise 512 Byte großen Blöcken, den so genannten Sektoren, abgelegt. Die binäre Information wird dabei auf das Medium eingepreßt, indem ein Schreibkopf die magne-

tisierbare Schicht auf der Spur in Drehrichtung oder entgegengesetzt zur Drehrichtung der Scheibe magnetisiert. Die kleinste schreibbare Einheit wird als Bitzelle bezeichnet. Bei der Datenrückgewinnung detektiert ein Lesekopf die Wechsel der Magnetisierungsrichtung (so genannte Transitionen).

Eine Festplatte enthält in der Regel mehrere Magnetscheiben (bis zu zwölf), die auf einer gemeinsamen Spindel untergebracht sind. Jede Scheibe ist auf der Ober- und Unterseite mit eigenen Leseköpfen versehen, die alle an einem gemeinsamen, kammförmigen Arm befestigt sind. Die Köpfe werden von einem Servomotor bewegt. Die Rotationsgeschwindigkeit moderner Festplatten liegt zwischen 3500 und 15.000 Umdrehungen pro Minute. Aufgrund der hohen Relativgeschwindigkeit schweben die Leseköpfe auf einem dünnen Luftpolster und ermöglichen so einen berührungslosen Lesevorgang. Die Flughöhe ist ein wichtiger Parameter beim Entwurf einer Festplatte, da eine Steigerung der Speicherdichte nur bei gleichzeitiger Reduktion der Flughöhe zu erzielen ist [Ash97].

2.1.2 Komponenten des magnetischen Kanals

Der Schreib- und Lesevorgang kann durch einen Kanal modelliert werden, der im Wesentlichen einem herkömmlichen Kommunikationsmodell aus der Nachrichtentechnik entspricht. In Abbildung 2.1 sind die Einzelkomponenten des magnetischen Kanals für den Schreib- und Lesevorgang getrennt zu sehen. Beim Schreibvorgang gelangen die Daten über das Host-Interface in die Festplatte und werden nach einer Codierung in einen Schreibstrom umgewandelt. Der Lesevorgang beginnt mit der Erfassung der Transitionen des Magnetfelds auf dem Medium. Nach der Detektion und der Decodierung wird das fehlerfreie Ursprungssignal an das Host-Interface weitergereicht. Im Folgenden werden die einzelnen Stufen eines Schreib- und eines Lesevorgangs kurz erläutert. Eine ausführliche Diskussion der einzelnen Komponenten erfolgt in den Kapiteln 4 bis 6.

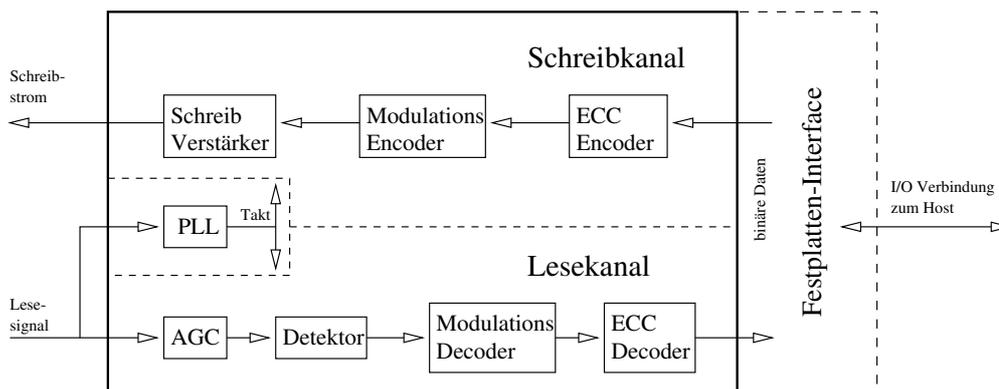


Abbildung 2.1: Die Komponenten des Schreib- und Lesekanals einer Festplatte

Die bei Festplatten erforderliche Bitfehler-Rate von mindestens 10^{-13} erfordert den Einsatz einer leistungsfähigen Fehlerkorrektur. Aus diesem Grund werden die zu schreibenden Daten zunächst mit einem geeigneten Fehlerschutz codiert, wobei Redundanz in den Datenstrom eingefügt wird. In der Regel erfolgt die Fehlerkorrektur in zwei Stufen. Die

erste Stufe wird während des Betriebs in Hardware korrigiert (so genannte On-the-Fly Korrektur). Die zweite Stufe ist in Software implementiert und wird nur bei einem Decodiersversagen der ersten Stufe verwendet. Meist kommen Reed-Solomon Codes zum Einsatz [MD95b], die sich hervorragend als schnelle On-the-Fly Fehlerkorrektur auf dem magnetischen Kanal eignen [HSW95]. In dieser Arbeit wird nur die On-the-Fly Korrektur betrachtet.

Nach dem Fehlerschutz-Encoder werden die Daten mit einem Modulationscode versehen. Dieser erzwingt zusätzliche Beschränkungen im Datenstrom und fügt dadurch weitere Redundanz in das System ein. Modulationscodes können aus mehreren Gründen eingesetzt werden. Einer der wichtigsten Gründe bei Festplatten ist, dass der maximale Abstand zweier Transitionen auf dem Medium beschränkt wird. Dadurch wird ein Minimum an Timing-Information im Lesesignal garantiert, das zur Rekonstruktion der Daten beim Lesen erforderlich ist. Nach dem Modulationsencoder wird schließlich aus den codierten Daten ein Schreibstrom erzeugt, der die Wechsel des Magnetfeldes auf der Festplatte erzeugt.

Beim Leseprozess müssen die auf den Scheiben festgehaltenen Transitionen des Magnetfeldes wieder in einen binären Datenstrom umgewandelt werden. Ein Lesekopf – früher ein Induktivkopf, heute ein magnetoresistives (MR) Element – erfasst das Magnetfeld des Mediums und erzeugt einen kontinuierlichen Lesestrom, der mit der gespeicherten Information korrespondiert. Eine Automatic Gain Control (AGC) hält die Ausgangsspannung, die zum Beispiel durch leicht unregelmäßige Material-Beschichtungen leicht schwankt, in festgelegten Grenzen. Der Detektor hat nun die Aufgabe, aus diesem Lesesignal die gespeicherten, modulierten Daten zurückzugewinnen. Aufgrund mehrerer Rausch- und Störeffekte, die beim Schreiben oder Lesen entstehen, ist das zurückgewonnene Signal fehlerhaft. Typischerweise werden Fehlerraten von etwa 10^{-5} erreicht [MD95a, Ash97]. Nach der Detektion müssen die Daten demoduliert werden. Bei diesem Prozess werden die mit den Beschränkungen des Modulationscodes versehenen Daten blockweise auf die ursprünglichen, noch mit dem Fehlerschutz codierten Daten abgebildet. Dabei kann ein einzelner Detektionsfehler zu mehreren fehlerhaften Bits führen. Dieser Punkt ist eine von mehreren Stellen, an denen der magnetische Kanal zu Fehlerfortpflanzung neigt. Hierbei entstehen die für den Kanal typischen Bündelfehler (engl. Burst Error). Bei der abschließenden Decodierung des Fehlerschutzcodes muss die Fehlerrate auf die eingangs erwähnten 10^{-13} bis 10^{-15} gesenkt werden. Üblicherweise wird das Ergebnis des Fehlerschutzcodes durch eine zweite Stufe, dem so genannten Cyclic Redundancy Check (CRC) Code auf Korrektheit überprüft [MD95a, Ash97]. Bei einem Fehler werden die Daten erneut von der Festplatte gelesen (engl. Re-Read) [MD95a, Ash97] oder an die zweite, in Software implementierte Fehlerkorrektur weitergereicht.

2.2 Schreibprozess

Nach diesem kurzen Überblick folgt eine detailliertere Behandlung des Schreib- und Leseprozesses. Dieser Vorgang ist nicht ideal und fehlerfrei. Im Folgenden wird das Augenmerk auf die verschiedenen Rauschquellen gelegt, die als Ursache von Detektionsfehlern gelten. Diese sind teilweise vorhersagbar, teilweise aber auch rein zufälliger Natur.

2.2.1 Aufbau und Arbeitsweise des Schreibkopfes

Die Speicherung der Daten auf der Magnetscheibe erfolgt über einen induktiven Schreibkopf. Dieser besteht aus einem Eisenring oder -joch, der einen schmalen Spalt besitzt (Abbildung 2.2). Um den Eisenring ist eine Spule gewickelt (in der Abbildung nicht zu sehen), an die der Schreibstrom angelegt wird. Im Bereich des Spaltes tritt ein magnetisches Streufeld aus dem Ring heraus. Beim Schreibvorgang wird der Schreibkopf in geringer Höhe über der Magnetscheibe geführt, so dass das aus dem Spalt austretende Streufeld in die Oberfläche eindringt. An den Bereichen der Oberfläche, an denen die magnetische Feldstärke die Koerzitivfeldstärke H_c des magnetischen Mediums übersteigt, erfolgt eine permanente Magnetisierung durch den Schreibkopf.

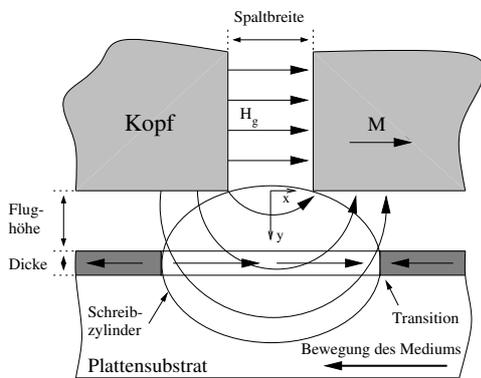


Abbildung 2.2: Darstellung des Schreibprozesses

Um Nichtlinearitäten aufgrund der Hysterese des magnetischen Mediums zu vermeiden, muss die Feldstärke im Medium so groß sein, dass Sättigung eintritt. Die Magnetisierung erfolgt in oder entgegengesetzt zur Drehrichtung der rotierenden Scheibe und somit longitudinal zur Oberfläche. Aus diesem Grund spielt nur die longitudinale Komponente H_x des Kopffeldes eine Rolle [WT99]. Die Umgebung des Spaltes, in der die Koerzitivkraft überschritten wird, wird als Schreibzylinder (engl. Write Bubble) bezeichnet. Die in Drehrichtung der Magnetscheibe nachlaufende Kante ist der Bereich, an dem der Einfluss des Schreibkopfes auf das Medium endet und kann somit in erster Näherung als die Stelle bezeichnet werden, an der eine Transition geschrieben wird [WT99].

2.2.2 Beeinträchtigungen des Schreibprozesses

Der Schreibprozess unterliegt mehreren Beeinträchtigungen, von denen die wichtigsten hier vorgestellt werden. Grundsätzlich lassen sie sich in zwei Kategorien unterteilen: Gleichverteiltes weißes oder annähernd weißes Rauschen und das so genannte Transitionsrauschen (engl. Jitter), das nur in Verbindung mit einem Wechsel der Magnetisierungsrichtung (Transition) auftritt und die Positionen der Magnetfeldwechsel leicht in einer nichtlinearen Weise verschiebt. Beim Schreibprozess dominieren die folgenden Jitter-Effekte:

– Bandbreitenbegrenzung des Schreibstroms

Ein idealer Schreibstrom ist exakt rechteckförmig und kann so exakte Übergänge zwischen den Transitionen generieren. In der Realität entsteht allerdings aufgrund des in der Bandbreite begrenzten Schreibstroms eine endliche Steigung. Aus diesem Grund variiert die Größe des Schreibzylinders dynamisch [WWH90, BvV91]. Die Feldstärke im Spalt steigt zunächst steil an, um sich danach asymptotisch an das Maximum anzunähern. Dieser Anstieg kann sich bei hohen Datendichten über die Breite mehrerer Bitzellen erstrecken. Deshalb hängt der exakte Verlauf

der Feldstärke von dem unmittelbar zuvor geschriebenen Datenmuster ab. Dies bewirkt, dass die Transitionen nicht exakt an den vorhergesehenen Positionen geschrieben werden.

– **Non-Linear Transition Shift (NLTS)**

Ein weiterer Grund, der ebenfalls zu einer Verschiebung der Transitionen führt, ist der so genannte "Nonlinear Transition Shift (NLTS)" [Ber94, Tar96]. Dieser ist auf Effekte benachbarter Transitionen zurückzuführen. Der Bereich zwischen zwei Transitionen ist nicht absolut homogen magnetisiert. Dadurch entstehen Demagnetisierungsfelder zwischen aufeinander folgenden Transitionen. Diese Felder überlagern sich beim Schreibprozess mit dem Kopffeld und vergrößern oder verkleinern dadurch den Schreibzylinder. Dies führt ebenfalls zu einer Verschiebung der Transition. Da der Effekt über mehrere benachbarte Transitionen hinweg Einfluss hat, werden die Transitionen in einer nicht-linearen Art beeinflusst.

Sowohl NLTS, als auch die zuvor beschriebene Bandbreitenbegrenzung des Schreibstroms bewirken eine nicht-lineare Verschiebung der Transitionen. Dennoch sind beide Effekte auf die zuvor geschriebenen Transitionen zurückzuführen und aus diesem Grund abschätzbar [Ash97]. Durch eine geeignete Anpassung des Schreibstroms in Abhängigkeit der vorher geschriebenen Daten können beide Effekte somit kompensiert werden (so genannte Write Precompensation) [Tar96].

Die folgenden beiden Jitter-Effekte bewirken ebenfalls eine Verschiebung der Transitionen, allerdings hängen sie nicht von den unmittelbar zuvor geschriebenen Daten ab und sind somit nicht durch eine geeignete Schaltung kompensierbar.

– **Überschreibeffekte (Hard Transitions)**

Ein bislang nicht betrachteter Effekt ist die zuvor auf das Medium eingeprägte Magnetisierung. Bei Festplatten gibt es keinen expliziten Löschvorgang – das gespeicherte Signal wird einfach überschrieben [MD95a, Ash97].

Falls das Medium entgegengesetzt zur bestehenden Magnetisierung beschrieben werden soll, entstehen unter dem Schreibkopf aufgrund der an beiden Seiten des Schreibzylinders geschriebenen Transitionen Demagnetisierungsfelder. Diese führen – ähnlich wie NLTS – zu einer Verschiebung der Transition. Dieser Überschreib-Effekt wird als Hard Transition bezeichnet [Ber94, Tar96]. Dieser Effekt führt dazu, dass die Bitzellen als breiter oder schmaler wahrgenommen werden.

Entspricht die durch den Schreibkopf eingeprägte Magnetisierungsrichtung der bisherigen Magnetisierung, so treten keine Demagnetisierungseffekte auf. In diesem Fall wird von einer Easy Transition gesprochen. Der Effekt der Hard Transitions ist nicht wie NLTS prädizierbar und tritt deshalb als eine Form des Rauschens auf.

– **Nichtlinearitäten des magnetischen Mediums**

Ein weiterer Rauscheffekt entsteht durch die Struktur des Mediums selbst. Das magnetische Medium lässt sich nicht in beliebig kleinen Bereichen magnetisieren. Stattdessen bilden sich so genannte Domänen mit homogener Magnetisierung aus [BMN83]. Dies hat zur Folge, dass die Bitzellen auf dem Medium keine exakten

Kanten besitzen, sondern eine nichtlineare, zickzack-förmige Struktur aufweisen [BGM85]. Aus diesem Grund verschiebt sich die effektive Position einer Transition – das heißt der Punkt, an dem der Lesekopf die Transition wahrnimmt – in einer nichtlinearen Weise.

Die genannten Effekte führen dazu, dass beim Lesen der Daten die Transitionen des Magnetfeldes an der falschen Stelle detektiert werden. Darüber hinaus entstehen auch beim Leseprozess weitere Nichtlinearitäten, die im Folgenden beschrieben werden.

2.3 Leseprozess

2.3.1 Aufbau und Arbeitsweise des Lesekopfes

Beim Lesen müssen die Daten aus den Transitionen rekonstruiert werden, die auf den Magnetscheiben gespeichert sind. Der erste Schritt ist die Detektion der Transitionen durch den Lesekopf. Dieser bestand bei älteren Festplatten noch aus einem induktiven Lesekopf. Er besaß einen dem Schreibkopf identischen Aufbau, so dass in älteren Modellen ein einziger Kopf für den Lese- und Schreibprozess verwendet wurde [Ash97].

Der Leseprozess ist im Prinzip eine Umkehrung des Schreibvorgangs: Das Feld der magnetisierten Schicht dringt in den Luftspalt des Lesekopfes ein. Bei einer Transition auf dem Medium wird aufgrund des Magnetfeldwechsels ein Strom im Lesekopf induziert. Auf diese Weise können die Transitionen wieder in einen kontinuierlichen Datenstrom umgesetzt werden, der mit der gespeicherten Information korrespondiert.

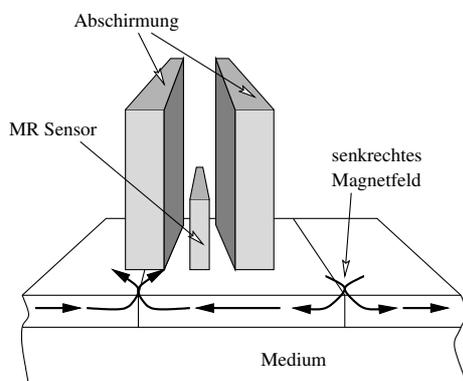


Abbildung 2.3: Aufbau eines magnetoresistiven Lesekopfes

Heutige Festplatten hingegen verwenden seit den frühen 1990er Jahren ein magnetoresistives (MR) Element als Lesekopf [GH96]. Es nutzt den Effekt aus, dass sich der elektrische Widerstand einiger Metalle im Magnetfeld ändert [Hun71]. In der Praxis werden ein oder zwei magnetoresistive Elemente zwischen zwei Abschirmungen in geringer Höhe über die Festplatte geführt (Abbildung 2.3). Die Elemente detektieren die senkrechte Komponente des Magnetfelds und somit die Transitionen auf den Scheiben.

Die Verwendung von MR-Köpfen hat viele Vorteile, die alle dazu beigetragen haben, dass die Datendichte heutiger Festplatten immens gesteigert werden konnte. Einer der Vorteile ist, dass das Signal eines MR-Elements unabhängig von der Rotationsgeschwindigkeit der Magnetscheiben ist. Dadurch ist die Signalamplitude über die gesamte Festplatte annähernd konstant und muss nicht wie bei Festplatten mit Induktivköpfen an die Kopfposition angepasst werden. Des Weiteren kann der MR-Effekt auch bei langsam rotierenden Platten ein ausreichend großes Lesesignal erzeugen. Dadurch eignen sich Festplatten mit MR-Elementen insbesondere für leistungsarme Implementierungen, zum Beispiel für den mobilen Einsatz [Str00]. Des Weiteren besitzt

das Lesesignal eines MR-Kopfes ein höheres Signal-Rauschleistungs Verhältnis (SNR) als das eines Induktivkopfes – eine wichtige Voraussetzung für sehr hohe Datendichten. Durch eine geringere Induktivität besitzt das Lesesignal zudem eine höhere Bandbreite. Allerdings ist eine Differenzierung des Lesesignals notwendig, um eine Lesepuls-Form wie bei einem Induktivkopf zu erreichen.

Beim Einsatz eines magnetoresistiven Lesekopfes ist allerdings ein getrennter, induktiver Schreibkopf erforderlich. Die dadurch resultierende höhere Hardware-Komplexität wird aber in Kauf genommen, da eine gezielte Optimierung von Lese- und Schreibprozess möglich ist [Ash97].

2.3.2 Charakterisierung des Lesesignals

Das im Lesekopf durch die Transitionen erzeugte Signal muss der gespeicherten Information zugeordnet werden. Für die Beschreibung dieses Signals existieren zwei äquivalente Notationen, die sich leicht ineinander überführen lassen. Das Medium wird im Abstand T abgetastet. Dieser minimale Abstand T zweier Transitionen wird als Bitintervall bezeichnet.

Für die erste Notation wird das Lesesignal direkt als eine Sequenz \underline{a} von Transitionen beschrieben. Dabei wird zum Zeitpunkt $t = i \cdot T$ eine Eins angenommen, wenn eine Transition vorliegt, während beim Ausbleiben einer Transition eine Null angenommen wird. Die sich bei dieser Notation ergebende Sequenz \underline{a} wird im Folgenden als Non Return To Zero Inverted (NRZI) bezeichnet. Die Elemente a_i ergeben sich zu

$$a_i = \begin{cases} 0 & \text{keine Transition zum Zeitpunkt } t = i \cdot T \\ \pm 1 & \text{sonst.} \end{cases} \quad (2.1)$$

Die zweite Notation entspricht dem Schreibstrom, der immer dann sein Vorzeichen wechselt, wenn eine Transition geschrieben wird ($b_i = \pm 1$). Die sich daraus ergebende Sequenz \underline{b} wird Non Return To Zero (NRZ) genannt. Die Differenz der letzten beiden NRZ-Werte entspricht dabei dem aktuellen NRZI-Wert:

$$a_i = \frac{1}{2}(b_i - b_{i-1}) \quad (2.2)$$

Die Form des Lesesignals bei einer isolierten Transition lässt sich sowohl für induktive, als auch für magnetoresistive Leseköpfe gut durch den so genannten Lorentzpuls charakterisieren (Abbildung 2.4). Der Puls stellt eine Näherung unter der Annahme eines zirkularen Kopffeldes dar [Ber94]. Eine charakteristische Kenngröße des Lorentzpulses ist die Breite bei halber Amplitude. Sie wird als PW_{50} bezeichnet und gibt ein einfaches Maß für die Datendichte auf dem Medium an. Je breiter der Puls in Relation zu einer elementaren Bitzelle auf dem Medium ist (PW_{50}/T), desto größer ist die lineare Datendichte. Der Lorentzpuls als Sprungantwort $g(t)$ bei einer isolierten Transition ergibt sich wie folgt:

$$g(t) \sim \frac{1}{1 + \left(\frac{2t}{PW_{50}}\right)^2} \quad (2.3)$$

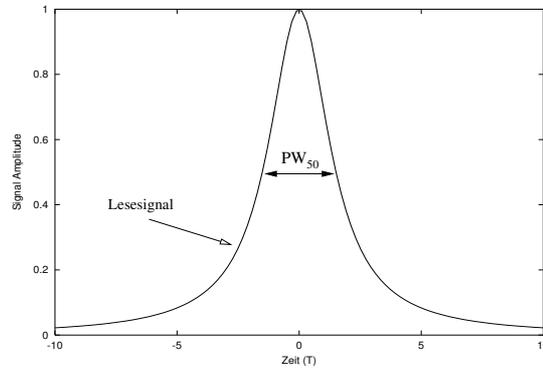


Abbildung 2.4: Durch Lorentzimpuls approximiertes Lesesignal bei einer isolierten Transition ($PW_{50}/T = 3.0$)

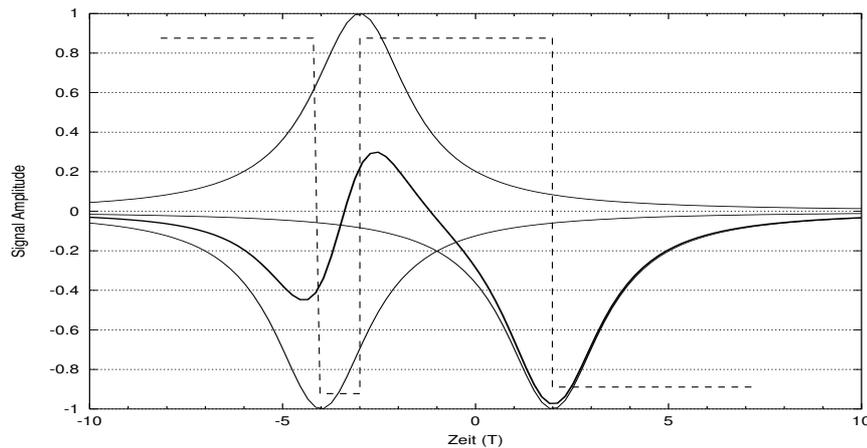


Abbildung 2.5: Intersymbolinterferenz bei $PW_{50}/T = 3.0$: Die einzelnen Pulse (dünne Linien) überlagern sich. Dies führt bei dicht beieinander liegenden Transitionen zu einer Reduktion der Amplitude und zu einer Verschiebung der Maxima gegenüber den idealen Positionen (gestrichelt).

Bei heutigen Festplatten entspricht der Wert von PW_{50} einem Vielfache der Breite T einer Bitzelle. Dies führt zu einer starken Überlagerung einzelner Lesepulse (Intersymbolinterferenz, ISI). Dadurch werden die Pulse in ihrer Amplitude verkleinert oder sogar teilweise ausgelöscht und die Maxima leicht verschoben (Abbildung 2.5). Allerdings ist die Überlagerung im Gegensatz zu den bisher beschriebenen Effekten linear und somit berechenbar.

Das ideale, rauschfreie Lesesignal $z(t)$ lässt sich über die Sprungantwort $g(t)$ oder die Impulsantwort $h(t)$ des magnetischen Kanals, die sich als Systemantwort auf zwei direkt

aufeinander folgende Transitionen ergibt, beschreiben. In Abhängigkeit von den Sequenzen der Transitionen \underline{a} bzw. der Schreibstromamplituden \underline{b} ergibt sich

$$z(t) = \begin{cases} \sum_i a_i \cdot g(t - iT) & \text{NRZI-Notation} \\ \sum_i b_i \cdot h(t - iT) & \text{NRZ-Notation} \end{cases} \quad (2.4)$$

$$\text{mit } h(t) = \frac{1}{2}(g(t) - g(t - T)). \quad (2.5)$$

2.3.3 Beeinträchtigungen des Leseprozesses

Aus dem im Lesekopf erzeugten Signal muss die binäre Information zurück gewonnen werden. Neben der Intersymbolinterferenz wird der Prozess noch von weiteren Effekten beeinträchtigt, die die spätere Umwandlung im Detektor erschweren.

Die erste Schwierigkeit besteht darin, dass die binären Daten ohne Synchronisierungsinformation gespeichert sind [Ash97]. Die Sektoren selbst enthalten zwar am Anfang so genannte Sync-Felder, die zu Beginn eines Lesevorgangs für eine richtige Synchronisierung sorgen, im weiteren Datenstrom innerhalb eines Sektors ist aber keine Timing-Information enthalten. Diese muss aus den Transitionen des Magnetfeldes generiert werden. Zu diesem Zweck kommt eine so genannte Phasen-Halte Schaltung (engl. Phase Lock Loop, PLL) Schaltung zum Einsatz [Ber96]. Die bereits beschriebenen Jitter-Effekte, die zu einer Verschiebung der Transitionen führen, erschweren somit die Taktrückgewinnung. Zudem kann aus dem Signal nur dann eine Timing-Information gewonnen werden, wenn eine Transition vorliegt. Das heißt, lange Sequenzen von Nullen – und somit das Ausbleiben einer Transition – erschweren diesen Vorgang. Aus diesem Grund werden Modulationscodes eingesetzt, die den Datenstrom dahingehend verändern, dass ein Minimum an Timing-Information garantiert ist. Eine eingehendere Diskussion der Modulationscodes erfolgt in Kapitel 5.

Ein weiterer Grund für Beeinträchtigungen des Lesesignals entsteht bereits im MR-Lesekopf. Die Kennlinie der MR-Elemente ist nicht linear. Dies kann zwar durch einen geeigneten Arbeitspunkt weitgehend kompensiert werden, allerdings sind die Impulsantworten beider möglicher Transitionen des Magnetfeldes nicht absolut identisch. Dieser Effekt lässt sich aber durch den Einsatz zweier Elemente kompensieren [ANB⁺94] und wird deshalb in dieser Arbeit vernachlässigt.

Die Erfassung der Transitionen durch eine Widerstandsänderung im MR-Kopf verursacht eine Empfindlichkeit des Lesesignals gegenüber thermischen Schwankungen. Eine Temperaturänderung kann eine Verschiebung des Arbeitspunktes bewirken. Ein besonders schwer wiegender Effekt ist der so genannte Thermal Asperity (TA) [Saw98]. Er entsteht bei einem Zusammenprall des Lesekopfes mit einem in der Festplatte eingeschlossenen Partikel. Derartige Einschlüsse können entweder bereits beim Fertigungsprozess oder durch Abrieb entstehen. Die bei einem Zusammenprall frei werdende Wärmeenergie führt zu einer starken Verschiebung des Arbeitspunktes im MR-Kopf und dadurch zu

einem sehr langen Bündelfehler (Burst Error). Thermal Asperity Effekte lassen sich in einem gewissen Rahmen durch eine Korrekturschaltung, die den Arbeitspunkt nachführt, kompensieren [GKP92, Tex98]. Dennoch stellen die entstehenden Fehlermuster eine neue Herausforderung für die Signalverarbeitung bei Festplatten – insbesondere für die Fehlerkorrektur – dar.

Die weiteren Rauschquellen bei Festplatten ergeben sich direkt aus dem Signalverarbeitungspfad. Der Datenstrom wird beim Weg durch den Lesekanal vom Rauschen des Mediums und vom Elektronikrauschen der eingesetzten Komponenten und Schaltungen überlagert [Ash97]. Die einzelnen Komponenten sind dabei unkorreliert und können deshalb als eine einzige Rauschquelle betrachtet werden. Dieses gleichverteilte, gaußförmige Rauschen (engl. Additive White Gaussian Noise, AWGN) ist für die Signalverarbeitung allerdings weniger kritisch als die im Vorfeld beschriebenen Jitter-Effekte.

2.4 Definition des Kanalmodells

Im Folgenden werden die bisher beschriebenen Beeinträchtigungen des Schreib- und Leseprozesses noch einmal zusammengefasst. Daraus werden dann die Parameter festgelegt, die als Grundlage für das in dieser Arbeit verwendete Kanalmodell [DSS] dienen.

2.4.1 Auswahl der modellierten Rauschkomponenten

Das gleichförmige Rauschen durch Medium, Bauteile und Elektronik ist unkorreliert und wird deshalb als eine einzelne AWGN-Rausch-Quelle nachgebildet.

Im Kanalmodell wird beim Lesen aus den einzelnen Pulsen ein quasi-kontinuierlicher Lesestrom erzeugt. Die Intersymbolinterferenz ergibt sich somit automatisch bei der Modellierung.

Die Beeinträchtigungen, die beim Schreiben durch die unmittelbar zuvor geschriebenen Transitionen verursacht werden (datenabhängiges Transitionsrauschen), lassen sich durch eine Write Precompensation kompensieren. Da der Schreibprozess als solcher nicht modelliert wird, werden diese Effekte nicht berücksichtigt und die Kompensation als ideal angenommen.

Das datenunabhängige Transitionsrauschen ist nicht vorhersagbar. Aus diesem Grund muss es in der Modellierung berücksichtigt werden. Grob lässt sich diese Rauschkomponente in zwei Kategorien unterteilen: Überschreibeffekte und Verschiebung der Transitionen durch Nicht-Linearitäten [Str00]. Die Überschreibeffekte lässt sich nur schwer modellieren. Zudem ist der Anteil dieser Komponente am Rausch-Spektrum relativ gering und sinkt sogar noch mit steigenden Datendichten [Str00]. Aus diesem Grund werden sie nicht berücksichtigt. Die Verschiebung der Transitionen hingegen besitzt einen bedeutend größeren Anteil am Rausch-Spektrum und muss daher in das Kanalmodell einbezogen werden.

Aufgrund verbesserter Fertigungsprozesse ist ein Thermal Asperity (TA) sehr selten [Saw98] und wird daher nicht direkt in das Modell einbezogen. Die On-the-Fly Fehlerkor-

rektur von Festplatten wird ohnehin nicht zur Korrektur von Thermal Asperity Fehlern ausgelegt. Statt dessen wird neben einer dynamischen Anpassung des Bias durch den MR-Kopf eine zweite, in Software implementierte Korrekturstufe eingesetzt.

Die in dieser Arbeit modellierten Rausch- und Störquellen sind in Tabelle 2.1 zusammengefasst.

Effekt	Ursache	Präd.	Modell.
Gleichverteiltes Rauschen	Medium, Elektronik, Lesekopf	nein	ja
Intersymbolinterferenz	Überlagerung der Lesepulse	ja	ja
Datenabhängiges Transitionsrauschen	– Bandbreitenbegrenzung des Schreibstroms – Non-Linear Transition Shift (NLTS)	ja ja	nein nein
Datenunabhängiges Transitionsrauschen	– Überschreibeffekte (Hard Transitions) – Nichtlinearitäten des Mediums	nein nein	nein ja
Thermal Asperity (TA)	Kollision mit Partikel	nein	nein

Tabelle 2.1: Übersicht der verschiedenen Rauschquellen beim Schreib- und Leseprozess. Die Spalte "Präd." gibt an, ob die Effekte vorhersagbar und somit kompensierbar sind. Die Intersymbolinterferenz wird bei der Berechnung der Eingangsdaten berücksichtigt und ist nicht kompensierbar. In der Spalte "Modell." steht, ob der Effekt im Kanalmodell der Arbeit berücksichtigt ist.

2.4.2 Kanalmodell

Die in dieser Arbeit berücksichtigten Rauschkomponenten führen zu einer Korruption des Lesesignals. Sofern nur die in Tabelle 2.1 gekennzeichneten Komponenten einfließen, ist ein relativ einfaches Kanalmodell möglich. Das Transitionsrauschen wird durch eine Verschiebung τ des Lesepulses mit der Standardabweichung σ_τ nachgebildet:

$$g(t, \tau) = \frac{1}{1 + \left(\frac{2(t-\tau)}{PW_{50}}\right)^2} \quad (2.6)$$

Das Lesesignal ergibt sich dann als lineare Superposition mehrerer benachbarter Pulse $g(t, \tau)$, die zusätzlich von einer additiven, gleichverteilten Komponente $\nu(t)$ mit der Standardabweichung σ_ν überlagert werden:

$$z(t) = \sum_i a_i \cdot g(t - iT, \tau_i) + \nu(t) \quad (2.7)$$

In dieser Arbeit werden die eingesetzten Algorithmen für verschiedene magnetische Kanäle getestet. Gleichung 2.6 erlaubt dabei durch die Veränderung von PW_{50}/T eine Anpassung der linearen Datendichte. Durch die Standardabweichungen σ_ν und σ_τ können unterschiedliche Signal-Rauschleistungs-Verhältnisse und Kanäle mit unterschiedlichen Anteilen von Transitionsrauschen nachgebildet werden.

2.5 Zusammenfassung

Die in diesem Kapitel erläuterten Beeinträchtigungen des Lese- und Schreibprozesses erzeugen Fehlermuster, die für den magnetischen Kanal charakteristisch sind. Grob lassen sich die Fehler in drei Gruppen einteilen:

- **Einzelbit-Fehler**

Durch verschiedene Rausch-Effekte können auf dem magnetischen Kanal zufällige Einzelbit-Fehler auftreten. Deren Ursache kann sowohl in dem gleichverteilten Elektronikrauschen, als auch im Transitionsrauschen liegen.

- **Kurze Bursts aufgrund von Fehlerfortpflanzung**

An mehreren Stellen des magnetischen Kanals kann eine Fehlerfortpflanzung auftreten. In einigen Detektoren, wie zum Beispiel dem Signalraumdetektor [MJ98], ist eine Feed-Back Struktur zur Kompensation der Intersymbolinterferenz eingebaut. Dadurch wird bei einem Einzelbitfehler ein falscher Wert zurückgekoppelt und kann somit weitere Bits kippen lassen. Durch diesen Effekt entstehen kurze Bündelfehler mit einer Länge von einigen Bits. Eine zweite Quelle für kurze Fehlerbursts ist der Modulationsdecoder, der den modulierten Datenstrom blockweise auf die ursprüngliche Datenfolge abbildet. An dieser Stelle ist die Fehlerlänge auf die Blockgröße des Modulationscodes beschränkt.

- **Lange Bursts aufgrund von Thermal Asperity (TA)**

Der bereits beschriebene Thermal Asperity ist unabhängig von den anderen beiden Fehler-Events und tritt nur aufgrund einer Kollision des Lesekopfes mit einem Partikel auf. Da Thermal Asperity in dem Kanalmodell dieser Arbeit nicht modelliert wird, erfolgt nur eine Abschätzung, welche Fehlerlängen ohne Unterbrechung des Leseprozesses (On-the-Fly) korrigierbar sind.

Die Verteilung dieser Fehler-Events ist charakteristisch für die eingesetzten Komponenten. Insbesondere erzeugt jeder Detektor ein typisches Fehlermuster. Aus diesem Grund erfolgt in Kapitel 7.5 eine Untersuchung der Fehlerstatistik für die betrachteten Detektoren. Im folgenden Kapitel werden die Kriterien für den Entwurf eines Festplatten-Lesekanals diskutiert. Dabei wird aufgezeigt, wie den besonderen Anforderungen und Charakteristiken des magnetischen Kanals begegnet werden kann.

Kapitel 3

Entwurfskriterien für einen Festplatten-Lesekanal

Bei der Entwicklung einer Festplatte müssen verschiedene Ziele berücksichtigt werden. Diese bewegen sich in einem Spannungsfeld, in dem eine Verbesserung an der einen Stelle zu einer Verschlechterung an einer anderen Stelle führen kann. Aus diesem Grund ist eine systemübergreifende Optimierung des Entwicklungsprozesses sinnvoll und notwendig. Komplizierter wird das Optimierungsproblem bei Festplatten für den mobilen Einsatz, da der Stromverbrauch zu einer weiteren, kritischen Größe wird. Bei mobilen Festplatten werden ähnlich hohe Datendichten wie bei den entsprechenden Desktop-Modellen erreicht. Dies erfordert Algorithmen und Konzepte bei der Signalverarbeitung, die nicht nur ein fehlerfreies Auslesen gewährleisten, sondern auch für einen Strom sparenden Betrieb ausgelegt sind. Im Folgenden werden die Anforderungen an den magnetischen Kanal beschrieben und ein Konzept für den Entwurf einer mobilen Festplatte mit hoher Datendichte entworfen.

3.1 Anforderungen an den magnetischen Kanal

Die Diskussion des magnetischen Kanals in Kapitel 2 hat gezeigt, dass das Signal beim Schreiben und Lesen durch zahlreiche Rausch- und Störquellen beeinflusst wird. Dennoch muss das ausgelesene Signal sehr hohen Ansprüchen in vielfacher Hinsicht genügen. Die wichtigsten Kriterien sind:

- **Signalzuverlässigkeit**

Das Signal, das von der Festplatte zum Interface des Hostcomputers weitergereicht wird, erfordert einen extrem hohen Quality-of-Service (QoS). Dies bedeutet, dass die Fehlerrate extrem niedrig sein muss. Im Gegensatz zur Übertragung von Musik- oder Sprachsignalen, bei denen eine geringe Fehlerrate im Signal zu vertreten sind, müssen die von der Festplatte gelesenen Daten fehlerfrei sein. Aus diesem Grund kommen leistungsfähige fehlerkorrigierende Codes zum Einsatz, die die Fehlerrate

auf den gewünschten Wert von mindestens 10^{-13} absenken. Aktuelle Modelle erreichen Werte zwischen 10^{-13} bei mobilen Platten und 10^{-15} bei Festplatten für den Server-Betrieb.

– Speicherkapazität/Speicherdichte

Der Bedarf an Speicherplatz steigt ständig an. Die Speicherkapazität von Festplatten konnte über einen großen Zeitraum jährlich um 60% gesteigert werden. In den vergangenen Jahren wurde dieser Anstieg sogar noch auf 100% p.a. erhöht (Abbildung 3.1). Motor für diese Entwicklung sind stark steigende Datendichten, bei denen inzwischen Werte von bis zu $80 \text{ Gbit}/\text{in}^2$ erreicht werden. Dabei konnten sowohl die lineare Dichte, als auch die Spurdichte enorm gesteigert werden (Abbildung 3.2). Prognosen gehen zwar davon aus, dass sich dieser Trend wieder leicht abschwächen wird, allerdings ist auch weiterhin von einem starken Anstieg auszugehen [TB00]. Aufgrund neuer Innovationen wird auch weiterhin ein Anstieg der Speicherdichte erwartet. Ein Beispiel hierfür ist das so genannte Perpendicular Recording, bei dem die Transitionen senkrecht zum magnetischen Medium geschrieben werden [TB00].

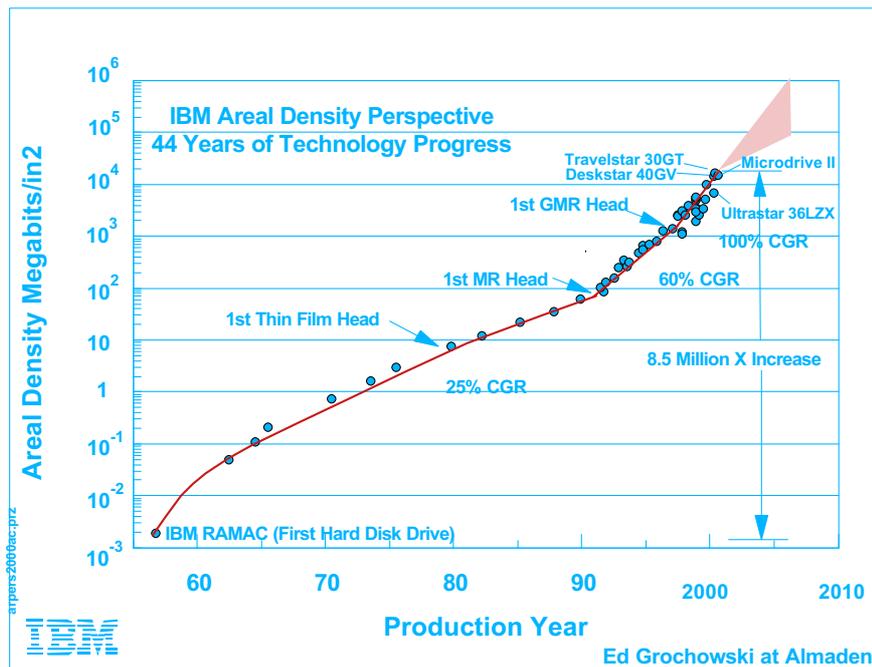


Abbildung 3.1: Entwicklung der Datendichte seit der ersten Festplatte 1957. Der jährliche Anstieg der Speicherkapazität konnte sukzessive von 25% auf 100% gesteigert werden. Zu sehen ist ferner, dass die mobilen Festplatten (zum Beispiel IBM Microdrive) zu den Modellen mit der höchsten Speicherdichte gehören.

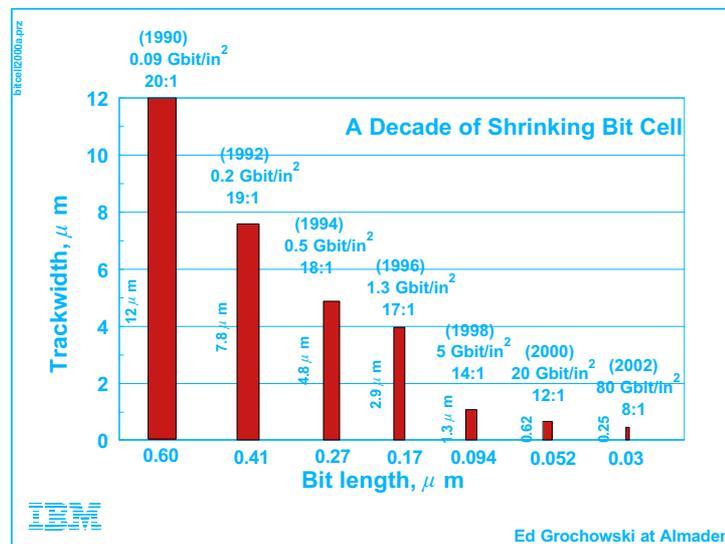


Abbildung 3.2: Die Entwicklung der Speicherdichte ist sehr anschaulich an der Größe einer Bitzelle zu erkennen (hier grafisch dargestellt). Seit der Einführung von PRML-LeseKanälen mit MR-LeseKöpfen sind sowohl die lineare Speicherdichte, als auch die Spurdichte massiv gestiegen.

– Datenübertragungsrate

Die maximale interne Datenübertragungsrate hängt von der linearen Speicherdichte und der Umdrehungszahl der Festplatte ab, so lange die Festplatten-Elektronik die ankommenden Daten weiter verarbeiten kann. Der starke Anstieg der Datendichte (Abbildung 3.1) hat am stärksten zu den hohen Übertragungsraten beigetragen. Die Umdrehungszahl konnte zwar auch deutlich auf bis zu 15000 Umdrehungen pro Minute gesteigert werden, allerdings ist der Zuwachs bei weitem nicht so groß wie bei der Datendichte. Die internen Datenübertragungsraten erreichen inzwischen Werte von über 1 *Gbit/s*.

– Kosten

Trotz steigender Speicherkapazitäten ist der Preis pro MByte stark gefallen und liegt inzwischen bei unter einem Cent. Dieser Umstand hat Magnetspeicher zu einem der günstigsten Massenspeicher gemacht. Der Preisdruck bleibt trotz der bereits immens gesunkenen Preise sehr groß und wird zu weiteren Vergünstigungen führen. Bei Festplatten für den mobilen Einsatz oder für Server-Anwendungen erfolgt eher eine gezielte Optimierung auf Stromverbrauch resp. die Performance. Die Kosten liegen deshalb höher.

– Leistungsverbrauch

Der zunehmende Einsatz von Festplatten in mobilen Geräten macht den Strom-

verbrauch zu einer wichtigen Größe, die beim Entwurf nicht mehr vernachlässigt werden darf. Die stark angewachsene Speicherdichte hat die Entwicklung von Festplatten mit kleinem Formfaktor bei gleichzeitig hoher Kapazität ermöglicht. Ein IBM Microdrive besitzt zum Beispiel bei den Ausmaßen einer Compact Flash Karte vom Typ II eine Kapazität von vier GByte. Der Einsatz in PDAs oder digitalen Kameras erfordert insbesondere für dieses Modell einen geringen Stromverbrauch. Dadurch muss die Optimierung der Festplatten jetzt noch einen weiteren Parameter berücksichtigen.

Die beschriebenen Anforderungen an den magnetischen Kanal beeinflussen sich gegenseitig und können somit nicht alle gleichermaßen verbessert werden. Dieser Umstand soll an zwei Beispielen verdeutlicht werden:

- Eine aufwändigere Fehlerkorrektur kann zu einer Verbesserung der Bitfehlerrate eingesetzt werden. Dies wird allerdings in der Regel mit einer höheren Hardware-Komplexität und demzufolge mit einem höheren Stromverbrauch einher gehen. Gleichzeitig wird – zumindest falls die Verbesserung durch das Hinzufügen weiterer Redundanz erreicht wurde – die Nutzdatenrate kleiner.
- Soll in erster Linie eine höhere Speicherkapazität erzielt werden, so kann dies durch eine größere lineare Bitdichte erreicht werden. Die Folge ist eine erschwerte Detektion und eine erhöhte Fehlerrate. Zur Kompensation müssen entweder aufwändigere Fehlerschutzcodes eingesetzt werden oder die Redundanz der bestehenden Codes erhöht werden. Die Konsequenz ist eine höhere Komplexität – mit steigender Verlustleistung und höheren Kosten – und im zweiten Fall zusätzlich eine wieder abnehmende Dichte der Nutzdaten.

Somit ist das Entwicklungsziel, einen geeigneten Kompromiss zu finden, der allen Kriterien gerecht wird. In Abhängigkeit von dem geplanten Einsatzzweck der Festplatte kann allerdings eine gezielte Optimierung auf einen Parameter erfolgen. Somit wird je nach Anwendung eine andere Eigenschaft zum wichtigsten Entwicklungsziel. Im Folgenden wird ein möglicher Weg für den Entwurf einer Festplatte unter Berücksichtigung des Leistungsverbrauchs aufgezeigt.

3.2 Leistungsarmer Entwurf bei hohen Datendichten

3.2.1 Motivation

Der Entwurf einer mobilen Festplatte unterscheidet sich von dem eines Desktop- bzw. Server-Modells. Im mobilen Einsatz ist der Leistungsverbrauch eine kritische Größe. Da dort in der Regel keine besonders leistungs- oder speicherintensiven Anwendungen eingesetzt werden, sind bezüglich der Speicherkapazität und der Datenübertragungsrate Kompromisse denkbar. Ebenso ist der Preisdruck nicht so hoch wie bei Desktop-Modellen. Die Zuverlässigkeit der ausgelesenen Daten muss allerdings auch bei mobilen Festplatten erhalten bleiben, um einen fehlerfreien Datenstrom zu gewährleisten.

Trotz der geringeren Speicherkapazität weisen mobile Festplatten aufgrund der kleineren Formfaktoren sehr hohe Datendichten auf. Beim IBM Microdrive ist diese zum Beispiel in etwa so hoch wie bei aktuellen Desktop-Festplatten (Abbildung 3.1). Wie eingangs beschrieben, ergibt sich aus einer höheren Datendichte ein höherer Aufwand für die Signalverarbeitung, woraus ein Anstieg der Schaltungskomplexität resultiert. Dieser Effekt wird sich mit weiter steigenden Datendichten verstärken. Das Ziel beim Entwurf eines Festplatten-Lesekanals für mobile Anwendungen ist, alle beschriebenen Anforderungen mit einem möglichst geringen Stromverbrauch zu erfüllen. Die nahe liegende Lösung, die Zuverlässigkeit durch Fehlerschutz-Codes mit größerer Redundanz zu verbessern, führt – wie eingangs beschrieben – zu einer höheren Hardware-Komplexität.

Eine viel versprechende Alternative ist das Ausnutzen zusätzlicher Informationen im Datenstrom. Dies kann über die Verwendung von so genannter Zuverlässigkeitsinformation (im Weiteren auch Soft-Information oder Soft-Output genannt) geschehen. Normalerweise wird bei der Detektion eine "harte" Entscheidung getroffen, die zur Ausgabe einer Null oder einer Eins führt. Dabei geht jedoch wichtige Information verloren. Die im Detektor eingesetzten Algorithmen, wie zum Beispiel der Viterbi-Algorithmus, lassen sich derart erweitern, dass zusätzlich zur Bitentscheidung eine Angabe über die Wahrscheinlichkeit dieser Entscheidung ausgegeben wird. Die Implementierung des so genannten Soft-Output Viterbi Algorithmus [HH89] hat zudem gezeigt, dass Zuverlässigkeiten mit nur geringen Veränderungen der Hardware angegeben werden können [NKRA96].

Das Ziel bei der Verwendung von Zuverlässigkeitsinformation ist eine Verbesserung der Fehlerkorrektur. Kommt dort ein Decoder zum Einsatz, der diese Information auswerten und verarbeiten kann, ist bei gleicher Coderate eine niedrigere Fehlerrate möglich. Durch diese so genannte Soft-Decodierung (oder Soft-Decoding) ist ein Codiergewinn von bis zu 3 dB auf dem magnetischen Kanal erreichbar [KWM91, CHIW98].

Die Implementierung eines Lesekanals mit Soft-Information erfordert Anpassungen und Modifikationen an allen signalverarbeitenden Komponenten. Viele der dazu geeigneten Algorithmen sind sehr komplex und müssen des Weiteren an den magnetischen Kanal angepasst werden. Dennoch ist es eine sehr leistungsfähige Methode zur Verbesserung der Zuverlässigkeit. Im Folgenden wird ein systemübergreifendes Konzept vorgestellt, das für die Konstruktion eines Lesekanals für mobile Festplatten verwendet wird. Daher ist eine geringe Komplexität der eingesetzten Komponenten und Algorithmen ein wichtiges Designziel, das durch eine Systemoptimierung erreicht werden soll.

3.2.2 Vorgehensweise

Bevor die Vorgehensweise bei der Konstruktion eines Soft-Decoding Lesekanals beschrieben wird, erfolgt eine formale Definition der Zuverlässigkeitsinformation [HOP96]. Für eine binäre Zufallsvariable $x \in X = \{+1, -1\}$, wobei $+1$ das neutrale Element der Addition ist, wird die Zuverlässigkeitsinformation (auch: Log-Likelihood Wert) $L(x)$ definiert als

$$L(x) = \ln \frac{P(x = +1)}{P(x = -1)}. \quad (3.1)$$

Das Vorzeichen von $L(x)$ gibt den binären Wert von x an und Betrag $|L(x)|$ gibt die Zuverlässigkeit für diese Variable an. Falls x von einer anderen Zufallsvariable y abhängt, ist der Log-Likelihood Wert $L(x|y)$ gegeben durch

$$L(x|y) = \ln \frac{P(x = +1|y)}{P(x = -1|y)}. \quad (3.2)$$

Mit Hilfe der Regel von Bayes ergibt sich

$$L_{app}(\hat{x}) = \ln \frac{P(y|x = +1)}{P(y|x = -1)} + \ln \frac{P(x = +1)}{P(x = -1)} \quad (3.3)$$

$$= L(y|x) + L(x)$$

$$= L_{ch}(y|x) + L_a(x). \quad (3.4)$$

Die Bedeutung der drei Terme aus (3.4) soll an einem Beispiel verdeutlicht werden. Über einen gestörten Kanal werde eine Variable $x \in X = \{+1, -1\}$ übertragen. Am Empfänger werde die verrauschte Variable y beobachtet. Über die Kanalcharakteristik kann $P(y|x = \pm 1)$ bestimmt werden [Bos99], also die Wahrscheinlichkeit, dass y empfangen wird, wenn $x = \pm 1$ gesendet wurde. Die sich daraus ergebende Zuverlässigkeitsinformation $L(y|x)$ wird als Kanalzuverlässigkeit $L_{ch}(y|x)$ bezeichnet. Der zweite Summand aus (3.4) drückt die Wahrscheinlichkeit für $x = \pm 1$ aus. Hiermit kann angegeben werden, wenn die beiden binären Werte ± 1 nicht mit der gleichen Wahrscheinlichkeit auftreten. Daher wird $L(x)$ als *a priori* Zuverlässigkeit $L_a(x)$ bezeichnet. Die Summe aus beiden Termen $L(x|y)$ ist die *a posteriori* Zuverlässigkeit $L_{app}(\hat{x})$, die während der Detektion oder während der Decodierung bestimmt wird.

Nach der formalen Definition der Zuverlässigkeitsinformation kann die Implementierung eines Soft-Decoding Lesekanals erfolgen. Dazu sind die folgenden Schritte notwendig:

- Die erste Aufgabe ist die Berechnung der Zuverlässigkeitsinformation im Detektor. An dieser Stelle muss ein geeignetes Kriterium entwickelt werden, durch das

jedem Bit eine Fehlerwahrscheinlichkeit (oder etwas dazu Äquivalentes) zugeordnet werden kann. Der Detektor muss die Zuverlässigkeitsinformation bei geringem, zusätzlichem Schaltungsaufwand extrahieren.

- Je nach eingesetztem Fehlerschutzcode muss ein Verfahren zur Handhabung korrekt detektierter Bits mit geringer Zuverlässigkeit entwickelt werden.
- Der Modulationsdecoder muss in der Lage sein, die Zuverlässigkeitsinformation zu verarbeiten und an den Fehlerschutzdecoder weiterzureichen.
- Der Fehlerschutzdecoder muss die Zuverlässigkeitsinformation auswerten können. Bei seiner Schaltungsimplementierung muss besonderer Wert auf eine geringe Verlustleistung gelegt werden.

3.2.3 Berechnung der Zuverlässigkeitsinformation

Bei der Detektion wird eine Entscheidung aufgrund einer Metrik getroffen. Diese ermöglicht es, das wahrscheinlichste Ausgabebit zu bestimmen. Die Berechnung von Zuverlässigkeitsinformation bzw. Soft-Outputs erfolgt im Allgemeinen durch die Betrachtung von Alternativen, indem die Wahrscheinlichkeiten für beide möglichen Detektorausgaben berücksichtigt werden. Sind beide Wahrscheinlichkeiten ähnlich groß, ist die Zuverlässigkeit der Ausgabe gering (vgl. (3.1)).

Bei Detektionsverfahren, die auf dem Viterbi-Algorithmus basieren, (zum Beispiel EPRA PRML) kann die Zuverlässigkeitsinformation recht einfach durch die Berücksichtigung verworfener Pfade im Trellis bestimmt werden [HH89]. Allerdings hat Stromberg [Str00] gezeigt, dass PRML Detektoren anderen Verfahren, wie zum Beispiel Signalraum- oder FDTS-Detektoren [BM96b, MJ98], in mehrfacher Hinsicht unterlegen sind. Der Signalraumdetektor, insbesondere der für farbiges Rauschen optimierte Noise Whitening Signal Space Detector (WSSD) [SHS00], ermöglicht sowohl eine leistungsärmere Implementierung als auch eine geringere Bitfehlerrate bei gegebenem SNR.

Der WSSD-Detektor verwendet einen geometrischen Ansatz zur Detektion. Zur Bestimmung eines einzelnen Bits müssen aufgrund der Intersymbol-Interferenz mehrere von der Festplatte gelesene Samples betrachtet werden. Die letzten L gelesenen Samples werden als Punkt in einem L -dimensionalen Signalraum aufgefasst. Der Signalraum wird zunächst aus Signalpunkten gebildet, die sich aus allen erlaubten, unverrauschten Sequenzen ergeben. Danach lässt sich der Signalraum in Bereiche partitionieren, die den beiden möglichen Detektorausgaben Null bzw. Eins entsprechen. Die Bereiche werden durch Hyperebenen – so genannte Entscheidungsebenen – getrennt. Die Detektion eines verrauschten Empfangspunktes erfolgt schließlich durch eine Lagebestimmung dieses Punktes relativ zu den Entscheidungsebenen.

Für Signalraumdetektoren wurde für den optischen Kanal einer DVD ein Kriterium hergeleitet, das aus dem Abstand eines Empfangspunktes zur begrenzenden Entscheidungsebene eine Berechnung der Zuverlässigkeit erlaubt [SMO01]. Je näher ein Empfangspunkt an einer Entscheidungsebene liegt, desto größer ist die Wahrscheinlichkeit eines Detektionsfehlers. Andererseits entspricht ein Empfangspunkt, der sehr nahe bei

den den Signalraum beschreibenden Vektoren liegt, einer Detektorausgabe mit einer sehr geringen Fehlerwahrscheinlichkeit.

Der Kanal einer DVD ist jedoch weit weniger komplex als der magnetische Kanal, so dass zur Konstruktion des Detektors eine einzelne Entscheidungsebene ausreicht [SMO01]. Das Verfahren wurde später auf den magnetischen Kanal angepasst [Ste02], dessen Signalraum aus mehreren Ebenen konstruiert ist [BM96b]. Die dabei vorgestellte Implementierung basiert jedoch nicht auf der sehr effizienten WSSD-Architektur. Statt dessen wird ein aufwändigerer Ansatz gewählt, der eine Neukonstruktion des Detektors zur Folge hat. In Kap. 4.5 wird daher ein Soft-Output WSSD-Detektor für den magnetischen Kanal entworfen, der nur geringe Hardware-Modifikationen im Vergleich zur Hard-Output Architektur erfordert und sich dadurch insbesondere für einen mobilen Soft-Decoding Kanal eignet.

3.2.4 Lokalisierung von Detektionsfehlern mit Soft-Information

Unabhängig von dem Kriterium zur Bestimmung der Soft-Outputs werden sowohl korrekte als auch falsche Bits niedrige Zuverlässigkeiten aufweisen. Falls der Abstand zu den Entscheidungsebenen betrachtet wird, ist dies auch anschaulich nachzuvollziehen: Wenn ein verrauschter Empfangsvektor sehr nahe an einer Entscheidungsebene liegt, so hat er unabhängig davon, ob er sich auf der korrekten Seite der Ebene befindet, eine geringe Zuverlässigkeit. Falls der Empfangsvektor auf der falschen Seite der Ebene liegt, ist er stärker verrauscht als auf der richtigen Seite. Aus diesem Grund werden statistisch gesehen mehr korrekte als falsche Bits mit einer geringen Zuverlässigkeit auftreten.

Dieser Umstand zeigt, dass die alleinige Betrachtung der Soft-Information nicht für eine zuverlässige Lokalisierung der Fehler ausreicht. Dies ist weniger kritisch, wenn fehlerkorrigierende Codes eingesetzt werden, die direkt Zuverlässigkeitinformation verarbeiten können, wie zum Beispiel Turbo Codes [BGT93] oder LDPC Codes [Gal62]. In diesem Fall reicht allein die statistisch höhere Fehlerwahrscheinlichkeit eines Bits mit geringer Zuverlässigkeit zu einer Verbesserung der Fehlerrate aus.

Andere Codes, zum Beispiel die bei Festplatten in der Regel eingesetzten Reed-Solomon Codes [RS60], sind zumindest in ihrer ursprünglichen Fassung nicht zur Verarbeitung von Zuverlässigkeitinformation in der Lage. In diesem Fall wird von Hard-Decoding gesprochen. Die Korrekturfähigkeit von Reed-Solomon Codes kann aber erheblich verbessert werden, wenn die exakten Fehlerpositionen bekannt sind (so genannte Erasures) und der Decoder die gesamte Redundanz zur Berechnung der Fehlerwerte nutzen kann. Die Fehlerpositionen sollen mittels der Zuverlässigkeitinformation identifiziert werden. Korrekte Bits mit geringer Zuverlässigkeit führen jedoch dazu, dass sie als Fehler betrachtet werden. Die Folge ist, dass Redundanz verschwendet wird, die zur Korrektur weiterer, unentdeckter Fehler verwendet werden könnte.

Aus diesem Grund wird ein Verfahren benötigt, das aus der Gruppe der unzuverlässigen Bits die Fehlerpositionen ermittelt. In Kapitel 6.4.1 wird zu diesem Zweck ein weiterer innerer Code verwendet. An dieser Stelle empfiehlt sich beispielsweise ein einfacher, über nur wenige Code-Symbole reichender Parity-Check Code. Dieser wird an die Fehlermuster des Detektors bzw. des Modulationsdecoders derart angepasst, dass häufig

auftretende Fehlermuster erkannt und an den entsprechenden Stellen Fehler lokalisiert werden können. Falls die Decodierung dieses inneren Codes fehlschlägt, können die unzuverlässigsten Symbole als Fehlerpositionen gekennzeichnet werden [SSHS03]. Ein solcher Parity-Check Code bietet zudem den Vorteil einer einfachen Implementierung, die sich mit einer niedrigen Verlustleistung betreiben lässt.

3.2.5 Entwicklung eines Soft-In/Soft-Out Modulationsdecoders

Bei der Verwendung eines Modulationscodes muss auch der Decoder für die Verarbeitung von Zuverlässigkeitsinformation ausgelegt werden. Dazu muss der Einfluss einzelner Eingangsbits auf den Decoderausgang berechnet werden. Dies bedeutet, dass die Zuverlässigkeitsinformation der Eingangsbits umgerechnet werden muss. Außerdem tritt auch im Modulationsdecoder Fehlerfortpflanzung auf, die bei der Berechnung der Soft-Outputs berücksichtigt werden muss.

Bisherige Untersuchungen haben gezeigt, dass der Einfluss des Modulationscodes auf die Zuverlässigkeit der Detektion gering ist: Bei einigen Verfahren tritt unter Verwendung eines Modulationscodes sogar eine Verschlechterung auf [Str00]. Bei Signalraumdetektoren kann durch die Wahl eines geeigneten MTR-Codes allerdings eine Reduktion der Ebenenanzahl und somit eine Reduktion des Hardware-Aufwands erreicht werden [Str00]. Hinsichtlich des Codiergewinns ist zumindest bei den aktuellen Datendichten und bei den derzeit verwendeten Modulationscodes auch bei Signalraumdetektoren keine entscheidende Verbesserung zu erzielen. Bei den steigenden Datendichten ist aber davon auszugehen, dass der Einfluss des Modulationscodes wieder zunimmt.

Dennoch muss auch bei Signalraumdetektoren die Verwendung eines MTR-Modulationscodes in Frage gestellt werden. Untersuchungen müssen zeigen, ob künftig die Verschlechterung der Coderate durch den Codiergewinn gerechtfertigt werden kann oder ob die Redundanz nicht im Bereich der Fehlerkorrektur besser genutzt werden könnte. Der in Kap. 3.2.4 vorgeschlagene Parity-Check Code kann beispielsweise derart modifiziert werden, dass er als Modulationscode fungiert. Somit könnte ein weiterer Modulationscode eingespart werden. Ein derart angepasstes System ist auch aus Sicht des Leistungsbedarfs von Vorteil.

Beide Varianten – der Einsatz eines Soft-In/Soft-Out Modulationsdecoders oder eines Parity-Check Codes als Modulationscode – werden sowohl hinsichtlich des Hardware-Aufwands, als auch mit Blick auf die Zuverlässigkeit des Gesamtsystems im weiteren Verlauf dieser Arbeit miteinander verglichen. Zu diesem Zweck wird in Kap. 5.2 zunächst ein Soft-In/Soft-Out Modulationscode entworfen. Dieser wird dann in einen Soft-Decoding Kanal integriert und in Kap. 7.5.2 mit einem Kanal verglichen, der auf einen Parity-Code als Modulationscode zurückgreift.

3.2.6 Auswahl eines Fehlerschutzcodes

Das Hauptziel bei der Verwendung von Zuverlässigkeitsinformation ist eine Verbesserung der Fehlerkorrektur. Die grundsätzliche Idee ist nicht neu. Bereits in den sechziger

Jahren hat Forney mit dem Generalized Minimum Distance (GMD) Decoding ein Verfahren zur Verwendung von Soft-Information veröffentlicht [For66]. Trotz einer Reihe weiterer, wichtiger Publikationen auf diesem Gebiet [BCJR74, BDG79] haben Decodierverfahren unter Verwendung von Soft-Information erst in den neunziger Jahren mit der Vorstellung der Turbo Codes [BGT93] wieder Aufmerksamkeit erlangt. Die dabei zum Einsatz kommenden Algorithmen verwenden eine Kombination aus iterativer Decodierung verketteter Codes mit Soft-Decoding und sind daher sehr komplex. Dennoch wurden Turbo Decoder auch unter dem Gesichtspunkt geringer Implementierungskosten [WLW01, TVGW02] oder hoher Datendurchsätze [TWR02, TGV⁺02] untersucht. Auch im Bereich magnetischer Aufzeichnung wird der Einsatz von Turbo Codes in Erwägung gezogen [Rya98, Hee98, MMH98, SFO⁺99]. Ein detaillierter Vergleich mit anderen Verfahren auf unterschiedlichen Lesekanälen steht allerdings noch aus.

Eine möglicherweise besser geeignete Variante stellen die Low-Density Parity-Check (LDPC) Codes dar [Gal62, Mac99]. Diese sind ebenso wie die Turbo Codes zu einer direkten Verarbeitung von Zuverlässigkeitsinformation in der Lage. Im Vergleich zu diesen besitzen LDPC Codes jedoch einige Vorteile, die bei einer Verwendung für den magnetischen Kanal wichtig sind. Zum einen haben sie gerade bei einem hohen SNR, wie es für den magnetischen Kanal üblich ist, eine bessere Decodierfähigkeit [Mac99]. Zum anderen lässt sich der Decodieralgorithmus gut parallelisieren. Dadurch sind Schaltungen mit hohem Datendurchsatz bei gleichzeitig geringer Taktfrequenz möglich [HB01]. Auch LDPC Codes waren bereits ein ausführlicher Forschungsgegenstand für die Signalverarbeitung auf Festplatten-Lesekanälen [STC01, OM01]. In Kap. 7 werden sowohl Turbo Codes als auch LDPC Codes untersucht. Die Verfahren müssen dazu zunächst hinsichtlich der Codelänge und der Coderate an den magnetischen Kanal angepasst werden.

Auch für die bei Festplatten üblicherweise eingesetzten Reed-Solomon Codes ist bereits eine Möglichkeit zur Verarbeitung von Soft-Information vorgestellt worden. Kamiya hat 1997 einen abgeänderten Berlekamp-Massey Algorithmus präsentiert, der Zuverlässigkeitsinformation direkt verarbeiten kann [Kam97]. Koetter hat darüber hinaus im Jahr 2002 ein algebraisches Verfahren zum Soft-Decoding von Reed-Solomon Codes präsentiert [KV02]. Beide Ansätze stellen aus algorithmischer Sicht sicherlich eine sehr gute Lösung dar. Die Problematik einer effizienten Implementierung beider Verfahren ist allerdings bislang ungelöst.

Mit der in Kap. 3.2.4 beschriebenen Erasure-Decodierung existiert allerdings eine einfachere Möglichkeit, Soft-Information bei Reed-Solomon Codes zu verarbeiten. Durch diese Maßnahme lässt sich die Korrekturfähigkeit des Codes im Idealfall verdoppeln [WB94]. Die Implementierung einer solchen Erasure-Decodierung ist relativ einfach: Das durch den Berlekamp-Massey Algorithmus berechnete Fehlerstellen-Polynom muss nur mit den bereits bekannten Positionen initialisiert werden.

Daher ist der Ansatz der Erasure-Decodierung im Hinblick auf eine leistungsarme Implementierung vielversprechend. Sie ist allerdings nur wirksam, wenn sehr selten korrekt erkannte Symbole als mögliche Fehlerpositionen ausgewiesen werden. Dies unterstreicht die Notwendigkeit für eine Lösung des bereits erläuterten Problems der Fehlerlokalisierung. Eine genaue Evaluierung der hier vorgestellten fehlerkorrigierenden Codes erfolgt in Kapitel 6 und 7. Die Codes werden in Bezug auf Decodierfähigkeit und Verlustleistung mit den bei Festplatten populären Reed-Solomon Codes mit Hard-Decoding sowie mit

den iterativen Turbo und LDPC Codes verglichen.

3.3 Zusammenfassung

Auf dem magnetischen Kanal werden bislang in erster Linie Reed-Solomon Codes in Verbindung mit PRML Detektoren eingesetzt. Zusätzlich ist zur Erzeugung von Timing-Information der Einsatz eines Modulationscodes unerlässlich. Diese Kombination hat sich als sehr leistungsfähig bei den auftretenden Fehlermustern erwiesen. Neben einer hohen Signalzuverlässigkeit müssen aber auch andere Voraussetzungen (Speicherkapazität, Datenübertragungsrate, Kosten) erfüllt sein. Der Einsatz von Festplatten in mobilen Geräten lässt die Verlustleistung zu einem weiteren wichtigen Designziel werden. Sollen bei zukünftigen Festplattengenerationen mit höheren Datendichten verbesserte Fehlerkorrekturverfahren zum Erhalt der Signalzuverlässigkeit eingesetzt werden, so müssen beim Entwurf der Festplatte auch die anderen Designziele berücksichtigt werden.

Der Einsatz von fehlerkorrigierenden Codes unter Verwendung von Zuverlässigkeitsinformation erlaubt beispielsweise eine Verbesserung der Fehlerraten ohne eine gleichzeitige Beeinträchtigung der Datenübertragungsrate. Dazu sind allerdings Veränderungen am gesamten System erforderlich: Die Zuverlässigkeitsinformation muss im Detektor berechnet und von den anderen Komponenten des magnetischen Kanals weiterverarbeitet oder ausgewertet werden. Darüber hinaus sind viele aus der Literatur bekannte Soft-Decoding Verfahren, wie zum Beispiel Turbo Codes, sehr komplex und müssen zudem an den magnetischen Kanal angepasst werden, da sie für andere Einsatzzwecke konzipiert wurden. Allerdings ermöglicht die Verwendung von Zuverlässigkeitsinformation einen Codiergewinn von bis zu 3 dB.

Um den Zusatzaufwand für die Signalverarbeitung so gering wie möglich zu halten, werden bei der Bestimmung und Weiterverarbeitung der Zuverlässigkeitsinformation häufig mathematische Näherungen in Betracht gezogen. In der Literatur wurden unter diesem Gesichtspunkt bereits zahlreiche algorithmische Vereinfachungen und Näherungen veröffentlicht [NKRA96, RVH95, Ber93], die eine sehr gute Decodierleistung bei moderater Komplexität erzielen. Auf solche Verfahren wird auch in dieser Arbeit zurückgegriffen. Eine zwingende Voraussetzung ist allerdings eine weitgehende Erhaltung der Signalzuverlässigkeit.

Bei einer späteren Implementierung des Systems in Hardware muss unter der Zielsetzung einer leistungsarmen Schaltung eine Realisierung von möglichst geringer Komplexität entstehen. Aus diesem Grund wird in dieser Arbeit ein systemübergreifendes Konzept zur Konstruktion eines Festplattenlesekanals präsentiert. Im Gegensatz zu einer alleinigen Optimierung der Einzelkomponenten erlaubt dieses Konzept durch die gleichzeitige Betrachtung aller Komponenten eine weitere Optimierung des Lesekanals unter Berücksichtigung aller Designziele.

Kapitel 4

Detektionsverfahren mit Zuverlässigkeitsinformation

Im Detektor erfolgt die Umwandlung des von der Festplatte gelesenen, kontinuierlichen Datenstroms in eine binäre Sequenz. Der Detektor besitzt somit eine große Bedeutung, da er mit einer möglichst fehlerfreien Signalverarbeitung eine Grundvoraussetzung für ein zuverlässiges Ausgangssignal schafft. Bei der Rekonstruktion der binären Information entstehen allerdings unweigerlich durch die in Kapitel 2.2.2 und 2.3.3 beschriebenen Störgrößen Fehler. Des Weiteren führen Feedback Schleifen in einigen Detektor-Architekturen zu Fehlerfortpflanzung.

In diesem Kapitel werden verschiedene Detektionsverfahren für den magnetischen Kanal miteinander verglichen. Wie in Kapitel 3.2 beschrieben, soll ein Detektor entwickelt werden, der eine niedrige Fehlerrate bereits bei einer geringen Hardware-Komplexität erreicht und zudem zu einer Ausgabe von Zuverlässigkeitsinformation in der Lage ist.

Seit den frühen neunziger Jahren werden in Festplatten Partial Response Maximum Likelihood (PRML) Lesekanäle eingesetzt. Die verwendeten Target-Polynome sind dabei immer an die Datendichte angepasst worden. Anfangs wurden Partial Response Polynome vom Typ IV (PR4) verwendet, die dann mit steigenden Datendichten durch Polynome höherer Ordnung ersetzt wurden. Für den Viterbi-Algorithmus, der bei PRML-Kanälen meist zur eigentlichen Detektion eingesetzt wird [For73], existiert eine Erweiterung zur Ausgabe von Zuverlässigkeitsinformation (Soft Output Viterbi Algorithmus, SOVA [HH89]).

Unter dem Gesichtspunkt einer möglichst einfachen Implementierung sind allerdings Lesekanäle nach dem Fixed Delay Tree Search (FDTS) Algorithmus besser geeignet [Str00]. Diese arbeiten ähnlich wie der Viterbi-Algorithmus, allerdings mit einer kurzen und festgelegten Suchtiefe. Die so genannten Signalraumdetektoren, als eine geometrische Interpretation eines FDTS-Detektors, lassen sich besonders einfach implementieren und sind somit gerade für einen Einsatz in mobilen Festplatten geeignet.

Der WSSD Detektor [SHS00], ein Signalraumdetektor mit einer Rausch-Entfärbung, wurde in Kapitel 3 als eine effiziente Architektur vorgestellt. An dieser Stelle erfolgt eine Erweiterung um die Ausgabe von Zuverlässigkeitsinformation. Dabei wird der von

Steingrimsson für den optischen Kanal einer DVD präsentierte Algorithmus [SMO01] für den WSSD Detektor angepasst. Im Gegensatz zu anderen Soft-Output Signalraumdetektoren [Ste02], wird in diesem Kapitel ein Ansatz verfolgt, der die Implementierung der Soft-Outputs mit nur sehr geringen Modifikationen der WSSD Hardware realisiert. Zunächst erfolgt aber eine allgemeine Beschreibung des Detektionsprozesses.

4.1 Detektion auf dem magnetischen Kanal

In Kapitel 2 wurde das Kanalmodell für einen Festplatten-Lesekanal diskutiert. Das verauschte Ausgangssignal des Lesekopfes wurde in Gleichung (2.7) als zeit- und wertekontinuierliche Funktion $z(t)$ beschrieben. Dieses Signal gelangt in den Detektor und wird zunächst gefiltert, um hochfrequente Rauschkomponenten zu unterdrücken (Abbildung 4.1). Durch diese Maßnahme wird Alias bei der späteren Signalverarbeitung vermieden. Danach wird das Signal im Abstand der Bitintervalle T abgetastet. Es ergibt sich eine Folge von quasi-kontinuierlichen Abtastwerten \underline{z} (4.1). Die heute übliche Datendichte verursacht eine starke Überlagerung der einzelnen Transitionen des Magnetfelds. Dadurch beeinflusst ein einzelner Lesepuls auf der Festplatte mehrere Abtastwerte (engl. Samples) z_i .

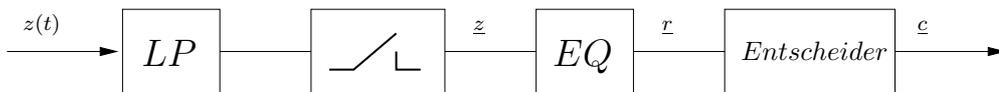


Abbildung 4.1: Die einzelnen Komponenten eines Detektors: Das Signal gelangt zunächst durch ein Low-Pass Filter (LP), um das Rauschspektrum zu beschneiden und wird anschließend abgetastet. Nach dem Equalizer (EQ) erfolgt die eigentliche Detektion im Entscheider.

$$z_i = z(iT) \quad (4.1)$$

$$= \nu_i + \sum_k a_k g((i-k)T, \tau_k) \text{ mit}$$

$$\nu_i = \nu(iT) \quad (4.2)$$

Um die Datenrückgewinnung zu vereinfachen wird im Equalizer eine weitere Filterung vorgenommen. Diese bewirkt, dass das abgetastete Signal einer einzelnen Transition nur zu wenigen, im Vorfeld definierten Zeitpunkten einen Wert ungleich Null besitzt. Durch diese Maßnahme kann die Signalverarbeitung bei der späteren Maximum Likelihood Detektion (siehe unten) vereinfacht werden. Die Ausgabe des Equalizers bei einer einzelnen Transition auf dem magnetischen Medium wird als Target $\underline{\Gamma}$ mit der Länge L_Γ bezeichnet.

Die durch den Equalizer vorgefilterten Abtastwerte gelangen in den Entscheider, der daraus die binäre Detektorausgabe berechnet. Da ein einzelnes Sample aufgrund der

Intersymbolinterferenz nicht genügend Information zur Bestimmung einer Detektorausgabe enthält, müssen mehrere Samples zur Bestimmung eines Ausgabebits berücksichtigt werden. Dabei wird anhand der von der Festplatte gelesenen Samples \underline{r} die Wahrscheinlichkeit für jedes Ausgabebit c_i getrennt maximiert. Damit ergibt sich die Detektorausgabe c_i wie folgt:

$$c_i = \arg \max_{\hat{c} \in \mathcal{C}} P(\hat{c} | \underline{r}) \quad (4.3)$$

Das Verfahren wird als Maximum A-Posteriori (MAP) Decodierung bezeichnet. Dabei steht $\hat{c} \in \mathcal{C}$ für die Menge der möglichen Detektor-Ausgaben – bei einem binären Alphabet für ”-1” oder ”+1”. Somit wird die bedingte Wahrscheinlichkeit $P(\cdot | \cdot)$ maximiert, so dass $P(c_i | \underline{r}) = \max_{\hat{c} \in \mathcal{C}} P(\hat{c} | \underline{r})$. Die wahrscheinlichste Detektor-Ausgabe ist dann $c_i \in \mathcal{C}$. Die Bestimmung von c_i ist allerdings mit aufwändigen Berechnungen verbunden, da (4.3) für jedes detektierte Bit separat bestimmt werden muss. Der nach seinen Erfindern Bahl, Cocke, Jelinek und Raviv benannte BCJR-Algorithmus [BCJR74] ist ein sehr bekanntes Beispiel für eine MAP-Decodierung. Der Algorithmus kann sowohl zur Detektion, als auch zur Decodierung von Block- oder Faltungscodes verwendet werden. Mit der Regel von Bayes lässt sich (4.3) umstellen:

$$P(\hat{c} | \underline{r}) = \frac{P(\underline{r} | \hat{c}) \cdot P(\hat{c})}{P(\underline{r})} \quad (4.4)$$

Die Terme $P(\hat{c})$ und $P(\underline{r})$ stehen für die a priori Wahrscheinlichkeiten der Zufallsvariablen \hat{c} und \underline{r} . So lange $P(\underline{r})$ unabhängig von den möglichen Datenbits aus \mathcal{C} ist, spielt der Term im Nenner von (4.4) für die Maximierung keine Rolle. Somit lässt sich (4.3) zu (4.5) vereinfachen:

$$c_i = \arg \max_{\hat{c} \in \mathcal{C}} P(\underline{r} | \hat{c}) \cdot P(\hat{c}) \quad (4.5)$$

Sofern alle möglichen Detektorausgaben gleich wahrscheinlich sind und nicht etwa durch einen Modulationscode eingeschränkt werden, kann der Term $P(\hat{c})$ ebenfalls entfallen. Der Term $P(\underline{r} | \hat{c})$ lässt sich über die Kanalcharakteristik bestimmen [Pro95]. In Kap. 4.2.2 wird eine Lösung für stationäres, weißes Rauschen (AWGN) angegeben.

Alternativ zur Maximierung der Wahrscheinlichkeit für jedes detektierte Bit kann eine ganze Sequenz abgeschätzt und somit mehrere Bits gleichzeitig bestimmt werden. Dieses Verfahren ist unter der englischen Bezeichnung Maximum Likelihood Sequence Estimation (MLSE) bekannt. Es ist weniger komplex als die zuvor beschriebenen MAP-Algorithmen und wird aus diesem Grund häufig zur Detektion auf dem magnetischen Kanal verwendet. Meist kommt zu diesem Zweck der Viterbi-Algorithmus zum Einsatz, der ursprünglich zur Decodierung von Faltungscodes konzipiert wurde [Vit67]. Forney konnte schließlich die Eignung des Viterbi-Algorithmus zur Detektion zeigen [For73]. Analog zu (4.3) ergibt sich:

$$\underline{c} = \arg \max_{\hat{c} \in \mathcal{C}} P(\hat{c} | \underline{r}) \quad (4.6)$$

Durch die Anwendung der Regel von Bayes und den oben getroffenen Vereinfachungen lässt sich (4.6) wie folgt vereinfachen:

$$\underline{c} = \arg \max_{\hat{c} \in \mathcal{C}} P(\underline{r} | \hat{c}) \quad (4.7)$$

Bei der Berechnung von c_i in (4.5) bzw. \underline{c} in (4.7) wird über das MAP- oder das MLSE Kriterium die wahrscheinlichste Detektorausgabe bestimmt. Wie bereits in Kapitel 3 beschrieben, soll in dieser Arbeit jedoch ein Detektor konstruiert werden, der zusätzlich zu der binären Ausgabe eine Zuverlässigkeitinformation ausgibt. Bei der Berechnung der Detektorausgabe ist es ausreichend, über die Maximierung des wahrscheinlichsten Ausgabebits oder die wahrscheinlichste Ausgabesequenz zu bestimmen. Weitere Informationen über die Wahrscheinlichkeit anderer Detektorausgaben können verworfen werden. Zur Angabe einer Zuverlässigkeitinformation müssen aber auch Alternativen betrachtet werden. Dazu wird bestimmt, wie groß die Wahrscheinlichkeit der Detektorausgabe c_i im Vergleich zur inversen Detektorausgabe $\tilde{c}_i = -c_i$ ist. Hagenauer *et.al.* [HOP96] haben die Zuverlässigkeit als so genanntes Log-Likelihood Verhältnis dieser beiden Wahrscheinlichkeiten beschrieben. Die Zuverlässigkeitinformation $L(c_i)$ für das detektierte Bit c_i ergibt sich damit als

$$L(c_i) = \ln \frac{P(c_i = +1 | \underline{r})}{P(c_i = -1 | \underline{r})}. \quad (4.8)$$

Das Vorzeichen von $L(c_i)$ kennzeichnet die binäre Detektorausgabe und der Absolutbetrag die Zuverlässigkeitinformation. Je unterschiedlicher die Wahrscheinlichkeiten in Zähler und Nenner sind, desto sicherer ist die vom Detektor getroffene Entscheidung. Sind beide Wahrscheinlichkeiten gleich groß, hat die Zuverlässigkeit den Wert Null – es ist keine Aussage möglich.

Dieses Prinzip lässt sich auf die verschiedenen Detektionsprinzipien anwenden, die im weiteren Verlauf dieses Kapitels vorgestellt werden. In Kap. 4.2 und 4.5 werden Soft-Output Algorithmen für PRML- und für Signalraumdetektoren vorgestellt.

4.2 Partial Response Maximum Likelihood (PRML)

Bis zu Beginn der neunziger Jahre wurden in Festplatten so genannte Peak-Detektoren eingesetzt, bei denen jedes Bit durch einen einzelnen, isolierten Puls detektiert wurde. Durch steigende Datendichten und die damit verbundene Intersymbolinterferenz wurde die Erkennung einzelner Pulse jedoch immer schwieriger. Schließlich waren die bestehenden Detektionsverfahren ein Hindernis für weiter steigende Datendichten. Schwellwert-Detektoren, die als Alternative zu den Peak-Detektionsverfahren eingesetzt wurden, haben sich ebenso als ungeeignet für hohe Datendichten erwiesen, da die Intersymbolinterferenz zu einer Verringerung der Signalamplitude (so genannter Roll-Off) und dadurch auch hier zu einer höheren Fehlerrate führt.

Heute werden in Festplatten in der Regel Partial Response Maximum Likelihood (PRML) Lesekanäle eingesetzt. Diese lassen erstmals die Intersymbolinterferenz bewusst zu, wodurch eine wichtige Voraussetzung zur Erzielung heutiger Datendichten erfüllt werden konnte. Ein PRML Lesekanal besteht aus zwei Hauptkomponenten: Ein Partial Response (PR) Vorfilter (Equalizer) und einen Maximum Likelihood (ML) Detektor.

Die PR-Equalizer haben die Aufgabe, die Intersymbolinterferenz in eine vorausberechnete Form zu bringen und ermöglichen somit, dass sie aus den gelesenen Signalen wieder heraus gerechnet werden kann. In Kap. 4.2.1 werden verschiedene Partial Response Kanäle vorgestellt. Die eigentliche Detektion erfolgt durch einen Maximum Likelihood Detektor. An dieser Stelle wird auf PRML-Kanälen meist der Viterbi-Algorithmus eingesetzt. Die Beschreibung dieses eigentlichen Detektionsvorgangs inklusive der Erweiterungen zur Soft-Output Detektion erfolgt in Kap. 4.2.2.

4.2.1 Partial Response-Entzerrung

Bei PRML Lesekanälen wird das Signal zunächst durch ein geeignetes Partial Response Filter derart geformt, dass die Abtastwerte der überlagerten Transitionen nur zu bestimmten Zeitpunkten einen Wert ungleich Null besitzen. Die Ausgabe des Equalizers bei einer einzelnen Transition auf dem magnetischen Medium wird als Target $\underline{\Gamma}$ mit der Länge L_Γ bezeichnet ($\underline{\Gamma} = [\Gamma_0, \dots, \Gamma_{L_\Gamma-1}]$). Der Equalizer-Ausgang r_i ergibt sich durch lineare Superposition der letzten Transitionen a_j des Magnetfeldes zu

$$r_i = \sum_{j=0}^{L_\Gamma-1} a_{i-j} \cdot \Gamma_j. \quad (4.9)$$

Somit lässt sich über $\underline{\Gamma}$ der Zusammenhang zwischen dem aktuellen Abtastwert r_i und den vorher gehenden Transitionen des Magnetfeldes beschreiben. Der aktuelle Abtastwert kann also als Zustand des Kanals angesehen werden, der sich aus den zuletzt gespeicherten Datenbits ergibt. Gleichzeitig wird durch die Partial Response Filterung der Wertebereich der Abtastwerte beschränkt, wodurch der Schaltungsaufwand der nachfolgenden Signalverarbeitung verringert wird (siehe Kap. 4.2.2).

Ein bekanntes Beispiel für einen Partial Response Kanal ist das so genannte Class IV Partial Response (PR4) Polynom. Bei diesem wird das von der Festplatte gelesene Signal derart gefiltert, dass eine isolierte Transition (das heißt die Sprungantwort des Kanals) nur an exakt zwei Stellen einen von Null verschiedenen Abtastwert besitzt: Zu dem Zeitpunkt, an dem die Transition geschrieben wurde ($t = 0$) und einen Abtastwert später ($t = T$). Das Target $\underline{\Gamma}$ ergibt sich somit zu $\underline{\Gamma} = [1, 1]$ mit der Länge $L_\Gamma = 2$. Sofern die Überlagerung der einzelnen Transitionen linear ist und die exakte Form des Lesesignals bei einer isolierten Transition bekannt ist¹, kann auch das Signal mehrerer direkt aufeinander folgender Transitionen bestimmt werden.

Analog zur Vorgehensweise beim Lesesignal $z(t)$ in Kap. 2.3.2 können zwei unterschiedliche Darstellungen des Targets definiert werden: Einerseits kann das Target den Zusam-

¹Im Falle des magnetischen Kanals wird das Signal über den Lorentz-Puls angenähert, siehe Kap. 2.

menhang zwischen den letzten Transitionen a_j des Magnetfeldes (das heißt den letzten Sprungantworten des Kanals) und dem aktuellen Abtastwert r_i herstellen (wie oben beschrieben). Dieses wird im Folgenden als NRZI-Target $\underline{\Gamma}^{(NRZI)}$ bezeichnet, mit dem sich (4.9) zu

$$r_i = \sum_{j=0}^{L_\Gamma-1} a_{i-j} \cdot \Gamma_j^{(NRZI)} \quad (4.10)$$

ergibt. Andererseits kann es hingegen den Zusammenhang zwischen den letzten Schreibstromamplituden b_j (das heißt den letzten Impulsantworten) und dem aktuellen Abtastwert r_i darstellen. In diesem Fall wird es als NRZ-Target $\underline{\Gamma}^{(NRZ)}$ bezeichnet. Die Impulsantwort ergibt sich als lineare Superposition zweier benachbarter Sprungantworten gegensätzlicher Polarität. Damit ergibt sich das NRZ-Target als die Differenz zweier NRZI-Targets zu

$$r_i = \sum_{j=0}^{L_\Gamma-1} b_{i-j} \cdot \Gamma_j^{(NRZ)}. \quad (4.11)$$

Für einen PR4-Kanal ist das NRZI-Target $\underline{\Gamma}^{(NRZI)} = [1, 1]$ mit der Länge $L_{\Gamma^{(NRZI)}} = 2$ und das NRZ-Target $\underline{\Gamma}^{(NRZ)} = [1, 0, -1]$ mit der Länge $L_{\Gamma^{(NRZ)}} = 3$. Daraus wiederum ergibt sich der aktuelle Abtastwert als $r_i = a_i + a_{i-1}$ für das NRZI-Target bzw. $r_i = b_i - b_{i-2}$ für das NRZ-Target.

Vielfach werden die PR-Targets über die so genannten Target-Polynome dargestellt. Diese verwenden die Z-Transformation [Bos99] einer Sequenz \underline{x}

$$\mathcal{Z}\{\underline{x}\} := X(z) = \sum_{t=0}^{\infty} x_t z^{-t}, \quad (4.12)$$

die mit dem Verzögerungsoperator $D = z^{-1}$ und mit

$$X(D) = \sum_{t=0}^{\infty} x_t D^t \quad (4.13)$$

das Targetpolynom $\underline{\Gamma}$ im Frequenzbereich als

$$P(D) = \sum_{j=0}^{L_\Gamma-1} \Gamma_j \cdot D^j \quad (4.14)$$

darstellt. Die PR4-Targetpolynome ergeben sich mit dem Verzögerungsoperator zu $(1 + D)$ für NRZI bzw. $(1 - D)(1 + D) = (1 - D^2)$ für NRZ. Der $(1 - D)$ -Term entsteht dadurch, dass beim Lesen des Signals von der Festplatte nur die Magnetfeldwechsel erfasst werden und daher die Differenz zweier NRZ-Datensignale gebildet wird.

Bei höheren Datendichten steigt der Einfluss der Intersymbolinterferenz, so dass eine isolierte Transition des Magnetfeldes mehr als zwei Abtastwerte beeinflusst. Aus diesem Grund werden Partial Response Polynome einer höheren Ordnung verwendet. Die unterschiedlichen Class IV Polynome P_n werden über ihre Ordnung n , wie in Gleichung (4.15) angegeben, charakterisiert.

$$P_n(D) = (1 - D) \cdot (1 + D)^n \quad (4.15)$$

In dieser Arbeit werden die folgenden Polynome betrachtet: PR4 ($n = 1$), Extended PR4 (EPR4, $n = 2$) sowie eine modifizierte Extended EPR4 (EEPR4, $n = 3$) Variante, das Modified EEPR4 (MEEPR4) Target [Con98] mit dem NRZI-Targetpolynom $2 + 4 \cdot D + 3 \cdot D^2 + D^3$. Dieses Polynom bietet eine besonders gute Übereinstimmung mit der ungefilterten Kanalantwort des magnetischen Kanals, erfordert jedoch aufgrund seiner unsymmetrischen Struktur einen größeren Filteraufwand [Con98]. Bei der Verwendung von PR-Polynomen höherer Ordnung erfordert eine Trellis-Repräsentation zudem mehr Zustände. Dies führt zwar zu einer Verbesserung der Detektionsgüte, aber natürlich auch zu einer deutlich höheren Hardware-Komplexität [Con98]. In der folgenden Tabelle sind die entsprechenden Target-Polynome für NRZI- sowie NRZ-Darstellung angegeben.

Polynom	NRZI-Target	NRZ-Target
PR4	$1 + D$	$1 - D^2$
EPR4	$1 + 2 \cdot D + D^2$	$1 + D - D^2 - D^3$
EEPR4	$1 + 3 \cdot D + 3 \cdot D^2 + D^3$	$1 + 2 \cdot D - 2 \cdot D^3 - D^4$
MEEPR4	$2 + 4 \cdot D + 3 \cdot D^2 + D^3$	$2 + 6 \cdot D + 7 \cdot D^2 + 4 \cdot D^3 + D^4$

Tabelle 4.1: Übersicht der in dieser Arbeit berücksichtigten Partial Response Polynome

Aus den durch die PR-Vorfilterung geformten Abtastwerten muss das Lesesignal zurückgewonnen werden. Nach dem Abtast-Theorem von Nyquist [Pro95] kann ein bandbreitenbegrenztes Analogsignal mit der Cut-Off Frequenz f_{max} aus seinen Abtastwerten vollständig rekonstruiert werden, wenn es im Abstand $T_{min} \leq \frac{2}{f_{max}}$ abgetastet wird. Insbesondere bei niedrigen Datendichten und entsprechend steilen Impulsantworten enthält das Spektrum der Lorentzpulse auch signifikante Komponenten bei Frequenzen oberhalb von $2/T$. Aus diesem Grund arbeiten PR-Kanäle unter diesen Bedingungen nicht optimal.

Die Frequenzen oberhalb von $2/T$ müssen aus dem Signalspektrum herausgefiltert werden, um Alias zu vermeiden. Bei höheren Datendichten und entsprechend breiteren und flacheren Lesepulsen konzentrieren sich die spektralen Anteile zunehmend auf niedrigere Frequenzen, so dass nur ein kleiner Teil des Spektrums abgeschnitten wird. Durch die Beschneidung des Spektrums wird zudem das Signal-Rausch-Verhältnis deutlich verbessert, da vor allem hochfrequente Rauschkomponenten entfernt werden [WT99].

Die Spektren der Class-IV Polynome weisen je nach Datendichte und Target eine gute Übereinstimmung zum Spektrum des Lorentz-Pulses auf. Aus diesem Grund lässt sich der magnetische Kanal mit relativ geringem Filteraufwand in einen Partial Response Kanal umwandeln. Die längeren Polynome bieten dabei insbesondere bei hohen Daten-

ichten eine bessere Anpassung an den Kanal. Im Gegenzug sind allerdings auch mehr Kanalzustände möglich, da ein einzelner Abtastwert von einer größeren Anzahl von gespeicherten Bits beeinflusst wird. Dies wiederum führt zu einer höheren Komplexität im Maximum Likelihood Detektor (siehe Kap. 4.2.2), so dass ein geeigneter Kompromiss zwischen der Target-Länge und der Komplexität der Detektionshardware gefunden werden muss.

4.2.2 Maximum Likelihood Detektion

Aus den gefilterten Abtastwerten müssen im Entscheider die ursprünglichen Datenbits zurückgewonnen werden. Durch die PR-Filterung sind alle Abtastwerte auf einen bestimmten Wertebereich beschränkt. Aus diesem Grund ist es auf Kanälen mit hohem SNR ausreichend, einen Schwellwert-Entscheider einzusetzen [WT99]. Allerdings erfordert der niedrige SNR bei den heute üblichen Datendichten aufwändigere Maximum Likelihood Verfahren. Das Prinzip einer Maximum Likelihood Detektion soll im Folgenden zunächst grob erläutert werden.

- Für die Berechnung eines einzelnen Ausgabebits muss wegen der Intersymbolinterferenz eine Sequenz von Abtastwerten betrachtet werden. Diese wird dann mit allen erlaubten Sequenzen verglichen, die sich auf einem unverrauschten Kanal ergeben können.
- Über ein geeignetes Distanzmaß, zum Beispiel durch Minimierung der Euklidischen Metrik, wird die erlaubte Sequenz gesucht, die der Sequenz von Abtastwerten am ähnlichsten ist.
- Da eine Sequenz zur Bestimmung des Detektionsergebnisses betrachtet wird, kann die Ausgabe erst verzögert erfolgen.

Ein vielfach zur Maximum Likelihood Detektion eingesetztes Verfahren ist der Viterbi-Algorithmus [For73]. Zusammen mit einem PR-Vorfilter bildet der Viterbi-Algorithmus die Kernkomponente eines PRML Lesekanals. Der Algorithmus wird im Folgenden beschrieben.

Viterbi Algorithmus

Grundlage zur Implementierung des Viterbi-Algorithmus ist eine Beschreibung des magnetischen Kanals mit Hilfe eines Zustandsdiagramms. Der Zustand $s^{(n)}$ eines Partial Response Kanals hängt aufgrund der Intersymbolinterferenz nicht nur von der aktuellen Magnetisierung des Mediums, sondern auch von den $L = L_{\Gamma} - 1$ vorhergehenden Medium-Magnetisierungen ab. Im Falle NRZ-codierter Daten entsprechen diese den letzten $L = L_{\Gamma} - 1$ Datenbits. In der folgenden Beschreibung wird von NRZ-codierten Daten ausgegangen.

Ein beliebiger Partial Response Kanal besitzt 2^L Zustände, die sich aus den Werten der letzten L NRZ-Datenbits $c_{i-1}, \dots, c_{i-L_{\Gamma}+1}$ ergeben. Da auf dem magnetischen Me-

dium nur eine binäre Information gespeichert wird, besitzt jeder Zustand genau zwei Folgezustände.

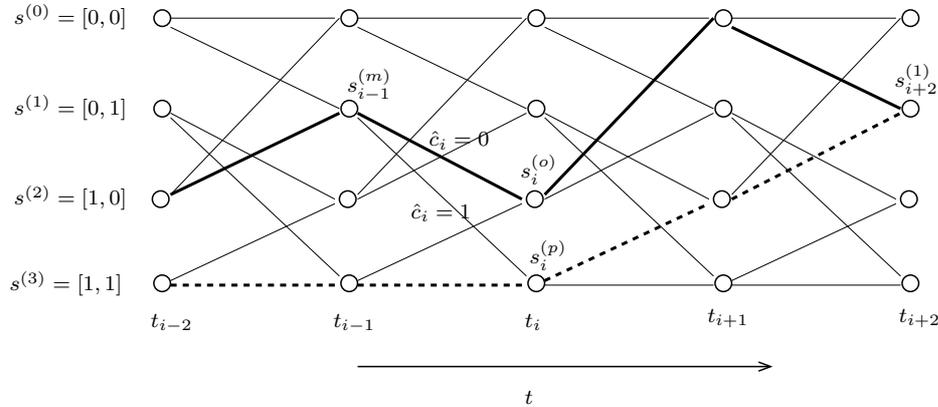


Abbildung 4.2: Die Zustandsänderungen eines PR4-Kanals werden im Trellis über der Zeit t aufgetragen. Jede Verzweigung korrespondiert mit einem Datenbit \hat{c} .

Für den Viterbi-Algorithmus werden die Zustände des magnetischen Kanals an einem so genannten Trellis dargestellt. Dieser beschreibt die Zustandsänderungen über die Zeit. In Abbildung 4.2 ist der Trellis eines PR4 Kanals zu sehen. Auf diesem Kanal beeinflusst eine isolierte Transition des Magnetfeldes zwei aufeinander folgende Abtastwerte. Dies bedeutet, dass die letzten beiden Magnetisierungsrichtungen des Mediums zur eindeutigen Bestimmung des Kanalszustands berücksichtigt werden müssen. Es ergeben sich $2^L = 2^2 = 4$ Zustände $s^{(n)}$ mit $s^{(0)} = [0, 0]$, $s^{(1)} = [0, 1]$, $s^{(2)} = [1, 0]$ und $s^{(3)} = [1, 1]$. Diese werden im Trellis übereinander dargestellt. Die beiden Ziffern entsprechen den zwei vorhergehenden Magnetisierungsrichtungen resp. NRZ-Datenbits. Aus diesem Grund korrespondiert der Zustand $s^{(n)}$ zum Zeitpunkt t_i mit dem aktuellen Abtastwert r_i und den vorherigen NRZ-Datenbits.

In der horizontalen Achse ist die Zeit zu den Abtastzeitpunkten $t_i = i \cdot T$ dargestellt. Zu jedem Abtastzeitpunkt t_i ergeben sich in Abhängigkeit von dem aktuellen Abtastwert r_i und den Kanalszuständen zum Zeitpunkt t_{i-1} 2^L neue Zustände $s^{(n)}$. Diese werden im Folgenden als $s_i^{(n)}$ bezeichnet. Von jedem Zustand gehen aufgrund der binären Daten genau zwei Verzweigungen aus, die den beiden möglichen NRZ-Datenbits entsprechen. Im Trellis entspricht dies zum Beispiel einer Verbindung von Zustand $s_{i-1}^{(m)}$ in den Zustand $s_i^{(o)}$ oder $s_i^{(p)}$ (Abb. 4.2). Auf diese Weise bildet jede erlaubte Datensequenz \hat{c} einen festen Weg durch den Trellis. Dadurch kann sie einer Sequenz von Zuständen $\underline{s}_{\hat{c}}$ und somit einer Sequenz von idealen Abtastwerten $\underline{r}_{\hat{c}}$ zugeordnet werden, die sich auf einem unverrauschten Kanal ergeben würden.

Bei der Decodierung des Viterbi-Algorithmus gelangt zu jedem Takt ein neuer von der Festplatte gelesener Abtastwert r_i in den Detektor. Die Sequenz der Abtastwerte \underline{r} wird im Trellis sequentiell mit allen idealen Sequenzen $\underline{r}_{\hat{c}}$ verglichen. Über ein geeignetes Distanzmaß wird die ähnlichste Sequenz $\underline{r}_{\hat{c}}$ gesucht. Dazu wird mit Hilfe des Euklidischen Abstands zu jedem Zeitpunkt t_i in allen 2^L Zuständen $s_i^{(n)}$ die Wahrscheinlichkeit für das aktuelle Sample r_i berechnet. Durch Multiplikation dieser Einzelwahrscheinlichkeiten für

die Abtastwerte r_i in Abhängigkeit von den idealen Abtastwerten $r_{\hat{c},i}$ ergibt sich eine Wahrscheinlichkeit für \underline{r} unter der Bedingung, dass die zu $\underline{r}_{\hat{c}}$ gehörige Datenfolge \hat{c} gespeichert wurde [Str00]:

$$P(\underline{r}|\hat{c}) = \prod_i \frac{1}{\sqrt{2\pi\sigma}} \cdot \exp \left[-\frac{(r_i - r_{\hat{c},i})^2}{2\sigma^2} \right] \quad (4.16)$$

Zur Bestimmung der wahrscheinlichsten NRZ-Datenfolge \underline{c} muss Gleichung (4.16), wie in Gleichung (4.7) angegeben, maximiert werden. Der Viterbi-Algorithmus nimmt jedoch statt der Maximierung von Gleichung (4.16) eine Minimierung der so genannten Metrik λ vor:

$$\lambda_{\hat{c}} = \sum_i (r_i - r_{\hat{c},i})^2 \quad (4.17)$$

Die Metrik aus Gleichung (4.17) muss zu jedem Zeitpunkt t_i für jeden der 2^L Zustände $s_i^{(n)}$ berechnet werden. Dies erfolgt rekursiv, indem zu jedem Zeitpunkt die Metrik für den aktuellen Zweig hinzuaddiert wird. Diese wird als Branch-Metrik β_i mit $\beta_i = (r_i - r_{\hat{c},i})^2$ bezeichnet.

Da jeder Zustand im Trellis zwei Folgezustände hat, treffen sich in jedem Zustand $s_i^{(n)}$ zwei Pfade. Dies ist in Abb. 4.2 für den Zustand $s_{i+2}^{(1)}$ skizziert. Dabei wird nur der Zweig mit der geringeren Metrik (und somit der höheren Wahrscheinlichkeit) weiter verfolgt und als Survivor bezeichnet. Der andere Pfad wird verworfen. Durch das Streichen von Pfaden wird der Pfad mit der geringsten Metrik $\lambda = \min_{\hat{c}} \lambda_{\hat{c}}$ rekursiv bestimmt. Somit kann der Rechenaufwand erheblich reduziert werden, da nur eine konstante Anzahl von 2^L Metriken resp. Pfaden gespeichert werden muss. Das Streichen von 2^L Pfaden zu jedem Decodierschritt i führt dazu, dass sich alle verbleibenden Pfade nach einer Verzögerung von τ Decodierschritten vereinigen. Es verbleibt ab dem Zeitpunkt $t_{i-\tau}$ nur ein einziger, wahrscheinlichster Pfad $\underline{r}_{\hat{c}}$. Dieser entspricht der wahrscheinlichsten Datensequenz \underline{c} . Der Wert von τ hängt sowohl vom verwendeten Target, als auch von der gespeicherten Datensequenz ab. Die maximale Länge, nach der sich ausgehend vom aktuellen Abtastwert alle Pfade vereinigen, wird als Survivor-Länge τ_{max} bezeichnet.

In der Praxis müssen für eine Implementierung des Viterbi-Algorithmus alle 2^L Pfade gespeichert werden. Um den benötigten Speicherplatz gering zu halten, werden die Pfade nur bis zu einer maximalen Pfadlänge M gespeichert. Danach wird das letzte Bit c_{i-M} des Pfades mit der größten Wahrscheinlichkeit ausgegeben, unabhängig davon, ob sich bereits alle Pfade vereinigt haben. Dadurch beeinflusst die Wahl der maximalen Pfadlänge M den Hardware-Aufwand und bei $M < \tau_{max}$ die Detektionsgüte. Auf dem Prinzip einer begrenzten Pfadlänge basieren auch die im weiteren Verlauf dieser Arbeit präsentierten Signalraumdetektoren (Kap. 4.3).

Soft-Output Viterbi Algorithmus (SOVA)

Der Viterbi-Algorithmus ist in der in Kap. 4.2.2 beschriebenen Variante nicht zu einer Verarbeitung von Zuverlässigkeitsinformation in der Lage. Hagenauer et al. haben jedoch eine einfache Erweiterung präsentiert, die die Ausgabe einer Wahrscheinlichkeit für jedes detektierte Bit c_j und somit einer Zuverlässigkeitsinformation $L(c_j)$ ermöglicht [HH89]. Der so genannte Soft-Output Viterbi-Algorithmus (SOVA) berechnet die Zuverlässigkeitsinformation während der Bestimmung der Survivor-Pfade. Die Wahrscheinlichkeit für einen bestimmten Pfad im Trellis wurde in Gleichung (4.16) bestimmt. Wenn sich im Zustand $s_i^{(n)}$ zwei Pfade treffen, wobei die Sequenz \underline{c}_s dem Survivor-Pfad entspricht und die Sequenz \underline{c}_v dem verworfenen Pfad, lässt sich die Wahrscheinlichkeit $P_f(s_i^{(n)})$ für die Auswahl des falschen Survivor-Pfads im Zustand $s_i^{(n)}$ wie folgt angeben [Bos99]:

$$P_f(s_i^{(n)}) = \frac{P(\underline{r}|\underline{c}_v)}{P(\underline{r}|\underline{c}_s) + P(\underline{r}|\underline{c}_v)} \quad (4.18)$$

Für die Wahrscheinlichkeiten aus Gleichung (4.18) gilt $P(\underline{r}|\hat{\underline{c}}) \sim e^{\lambda_{\hat{\underline{c}}}}$ [Bos99]. Somit lässt sich die Fehlerwahrscheinlichkeit $P_f(s_i^{(n)})$ wie folgt ausdrücken:

$$P_f(s_i^{(n)}) = P_f(\Delta_i^{(n)}) = \frac{e^{\lambda_v}}{e^{\lambda_v} + e^{\lambda_s}} = \frac{1}{1 + e^{\lambda_s - \lambda_v}} = \frac{1}{1 + e^{\Delta_i^{(n)}}} \quad (4.19)$$

mit $\Delta_i^{(n)} = \lambda_s - \lambda_v$

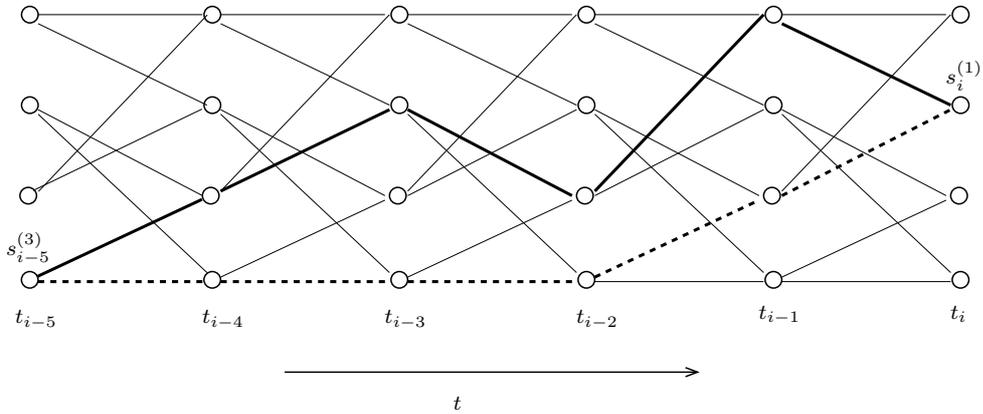


Abbildung 4.3: Bestimmung der Zuverlässigkeitsinformation mit Hilfe des SOVA. Die Auswahl eines Survivor-Pfads \underline{c}_s (fett) im Zustand $s_i^{(1)}$ bestimmt die Zuverlässigkeitsinformation aller Bits, die sich auf dem Survivor \underline{c}_s und dem verworfenen Pfad \underline{c}_v (gestrichelt) unterscheiden.

Die Wahrscheinlichkeit, im Zustand $s_i^{(n)}$ zum Zeitpunkt t_i einen falschen Survivor-Pfad auszuwählen ist nach Gleichung (4.19) von der Differenz der Pfadmetriken $\Delta_i^{(n)}$ in diesem Zustand abhängig. Daraus soll im Folgenden die daraus resultierende Fehlerwahrscheinlichkeit $P_i^{(n)}(c_j)$ für ein Ausgabebit c_j bestimmt werden. Dieser Schritt muss bei

der Durchführung des Viterbi-Algorithmus zu jedem Zeitpunkt t_i für alle Zustände $s_i^{(n)}$ durchgeführt werden.

Spätestens nach der Survivor-Länge τ vereinigen sich Survivor und verworfener Pfad im Trellis. Aus diesem Grund beeinflusst die Entscheidung für einen Survivor-Pfad im Zustand $s_i^{(n)}$ höchstens die Bits $c_j \in \underline{c}_s$, für die gilt $j \in \{i - \tau, \dots, i\}$. Des Weiteren unterscheiden sich nicht notwendigerweise alle Bits $c_{i-\tau}, \dots, c_i$ auf dem Survivor-Pfad und dem verworfenen Pfad. Die Entscheidung für einen Survivor im Zustand $s_i^{(n)}$ betrifft somit nur die Bits $c_{s,j}$ mit $j \in \{i - \tau, \dots, i\}$ auf dem Survivor-Pfad \underline{c}_s , für die gilt $c_{s,j} \neq c_{v,j}$.

Dieser Umstand ist in dem Trellis in Abb. 4.3 skizziert. Im Zustand $s_i^{(1)}$ treffen sich zwei Pfade, wobei der Survivor \underline{c}_s fett und der verworfene Pfad \underline{c}_v gestrichelt gezeichnet ist. Beide Pfade treffen sich wieder im Zustand $s_{i-5}^{(3)}$. Somit sind von der Auswahl des Survivors alle Bits $c_{s,j}$ aus dem Survivor-Pfad \underline{c}_s für $t_{i-5} < t_j \leq t_i$ betroffen, für die gilt $c_{s,j} \neq c_{v,j}$. Des Weiteren ist zu berücksichtigen, dass der Ausgangs-Zustand – in Abb. 4.3 also $s_{i-1}^{(0)}$ – aufgrund einer vorherigen falschen Entscheidung bereits falsch sein kann. Die Wahrscheinlichkeit hierfür wird im Folgenden mit $P_{i-1}^{(0)}(c_j)$ bezeichnet. Daraus lässt sich jetzt die gesuchte Wahrscheinlichkeit $P_i^{(1)}(c_j)$ bestimmen.

Allgemein lässt sich für den Zustand $s_i^{(n)}$ die Wahrscheinlichkeit $P_i^{(n)}(c_j)$, dass Bit c_j falsch ist, wie folgt angeben:

- Wenn die Entscheidung für einen Survivor im Zustand $s_i^{(n)}$ keinen Einfluss auf Bit c_j hat, das heißt $c_{s,j} = c_{v,j}$, bleibt die Fehlerwahrscheinlichkeit $P_i^{(n)}(c_j)$ unverändert bei $P_{i-1}^{(n)}(c_j)$.
- Wenn die Entscheidung für einen Survivor im Zustand $s_i^{(n)}$ Bit c_j beeinflusst ($c_{s,j} \neq c_{v,j}$), muss aufgrund der binären Eigenschaft von c_j zwischen zwei Fällen unterschieden werden. Zum einen kann c_j falsch sein, weil bereits der vorausgehende Zustand $s_{i-1}^{(n)}$ falsch war. Zum anderen kann eine falsche Entscheidung im aktuellen Zustand $s_i^{(n)}$ zu einem falschen Wert von c_j führen. Wenn jedoch beide Ereignisse eintreten, ist das Bit c_j wieder korrekt.

Diese Prozedur muss für jeden Zustand $s_i^{(n)}$ und zu jedem Zeitpunkt t_i durchgeführt werden. Formal lässt sich die Fehlerwahrscheinlichkeit somit wie folgt ausdrücken:

$$P_i^{(n)}(c_j) = \begin{cases} P_{i-1}^{(n)}(c_j) \cdot (1 - P_f(\Delta_i^{(n)})) \\ \quad + (1 - P_{i-1}^{(n)}(c_j)) \cdot P_f(\Delta_i^{(n)}) & \text{falls } c_{s,j} \neq c_{v,j} \\ P_{i-1}^{(n)}(c_j) & \text{falls } c_{s,j} = c_{v,j} \end{cases} \quad (4.20)$$

Sobald sich zum Zeitpunkt $t_{i-\tau}$ alle Pfade im Trellis vereinigt haben, verbleibt nur noch ein Weg durch den Trellis: der Survivor-Pfad. Die Fehlerwahrscheinlichkeiten aus

(4.20) für diesen Pfad werden als $P(c_j)$ bezeichnet. Damit lässt sich für $j < i - \tau$ die Zuverlässigkeitsinformation $L(c_j)$ für das Bit c_j bestimmen:

$$L(c_j) = \ln \frac{1 - P(c_j)}{P(c_j)} \quad \forall j < t - \tau \quad (4.21)$$

Alternativ kann bei der Decodierung des Viterbi-Algorithmus auch direkt Zuverlässigkeitsinformation verwendet werden. Mit Hilfe des Zusammenhangs in (4.21) und der Log-Likelihood Algebra [HOP96] kann auch die iterativ bestimmte Wahrscheinlichkeit $P_i^{(n)}(c_j)$ aus (4.20) als Zuverlässigkeitsinformation $L_i^{(n)}(c_j)$ ausgedrückt werden [Bos99]. Die resultierenden Terme lassen sich durch eine einfache Minimum-Bildung annähern, so dass sich eine deutlich einfachere Berechnung ergibt, die für Implementierungen des SOVA besonders geeignet ist [Bos99]:

$$L_i^{(n)}(c_j) = \begin{cases} \ln \frac{1 + \exp(L_{i-1}^{(n)}(c_j) + \Delta_i^{(n)})}{\exp(L_{i-1}^{(n)}(c_j)) + \exp(\Delta_i^{(n)})} \\ \approx \min(L_{i-1}^{(n)}(c_j), \Delta_i^{(n)}) & \text{falls } c_{s,j} \neq c_{v,j} \\ L_{i-1}^{(n)}(c_j) & \text{falls } c_{s,j} = c_{v,j} \end{cases} \quad (4.22)$$

Bisher wurden die in heutigen Festplatten als Standard eingesetzten PRML-Detektoren inklusive des Soft-Output Viterbi Algorithmus zur Bestimmung von Zuverlässigkeitsinformation präsentiert. Wenn ein Detektor mit dem Ziel einer möglichst einfachen Implementierung ausgewählt werden soll, bieten sich allerdings so genannte Fixed Delay Tree Search (FDTS) Detektoren an, die im folgenden Kapitel vorgestellt werden.

4.3 Signalraumdetektion

Bei der Beschreibung des Viterbi-Algorithmus in Kapitel 4.2.2 wurde bereits eine Implementierung mit einer begrenzten Survivor-Länge M diskutiert. Basierend auf dieser Idee kann ein Detektor mit geringer Hardware-Komplexität konstruiert werden. Das Prinzip ist in der Literatur als Fixed Delay Tree Search (FDTS) bekannt und wird in diesem Kapitel erläutert. Die geometrische Interpretation des FDTS-Prinzips – die so genannte Signalraumdetektion – führt zu einer besonders einfachen und parallelen Architektur [BM97a].

4.3.1 Fixed Delay Tree Search (FDTS) Detektoren

Ähnlich wie beim Viterbi-Algorithmus mit begrenzter Survivor-Länge (Kap. 4.2.2), wird bei einem FDTS-Detektor nach einer festen Verzögerung von M Taktzyklen eine Detektorentscheidung ausgegeben. Des Weiteren wird davon ausgegangen, dass die Pfadlänge M ausreichend ist und somit vor dem Zeitpunkt t_{i-M} nur noch ein einzelner Pfad im

Trellis existiert (siehe Abb. 4.4). Unter dieser Voraussetzung erfüllt auch ein FDTS-Detektor das MLSE-Kriterium aus (4.7).

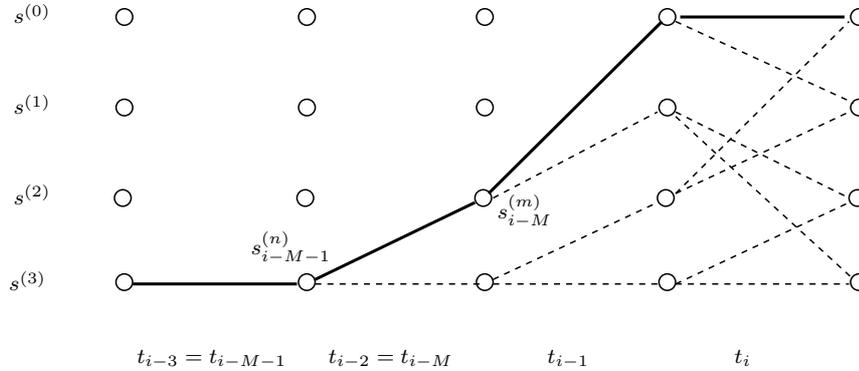


Abbildung 4.4: Das Prinzip eines FDTS Detektors mit $M = 2$ an einem Trellis: Im Zustand $s_{i-M-1}^{(n)}$ haben sich alle Pfade zu einem einzelnen Survivor-Pfad vereinigt. Von dort aus starten verschiedene Wege durch den Trellis. Zum Zeitpunkt t_i wird der Pfad mit der geringsten Metrik ausgewählt (fett). Dadurch kann eine eindeutige Verzweigung zum Zeitpunkt t_{i-M} in den Zustand $s_{i-M}^{(m)}$ festgelegt werden.

Der Decodierungsprozess erfolgt analog zum Viterbi-Algorithmus über eine rekursive Minimierung der Pfadmetriken. Aus dem Zustand $s_{i-M-1}^{(n)}$ starten verschiedene Wege durch den Trellis. Zu jedem Zeitpunkt t_i gelangt ein weiterer Abtastwert r_i in den Detektor. Zur Bestimmung der Detektorausgabe werden alle Pfade, die im Zustand $s_{i-M-1}^{(n)}$ beginnen und zum Zeitpunkt t_i in einem der 2^L Zustände $s_i^{(n)}$ enden, miteinander verglichen und der Pfad mit der geringsten Metrik $\lambda = \min_{\hat{c}} \lambda_{\hat{c}}$ als Survivor ausgewählt (fett in Abb. 4.4). Durch diese Auswahl wird zum Zeitpunkt t_{i-M} eine Verzweigung von Zustand $s_{i-M-1}^{(n)}$ in einen Zustand $s_{i-M}^{(m)}$ festgelegt. Das zu dieser Verzweigung gehörige Bit c_{i-M} ist die Detektorausgabe. Da sich die Pfade zu allen vorherigen Zeitpunkten $t \leq t_{i-M-1}$ nicht unterscheiden, ist eine Betrachtung der Zustände zu diesen Zeitpunkten nicht notwendig. Somit lässt sich die Pfadmetrik wie folgt bestimmen:

$$\lambda_{\hat{c}} = \sum_{i-M}^i (r_i - r_{\hat{c},i})^2 \quad (4.23)$$

Im Gegensatz zu der Pfadmetrik aus (4.17) wird in (4.23) nicht über alle vorhergehenden Zustände $s_i^{(n)}$ aufsummiert. Bei realen Implementierungen wird die Pfadmetrik jedoch über einen rekursiven Ansatz bestimmt, der zum Zeitpunkt t_i einfach die aktuelle Pfadmetrik hinzuaddiert. Dies führt dazu, dass der Absolutbetrag der Metrik im Verlauf der Zeit immer größer wird und durch geeignete Maßnahmen kompensiert werden muss.

Alternativ zur rekursiven Formulierung kann auch eine interne Rückkopplungsschleife eingesetzt werden, um den Einfluss vorheriger Entscheidungen c_i auf die Pfadmetriken zu kompensieren. Über das Target Γ der Länge L_Γ kann dann der Einfluss der Inter-symbolinterferenz herausgerechnet werden. In diesem Fall werden an Stelle der von der Festplatte gelesenen Abtastwerte r'_i die modifizierten Samples r_i verwendet:

$$r_j = r'_j - \sum_{j-L_\Gamma+1 \leq j-k \leq i-M-1}^k c_{j-k} \Gamma_k. \quad (4.24)$$

Der Vorteil dieser Methode ist, dass die Pfadmetriken immer von einer festen Anzahl von Verzweigungen abhängen. Aus diesem Grund steigen sie im Verlauf der Decodierung nicht an. Eine Kompensation, wie im Falle der rekursiven Berechnung ist somit nicht notwendig. Das beschriebene Verfahren wird als Fixed Delay Tree Search mit Decision Feedback (FDTS/DF) bezeichnet. Im Folgenden werden die Signalraumdetektoren (SSD) als eine geometrische Interpretation der FDTS/DF Detektoren präsentiert.

4.3.2 Signalraumdetektoren (SSD)

In der Literatur wurden SSDs für Kanäle untersucht, die weißem und farbigem Rauschen unterliegen [Pat91, BM97a, MJ98, SHS00]. Ein Signalraumdetektor erlaubt eine geometrische Interpretation des Detektionsprozesses. Wie beim FDTS-Detektor wird das Ausgabebit c_{i-M} eindeutig durch die letzten $M+1$ Abtastwerte $\underline{r} = \{r_{i-M}, \dots, r_i\}$ bestimmt, von denen die ISI-Terme gemäß (4.24) entfernt wurden.

Die eigentliche Detektion erfolgt in einem $M+1$ -dimensionalen Vektorraum, dem so genannten Signalraum $\mathcal{S} = \mathcal{R}^{M+1}$. In diesem Signalraum werden die zu der Datensequenz $\hat{c} \in \mathcal{C}$ gehörigen Sequenzen $\underline{r}_{\hat{c}}$ der idealen Abtastwerte als Punkte dargestellt und als erlaubte Signalpunkte (engl. admissible signal point) $\underline{s}_{\hat{c}}$ bezeichnet. Die Menge \mathcal{C} enthält alle vom Modulationscode zugelassenen binären Sequenzen der Länge $M+1$. Die Komponenten s_j des erlaubten Signalpunkts $\underline{s}_{\hat{c}} = \{s_{i-M}, \dots, s_i\}$ ergeben sich durch die Faltung der Datensequenz \hat{c} mit dem Target Γ des magnetischen Kanals:

$$s_j = \sum_{k=0}^{L_\Gamma-1} \hat{c}_{j-k} \Gamma_k. \quad (4.25)$$

Eine beliebige verrauschte Sequenz \underline{r} kann ebenfalls im Signalraum als so genannter beobachteter Signalpunkt (engl. observed signal point) \underline{r} dargestellt werden. Im Folgenden wird die Detektorausgabe unter der Annahme von gleichverteiltem weißem Rauschen mit der Kovarianzmatrix $\underline{\underline{\mathbf{R}}}$ und für eine begrenzte Pfadlänge M berechnet. Gemäß des Maximum Likelihood Kriteriums bestimmt der Signalraumdetektor mit

$$(\underline{r} - \underline{s}_{\hat{c}})^T \underline{\underline{\mathbf{R}}}^{-1} (\underline{r} - \underline{s}_{\hat{c}}) = \min_{\underline{s}_{\hat{c}}} \left[(\underline{r} - \underline{s}_{\hat{c}})^T \underline{\underline{\mathbf{R}}}^{-1} (\underline{r} - \underline{s}_{\hat{c}}) \right] \quad (4.26)$$

zunächst die wahrscheinlichste Sequenz $\underline{c} = \{c_{i-M}, \dots, c_i\} \in \mathcal{C}$ und gibt schließlich c_{i-M} als Entscheidung aus. Sofern der Einfluss eines Modulationscodes vernachlässigt wird, ist die Menge der in (4.26) benötigten erlaubten Signalpunkte laut (4.25) zeitinvariant und bildet einen festen Satz von Punkten im Signalraum $\mathcal{S} = \mathcal{R}^{M+1}$.

Somit kann mit Hilfe von (4.26) jedem beobachteten Signalpunkt \underline{r} ein erlaubter Signalpunkt $\underline{s}_{\hat{c}}$ und eine dazu gehörige Detektorausgabe \hat{c}_{i-M} zugeordnet werden. Die

Menge aller Signalpunkte, die einem festen erlaubten Signalpunkt $\underline{s}_{\hat{c}}$ zugeordnet wird, bildet eine homogene Partition (so genannte Voronoi-Region). Diese wird im Folgenden als Entscheidungsregion $\mathcal{R}_{\hat{c}}$ bezeichnet. Wie in Abb. 4.5 für einen zweidimensionalen Signalraum zu sehen, werden die Entscheidungsregionen durch Hyperebenen begrenzt. Jede Hyperebene trennt dabei zwei erlaubte Signalpunkte, die sich in der dazu gehörigen Detektorausgabe \hat{c}_{i-M} unterscheiden [Pat91]. Geometrisch ergeben sich die Hyperebenen im zweidimensionalen Fall und unter der Annahme von AWGN als die Mittelsenkrechten der Verbindungslinien der entsprechenden erlaubten Signalpunkte. Dicht beieinander liegende Hyperebenen können jedoch zur Vereinfachung einer Detektorimplementierung ohne nennenswerte Einbußen bei der Zuverlässigkeit weggelassen werden [MJ98]. Aus diesem Grund wird der Einfluss nicht implementierter Ebenen auch bei der Konstruktion des Soft-Output SSDs im weiteren Verlauf dieses Kapitels nicht berücksichtigt.

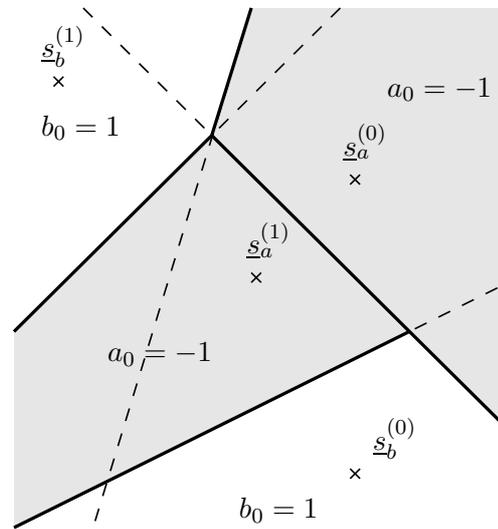


Abbildung 4.5: Konzept der Signalraumdetektion in einem zweidimensionalen Signalraum. Die Entscheidungsebenen (gestrichelt) formen die Entscheidungsregionen (fett).

Die optimale Lage der Entscheidungsebenen lässt sich im Voraus berechnen, da die Positionen der erlaubten Signalpunkte bekannt sind. Die Minimierung von (4.26), die zur Bestimmung der Detektorausgabe notwendig ist, entspricht im Signalraum der Identifikation der Entscheidungsregion \mathcal{R}_c , in dem sich der Signalpunkt befindet. In der SSD-Architektur geschieht dies, indem die relative Lage des empfangenen Signalpunkts \underline{r} zu den Entscheidungsebenen

$$E_j = \{\underline{x} | \underline{n}_j^T \underline{x} - k_j = 0\}, j = 1, \dots, N \quad (4.27)$$

bestimmt wird. Die relative Position von \underline{r} zu allen Entscheidungsebenen kann dann durch einen Booleschen Vektor $\underline{\epsilon} = [\epsilon_1, \dots, \epsilon_N]$ mit

$$\epsilon_j = (\underline{n}_j^T \underline{r} - k_j > 0). \quad (4.28)$$

festgelegt werden. Über den Wert von $\underline{\epsilon}$ kann der empfangene Signalpunkt dann einer Entscheidungsregion \mathcal{R}_c zugeordnet werden. Dadurch wird die eigentliche Detektorausgabe c_{i-M} mit Hilfe der Booleschen Abbildung $O(\underline{\epsilon})$ nach der Vorschrift

$$O(\underline{\epsilon}) = c_{i-M} \quad (4.29)$$

bestimmt. Wenn \underline{r} elementweise in den Detektor gespeist wird, kann die Lösung von (4.28) mit Hilfe einer FIR-Filter Struktur, gefolgt von einem Schwellwert-Entscheider, bestimmt werden [BM97a]. Diese Struktur bildet die Ebenengleichungen im Signalraum nach. Eine solche Architektur ist schematisch in Abb. 4.6 zu sehen.

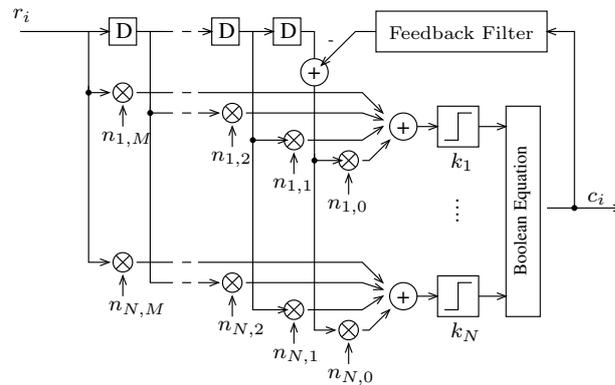


Abbildung 4.6: Schematische Illustration einer SSD Architektur.

Durch die parallele Auswertung aller Entscheidungsebenen ist die vorgestellte Architektur besonders effizient. Das Konzept der Signalraumdetektion lässt sich zudem einfach zu einem Soft-Output Detektor erweitern. Das dabei eingesetzte Verfahren wird im folgenden Kapitel vorgestellt.

4.4 Soft-Output Signalraumdetektion

Die Bestimmung von Soft-Outputs lässt sich leicht im Signalraum veranschaulichen. Allgemein kann die Zuverlässigkeit $L(c_{i-M})$ der Detektorausgabe c_{i-M} in Abhängigkeit vom empfangenen Signalpunkt \underline{r} über das Log-Likelihood Verhältnis der *a posteriori*-Wahrscheinlichkeiten (engl. *a posteriori* probabilities (APP)) für die beiden möglichen Detektorausgaben "+1" und "-1" bestimmt werden (vgl. (4.8)). Für einen Signalraum, der durch eine einzelne Hyperebene partitioniert werden kann, ergibt sich

$$L(c_{i-M}) = \ln \frac{P(c_{i-M} = +1 | \underline{r})}{P(c_{i-M} = -1 | \underline{r})} = x \cdot d, \quad (4.30)$$

wobei x ein durch Simulation bestimmbarer Faktor und d der geometrische Abstand des empfangenen Signalpunktes zur Entscheidungsebene ist [SMO01]. Die Abhängigkeit vom

Abstand d lässt sich auch anschaulich verdeutlichen: Liegt der empfangene Signalpunkt \underline{r} dicht an einem erlaubten Signalpunkt, so entspricht dies einer zuverlässigen Entscheidung, weil eine sehr gute Übereinstimmung mit einem unverrauschten Punkt vorliegt. Im Gegensatz dazu kann ein Signalpunkt, der genau auf einer Entscheidungsebene liegt, keiner Detektorausgabe zugeordnet werden. Die beiden Wahrscheinlichkeiten im Zähler und im Nenner von (4.30) sind dann aufgrund der Definition der Entscheidungsebene (Siehe Kap. 4.3.2) gleich groß und die Zuverlässigkeit $L(c_{i-M})$ hat den Wert Null.

Das Prinzip aus (4.30) wurde für den optischen Kanal einer DVD entwickelt, dessen Signalraumkonstellation mit einer einzigen Hyperebene partitioniert werden kann [SMO01]. Auf dem magnetischen Kanal sind jedoch typischerweise mehrere Entscheidungsebenen notwendig [BM96b]. Aus diesem Grund ist (4.30) nicht anwendbar. In diesem Fall kann $L(c_{i-M})$ bestimmt werden, indem der Abstand des empfangenen Signalpunktes zu allen erlaubten Signalpunkten betrachtet wird [Ste02]. Dazu wird die Menge der erlaubten Signalpunkte in zwei Gruppen partitioniert, die mit den beiden möglichen Detektorausgaben korrespondieren: $\underline{s}_{\hat{a}}$ entspricht einem erlaubten Signalpunkt mit der dazugehörigen Datensequenz $\hat{a} \in \mathcal{A}$ mit $\mathcal{A} = \{\hat{a} \in \mathcal{C} \mid \hat{a}_{i-M} = +1\}$ und $\underline{s}_{\hat{b}}$ entspricht einem erlaubten Signalpunkt mit der dazugehörigen Datensequenz $\hat{b} \in \mathcal{B}$ mit $\mathcal{B} = \{\hat{b} \in \mathcal{C} \mid \hat{b}_{i-M} = -1\}$. $P(\underline{s}_{\hat{a}})$ und $P(\underline{s}_{\hat{b}})$ sind die a priori-Wahrscheinlichkeiten für die entsprechenden Signalpunkte. Mit Hilfe von (4.26) lassen sich über die geometrischen Abstände zwischen erlaubten und empfangenen Signalpunkten die bedingten Wahrscheinlichkeiten $P(\underline{r}|\underline{s}_{\hat{a}})$ und $P(\underline{r}|\underline{s}_{\hat{b}})$ in Abhängigkeit von \underline{r} angeben. Es ergibt sich

$$L(c_{i-M}) = \ln \frac{\sum_{\underline{s}_{\hat{a}}} P(\underline{r}|\underline{s}_{\hat{a}})P(\underline{s}_{\hat{a}})}{\sum_{\underline{s}_{\hat{b}}} P(\underline{r}|\underline{s}_{\hat{b}})P(\underline{s}_{\hat{b}})}. \quad (4.31)$$

Der Ansatz aus (4.31) führt zu sehr guten Ergebnissen [Ste02], erfordert mit den bedingten Wahrscheinlichkeiten jedoch Parameter, die in der vorgestellten Hard-Output SSD Architektur nicht zur Verfügung stehen. Aus diesem Grund sind aufwändige Modifikationen notwendig, die unter dem Gesichtspunkt einer möglichst geringen Verlustleistung vermieden werden sollten. Alternativ zu der Implementierung von (4.31) wurde in der gleichen Veröffentlichung eine Approximation vorgestellt, mit deren Hilfe die Zuverlässigkeit über eine parametrische Modellierung bestimmt wurde [Ste02]. Dennoch sind auch hier kostspielige Modifikationen der Detektorhardware nötig. Im folgenden Abschnitt wird aus diesem Grund eine andere Soft-Output Architektur vorgestellt, die praktisch keine Veränderungen an der Hard-Output Architektur erfordert.

4.5 Low-Complexity Soft-Output Signalraumdetektoren

Bei der Konstruktion eines Low-Complexity Soft-Output Signalraumdetektors [SSHS03] (im Folgenden als S³D bezeichnet) soll bei der Bestimmung der Zuverlässigkeitsinformation möglichst von Größen ausgegangen werden, die schon bei einem normalen Hard-Output Detektor benötigt werden. Zu diesem Zweck wird ein Detektor konstruiert, der zur Bestimmung der Soft-Outputs eine einzelne, geeignete Entscheidungsebene auswählt.

Der normalisierte Abstand zu dieser Ebene wird schließlich zur Berechnung der Zuverlässigkeitsinformation verwendet. Im Einzelnen müssen zur Konstruktion des S³D - Detektors die folgenden Aufgaben gelöst werden [SSHS03]:

– **Berechnung der Soft-Outputs für AWGN**

Da ein SSD für den magnetischen Kanal aus mehreren Entscheidungsebenen konstruiert ist, kann es passieren, dass ein empfangener Signalpunkt von mehreren Entscheidungsebenen umgeben ist. In Kap. 4.5.1 wird für diesen Fall ein Verfahren zur Auswahl einer einzelnen Ebene entwickelt. Die Berechnung erfolgt zunächst unter der Annahme von stationärem, weißem Rauschen (AWGN).

– **Berechnung der Soft-Outputs für farbiges Rauschen**

Signalraumdetektoren verwenden in der Regel einen Forward-Equalizer, so dass das Rauschspektrum am Eingang des Detektors gefärbt ist. Diese Rauschfärbung bewirkt eine Verschiebung der Ebenenkonstellation, die bei der Berechnung der Zuverlässigkeitsinformation berücksichtigt werden muss (Kap. 4.5.2).

– **Betrachtung konkaver Entscheidungsregionen**

In konkaven Entscheidungsregionen entspricht der Abstand zur Entscheidungsregion nicht immer dem Abstand zur nächsten Entscheidungsebene. Für diesen Fall wird in Kap. 4.5.3 eine Näherung vorgestellt.

– **Berücksichtigung von Fehlerfortpflanzung**

Ein Signalraumdetektor ist ein FDTS-Detektor mit Decision Feedback (FDTS/DF). Die Feedback-Schleife führt zu Fehlerfortpflanzung, die die Soft-Outputs beeinflusst. In Kap. 4.5.4 wird ein Algorithmus aufgezeigt, der – ähnlich zu der Berechnung beim Soft-Output Viterbi-Algorithmus (SOVA, Kap. 4.2.2) – die Zuverlässigkeitsinformation anpasst.

– **Entwurf einer Detektor-Architektur**

Abschließend wird in Kap. 4.5.5 eine Architektur präsentiert. Eine Ausgabe der in Kap. 4.5.2 berechneten Zuverlässigkeitsinformation kann praktisch ohne Veränderungen an der bestehenden Detektorhardware realisiert werden.

4.5.1 Berechnung der Zuverlässigkeitsinformation für AWGN

Bei der Berechnung der Zuverlässigkeitsinformation wird zunächst von (4.31) ausgegangen. Somit wird der geometrische Abstand eines empfangenen Signalpunktes zu den anderen Signalpunkten als Maß für die Zuverlässigkeit der Detektorausgabe verwendet. Durch die Anwendung einer üblichen Max-Log-Approximation kann (4.31) zu

$$L(c_{i-M}) \approx \ln \frac{\max_{\hat{s}_a} (P(r|\hat{s}_a))}{\max_{\hat{s}_b} (P(r|\hat{s}_b))} \quad (4.32)$$

vereinfacht werden. Somit kann die Zuverlässigkeit der Detektorausgabe c_{i-M} durch die maximalen Wahrscheinlichkeiten für die beiden möglichen Detektorausgaben ”+1” und ”-1” ausgedrückt werden. Unter dem Einfluss von weißem Rauschen (AWGN) mit der Varianz σ^2 können beide Terme aus (4.32) mit Hilfe von

$$P(\underline{r}|\underline{s}_c) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \exp\left(-\frac{(\underline{r} - \underline{s}_c)^T \cdot (\underline{r} - \underline{s}_c)}{2\sigma^2}\right) \quad (4.33)$$

einfach bestimmt werden. Der Detektor bestimmt die wahrscheinlichste Datensequenz \underline{c} , die mit dem erlaubten Signalpunkt \underline{s}_c assoziiert ist und gibt c_{i-M} als Entscheidung aus. Dies bedeutet, dass entweder

$$P(\underline{r}|\underline{s}_c) = \max_{\underline{s}_a} P(\underline{r}|\underline{s}_a) \quad (\text{für } c_{i-M} = +1) \text{ oder} \quad (4.34)$$

$$P(\underline{r}|\underline{s}_c) = \max_{\underline{s}_b} P(\underline{r}|\underline{s}_b) \quad (\text{für } c_{i-M} = -1) \quad (4.35)$$

während der normalen Detektoroperation bestimmt werden. Wenn also $c_{i-M} = +1$, bestimmt der Detektor

$$|L(c_{i-M})| = \ln \frac{P(\underline{r}|\underline{s}_c)}{\max_{\underline{s}_b} P(\underline{r}|\underline{s}_b)}, \quad (4.36)$$

im anderen Fall, wenn $c_{i-M} = -1$, wird

$$|L(c_{i-M})| = \ln \frac{P(\underline{r}|\underline{s}_c)}{\max_{\underline{s}_a} P(\underline{r}|\underline{s}_a)} \quad (4.37)$$

berechnet. Da der erlaubte Signalpunkt \underline{s}_c der wahrscheinlichsten Sequenz \underline{c} entspricht, ist der Logarithmus $\ln(\cdot)$ in (4.36) oder (4.37) in jedem Fall positiv, da der Zähler größer als der Nenner ist. Aus diesem Grund wird der Absolutbetrag der Zuverlässigkeitsinformation $|L(c_{i-M})|$ berechnet. Dennoch kann mit $L(c_{i-M}) = c_{i-M} \cdot |L(c_{i-M})|$ leicht das Vorzeichen bestimmt werden. Im Nenner von (4.36) oder (4.37), wird durch die Maximierung die wahrscheinlichste Datensequenz \underline{a} oder \underline{b} gesucht, deren erstes Element a_{i-M} oder b_{i-M} nicht der Detektorausgabe c_{i-M} entspricht. In diesem Fall werden die erlaubten Signalpunkte \underline{s}_c und \underline{s}_a bzw. \underline{s}_b von einer Entscheidungsebene getrennt, die die Entscheidungsregion \mathcal{R}_c begrenzt. Eine Ebene, die diese Eigenschaft besitzt, wird *entscheidend* genannt. Die Menge aller entscheidenden Ebenen, die die Entscheidungsregion \mathcal{R}_c um \underline{s}_c begrenzen, wird als \mathcal{D}_c bezeichnet.

In Abb. 4.7 ist ein zweidimensionaler Signalraum mit zwei Entscheidungsebenen zu sehen, die den Raum in zwei Entscheidungsregionen unterteilen. An dieser Konstellation wird eine geometrische Lösung für (4.32) hergeleitet, die auf (4.30) zurückgeführt werden kann und dadurch eine Bestimmung der Konstante x aus (4.30) erlaubt. In der Abbildung liegt ein empfangener Signalpunkt \underline{r} in der nicht schattierten Region und es gilt $e > f$. Es wird AWGN mit der Varianz σ^2 vorausgesetzt.

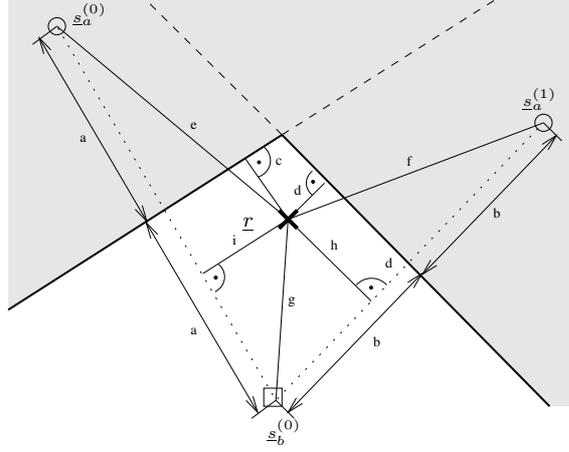


Abbildung 4.7: Bestimmung der Max-Log Zuverlässigkeitsinformation $L(c_{i-M})$ für einen erlaubten Signalpunkt \underline{r} , der von zwei Entscheidungsebenen umgeben ist.

Wenn die Max-Log Approximation aus (4.32) angewendet wird, wird die Zuverlässigkeitsinformation nur anhand des Abstands zweier Signalpunkte \underline{s}_a und \underline{s}_b zu \underline{r} bestimmt. Im Folgenden wird hergeleitet, dass in diesem Fall der geometrische Abstand von \underline{r} zu der Entscheidungsebene, die \underline{s}_a und \underline{s}_b trennt, proportional zur Zuverlässigkeit ist. Für diese Konstellation wird die Notation $L_{\underline{s}_a^{(1)}, \underline{s}_b^{(0)}}(c_{i-M})$ verwendet.

Durch Einsetzen von (4.33) in (4.32) ergibt sich

$$\begin{aligned}
 |L(c_{i-M})| &= |L_{\underline{s}_a^{(1)}, \underline{s}_b^{(0)}}(c_{i-M})| \\
 &= \left| \ln \frac{\frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{g^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{f^2}{2\sigma^2}}} \right| \\
 &= \left| \frac{f^2 - g^2}{2\sigma^2} \right| \\
 &= \frac{(h^2 + (b + d)^2) - (h^2 + (b - d)^2)}{2\sigma^2} \\
 &= 2bd/\sigma^2.
 \end{aligned} \tag{4.38}$$

Somit ergibt sich mit

$$x = \frac{2b}{\sigma^2} \tag{4.39}$$

und mit $L(c_{i-M}) = c_{i-M} \cdot |L(c_{i-M})|$ die Zuverlässigkeitsinformation zu

$$L(c_{i-M}) = c_{i-M} \cdot x \cdot d. \tag{4.40}$$

Die Rauschvarianz σ^2 ändert sich in den Simulationen bei einer Variation des Signal-Rausch-Verhältnisses. Sie kann entweder gemessen oder die Kanalcharakteristik ab-

geschätzt werden [HW99]. In einer praktischen Implementierung reicht allerdings das Verhältnis aufeinander folgender Bitzuverlässigkeiten für einen zuverlässigen Betrieb des Detektors aus. In diesem Fall ist die Rauschvarianz irrelevant.

Wenn die zur Berechnung der Soft-Outputs betrachtete Ebenengleichung E_j derart normiert wird, dass der Normalenvektor die Länge $|\underline{n}| = x$ hat, ist $L(c_{i-M})$ bei leichten Modifikationen der Detektor-Architektur direkt am Ausgang erhältlich (Siehe Kap. 4.5.5). Es ergibt sich:

$$L(c_{i-M}) = c_{i-M} \cdot |\underline{n}_j^T \underline{r} - k_j| \quad (4.41)$$

In der Praxis wird jedoch jede Entscheidungsregion \mathcal{R}_c von mehreren Entscheidungsebenen begrenzt. Die Menge der für die Detektorausgabe relevanten Ebenen wurde in diesem Kapitel als \mathcal{D}_c definiert. Wenn eine Entscheidungsregion \mathcal{R}_c von j entscheidenden Ebenen umgeben ist, kann die Max-Log Zuverlässigkeit aus (4.32) bestimmt werden, indem (4.41) über alle entscheidenden Ebenen E_j aus \mathcal{D}_c minimiert wird:

$$L(c_{i-M}) = c_{i-M} \cdot \min_{j \in \mathcal{D}_c} |\underline{n}_j^T \underline{r} - k_j| \quad (4.42)$$

Dies entspricht der Maximierung des Nenners in (4.36) bzw. (4.37). Die Menge der entscheidenden Ebenen kann während des Detektordesigns vorausberechnet werden, so dass im Betrieb des Detektors an dieser Stelle kein Rechen-Overhead entsteht.

4.5.2 Berechnung der Zuverlässigkeitsinformation für farbiges Rauschen

Die durch den Equalizer verursachte Rauschfärbung bewirkt, dass sich die ideale Position der Entscheidungsebenen verschiebt. Dennoch kann auch für gefärbtes, stationäres Rauschen ein S^3D konstruiert werden. Zu diesem Zweck wird die Signalraumtransformation angewendet, die zur Konstruktion des so genannten Noise-Whitening SSD (WSSD) entwickelt wurde [SHS00]. Dazu wird die Hauptachsentransformation

$$\underline{x} = \underline{\underline{H}} \cdot \underline{x}' \quad (4.43)$$

angewendet, die jeden Punkt $\underline{x} \in \mathcal{S}$ in die so genannte White Noise Domäne \mathcal{S}' überführt, in der das Rauschen weiß ist. $\underline{\underline{H}}$ ist die dazugehörige Transformationsmatrix [SHS00].

Die Entscheidungsebenen können dann nach den in Kap. 4.5.1 vorgestellten Methoden in der Form $E' = \{\underline{x}' | \underline{n}'^T \cdot \underline{x}' - k' = 0\}$ konstruiert werden. Die inverse Transformation der Ebenen führt schließlich zu den Entscheidungsebenen, die im WSSD implementiert werden müssen:

$$E = \{\underline{x} | \underline{n}^T \cdot \underline{x} - k = 0\} \quad \text{mit} \quad (4.44)$$

$$\underline{n} = (\underline{\underline{H}}^{-1})^T \cdot \underline{n}' \quad \text{und} \quad k = k'. \quad (4.45)$$

Wichtig für das Design eines WSSD Detektors ist, dass Wahrscheinlichkeiten durch die Signalraumtransformation erhalten bleiben und somit gilt: $P(\underline{x}' \in \mathcal{U}' \subseteq \mathcal{S}') = P(\underline{x} \in \mathcal{U} \subseteq \mathcal{S})$. \mathcal{U}' ist die Abbildung einer Region $\mathcal{U} \in \mathcal{S}$ in den Signalraum \mathcal{S}' . Daraus folgt die wichtige Eigenschaft, dass die Performance eines WSSD Detektors identisch mit der eines SSD Detektors ist, der in der White Noise Domäne konstruiert wurde. Dies bedeutet, dass auch die Bitzuverlässigkeiten des WSSD und des in \mathcal{S}' konstruierten SSD gleich sind:

$$|L(c_{i-M})| = |\underline{n}'^T \cdot \underline{r}' - k'| \quad \text{mit} \quad (4.46)$$

$$|\underline{n}'| = 2b'/\sigma'^2, \quad (4.47)$$

wobei b' der Abstand des empfangenen Signalpunkts \underline{r}' zu der betrachteten Entscheidungsebene ist und σ'^2 die Rauschvarianz in der White Noise Domäne. \underline{H} kann so gewählt werden, dass $\sigma' = 1$ [SHS00]. Wenn \underline{r}' ersetzt, wird ergibt sich mit (4.45)

$$|L(c_{i-M})| = |\underline{n}'^T \cdot \underline{H}^{-1} \cdot \underline{r} - k'| \quad (4.48)$$

$$= |\underline{n}^T \cdot \underline{r} - k|. \quad (4.49)$$

Wie bereits bei der Berechnung der Ebenen beim SSD in (4.41), ist die Bestimmung von (4.49) auch im Hard-Output WSSD notwendig. Somit sind auch im Falle stationären, farbigen Rauschens keine weiteren Hardware-Modifikationen zur Berechnung der Zuverlässigkeitsinformation einer einzelnen Ebene erforderlich.

Des Weiteren ergibt sich auch beim Soft-Output WSSD in (4.47) eine Normierung von $|\underline{n}'|$, analog zu der Normierung von $|\underline{n}|$ in (4.41). Dies ist in beiden Fällen die einzige Hardware-Modifikation bei der Konstruktion der Entscheidungsebenen. Die abschließende Berechnung des Soft-Outputs $|L(c_{i-M})|$ ist analog zu (4.42).

4.5.3 Betrachtung konkaver Entscheidungsregionen

In konkaven Entscheidungsregionen kann (4.42) nicht direkt angewendet werden. Dieser Umstand ist in Abb. 4.8 skizziert. Nur der durchgezogene Teil der Entscheidungsebenen E_1 und E_2 begrenzt die konvexe Entscheidungsregion \mathcal{R}_a (weiß) um den erlaubten Signalpunkt $\underline{s}_a^{(0)}$. Der gestrichelte Teil der Ebenen liegt in einer konkaven Entscheidungsregion \mathcal{R}_b (grau schattiert), die mit den beiden erlaubten Signalpunkten $\underline{s}_b^{(0)}$ und $\underline{s}_b^{(1)}$ assoziiert ist. In einem Teil dieser konkaven Region, zwischen den gestrichelten Teilen der Ebenen, haben weder E_1 , noch E_2 einen Einfluss auf die Detektorentscheidung. Aus diesem Grund sind beide Ebenen dort nicht entscheidend für die Bestimmung der Zuverlässigkeitsinformation mit Hilfe von (4.42).

Dieser Umstand soll an zwei Beispielen verdeutlicht werden. In der Abbildung sind zwei empfangene Signalpunkte \underline{r}_1 und \underline{r}_2 zu sehen, die beide in der Entscheidungsregion \mathcal{R}_b liegen. Im Falle von \underline{r}_1 begrenzt nur E_1 die Entscheidungsregion \mathcal{R}_b und ist somit als

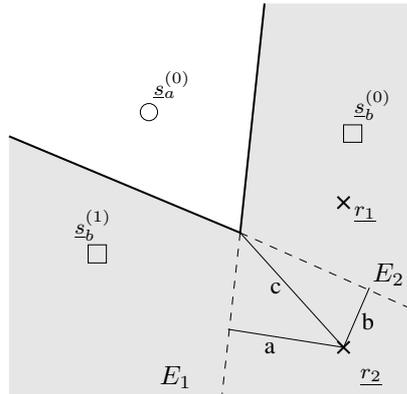


Abbildung 4.8: E_1 und E_2 bilden eine konkave Entscheidungsregion \mathcal{R}_b (grau schattiert) und eine konvexe Entscheidungsregion \mathcal{R}_a (weiß dargestellt). In diesem Fall sind beide \underline{r}_2 umgebende Ebenen nicht entscheidend für die Bestimmung der Zuverlässigkeitsinformation.

einzig entscheidend. Aus diesem Grund muss E_1 für die Berechnung der Zuverlässigkeitsinformation in (4.42) verwendet werden, obwohl der Abstand von \underline{r}_1 zu E_2 kleiner ist. Die Menge der entscheidenden Ebenen enthält nur E_1 , das heißt $D_b = \{E_1\}$.

Für den empfangenen Signalpunkt \underline{r}_2 begrenzen indes weder E_1 noch E_2 die Entscheidungsregion \mathcal{R}_b . Dies bedeutet, dass in konkaven Entscheidungsregionen Bereiche existieren, die von keiner entscheidenden Ebene umgeben sind. In diesem Fall ist die Menge der entscheidenden Ebenen leer: $D_b = \{0\}$ – es stehen somit keine Ebenen für die Minimierung in (4.42) zur Verfügung. Die Entscheidungsregion wird statt dessen von beiden Ebenen aus der Menge $E_i = \{E_1, E_2\}$ gemeinsam begrenzt. Über die in Kap. 4.3.2 eingeführte Boolesche Abbildung $O(\underline{\epsilon})$ kann dieser Umstand verdeutlicht werden: Die Abbildungen

$$O(\epsilon_1, \epsilon_2) = O(\bar{\epsilon}_1, \epsilon_2) = O(\epsilon_1, \bar{\epsilon}_2) = c_{i-M} \quad (4.50)$$

führen alle zu der gleichen Detektorausgabe c_{i-M} . Erst $O(\bar{\epsilon}_1, \bar{\epsilon}_2)$ ergibt eine andere Detektorausgabe. Somit entspricht der minimale Abstand eines empfangenen Signalpunktes zu der Grenze der Entscheidungsregion nicht dem Abstand zu einer einzelnen Ebene. Statt dessen müsste für eine exakte Berechnung die Entfernung zu dem Schnittpunkt der Entscheidungsebenen betrachtet werden. Im Fall von \underline{r}_2 in Abb. 4.8 ist das der mit c bezeichnete Abstand von \underline{r}_2 zur Entscheidungsregion \mathcal{R}_a um $\underline{s}_a^{(0)}$. Dieses Maß steht in der Hard-Output SSD Architektur jedoch nicht zur Verfügung, da die Schnitt- oder Kreuzungspunkte der Ebenen in einer Hard-Output Implementierung nicht benötigt werden. Aus diesem Grund kann die gesuchte Größe auch nicht ohne weiteres aus dem Detektionsprozess extrahiert werden.

Ziel dieses Kapitels ist die Konstruktion eines Soft-Output Detektors, der möglichst wenige Veränderungen an der Hard-Output Architektur erfordert. Aus diesem Grund soll eine Näherung gewählt werden, die eine niedrige Schranke für die Bitzuverlässigkeit darstellt. Dazu wird in der Situation in Abb. 4.8 an Stelle von c der maximale Abstand

zu allen umgebenden, nicht entscheidenden Ebenen $E_i = \{E_1, E_2\}$ zur Berechnung von $L(c_{i-M})$ herangezogen.

In der Abbildung ist $a > b$. Des Weiteren gilt, dass $c \geq a$ und $c \geq b$. Somit wird der Abstand a als untere Schranke zur Berechnung der Zuverlässigkeit verwendet. Dieser Ansatz ist sehr einfach, ermöglicht jedoch eine Implementierung ohne weit reichende Veränderungen an der bestehenden Hardware.

Das Prinzip kann leicht für Signalräume mit einer höheren Dimension erweitert werden. In einem N -dimensionalen Signalraum $\mathcal{S} = \mathcal{R}^N$ können bis zu N Hyperebenen eine konvexe Entscheidungsregion formen. Wenn keine für einen empfangenen Signalpunkt entscheidende Hyperebene zur Verfügung steht, kann analog zum zweidimensionalen Signalraum der größte Abstand von \underline{r} zu den N umgebenden Hyperebenen als niedrige Schranke für $L(c_{i-M})$ gewählt werden. Die Menge der entscheidenden Ebenen kann ebenso wie im Fall von Kap. 4.5.1 statisch vorberechnet werden.

4.5.4 Berücksichtigung von Fehlerfortpflanzung

Die Feedback-Schleife in der Detektor-Architektur (Abb. 4.6) kann aufgrund der Rückkopplung fehlerhafter Detektorausgaben zu Fehlerfortpflanzung führen. Dieser Effekt beeinflusst natürlich die Berechnung der Zuverlässigkeitsinformation, was in (4.42) nicht berücksichtigt ist. Aus diesem Grund muss der Einfluss der vorhergehenden Detektorentscheidung c_{-1} auf das aktuelle Ausgabebit c_0 berechnet werden². Detektionsfehler können während des Decodiervorgangs auf zwei unterschiedliche Ursachen zurückgeführt werden:

1. Ein empfangener Signalpunkt kann durch Rauschen derart verändert werden, dass er in eine andere Entscheidungsregion verschoben wird. Die Wahrscheinlichkeit für dieses Ereignis wird als $P_{c_0}^E$ bezeichnet.
2. Eine falsche Detektorausgabe c_{-1} führt zu Fehlerfortpflanzung, die die Detektorausgabe von c_0 zu \tilde{c}_0 kippen lässt. Bei der Trellis-Repräsentation des Detektionsprozesses entspricht dies einem falschen Ausgangszustand zum Zeitpunkt t_{-1} . Der Fehler kann also bereits vor dem Zeitpunkt t_{-1} aufgetreten sein. Dieses Ereignis tritt mit der Wahrscheinlichkeit $P_{c_{-1}}^E$ auf.

Die Detektorausgabe c_0 ist binär und kann somit nur einem der beiden Datensymbole ”+1” oder ”-1” entsprechen. Aus diesem Grund ist c_0 nur dann falsch, wenn genau eines der beschriebenen Ereignisse auftritt. Treten hingegen beide Ereignisse simultan auf, ist die Detektorausgabe aufgrund ihrer binären Form wieder korrekt. Des Weiteren führt eine falsche Entscheidung c_{-1} nicht notwendigerweise zu einer falschen Entscheidung c_0 .

Im Folgenden sei c_0 die Detektorausgabe und \tilde{c}_0 die Detektorausgabe unter der Annahme, dass das Feedback invertiert wurde. Falls beide Detektorausgaben verschieden sind, das heißt $c_0 \neq \tilde{c}_0$, muss Fehlerfortpflanzung berücksichtigt werden. In diesem Fall führt eine

²Zur einfacheren Darstellung wird im Folgenden davon ausgegangen, dass $i = M$. Somit ist die vorherige Detektorausgabe c_{-1} und die aktuelle c_0 .

falsche Entscheidung c_{-1} auch zu einer falschen Entscheidung c_0 . Sind hingegen beide Detektorausgaben identisch ($c_0 = \tilde{c}_0$), so hat c_{-1} keinen Einfluss auf c_0 . In diesem Fall kann die Fehlerfortpflanzung vernachlässigt werden.

Im Folgenden soll dies formal ausgedrückt werden, indem die Wahrscheinlichkeit P_{EP} für eine korrekte Detektorausgabe c_0 unter Berücksichtigung von Fehlerfortpflanzung und in Abhängigkeit von den Bitfehler-Wahrscheinlichkeiten $P_{c_0}^E$ und $P_{c_{-1}}^E$ bestimmt wird.

$$P_{EP}(c_0) = \begin{cases} (1 - P_{c_0}^E) \cdot (1 - P_{c_{-1}}^E) \\ \quad + P_{c_0}^E \cdot P_{c_{-1}}^E & \text{falls } c_0 \neq \tilde{c}_0 \\ 1 - P_{c_0}^E & \text{falls } c_0 = \tilde{c}_0 \end{cases} \quad (4.51)$$

Der in diesem Kapitel vorgestellte Detektor produziert jedoch Log-Likelihood-Werte und keine Wahrscheinlichkeiten. Aus diesem Grund ist eine Detektorimplementierung einfacher, wenn statt der Wahrscheinlichkeiten die Zuverlässigkeitsinformationen $L(c_0)$ und $L(c_{-1})$ verwendet werden. Diese können jedoch mit der Log-Likelihood Algebra [HOP96] leicht ineinander überführt werden. Der erste Term in (4.51) ($c_0 \neq \tilde{c}_0$) kann durch die kleinste Zuverlässigkeit der beiden Entscheidungen c_0 und c_{-1} approximiert werden [HOP96]:

$$\begin{aligned} L_{EP}(c_0) &= \ln \frac{1 + e^{L(c_0) + L(c_{-1})}}{e^{L(c_0)} + e^{L(c_{-1})}} \\ &\approx \text{sign}(L(c_0)) \cdot \text{sign}(L(c_{-1})) \\ &\quad \cdot \min(|L(c_0)|, |L(c_{-1})|) \end{aligned} \quad (4.52)$$

Der zweite Term in (4.51) ($c_0 = \tilde{c}_0$) kann direkt mit $L_{EP}(c_0) = L(c_0)$ angegeben werden. Die gesuchte Zuverlässigkeitsinformation $L_{EP}(c_0)$ ergibt sich schließlich zu

$$|L_{EP}(c_0)| = \begin{cases} \min(|L(c_0)|, |L(c_{-1})|) & \text{falls } c_0 \neq \tilde{c}_0 \\ |L(c_0)| & \text{falls } c_0 = \tilde{c}_0 \end{cases} \quad (4.53)$$

Somit kann auch unter Berücksichtigung von Fehlerfortpflanzung die Zuverlässigkeitsinformation durch eine einfache Minimum-Bestimmung angegeben werden.

4.5.5 Detektor-Architektur

In diesem Kapitel werden die zur Implementierung eines Soft-Output Detektors notwendigen Hardware-Modifikationen skizziert. Wie bereits in Kap. 4.5.1 erwähnt, ergibt sich die Zuverlässigkeitsinformation $|L(c_0)|$ für die Detektorausgabe c_0 durch eine geeignete Normierung der Ebenengleichungen (siehe (4.41) und (4.42)). Die veränderte

Architektur ist in Abb. 4.9 für eine einzelne Entscheidungsebene E_1 zu sehen. Wenn die Ebenengleichungen gemäß (4.41) normiert sind, muss nur die Konstante k an den Eingängen der Schwellwert-Entscheider hinzuaddiert werden um $|L(c_0)|$ zu erhalten. Die Berechnung des Absolutbetrags ist notwendig, da das Vorzeichen nur die relative Lage des empfangenen Signalpunktes zur Ebene angibt und nicht den Hard-Output. Das korrekte mit der Detektorausgabe $c_0 \in \{+1, -1\}$ korrespondierende Vorzeichen kann mit $L(c_0) = c_0 \cdot |L(c_0)|$ einfach bestimmt werden.

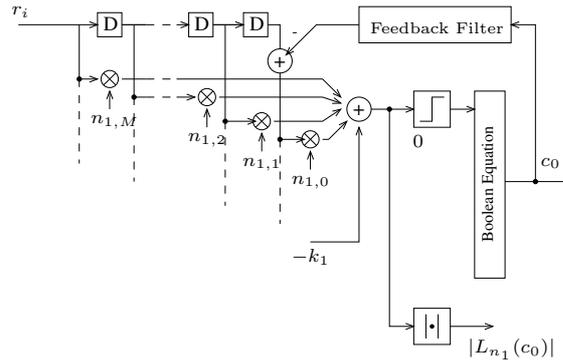


Abbildung 4.9: Berechnung der Zuverlässigkeitsinformation in der SSD-Architektur: Die Log-Likelihood-Zuverlässigkeit ist direkt an den Eingängen der Schwellwert-Entscheider erhältlich.

Eine weitere Logik (in Abb. 4.9 nicht gezeigt) muss zur Auswahl der korrekten Entscheidungsebene für die Minimierung in (4.42) und zur Berücksichtigung der Fehlerfortpflanzung laut (4.53) eingesetzt werden. Für die Berechnung der Fehlerfortpflanzung mit (4.53) müssen die Schwellwertentscheider für beide möglichen Feedback-Werte ausgewertet werden, was zu einer Erhöhung der Hardware-Komplexität führt.

Daher ergibt sich ein Soft-Output Detektor mit geringerer Komplexität (im Folgenden Low-Complexity S³D genannt), wenn der Einfluss von Fehlerfortpflanzung vernachlässigt wird. Falls ein solcher Detektor mit nicht-binären, fehlerkorrigierenden Codes kombiniert wird und die durch Fehlerfortpflanzung hervorgerufenen Bündelfehler deutlich kleiner als die Symbolgröße des Codes sind, wird der Anstieg der Fehlerrate unter Umständen nur sehr gering sein.

Hard-Output Signalraumdetektoren wurden bereits in Hinblick auf die Hardware-Komplexität mit EPR4-PRML Detektoren verglichen. Wenn beide Detektoren den gleichen beschränkten Lookahead von M besitzen, ist insbesondere der Bedarf an digitalem Speicher bei den auf dem Viterbi-Algorithmus basierenden PRML-Verfahren deutlich höher [DSS00]. Dieser Umstand ist umso bedeutender, da der Signalraumdetektor deutlich geringere Fehlerraten aufweist.

Wenn beide Detektorvarianten als Soft-Output Detektoren ausgelegt sind, vergrößert sich der Abstand weiter. Der in Kap. 4.2.2 vorgestellte Soft-Output Viterbi Algorithmus (SOVA) kann zur Berechnung der Zuverlässigkeitsinformation zwar auch einfache Minimum-Berechnungen verwenden (siehe (4.22)), allerdings wird zur Implementierung ein zweiter Trellis benötigt, der verworfene Pfade zur Berechnung der Soft-Information

betrachtet [Ber93]. Insgesamt wird dadurch die Hardware-Komplexität im Vergleich zum Hard-Output Viterbi Algorithmus in etwa verdoppelt [Ber93]. Im Gegensatz dazu kann beim Hard-Output SSD der größte Teil der Hardware für den S^3D wiederverwendet werden – insbesondere beim Low-Complexity S^3D . Bei letzterem ist nur eine weitere Boolesche Logik für die Minimierung in (4.42) notwendig. Aus diesem Grund bleibt die Gesamtarchitektur weniger komplex als die eines PRML-SOVA Detektors.

Alternativ zu den beschriebenen Verfahren kann auch der BCJR-Algorithmus [BCJR74] zur Detektion eingesetzt werden, der eine Maximum *a posteriori* (MAP) Decodierung durchführt und seine bessere Performance nur aufgrund einer deutlich höheren Komplexität erzielt [RVH95].

4.5.6 Adaptiver Signalraumdetektor

Ein Nachteil der Detektorarchitektur liegt darin, dass bei einer Änderung von Parametern, wie der Datendichte, ein komplettes Re-Design notwendig ist [DSS02]. Des Weiteren ist der Signalraum dynamisch: Die Lage und die Position der Entscheidungsebenen ändert sich mit den Parametern des Equalizers und mit der Rauschkorrelation. Aus diesem Grund ist eine adaptive Detektorarchitektur von Vorteil.

Die Struktur der Detektor-Komponenten ermöglicht die einfache Konzeption eines adaptiven Signalraumdetektors: Die einzelnen Komponenten, wie der Equalizer und die Entscheidungsebenen, ähneln Perzeptronen. Ein Perzeptron ist ein einfaches neuronales Netzwerk, bei dem ein Eingangsvektor mit einem Gewichtsvektor multipliziert wird. Das Ergebnis der Multiplikation wird dann mit einer Aktivierungsfunktion A gewichtet.

Ähnliche Operationen sind sowohl im Equalizer zur Signalfilterung, als auch bei der Implementierung der Entscheidungsebenen notwendig, so dass der Detektor leicht als Multi-Layer Perzeptron (MLP) aufgebaut werden kann [DSS02]. Ein solches adaptives Netzwerk ist in der Lage, sich an die zeitvarianten Parameter des Kanals anzupassen. Dabei werden die Gewichtsvektoren zum Beispiel nach dem so genannten Backpropagation-Lernverfahren verändert.

Durch eine einfache Modifikation der Perzeptronen kann zudem eine Erweiterung zu einem adaptiven Soft-Output Signalraumdetektor (AS^3D) realisiert werden [DS03]. Zu diesem Zweck muss die in Kap. 4.5.1 eingeführte Normierung der Entscheidungsebenen (Gleichung (4.39)) auch bei der Konzeption der Perzeptronen berücksichtigt werden. In diesem Fall kann – ähnlich wie bei der statischen Architektur aus Kap. 4.5.5 – vor der Aktivierungsfunktion des Perzeptrons die Zuverlässigkeitsinformation direkt abgegriffen werden [DS03].

Die Architektur des adaptiven Detektors zeichnet sich dann allerdings nicht mehr durch eine so niedrige Komplexität wie beim statischen Detektor aus. Allerdings kann der aufwändige Adaptionsprozess auch offline erfolgen, so dass der Detektor im Normalbetrieb statisch arbeitet. Außerdem ist der Detektor nicht nur zu einer zuverlässigen Adaption an veränderte Kanalparameter sondern auch an Parameter-Variationen der eingesetzten Bauteile in der Lage [DS03]. Somit ergibt sich auch beim Einsatz des adaptiven Detektors ein Einsparpotenzial durch die Verwendung preisgünstigerer Hardware-

Komponenten. Des Weiteren wird in Kap. 7.5.3 ein adaptiver Soft-Decoding Lesekanal vorgestellt, der mit einem nur geringen Zusatzaufwand im Vergleich zu einem statischen Hard-Decoding Kanal implementiert werden kann.

Kapitel 5

Modulationscodes

Wie in Kap. 2.1.2 beschrieben, werden auf dem magnetischen Kanal Modulationscodes eingesetzt. Deren Verwendung bietet eine Reihe von Vorteilen, die im weiteren Verlauf dieses Kapitels beschrieben werden. Grundsätzlich besteht die Aufgabe von Modulationscodes in einer Vereinfachung der Signalverarbeitung. Sie werden im Gegensatz zu den in Kapitel 6 beschriebenen fehlerkorrigierenden Codes nicht zur Behebung, sondern vorrangig zur Vermeidung von Fehlern eingesetzt. Dies kann durch das Ersetzen kritischer Sequenzen, das Hinzufügen von Timing-Information oder durch eine Anpassung des Signalspektrums geschehen. Seit der Verwendung von Partial Response Kanälen ist der Hauptgrund für den Einsatz von Modulationscodes die Generierung von Timing Information. Diese kann gewonnen werden, wenn in gewissen Mindestabständen Transitionen auf dem magnetischen Medium detektiert werden.

Allerdings haben Modulationscodes den Nachteil, dass sie weitere Redundanz in den Datenstrom einfügen und dadurch die Nutzdatendichte verringern. Des Weiteren ergibt sich bei einem Soft-Decoding Kanal das Problem, dass die Zuverlässigkeitsinformation des Detektors über den Modulationsdecoder zum Fehlerschutzdecoder propagiert werden muss. Daher ist ein geeigneter Soft-In/Soft-Out Modulationsdecoder erforderlich. Dieser erhöht die Komplexität des Lesekanals und beeinflusst die Soft-Outputs des Detektors.

Das in Kap. 3.2 präsentierte Konzept sieht die Konstruktion von leistungsarmen Lesekanälen mit Soft-Decoding vor. In diesem Kapitel werden zwei Konzepte zur Integration eines Modulationscodes in einen solchen Kanal präsentiert. Die erste Variante ist die Verwendung eines Soft-In/Soft-Out Modulationsdecoders. Nach einer Beschreibung bisheriger Vorarbeiten auf diesem Gebiet wird ein Soft-In/Soft-Out Modulationsdecoder für die bei heutigen Festplatten vorteilhaften Maximum Transition Run (MTR) Codes [MB96] vorgestellt [SS04]. Als zweite Möglichkeit wird ein Kanal vorgeschlagen, bei dem ein Parity-Check Code als Modulationscode dient [RMAAY00]. Der Parity Code wird im Fehlerschutz-Decoder zur Markierung von Detektionsfehlern eingesetzt (vgl. Kap. 3.2.4 und 6.4.2), kann allerdings darüber hinaus auch die Generierung von Timing-Information gewährleisten. Zunächst erfolgt jedoch ein Überblick über die Einsatzzwecke von Modulationscodes auf dem magnetischen Kanal und die dazu eingesetzten Verfahren.

5.1 Modulationscodes auf dem magnetischen Kanal

Die Einsatzgründe von Modulationscodes auf dem magnetischen Kanal sind vielfältig und haben sich im Laufe der Weiterentwicklung von Festplatten verändert. Die wichtigsten Gründe werden im Folgenden aufgelistet:

– Generierung von Timing-Information

Der Lesekanal einer Festplatte ist selbstsynchronisierend, das heißt die Information zur Rückgewinnung des Taktes wird aus dem Lesesignal generiert. Dies erfolgt mit Hilfe einer Phase-Lock-Loop (PLL) Schaltung. Voraussetzung dafür ist ein Mindestmaß an Timing-Information im Lesesignal. Diese kann dann gewonnen werden, wenn auf dem Medium ein Magnetfeldwechsel detektiert wird, also immer, wenn im Falle einer NRZI-Vorcodierung eine gespeicherte Eins gelesen wird.

Somit steigt die Wahrscheinlichkeit eines Synchronisierungsfehlers mit dem Abstand zweier aufeinander folgender Einsen. Umgekehrt können auch zu dicht beieinander liegende Einsen die Datenrückgewinnung erschweren: Aufgrund der Inter-symbolinterferenz (ISI) löschen sich zwei unmittelbar aufeinander folgende Transitionen teilweise aus, so dass eine Detektion erschwert wird.

Aus diesem Grund wurden bei Festplatten über einen langen Zeitraum so genannte Runlength-Limited (RLL) Codes eingesetzt [SW91]. Diese lassen sich über zwei charakteristische Größen (d, k) beschreiben. Der Parameter d markiert den Minimalabstand zwischen zwei Transitionen: $d = 0$ bedeutet zum Beispiel, dass zwei Einsen unmittelbar aufeinander folgen dürfen. Im Gegensatz dazu kennzeichnet der Parameter k den maximal zulässigen Abstand zweier Einsen.

RLL Codes hatten vor allem eine große Bedeutung, als auf dem magnetischen Kanal noch Peak-Detektoren eingesetzt wurden, bei denen jede Transition isoliert detektiert werden musste [Sie85]. Da seit der Verwendung von PRML-Kanälen die Überlagerung einzelner Lesepulse zugelassen wird, sind zwei unmittelbar aufeinander folgende Transitionen – und somit ein Parameter $d = 0$ – zulässig. Heutige Modulationscodes sind also RLL(0, k) Codes, bei denen zunehmend auch die Beschränkung des Parameters k gelockert wird, um höhere Coderaten zu erzielen [Bri98].

– Erhöhung der Detektorzuverlässigkeit

Ein weiterer wichtiger Einsatzzweck für Modulationscodes ist die Erhöhung der Detektorzuverlässigkeit. Die Wahrscheinlichkeit eines Detektions- oder Synchronisierungsfehlers ist – wie bereits im vorherigen Abschnitt beschrieben – eng mit der Wahl der Parameter (d, k) verknüpft. Nicht nur der Abstand zwischen den Transitionen des Magnetfeldes hat einen Einfluss auf die Detektionsgüte, sondern auch das Muster aufeinander folgender Transitionen. Die Untersuchung von Fehlermustern bei hoher Datendichte hat ergeben, dass vor allem bei Sequenzen mit vielen direkt aufeinander folgenden Transitionen Fehler entstehen [MB96]. Diese Muster lassen sich durch die Verwendung von RLL(1, k)-Modulationscodes vermeiden.

Die so genannten Maximum Transition Run (MTR) Codes [MB96] sind ebenfalls entwickelt worden, um Ketten von Transitionen zu unterbinden. Im Gegensatz zu herkömmlichen RLL(1, k)-Codes lassen sie sich mit einer höheren Coderate konstruieren. MTR Codes sind im Prinzip RLL(0, k)-Codes, die jedoch darüber hinaus einen Parameter j besitzen, der die maximale Anzahl direkt aufeinander folgender Transitionen festlegt. Da der d -Parameter ohnehin den Wert Null annimmt, werden MTR-Codes in der Literatur über die Notation MTR(j, k) beschrieben [BM97b].

Der Einsatz von Modulationscodes bietet vor allem bei sehr hohen Datendichten Vorteile [Str00]. Der Anteil an Transitionsrauschen im Lesesignal steigt mit der Datendichte an. Dies hat einen starken Anstieg der Fehlerrate zur Folge. Durch den Einsatz eines Modulationscodes wird die durchschnittliche Anzahl von Transitionen reduziert. Dadurch ist eine Absenkung der Fehlerrate möglich. Auf die Nutzdatenrate normiert, sind allerdings nur bei sehr hohen Datenraten und bei Verwendung eines MTR-Codes Verbesserungen zu beobachten [Str00].

– Vereinfachung der Detektorarchitektur

Neben einer Absenkung der Fehlerrate können Modulationscodes auch zu einer Vereinfachung der Detektorarchitektur eingesetzt werden. Bei den in Kap. 4.3 vorgestellten Signalraumdetektoren kann mit Hilfe eines geeigneten Modulationscodes die Position der erlaubten Signalepunkte verändert werden. Dadurch ist es möglich, entweder die Anzahl der Entscheidungsebenen zu reduzieren oder deren Lage zu verändern [BM98]. Wenn die Ebenen möglichst parallel ausgerichtet werden können, vereinfacht sich die Architektur des Detektors. Dadurch ergibt sich eine einfache Methode zur Minimierung der Verlustleistung [DSS00]. Des Weiteren wird durch MTR-Codes die Implementierung von Write-Precompensation vereinfacht, da eine Transition von einer geringeren Anzahl benachbarter Transitionen beeinflusst wird [MB96].

– Anpassung des Signalspektrums

Des Weiteren können Modulationscodes zu einer Anpassung des Signalspektrums eingesetzt werden. Durch diese Maßnahme werden Frequenzen, die besonders ver-räuscht sind, aus dem Signal herausgefiltert. Ein Beispiel hierfür sind Gleichstrom-freie Modulationscodes, die so genannten Matched Spectral Null (MSN) Codes [SW91].

Die Betrachtung der verschiedenen Einsatzgründe von Modulationscodes hat gezeigt, dass bei heutigen Festplatten vor allem MTR-Codes eine Reihe von Vorteilen bieten. Aus diesem Grund werden im weiteren Verlauf dieser Arbeit keine anderen Modulationscodes betrachtet.

5.2 Modulationscodes auf Soft-Decoding Kanälen

Wenn ein MTR-Code auf einem Kanal mit Zuverlässigkeitsinformation verwendet werden soll, muss der Decoder die Soft-Outputs des Detektors weiterverarbeiten können.

Der Decoder eines Modulationscodes kann über ein System von Booleschen Gleichungen beschrieben werden. Dabei wird jedes Ausgabebit durch einen Satz von Eingangsbits bestimmt. Für einen Soft-In/Soft-Out Modulationsdecoder bedeutet dies, dass die Zuverlässigkeitsinformation dieser Eingangsbits zu der Zuverlässigkeitsinformation des Ausgabebits beiträgt.

Das System von Booleschen Gleichungen kann auf einen Satz von rudimentären Booleschen Termen *NOT*, *AND*, *OR* sowie *exclusive – OR* (*XOR*) zurückgeführt werden. In der Vergangenheit wurde ein großer Aufwand zur Minimierung dieser Booleschen Terme für verschiedene Modulationscodes betrieben [ACH83]. Wenn eine Methode zur Weiterverarbeitung der Zuverlässigkeitsinformation in den Booleschen Termen existiert, kann daraus ein Soft-In/Soft-Out Modulationsdecoder konstruiert werden, dessen Struktur sich direkt aus dem Hard-Output Decoder ergibt.

Ziel dieses Kapitels ist der Entwurf von logischen Gattern, die Soft-Informationen an den Eingängen verarbeiten können und am Ausgang einen Log-Likelihood-Wert der logischen Verknüpfung zur Verfügung stellen. Mit Hilfe dieser so genannten Log-Likelihood Gatter wird dann ein Soft-In/Soft-Out Modulationsdecoder konstruiert [SS04]. Der Decoder basiert auf einem MTR-Code von Brickner und Moon [MB96]; das Prinzip kann jedoch auch auf andere Modulationscodes übertragen werden. Zunächst erfolgt jedoch eine Beschreibung bisheriger Arbeiten auf diesem Gebiet und eine Betrachtung des verwendeten MTR-Codes.

5.2.1 Integration eines Modulationscodes

Die wichtigste Ausgabe eines Modulationscodes ist die Erzeugung von Timing-Information mit Hilfe eines ausreichend kleinen k -Parameters. Die einfachste Methode ist das Einfügen einer Eins nach k Datenbits [RMAAY00]. Somit wird ein einfacher Modulationscode der Rate $\frac{k}{k+1}$ konstruiert. Der Code ist symmetrisch und kann durch Entfernen der Einsen sehr einfach decodiert werden. Die Zuverlässigkeitsinformation der k Datenbits wird bei der Decodierung unverändert gelassen, die der eingefügten Eins wird verworfen. In Kombination mit iterativen fehlerkorrigierenden Codes, wie zum Beispiel Turbo Codes [BGT93] oder Low-Density Parity-Check Codes [Gal62], bietet ein solcher Code zudem den Vorteil, dass die feste Position von Einsen im Datenstrom für Bits mit einer sehr hohen Zuverlässigkeit sorgt. Dies ist bei dem iterativen Decodierprozess von Vorteil [RMAAY00]. Alternativ kann ein einfacher Parity-Check Code mit der Regel Odd-Parity als Modulationscode verwendet werden [AAM01]. Dieser generiert durch die Parity-Bits auch dann Einsen im Datenstrom, wenn ausschließlich Nullen gespeichert werden. Somit wird ein Minimum von Timing-Information im Datenstrom garantiert. In Kap. 5.3 wird darüber hinaus gezeigt, wie ein einfacher Parity-Check Code mit einem Erasure-korrigierenden Reed-Solomon Code kombiniert werden kann.

Des Weiteren hat Anim-Appiah [AAM01] einen allgemeinen Ansatz für die Berechnung der Zuverlässigkeitsinformation am Ausgang eines Modulationsdecoders aufgestellt: Die Log-Likelihood Werte der Ausgabebits des Decoders werden über die übertragenen Codewörter des Modulationscodes ausgedrückt. Für den Ansatz sind jedoch komplexe Berechnungen notwendig, da – ähnlich wie beim Viterbi-Algorithmus – die Bestimmung

einer Metrik notwendig ist.

Fan et al. [FC98] haben das Problem zur Integration des Modulationscodes in einen Soft-Decoding Lesekanal durch eine umgekehrte Verkettung [Bli81] von Modulations- und Fehlerschutzcode gelöst. Durch diese Maßnahme wird der Fehlerschutzcode, der in diesem Fall systematisch sein muss, zuerst decodiert. Allerdings wird ein separater Modulationsdecoder für die Parity-Information des Fehlerschutzcodes benötigt [FC98].

Ein weiterer Weg ist das absichtliche Einfügen von Bitfehlern auf der Encoderseite, wenn der geforderte k -Parameter nicht eingehalten wird [VP03]. Durch diese Maßnahme kann ein Modulationscode komplett eingespart werden. Die eingesparte Redundanz wird dann zur Verbesserung der Fehlerkorrektur eingesetzt, so dass die zusätzlich eingebauten Fehler korrigiert werden können. Bei Verwendung von LDPC-Codes konnte dieser künstlich erzeugte Error Floor durch den Einsatz von niederratigen Codes mit besseren Distanzeigenschaften kompensiert werden [VP03]. Ein Nachteil dieses Verfahrens ist jedoch, dass die Anzahl der eingefügten Fehler von der gespeicherten Datensequenz abhängig ist. Dadurch kann unter Umständen an einigen Stellen im Datenstrom eine nicht korrigierbare Fehleranzahl erzeugt werden, obwohl die gemittelte Fehlerrate sehr niedrig ist.

5.2.2 Arbeitsweise eines MTR-Decoders

Die Diskussion in Kap. 5.1 hat gezeigt, dass die Verwendung von MTR-Codes Vorzüge in Hinblick auf Zuverlässigkeit und Detektor-Design bietet. Aus diesem Grund werden im weiteren Verlauf dieser Arbeit ausschließlich diese Codes betrachtet. Für den hier präsentierten MTR-Code soll später mit Hilfe von Log-Likelihood-Gattern ein Soft-In/Soft-Out Decoder entwickelt werden.

MTR-Codes lassen sich mit einer relativ hohen Coderate als Block-Codes fester Länge implementieren [MB96]. Die Erzeugung und die Decodierung kann dann mit Hilfe von Tabellen erfolgen. Dabei wird eine unbeschränkte Datensequenz blockweise auf eine Sequenz abgebildet, die die Beschränkungen j und k des MTR Codes erfüllt.

Im Decoder werden die ursprünglichen Daten über ein minimiertes Boolesches Gleichungssystem wieder hergestellt [BM97b]. Auf einem Kanal mit Zuverlässigkeitsinformation bestehen die Eingangsdaten des Decoders aus Log-Likelihood-Werten, bei denen das Vorzeichen den binären Wert und der Betrag die Zuverlässigkeit markiert [HOP96]. Wenn die benötigten Booleschen Operationen durch Gatter durchgeführt werden, die für die Verarbeitung von Log-Likelihood Werten angepasst sind, kann ein Soft-In/Soft-Out Modulationsdecoder konstruiert werden.

5.2.3 Definition Boolescher Operationen mit Log-Likelihood-Werten

In diesem Kapitel werden für die Booleschen Operationen *NOT*, *AND*, *OR* sowie *exclusive-OR* (*XOR*) Log-Likelihood-Gatter definiert. Diese können zu der Konstruktion eines beliebigen Modulationscodes verwendet werden, dessen Abbildungsvorschrift sich über ein Boolesches Gleichungssystem beschreiben lässt. In einem zweiten Schritt wird dann in Kap. 5.2.4 die Zuverlässigkeitsinformation für die Ausgänge der Gatter berechnet.

Dadurch, dass bei der Log-Likelihood-Darstellung der binäre Wert durch das Vorzeichen repräsentiert wird, müssen auch die Tabellen zur Bestimmung des Ausgangs der Gatter angepasst werden. Die entsprechenden Anpassungen sind den Tabellen 5.1 bis 5.4 zu entnehmen.

a	\bar{a}	\implies	a	\bar{a}
0	1		$a > 0$	$\bar{a} < 0$
1	0		$a < 0$	$\bar{a} > 0$

Tabelle 5.1: NOT-Operation in binärer und Log-Likelihood-Darstellung für eine Variable a .

a	b	$a \wedge b$	\implies	a	b	$a \wedge b$
0	0	0		$a > 0$	$b > 0$	$a \wedge b > 0$
0	1	0		$a > 0$	$b < 0$	$a \wedge b < 0$
1	0	0		$a < 0$	$b > 0$	$a \wedge b < 0$
1	1	1		$a < 0$	$b < 0$	$a \wedge b < 0$

Tabelle 5.2: AND-Operation in binärer und Log-Likelihood-Darstellung für zwei Variablen a und b .

a	b	$a \vee b$	\implies	a	b	$a \vee b$
0	0	0		$a > 0$	$b > 0$	$a \vee b > 0$
0	1	1		$a > 0$	$b < 0$	$a \vee b < 0$
1	0	1		$a < 0$	$b > 0$	$a \vee b < 0$
1	1	1		$a < 0$	$b < 0$	$a \vee b < 0$

Tabelle 5.3: OR-Operation in binärer und Log-Likelihood-Darstellung für zwei Variablen a und b .

5.2.4 Berechnung von Soft-Outputs für Boolesche Operationen

Für die Berechnung der Soft-Outputs eines logischen Gatters mit der Operation op muss der Log-Likelihood-Wert $L(a \text{ op } b \mid a, b)$ für den Gatter-Ausgang $a \text{ op } b$ berechnet werden. Dazu werden gemäß der Log-Likelihood Algebra [HOP96] die Wahrscheinlichkeiten, dass der Ausgang in Abhängigkeit von den Eingängen a und b den Wert Null bzw. Eins annimmt, gegenübergestellt:

$$L(a \text{ op } b \mid a, b) = \ln \frac{P(a \text{ op } b = 0 \mid a, b)}{P(a \text{ op } b = 1 \mid a, b)} \quad (5.1)$$

Ein Log-Likelihood Gatter verarbeitet dann als Eingangsdaten die Zuverlässigkeitsinformationen $L(a)$ und $L(b)$ und berechnet mit (5.1) eine Zuverlässigkeitsinformation für die Boolesche Operation $a \text{ op } b$. Im Folgenden werden derartige Gatter für die Booleschen Operationen *NOT*, *AND*, *OR* und *XOR* definiert.

a	b	$a \oplus b$	\implies	$a > 0$	$b > 0$	$a \oplus b > 0$
0	0	0	\implies	$a > 0$	$b > 0$	$a \oplus b > 0$
0	1	1	\implies	$a > 0$	$b < 0$	$a \oplus b < 0$
1	0	1	\implies	$a < 0$	$b > 0$	$a \oplus b < 0$
1	1	0	\implies	$a < 0$	$b < 0$	$a \oplus b > 0$

Tabelle 5.4: XOR-Operation in binärer und Log-Likelihood-Darstellung für zwei Variablen a und b .

Konstruktion eines Log-Likelihood NOT-Gatters

Da ein NOT-Gatter nur einen Eingang besitzt, ist eine einfache Bestimmung des Soft-Outputs möglich. Der Ausgang des Log-Likelihood NOT-Gatters wird durch eine einfache Invertierung des Eingangs bestimmt (Tab. 5.1) und kann dadurch nur den entgegengesetzten Wert des Eingangs annehmen. Somit gilt

$$\begin{aligned}
 P(\bar{a} = 0 \mid a) &= P(a = 1) \\
 &\text{ sowie} \\
 P(\bar{a} = 1 \mid a) &= P(a = 0).
 \end{aligned}
 \tag{5.2}$$

Die Zuverlässigkeitsinformation $L(\bar{a})$ kann daher mit

$$L(\bar{a} \mid a) = \ln \frac{P(\bar{a} = 0 \mid a)}{P(\bar{a} = 1 \mid a)} = \ln \frac{P(a = 1)}{P(a = 0)} = -L(a)
 \tag{5.3}$$

durch eine einfache Multiplikation von $L(a)$ mit dem Faktor (-1) berechnet werden.

Konstruktion eines Log-Likelihood AND-Gatters

Im Fall eines AND-Gatters muss die Zuverlässigkeit in Abhängigkeit beider Eingänge a und b bestimmt werden. Nach Tab. 5.2 gilt für die Wahrscheinlichkeit, dass der Ausgang des Gatters den Wert Null bzw. Eins annimmt:

$$\begin{aligned}
 P(a \wedge b = 0 \mid a, b) &= P(a = 0)P(b = 0) + P(a = 0)P(b = 1) + P(a = 1)P(b = 0) \\
 &\text{ und}
 \end{aligned}
 \tag{5.4}$$

$$P(a \wedge b = 1 \mid a, b) = P(a = 1)P(b = 1) = 1 - P(a \wedge b = 0 \mid a, b).$$

Mit

$$\begin{aligned}
L(a) &= \ln \frac{P(a=0)}{P(a=1)} \\
\Leftrightarrow P(a=1) &= \frac{1}{1 + e^{L(a)}} \\
\Leftrightarrow P(a=0) &= \frac{e^{L(a)}}{1 + e^{L(a)}}
\end{aligned} \tag{5.5}$$

und (5.4) lässt sich durch Einsetzen zeigen, dass

$$L(a \wedge b \mid a, b) = \ln \left(e^{L(a)} + e^{L(b)} + e^{L(a)} e^{L(b)} \right). \tag{5.6}$$

Der Verlauf von $L(a \wedge b \mid a, b)$ ist in Abb. 5.1 für einen Wertebereich von $-15 < L(a) < 15$ eingezeichnet. Der Wert von $L(b)$ ist auf -2 und $+2$ fest eingestellt.

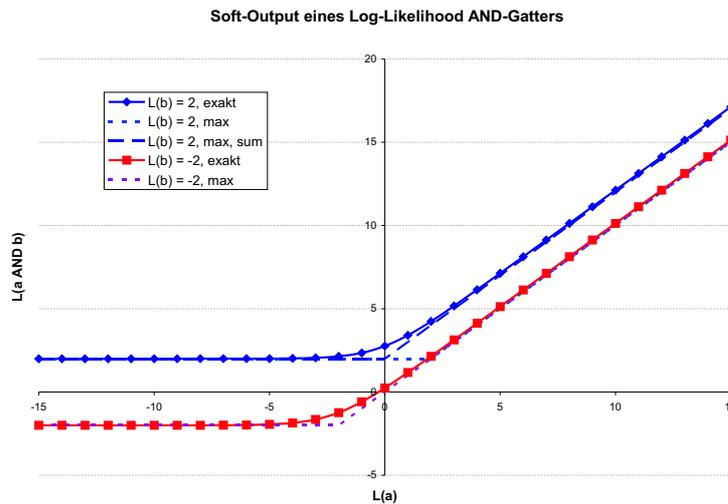


Abbildung 5.1: Der Verlauf von $L(a \wedge b \mid a, b)$ für $-15 < L(a) < 15$. $L(b)$ ist fest auf die Werte -2 und $+2$ eingestellt.

Für die Implementierung eines Log-Likelihood *AND*-Gatters ist die Berechnung der Logarithmen und Exponentialfunktionen in (5.6) bei digitaler Logik mit einem hohen Hardware-Aufwand verbunden. Allerdings kann der Verlauf von $L(a \wedge b \mid a, b)$ angenähert werden. Die absolut größte Zuverlässigkeit in (5.6) dominiert den Soft-Output, so dass $L(a \wedge b \mid a, b)$ mit

$$L(a \wedge b \mid a, b) = \max(L(a), L(b)) \tag{5.7}$$

approximiert werden kann. Die Güte dieser Näherung ist in Abb. 5.1 für verschiedene Werte von $L(a)$ und $L(b)$ abzulesen. Falls beide Werte positiv sind, ist die Approximation

$$L(a \wedge b | a, b) = L(a) + L(b) \quad \forall L(a), L(b) > 0 \quad (5.8)$$

besser geeignet (siehe Abb. 5.1). Zur Vereinfachung kann aber auch in diesem Fall (5.7) verwendet werden. Die zugewiesene Zuverlässigkeit ist dann etwas zu klein. Dies ist jedoch weniger kritisch als ein zu großer Wert. In diesem Fall reduziert sich die Berechnung des Soft-Outputs also auf eine einfache Maximierung. Diese kann dann mit relativ geringem Schaltungsaufwand implementiert werden.

Konstruktion eines Log-Likelihood OR-Gatters

Für die Berechnung der Zuverlässigkeit eines OR-Gatters ergibt sich eine sehr ähnliche Berechnung wie beim AND-Gatter. Nach Tab. 5.3 gilt für die Wahrscheinlichkeit, dass der Ausgang des Gatters den Wert Null bzw. Eins annimmt:

$$P(a \vee b = 1 | a, b) = P(a = 0)P(b = 1) + P(a = 1)P(b = 0) + P(a = 1)P(b = 1)$$

und (5.9)

$$P(a \vee b = 0 | a, b) = P(a = 0)P(b = 0) = 1 - P(a \vee b = 1 | a, b).$$

Mit (5.5) und (5.9) lässt sich durch Einsetzen zeigen, dass

$$L(a \vee b | a, b) = \ln \frac{e^{L(a)}e^{L(b)}}{1 + e^{L(a)} + e^{L(b)}}. \quad (5.10)$$

In Abb. 5.2 ist der Verlauf von $L(a \vee b | a, b)$ für einen Wertebereich von $-15 < L(a) < 15$ eingezeichnet. Der Wert von $L(b)$ ist wiederum auf die Werte von -2 und $+2$ fest eingestellt.

Analog zum Vorgehen beim AND-Gatter, kann auch der Verlauf von $L(a \vee b | a, b)$ angenähert werden. Die absolut kleinste Zuverlässigkeit in (5.10) dominiert den Soft-Output, so dass $L(a \vee b | a, b)$ mit

$$L(a \vee b | a, b) = \min(L(a), L(b)) \quad (5.11)$$

approximiert werden kann. In Abb. 5.2 ist für verschiedene Werte von $L(a)$ und $L(b)$ zu erkennen, dass die Näherung den exakten Verlauf gut wiedergibt. Falls beide Werte negativ sind, lässt sich die Kurve allerdings besser mit der Approximation

$$L(a \vee b | a, b) = L(a) + L(b) \quad \forall L(a), L(b) < 0 \quad (5.12)$$

annähern. Wie schon beim AND-Gatter kann aber zur Vereinfachung (5.11) verwendet werden.

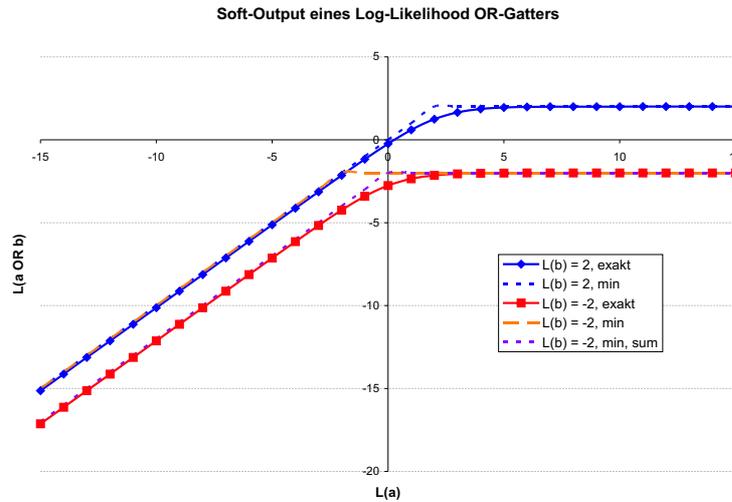


Abbildung 5.2: Der Verlauf von $L(a \vee b | a, b)$ für $-15 < L(a) < 15$. $L(b)$ ist fest auf die Werte -2 und $+2$ eingestellt.

Konstruktion eines Log-Likelihood *XOR*-Gatters

Analog zu dem Vorgehen beim *AND*- und beim *OR*-Gatter, lässt sich die Zuverlässigkeitsinformation eines *XOR*-Gatters mit Hilfe von Tab. 5.4 bestimmen. Die Wahrscheinlichkeit, dass der Ausgang des Gatters den Wert Null bzw. Eins annimmt beträgt

$$P(a \oplus b = 0 | a, b) = P(a = 0)P(b = 0) + P(a = 1)P(b = 1)$$

$$\text{und} \tag{5.13}$$

$$P(a \oplus b = 1 | a, b) = P(a = 1)P(b = 0) + P(a = 0)P(b = 1) = 1 - P(a \oplus b = 0 | a, b).$$

Mit (5.5) und (5.13) lässt sich durch Einsetzen zeigen, dass

$$L(a \oplus b | a, b) = \ln \frac{1 + e^{L(a)}e^{L(b)}}{e^{L(a)} + e^{L(b)}}. \tag{5.14}$$

Der Verlauf von $L(a \oplus b | a, b)$ ist in Abb. 5.3 für einen Wertebereich von $-15 < L(a) < 15$ eingezeichnet. Der Wert von $L(b)$ ist auf -2 und $+2$ fest eingestellt.

Analog zum Vorgehen beim *AND*- und beim *OR*-Gatter, kann auch der Verlauf von $L(a \oplus b | a, b)$ angenähert werden. Der kleinste Absolutbetrag der Zuverlässigkeiten in (5.14) dominiert den Soft-Output, so dass $L(a \oplus b | a, b)$ mit

$$L(a \oplus b | a, b) = \text{sign}(L(a)) \cdot \text{sign}(L(b)) \cdot \min(|L(a)|, |L(b)|) \tag{5.15}$$

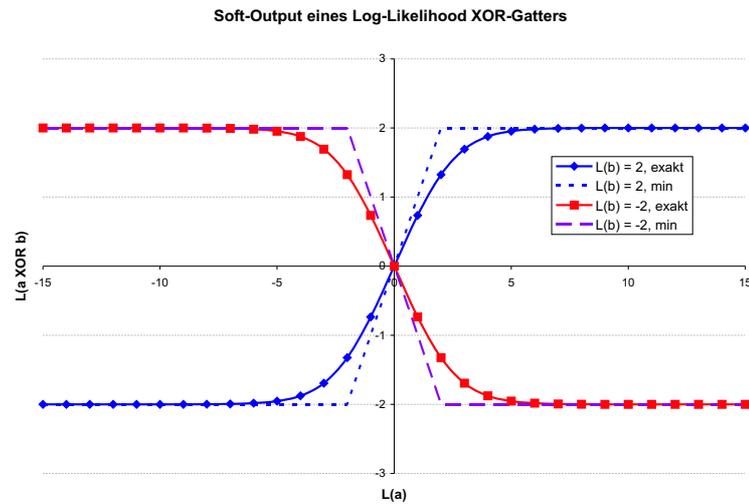


Abbildung 5.3: Der Verlauf von $L(a \oplus b | a, b)$ für $-15 < L(a) < 15$. $L(b)$ ist fest auf die Werte -2 und $+2$ eingestellt.

approximiert werden kann. Die Güte dieser Näherung ist in Abb. 5.3 für verschiedene Werte von $L(a)$ und $L(b)$ ablesbar.

Der strukturelle Aufbau der Log-Likelihood Gatter ist in Abb. 5.4 zu sehen. Sofern die Approximationen aus (5.7), (5.11) und (5.15) verwendet werden, ist eine Implementierung mit einfachen Minimum/Maximum Operationen möglich. Diese können in digitaler Schaltungstechnik mit Hilfe von Komparator-Schaltungen einfach in Hardware realisiert werden. Für das *XOR*-Gatter muss zusätzlich der Absolutbetrag der Eingänge gebildet werden. Zu diesem Zweck muss höchstens das Vorzeichenbit invertiert werden. Mit Hilfe der Log-Likelihood Gatter aus Abb. 5.4 wird im folgenden Kapitel ein Soft-In/Soft-Out Modulations-Decoder aufgebaut.

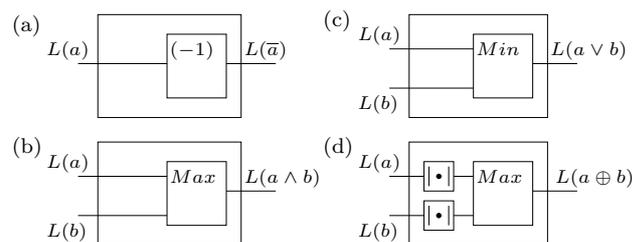


Abbildung 5.4: Struktur der Log-Likelihood Gatter: (a) *NOT*-Gatter, (b) *AND*-Gatter, (c) *OR*-Gatter, (d) *XOR*-Gatter

5.2.5 Konstruktion des Soft-In/Soft-Out Decoders

Mit Hilfe der in Kap. 5.2.4 definierten logischen Log-Likelihood Gatter kann der Modulationsdecoder aus Kap. 5.2.2 schließlich in einen Soft-In/Soft-Out Decoder umgewandelt

werden. Dazu müssen alle Booleschen Operationen, mit denen der Decoder beschrieben wird, durch die entsprechenden Operationen aus (5.3), (5.7) sowie (5.11) ersetzt werden. Durch diese Maßnahme wird die Komplexität des Decoders erhöht. Allerdings lassen sich die Berechnungen gut durch einfache Minimum- bzw. Maximum-Bildungen annähern.

Der beschriebene Soft-In/Soft-Out MTR Modulationsdecoder wird in Kap. 7.5 untersucht. Dabei wird ein Soft-Decoding Kanal unter Verwendung eines MTR-Codes mit einem Kanal verglichen, bei dem ein in den Datenstrom eingefügter Parity-Check Code als Modulationscode dient. Im folgenden Abschnitt wird das Konzept eines solchen Kanals erläutert.

5.3 Verwendung von Parity-Checks als Modulationscode

Heutige Detektoren erlauben seit der Einführung von Partial-Response Kanälen die Wahl eines d -Parameters mit dem Wert Null. Des Weiteren kann die auf die Nutzdatendichte normierte Fehlerrate eines magnetischen Kanals nur in den seltensten Fällen durch einen Modulationscode verbessert werden [Str00]. Der Hauptgrund für den Einsatz von Modulationscodes ist inzwischen die Gewährleistung von ausreichend Timing-Information im Lesesignal durch eine geeignete Wahl des Parameters k . Um höhere Coderaten und damit auch höhere Nutzdatenraten zu erreichen, wird zunehmend auch die Beschränkung von k gelockert [Bri98].

Dies führt zu der Idee eines Lesekanals ohne die Verwendung eines MTR- oder RLL-Codes. In Kap. 3.2.4 wurde bereits ein Konzept für einen Soft-Decoding Kanal unter Verwendung eines Erasure-korrigierenden Reed-Solomon Codes präsentiert. Dieser Kanal verwendet einen zusätzlichen Parity-Check Code, der als innerer fehlererkennender Code (engl. Error Detecting Code, EDC) zur Identifizierung von Detektionsfehlern dient. Der Parity-Code wird so ausgelegt, dass die typischen Fehlermuster des Detektors erkannt werden. Der genaue Decodierprozess dieses Verfahrens wird in Kap. 6.4.2 beschrieben. In diesem Kapitel soll gezeigt werden, dass der Parity-Check Code als RLL(0, k)-Modulationscode verwendet werden kann.

Bei der Kombination eines Erasure-korrigierenden Reed-Solomon Codes mit einem inneren Parity-Check Code werden zwischen den 8- oder 10-Bit Codesymbolen des RS-Codes Parity-Bits in den Datenstrom eingefügt (Abb. 5.5).

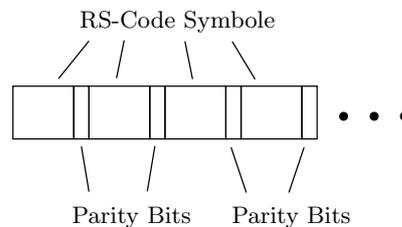


Abbildung 5.5: Wenn hinter jedem Codesymbol des Reed-Solomon (RS) Codes ein Parity-Bit eingefügt wird, werden in jedem Fall Einsen gespeichert und somit Timing-Information erzeugt.

Wenn die Parity-Bits als Odd-Parity¹ abgespeichert werden, sorgen die eingefügten Bits dafür, dass in jedem Fall Einsen gespeichert werden. Wenn für jedes Codesymbol ein Parity-Bit generiert wird und die Anordnung wie in Abb. 5.5 erfolgt, kann bei 8-Bit Symbolen ein k -Parameter von 15 garantiert werden. Der Parity-Check Code ist somit ein $RLL(0, 15)$ -Code. Der Worst-Case von 15 aufeinander folgenden Nullen ist in Abb. 5.6 zu sehen. Bei Festplatten werden häufig Reed-Solomon Codes mit 10-Bit Codesymbolen eingesetzt. In diesem Fall kann nur ein k -Parameter von 19 garantiert werden.

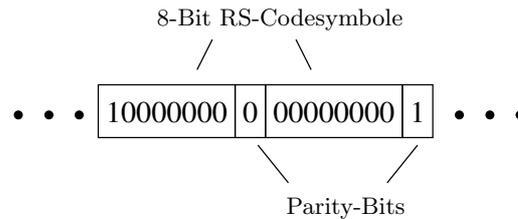


Abbildung 5.6: Das Verfahren aus Abb. 5.5 kann bei 8-Bit RS-Codesymbolen und Odd-Parity $k = 15$ garantieren. Der Worst-Case tritt ein, wenn die beiden gezeigten Codesymbole aufeinander folgen.

Der Parity-Check Code muss nicht notwendigerweise wie in Abb. 5.5 konstruiert werden. Wenn auch 2-Bit Fehler erkannt werden sollen, müssen zwei Parity-Bits eingesetzt werden. Dabei wird das erste über die geraden Bits und das andere über die ungeraden Bits gebildet. Um die Coderate nicht abzusenken, kann der Parity-Code über zwei RS-Symbole gebildet werden (Abb. 5.7 oben). Dennoch kann in diesem Fall die Anordnung im Codewort wie in Abb. 5.5 erfolgen, um weiterhin $k = 15$ zu ermöglichen (Abb. 5.7 unten). Der Generierungsaufwand erhöht sich in diesem Fall jedoch leicht, da eines der Parity-Bits zwischen den beiden RS-Code Symbolen, über die es gebildet wird, eingefügt werden muss.

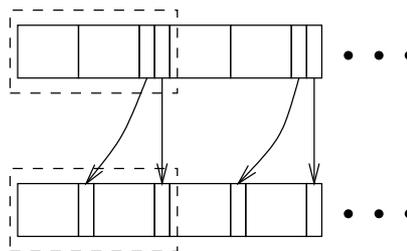


Abbildung 5.7: Wenn der Parity-Check Code mit mehreren Parity-Bits und über mehrere RS-Symbole gebildet wird (gestrichelt), wird der k -Parameter größer (oben). Durch eine andere Anordnung der Parity-Bits kann k aber konstant gehalten werden (unten).

Im Vergleich zu dem in Kap. 5.2 vorgestellten MTR-Code besitzt der hier gezeigte Code ein größeres k . Daher steigt die Wahrscheinlichkeit eines Synchronisierungsfehlers. Der Trend bei Festplatten geht allerdings dahin, ein höheres k zuzulassen, um damit den Einsatz von hochratigeren Modulationscodes zu ermöglichen [Bri98]. Da die Daten zusätzlich zur Fehlerkorrektur durch einen Cyclic Redundancy Check (CRC) abgesichert werden,

¹Die Summe der Einsen in einem Codewort ist ungerade.

kann ein Synchronisierungsfehler sicher erkannt werden. Falls der CRC fehlschlägt, wird der entsprechende Sektor erneut gelesen (engl. Re-Read).

Des Weiteren werden keine Beschränkungen auf die erlaubte Anzahl direkt aufeinander folgender Transitionen gelegt ($j = \infty$). Daher ist ein Lesekanal mit dem vorgeschlagenen Modulationscode wahrscheinlich anfälliger für die von Brickner et al. beschriebenen typischen Fehlermuster [MB96]. Allerdings kann die Redundanz des Modulationscodes, wie beschrieben, zur Verbesserung der Fehlerkorrektur genutzt werden.

Das Prinzip wurde in dieser Arbeit exemplarisch für einen einfachen Parity-Check Code erläutert. Neben dieser sehr einfachen Lösung ist auch der Einsatz komplexerer fehlererkennender Codes denkbar, die beispielsweise strengere Beschränkungen auf k oder auch j ermöglichen.

Die einfache Implementierbarkeit eines Parity-Check Codes im Vergleich zu einem MTR-Code ist auch in Hinblick auf die Verlustleistung von Vorteil. Aus diesem Grund werden beide Varianten in Kap. 7.5 in Bezug auf die erzielbare Bitfehlerrate und auf die Hardware-Komplexität miteinander verglichen.

Kapitel 6

Fehlerkorrigierende Codes mit Zuverlässigkeitsinformation

Stark steigende Datendichten erfordern einen höheren Aufwand für die Fehlerkorrektur eines magnetischen Kanals, um auch weiterhin die erforderlichen Bitfehlerraten von mindestens 10^{-13} gewährleisten zu können (siehe Kap. 3.1). Dies kann zum Beispiel durch die Verwendung von fehlerkorrigierenden Codes mit einer niedrigeren Coderate erfolgen. Dies hat allerdings sowohl auf die Hardwarekomplexität sowie auf die erzielbare Datenübertragungsrate einen negativen Einfluss. Aus diesem Grund sind insbesondere für Festplatten im mobilen Einsatz neue Ansätze erforderlich, da die Aufrechterhaltung der Zuverlässigkeit nicht mit einer Steigerung der Verlustleistung einhergehen darf.

Seit der Vorstellung von Turbo Codes durch Berrou et al. [BGT93] haben fehlerkorrigierende Verfahren, die auf der Basis von Zuverlässigkeitsinformation arbeiten, viel Aufmerksamkeit erlangt und eine Vielzahl von Veröffentlichungen auf diesem Gebiet stimuliert. Es hat sich herausgestellt, dass durch die Verwendung von Zuverlässigkeitsinformation ein Codiergewinn von 1,5-3 dB gegenüber gewöhnlicher Hard-Decodierung möglich ist [CHIW98]. Aus diesem Grund werden diese so genannten Soft-Decoding Verfahren in der Zukunft in den meisten Einsatzgebieten verwendet werden.

Auch im Bereich der Datenspeicherung – und hier insbesondere auf dem Gebiet der magnetischen Aufzeichnung – sind bereits verschiedene fehlerkorrigierende Codes unter der Verwendung von Zuverlässigkeitsinformation untersucht worden, so zum Beispiel Turbo Codes [RMM98, SFO⁺99] oder Low-Density Parity-Check (LDPC) Codes [STC01, OM01]. Dennoch bleiben Reed-Solomon Codes aufgrund ihrer hervorragenden Eigenschaften auf Kanälen mit Bündelfehlern bei Festplatten erste Wahl.

In diesem Kapitel werden verschiedene fehlerkorrigierende Verfahren für den magnetischen Kanal untersucht. Die Basis für diese Untersuchungen bilden Reed-Solomon Codes, die mittels Erasure-Decodierung um eine einfache Variante der Verarbeitung von Zuverlässigkeitsinformation erweitert werden [SSHS03]. Als Vergleich werden mit LDPC Codes und Turbo Codes zwei der bekanntesten Soft-Decoding Algorithmen herangezogen. Zunächst erfolgt jedoch eine allgemeine Betrachtung fehlerkorrigierender Codes und ihrer grundlegenden Eigenschaften.

6.1 Eigenschaften und Definitionen

Das grundsätzliche Prinzip von fehlerkorrigierenden Codes ist das Hinzufügen redundanter Information in den Datenstrom. Durch diese Maßnahme werden künstlich Lücken zwischen gültigen Codewörtern geschaffen, die im Falle einer teilweise fehlerhaften Übertragung die Rekonstruktion des ursprünglichen Datenwortes ermöglichen. Dazu wird ein aus k Informationssymbolen bestehendes Datenwort auf ein Codewort mit n Codesymbolen abgebildet, wobei gilt, $n > k$. Das Verhältnis $\frac{k}{n}$ wird als Coderate R bezeichnet. Ein auf diese Art erzeugter Code ist ein Block-Code.

Wenn die Summe zweier beliebiger Codewörter ebenfalls wieder ein Codewort ist, wird der Code als linear bezeichnet. Ein Code ist symmetrisch, wenn das Datenwort unverändert und zusammenhängend einen Teil des Codeworts bildet. Es existieren auch Codes mit $k = n$ und der Coderate 1, zum Beispiel kann der PRML-Kanal aus Kap. 4.2 als ein solcher Code betrachtet werden. Diese sind dann allerdings nicht zur Korrektur von Übertragungsfehlern in der Lage. Allgemein kann alles als ein Code bezeichnet werden, das ein Gedächtnis in ein System einfügt [Hag99].

Die Symbole eines fehlerkorrigierenden Codes stammen in der Regel aus einem Galois-Feld¹ (endlicher Körper). Bei den vielfach anzutreffenden binären Codes stammen die Elemente aus dem Galois-Feld $GF(2)$, das heißt der Menge $\{0, 1\}$. Für einige Codes existieren Erweiterungen auf mehr-bittige Symbole aus einem Galois-Feld $GF(q)$ mit den Elementen $\{0, 1, \dots, q - 1\}$. Jedes Galois-Feld besitzt ein primitives Element α , dessen Potenzen α^j mit $j = \{0, \dots, q - 1\}$ alle Elemente des Feldes erzeugen. Für die digitale Signalverarbeitung bietet sich die Verwendung eines Erweiterungsfeldes $GF(2^m)$ an, weil damit ein m -bittiges binäres Signal mathematisch beschrieben werden kann [Bos99].

Eine weitere wichtige Größe ist die Hammingdistanz zweier Codewörter. Diese gibt die Anzahl der Codesymbole an, in der sich zwei Codewörter unterscheiden. Die kleinste Hammingdistanz zweier Codewörter eines Codes ist eine charakteristische Größe und wird als Minimaldistanz des Codes bezeichnet. Sie gilt als wichtige Größe für die Beurteilung der Leistungsfähigkeit eines fehlerkorrigierenden Codes. Das Hamminggewicht wiederum gibt die Anzahl der von Null verschiedenen Codesymbole an.

6.2 Decodierprinzipien

Im Decoder müssen die gespeicherten Daten aus der von der Festplatte gelesenen Sequenz rekonstruiert werden. Dabei sind drei verschiedene Decodier-Ergebnisse möglich:

- **Erfolgreiche Korrektur:** Die empfangenen Daten können dem übertragenen Codewort korrekt zugeordnet werden. In diesem Fall ist der Datenstrom nach der Decodierung fehlerfrei.
- **Decodierversagen:** Die empfangenen Daten können keinem gültigen Codewort zugeordnet werden. Dieser Fehler wird jedoch bei der Decodierung erkannt. Somit

¹Die Theorie der Galois-Felder (endliche Körper) dient als Grundlage vieler fehlerkorrigierender Codes. Eine Einführung ist in vielen Büchern zum Thema Codierung zu finden [LC83, Wic95, Bos99].

können geeignete Maßnahmen getroffen werden, zum Beispiel das erneute Lesen der Information von der Festplatte.

- **Falschkorrektur:** Die empfangenen Daten werden einem falschen Codewort zugeordnet. Dieser Fehler wird nicht erkannt, da das Ergebnis der Decodierung ein gültiges Codewort ist. Bei Festplatten wird zur Erkennung von Falschkorrekturen ein zusätzlicher Cyclic Redundancy Check (CRC) durchgeführt.

Seit Shannons für die Codierungstheorie grundlegender Arbeit aus dem Jahr 1948 [Sha48] sind eine Vielzahl von Verfahren zur Erzeugung und Decodierung von fehlerkorrigierenden Codes entwickelt worden. Die für diese Arbeit relevanten Aspekte werden in diesem Abschnitt vorgestellt. Fehlerkorrigierende Codes können nach verschiedenen Decodierprinzipien eingeteilt werden:

- **Fehlererkennung:** Bei einer einfachen Fehlererkennung kann ein Datenwort – wie der Name bereits nahe legt – nur auf Fehlerfreiheit überprüft werden. Eine Korrektur etwaiger Übertragungsfehler ist nicht möglich. Beispiele für fehlererkennende Codes sind ein einfacher Parity-Check Code oder der bereits erwähnte Cyclic Redundancy Check (CRC).
- **Bounded Minimum Distance (BMD) Decodierung:** Im Falle einer BMD-Decodierung wird ein fehlerhaftes Codewort nur dann einem gültigen Codewort zugeordnet, wenn es nicht weiter als die halbe Minimaldistanz von diesem entfernt ist. Das Verfahren findet in erster Linie bei Block-Codes Anwendung, deren Decodierprozess auf einem algebraischen Algorithmus basiert, zum Beispiel bei Reed-Solomon Codes. Ist die Distanz zum nächstgelegenen Codewort größer als die halbe Minimaldistanz, wird der Decodiervorgang mit dem Ergebnis Decodierversagen abgebrochen. Eine Falschkorrektur ist ebenfalls möglich.
- **Maximum Likelihood (ML) Decodierung:** Das Prinzip der Maximum Likelihood Decodierung wurde bereits in Kap. 4.1 vorgestellt. Hierbei wird ein empfangenes Codewort immer dem ähnlichsten gültigen Codewort zugeordnet. Ein Decodierversagen ist daher nicht möglich, eine Falschkorrektur hingegen schon. Der Viterbi-Algorithmus [Vit67] ist das bekannteste Beispiel einer ML-Decodierung (siehe Kap. 4.2.2).
- **Symbolweise Maximum A Posteriori (MAP) Decodierung:** Ein MAP-Decoder maximiert die a posteriori Wahrscheinlichkeit für jedes übertragene Codesymbol getrennt. Auch dieses Verfahren ist bereits in Kap. 4.1 vorgestellt worden. In Gegensatz zur Maximum Likelihood Decodierung ist das Ergebnis nicht notwendigerweise ein gültiges Codewort. Somit sind sowohl Falschkorrektur, als auch Decodierversagen möglich. Ein Beispiel für eine MAP-Decodierung ist der bei Turbo Codes verwendete BCJR-Algorithmus [BCJR74], der mittels eines probabilistischen Verfahrens die Wahrscheinlichkeit der einzelnen Codesymbole maximiert (siehe Kap. 6.5).
- **Iterativer Decodieralgorithmus (IDA):** Der iterative Decodieralgorithmus ist ebenfalls eine symbolweise Decodierung, bei der die a posteriori Wahrscheinlichkeit

ten allerdings nur angenähert werden. Das Verfahren wird häufig mit einer Verkettung von zwei oder mehr fehlerkorrigierenden Codes kombiniert. Die Decodierung erfolgt dann in einem iterativen Prozess, bei dem mittels probabilistischer Verfahren Nachrichten oder Zuverlässigkeitsinformationen ausgetauscht werden. Die in dieser Arbeit betrachteten Turbo Codes und LDPC Codes gehören zu den iterativen Decodierverfahren.

6.3 Kriterien zur Auswahl eines geeigneten Codes

Die Auswahl eines geeigneten fehlerkorrigierenden Codes hängt von vielen Parametern, wie der vorgesehenen Anwendung, den veranschlagten Kosten und der zulässigen Verlustleistung, ab. Die Abhängigkeiten, die sich auf dem magnetischen Kanal ergeben, wurden bereits in Kap. 3.1 diskutiert. In diesem Abschnitt wird erläutert, welche Konsequenzen sich daraus für die Auswahl und Dimensionierung eines geeigneten fehlerkorrigierenden Codes ergeben. Abschließend erfolgt ein Ausblick auf Trends und mögliche zukünftige Entwicklungen bei Festplatten.

Die Daten werden auf einer Festplatte in Form von Sektoren mit einer Größe von derzeit 512 Byte gespeichert. Ein Sektor ist das physikalische kleinste Datenpaket, das von einer Festplatte gelesen werden kann. Aus diesem Grund werden die Codeworte typischerweise auf einen Sektor und somit auf eine Länge von 512 Byte beschränkt. Aus informationstheoretischer Sicht ist dies eine Einschränkung: Längere Codes besitzen bessere Distanzeigenschaften – und damit eine höhere Korrekturfähigkeit – als kurze Codes gleicher Coderate.

Die Auswahl eines geeigneten Fehlerkorrekturverfahrens wird bei Festplatten durch die strengen Anforderungen an den Quality-of-Service (QoS), die Datenübertragungsrate und die erforderliche Speicherdichte (siehe Kap. 3.1) stark eingeschränkt. Die Qualität des von der Festplatte gelesenen Signals ist jedoch im Vergleich zu anderen Kanälen, wie zum Beispiel dem Funkkanal, relativ hoch, so dass Codes mit sehr hohen Coderaten verwendet werden können. Der gesuchte Code muss also bei einer hohen Coderate und einem hohen Signal-Rausch-Abstand (SNR) eine sehr gute Korrekturleistung bieten. Je nach Qualität des Übertragungskanals bestimmen andere Codeeigenschaften die erzielbare Fehlerrate: Während bei hohem SNR die Minimaldistanz des Codes entscheidend ist, so bestimmt bei niedrigem SNR die Anzahl sehr ähnlicher Codewörter maßgeblich das erzielbare Ergebnis [CHIW98]. Des Weiteren beeinflussen die in Kap. 2.5 beschriebenen typischen Fehlermuster des magnetischen Kanals die Auswahl: Der Code muss in der Lage sein, sowohl die zufälligen Einzelbitfehler, als auch die kurzen Bündelfehler zu korrigieren.

Festplatten erreichen inzwischen sehr hohe Datenübertragungsraten (Kap. 3.1). Dadurch bleibt die Auswahl eines fehlerkorrigierenden Codes auf Verfahren beschränkt, die sich On-the-Fly mit hohen Datenraten decodieren lassen. Eine hohe Coderate des eingesetzten Verfahrens begünstigt diese Eigenschaft ebenso wie eine möglichst hohe Speicherdichte. Der enorme Preisdruck bei Festplatten erfordert zudem, dass sich die verwendeten Algorithmen günstig in Hardware implementieren lassen. Beim mobilen Einsatz kommt als weiterer entscheidender Aspekt die Verlustleistung der verwendeten Schaltungen hin-

zu.

Die beschriebenen Kriterien haben dazu geführt, dass bei Festplatten in erster Linie Reed-Solomon Codes eingesetzt werden. Diese bieten auch bei den auf dem magnetischen Kanal erforderlichen hohen Coderaten eine hervorragende Korrekturfähigkeit. Des Weiteren lassen sie sich in schnellen On-the-Fly Korrekturschaltungen implementieren [HSW95]. Seit der Vorstellung von Reed-Solomon Codes sind eine Vielzahl von Implementierungen entwickelt worden, die auch mit den für Festplatten erforderlichen hohen Datenraten effizient betrieben werden können. Eine Übersicht findet sich zum Beispiel in dem Buch von Wicker und Bhargava [WB94].

Der Decodieralgorithmus von Reed-Solomon Codes basiert auf einem algebraischen Verfahren unter Verwendung von Hard-Decoding. Der Trend in der Codierungstheorie geht allerdings dahin, zunehmend probabilistische Verfahren unter Verwendung von Soft-Decoding einzusetzen. Diese versprechen einen Codiergewinn von bis zu 3 dB [CHIW98]. Somit ist die Entwicklung von Soft-Decoding Lesekanälen für Festplatten eine interessante Herausforderung. Allerdings sind die Maximum Likelihood bzw. Maximum A Posteriori Algorithmen in der Praxis oft zu komplex. Daher ist eine Suche nach besser implementierbaren, suboptimalen Verfahren erforderlich. Dies kann die Verwendung mathematischer Näherungen ebenso umfassen wie den Einsatz von Interleaving oder Codeverkettung zur Verbesserung der Korrekturfähigkeit kurzer Codes.

In den folgenden Kapiteln werden daher drei verschiedene fehlerkorrigierende Verfahren vorgestellt. Alle drei haben die Eigenschaft, dass die Zuverlässigkeitsinformation des Detektors zu einer Verbesserung der Decodierleistung eingesetzt werden kann. Zu Beginn wird das bereits in Kap. 5.3 erwähnte Verfahren zur Erweiterung von Reed-Solomon Codes vorgestellt. Dabei kommt ein innerer Parity-Check Code zum Einsatz, mit dessen Hilfe unzuverlässige Codesymbole als Erasures markiert werden. Durch diese einfache Maßnahme können die ursprünglich nur zum Hard-Decoding geeigneten Reed-Solomon Codes leicht in einen Soft-Decoding Kanal integriert werden. Als Alternative werden sowohl Low-Density Parity-Check (LDPC) Codes, als auch Turbo Codes betrachtet. Beide Verfahren basieren auf einem iterativen Decodierprozess und sind direkt zur Verarbeitung von Zuverlässigkeitsinformation in der Lage.

6.4 Reed-Solomon Codes

Reed-Solomon Codes sind im Jahre 1960 von Reed und Solomon präsentiert worden [RS60]. Ursprünglich waren Reed-Solomon Codes für binäre Elemente ausgelegt, erst später erfolgte die Erweiterung auf ein beliebiges Galois-Feld $GF(2^m)$ [GZ61]. Die Länge eines Reed-Solomon Codes ist durch die Größe der Codesymbole festgelegt: Wenn m -Bit Symbole verwendet werden, ergibt sich eine Codelänge von $n = 2^m - 1$ Symbolen.

Die Anzahl k der Informationssymbole ist für alle $k < n$ frei wählbar. Somit kann bei gegebener Codelänge n eine beliebige Coderate $\frac{n-k}{n}$ gewählt werden. Es ergibt sich dann ein $RS(n, k)$ -Code mit einer Minimaldistanz von $n - k + 1$ [RS60]. Ein solcher Code kann $t = \lfloor \frac{n-k}{2} \rfloor$ Fehler pro Codewort garantiert korrigieren.

Zur Datenspeicherung bieten sich Symbole der Größe $m = 8$ Bit (oder 1 Byte) an. Dies

erfordert aber, dass mehrere Codewörter pro Sektor gespeichert werden müssen, da die Codewortlänge $n = 2^8 - 1 = 255$ kleiner als ein Sektor ist. Die Codewörter werden ineinander verschachtelt abgespeichert (Interleaving). Um einen Sektor möglichst komplett durch ein einzelnes Codewort abzudecken sind die Festplatten-Hersteller inzwischen dazu übergegangen, Reed-Solomon Codes auf der Basis von 10 Bit Codesymbolen einzusetzen. Diese sind so konstruiert, dass die Informationssymbole auch weiterhin auf 8 Bit beschränkt sind²; nur die Parity-Symbole haben eine Größe von 10 Bit.

6.4.1 Erzeugung und Decodierung von Reed-Solomon Codes

Die Beschreibung von Reed-Solomon Codes erfolgt mit Hilfe von Polynomen. Ein Informationswort der Länge k wird durch ein Polynom $i(x)$ mit dem Grad $k - 1$ beschrieben und im Encoder auf ein Codepolynom $c(x)$ mit dem Grad $n - 1$ abgebildet. Die Koeffizienten der Polynome entsprechen den Codesymbolen und stammen aus dem Galois-Feld $GF(q)$.

Zur Codeerzeugung wird ein Generator-Polynom $g(x)$ vom Grad $n - k$ verwendet. Dieses Polynom ist so definiert, dass es an $n - k$ aufeinander folgenden Potenzen des primitiven Elementes α Nullstellen besitzt:

$$g(x) = \prod_{j=0}^{n-k-1} (x - \alpha^j) = \sum_{j=0}^{n-k} g_j x^j \quad (6.1)$$

Das Codepolynom $c(x)$ wird durch Multiplikation des Informationspolynoms $i(x)$ mit dem Generatorpolynom $g(x)$ (6.1) erzeugt:

$$c(x) = i(x) \cdot g(x) \quad (6.2)$$

Der so erzeugte Code besitzt die gleichen Nullstellen wie das Generatorpolynom selbst. Allerdings ist der Code nicht systematisch, das heißt, nach der Decodierung muss noch das Informationswort berechnet werden. Alternativ kann ein Reed-Solomon Code auch mit der Vorschrift

$$c(x) = i(x) \cdot x^{n-k} - \left(i(x) \cdot x^{n-k} \bmod g(x) \right) \quad (6.3)$$

in systematischer Form erzeugt werden.

Bei der Decodierung muss ein verrauschtes Empfangspolynom $r(x)$ wieder einem gültigen Codepolynom $\hat{c}(x)$ zugeordnet werden. Der Algorithmus bestimmt dazu ein Fehlerpolynom $e(x)$ von kleinstmöglichem Grad, das mit der Vorschrift

$$r(x) = \hat{c}(x) + e(x) = \hat{i}(x) \cdot g(x) + e(x) \quad (6.4)$$

²Die beiden restlichen Bits haben daher immer den Wert Null.

mit einem gültigen Codepolynom $\hat{c}(x)$ das Empfangspolynom $r(x)$ erzeugt. Der gesamte Decodierprozess unterteilt sich in die drei folgenden Schritte:

1. Berechnung eines nur vom Fehler $e(x)$ abhängigen Syndromvektors \underline{s} . Die Syndrome s_j sind nur dann ungleich Null, wenn das Empfangspolynom kein Reed-Solomon Codewort ist.
2. Aufstellen und Lösen der so genannten Schlüsselgleichung mit Hilfe der Syndrome. Die Schlüsselgleichung besteht aus einem Fehlerstellenpolynom (engl. Error Locator Polynomial), dessen Nullstellen den Fehlerpositionen entsprechen und einem Fehlerwertpolynom (engl. Error Evaluator Polynomial), das die Berechnung der Fehlerwerte ermöglicht.
3. In der letzten Phase muss schließlich anhand der beiden Polynome der Fehlervektor $e(x)$ bestimmt werden. Anschließend kann mit Hilfe von (6.4) das wahrscheinlichste Codepolynom $\hat{c}(x)$ angegeben werden.

Seit der Vorstellung von Reed-Solomon Codes sind eine Reihe von Algorithmen zur Decodierung entworfen worden. Diese unterscheiden sich vor allem in der Berechnung der Schritte (2.) und (3.). Im Folgenden wird der konzeptionelle Weg bei der Decodierung beschrieben, eine exakte Übersicht der unterschiedlichen Decodieralgorithmen findet sich zum Beispiel in dem Buch *Reed-Solomon Codes and their Applications* von Wicker und Bhargava [WB94].

Berechnen der Syndrome

Als erstes wird der Anteil des gültigen Codepolynoms $\hat{c}(x)$ aus dem Empfangspolynom $r(x)$ herausgerechnet. Zu diesem Zweck wird die Eigenschaft ausgenutzt, dass das Codepolynom die gleichen Nullstellen wie das Generatorpolynom besitzt. Durch Einsetzen der $n - k$ Potenzen α^j in das Empfangspolynom $r(x)$ können die Komponenten

$$s_j = r(\alpha^j) \quad \text{mit } j = 0, 1, \dots, n - k - 1 \quad (6.5)$$

des so genannten Syndromvektors \underline{s} bestimmt werden. Wenn bereits das Empfangspolynom $r(x)$ ein gültiges Codepolynom ist, hat der Syndromvektor den Wert Null. Bei einem systematischen Reed-Solomon Code kann dann direkt das Informationspolynom $\hat{i}(x)$ ausgegeben werden. Wenn allerdings das Fehlerpolynom $e(x)$ ein gültiges Codepolynom ist, so ist $r(x)$ aufgrund der Linearität der Reed-Solomon Codes ein gültiges Codewort. Dadurch kann der Fehler nicht erkannt werden – es tritt eine Falschkorrektur auf.

Berechnen der Fehlerpositionen

Mit Hilfe des Syndromvektors \underline{s} kann ein Gleichungssystem zur Bestimmung des Fehlerpolynoms $e(x)$ aufgestellt werden:

$$s_j = e(\alpha^j) = \sum_{k=0}^{n-1} e_k \cdot (\alpha^j)^k \quad \text{mit } j = 0, 1, \dots, n-k-1 \quad (6.6)$$

Gesucht ist allerdings das Fehlerpolynom $e(x)$ von kleinstem Grad, das mit Hilfe von (6.4) zum Empfangspolynom $r(x)$ führt. Dies entspricht der Suche nach dem ähnlichsten Codewort $\hat{c}(x)$.

Das Gleichungssystem wird dazu nur über die ν Koeffizienten e_{k_l} aus (6.6) gebildet, die ungleich Null sind. Wenn diese von 1 bis ν durchnummeriert werden, ergibt sich mit $Y_l = e_{k_l}$ und $X_l = \alpha^{k_l}$

$$s_j = \sum_{l=1}^{\nu} e_{k_l} \cdot (\alpha^j)^{k_l} = \sum_{l=1}^{\nu} e_{k_l} \cdot (\alpha^{k_l})^j := \sum_{l=1}^{\nu} Y_l \cdot X_l^j \quad \text{mit } j = 0, 1, \dots, n-k-1. \quad (6.7)$$

Dabei steht Y_l für die Fehlerwerte und X_l für die Fehlerpositionen im Empfangspolynom $r(x)$. Die Lösung des nichtlinearen Gleichungssystems aus (6.7) ist die Aufgabe des Reed-Solomon Decoders. Neben den Fehlerstellen und den Fehlerwerten muss auch die zunächst unbekannte Anzahl ν der aufgetretenen Fehler bestimmt werden.

Die direkte Lösung der nicht-linearen Gleichung (6.7) ist aufwändig. Daher sind für eine effizientere Decodierung Zwischenschritte notwendig. Zu diesem Zweck wird das bereits erwähnte Fehlerstellenpolynom $\Lambda(x)$ vom Grad ν eingeführt [WB94]. Es besitzt Nullstellen an den inversen Fehlerpositionen X_l^{-1} , das heißt

$$\Lambda(x) = 1 + \sum_{i=1}^{\nu} \Lambda_i x^i = \prod_{l=1}^{\nu} (1 - x X_l) \quad \text{mit } X_l = \alpha^{k_l}. \quad (6.8)$$

Mit (6.7) und (6.8) kann dann ein Gleichungssystem nach der Form

$$s_j = - \sum_{i=1}^{\nu} \Lambda_i s_{j-i} \quad \text{mit } j = \nu+1, \nu+2, \dots, 2\nu \quad (6.9)$$

aufgestellt werden [WB94]. Mit dem Berlekamp-Massey Algorithmus existiert ein effizientes Verfahren, welches das jetzt lineare Gleichungssystem (6.9) mit Hilfe eines iterativen Verfahrens löst und in $n-k$ Iterationen die Koeffizienten Λ_i des Fehlerstellenpolynoms bestimmt [WB94].

Die Nullstellen des Fehlerstellenpolynoms können dann mit der so genannten Chien-Search bestimmt werden [WB94]. Dies ist eine einfache Trial-and-Error Methode, bei der für alle $j = 0, 1, \dots, n-1$ durch Einsetzen $\Lambda(\alpha^j)$ gebildet wird und anschließend überprüft wird, ob eine Nullstelle vorliegt. Bei binären Codes kann an dieser Stelle direkt das wahrscheinlichste Codewort bestimmt werden, da die alleinige Kenntnis der Fehlerorte zur Korrektur ausreicht. Im Falle von Codes über einem allgemeinen Galois-Feld $GF(q)$ müssen jedoch die Fehlerwerte explizit berechnet werden.

Berechnen der Fehlerwerte

Die Berechnung der Fehlerwerte erfolgt nach der Auswertung des Fehlerstellenpolynoms $\Lambda(x)$ über die so genannte Forney-Formel [WB94]. Als erstes muss die Schlüsselgleichung aufgestellt werden, mit der das Fehlerwertpolynom $\Omega(x)$ mit Hilfe des Syndromvektors

$$s(x) = \sum_{j=1}^{n-k} s_j x^j \quad (6.10)$$

und des Fehlerstellenpolynoms $\Lambda(x)$ zu

$$\Omega(x) = (1 + s(x)) \Lambda(x) \pmod{x^{n-k+1}} \quad (6.11)$$

definiert wird. $\Omega(x)$ ist über die Beziehung

$$\Omega(X_l^{-1}) = Y_l \prod_{i \neq l} (1 - X_i X_l^{-1}) \quad (6.12)$$

mit den Fehlerpositionen X_l und den Fehlerwerten Y_l verknüpft [WB94], die dann schließlich mit

$$Y_l = -X_l \frac{\Omega(X_l^{-1})}{\Lambda'(X_l^{-1})} \quad (6.13)$$

berechnet werden können [WB94]. $\Lambda'(x)$ ist die Ableitung von $\Lambda(x)$. Somit ist das Fehlerpolynom $e(x)$ komplett bestimmt. Abschließend können die berechneten Fehlerwerte von dem Empfangspolynom abgezogen werden um das wahrscheinlichste Codepolynom $\hat{c}(x)$ zu bestimmen.

6.4.2 Erasure-Decodierung von Reed-Solomon Codes

Der in Kap. 6.4.1 vorgestellte Algorithmus zur Decodierung von Reed-Solomon Codes ist nicht zur einer Verarbeitung von Zuverlässigkeitsinformation in der Lage. Falls allerdings die Position einiger fehlerhafter Symbole bekannt ist, kann der Algorithmus mit dieser Information initialisiert werden. Dadurch fallen einige Unbekannte aus dem Gleichungssystem (6.7) weg (so genannte Erasures). Dies hat zur Folge, dass eine größere Anzahl von Fehlern korrigiert werden kann. Ein $RS(n, k)$ -Code mit einer Minimaldistanz von $n - k + 1$ ist in der Lage, ein Codewort mit maximal $n - k$ bekannten Fehlerpositionen (Erasures) zu korrigieren. Der Code ist ebenfalls zur gleichzeitigen Korrektur von Fehlern und Erasures in der Lage: Jede Kombination aus ν unbekanntem Fehlern und ρ Erasures ist garantiert korrigierbar, falls $2\nu + \rho \leq n - k$.

Ein Reed-Solomon Code kann relativ einfach in einen Error/Erasure korrigierenden Code umgewandelt werden. Die Position der Erasures wird über das Erasure-Stellen-Polynom $\Gamma(x)$ festgelegt, wobei die j_i mit $i = 1, 2, \dots, \rho$ die Erasure-Positionen kennzeichnen:

$$\Gamma(x) = \prod_{i=1}^{\rho} (1 - x\alpha^{j_i}) \quad (6.14)$$

$\Gamma(x)$ wird dann mit dem Fehlerstellen-Polynom $\Lambda(x)$ zu dem Errata-Stellen-Polynom

$$\hat{\Lambda}(x) = \Gamma(x) \cdot \Lambda(x) \quad (6.15)$$

kombiniert. Wenn der Decodier-Algorithmus mit dem Errata-Stellen-Polynom $\hat{\Lambda}(x)$ initialisiert wird, kann eine Korrektur von Fehlern und Erasures erfolgen. Für die Durchführung des Berlekamp-Massey Algorithmus sind zudem wegen der bereits bekannten Fehlerpositionen zur noch $n - k - \rho$ Iterationen notwendig. Weitere Modifikationen am Decodieralgorithmus fallen nicht an.

6.4.3 Soft-Decodierung mit Hilfe eines inneren Parity-Check Codes

Wenn also die Position von Übertragungsfehlern im Datenstrom sicher erkannt werden kann, ist durch Erasure-Decodierung eine Korrektur der doppelten Fehlerzahl möglich. Umgekehrt kann die Korrekturfähigkeit eines Codes auch bei einer Halbierung der Redundanz aufrecht erhalten werden. Dieser Umstand kann also einerseits zu einer einfach implementierbaren, deutlichen Verbesserung der Korrekturfähigkeit eingesetzt werden oder andererseits zur Einsparung von Hardware – und somit von Verlustleistung – bei gleichzeitiger Aufrechterhaltung der Korrekturleistung.

Voraussetzung dafür ist eine möglichst sichere Erkennung von fehlerhaften Codesymbolen. Dazu soll die Zuverlässigkeitsinformation eines Soft-Output Detektors genutzt werden: Ein Codesymbol mit geringer Zuverlässigkeit besitzt eine höhere Fehlerwahrscheinlichkeit als ein Codesymbol mit hoher Zuverlässigkeit. Dabei ist jedoch zu beachten, dass sowohl korrekte, als auch fehlerbehaftete Codesymbole eine geringe Zuverlässigkeitsinformation aufweisen können. Insbesondere auf dem magnetischen Kanal, der ein sehr hohes Signal-Rausch-Verhältnis aufweist, sind die meisten Codesymbole mit geringer Zuverlässigkeit dennoch fehlerfrei. Aus diesem Grund ist eine alleinige Betrachtung der Zuverlässigkeitsinformation auf dem magnetischen Kanal nicht hinreichend, um Detektionsfehler zu erkennen.

Stattdessen muss die Position möglicher Fehler im Codewort eingegrenzt werden. Dies geschieht mit Hilfe eines Parity-Check Codes, der als innerer fehlererkennender Code (engl. Error Detecting Code, EDC) in den Datenstrom eingefügt wird [SSHS03]. Ein Codewort des inneren EDC besteht aus einer Gruppe von jeweils i hintereinander liegenden m -Bit Reed-Solomon Codesymbolen und j Parity Bits. Die Coderate des EDC beträgt somit $\frac{i \cdot m}{j + i \cdot m}$. Die Parity-Bits werden, wie in Abb. 6.1 zu sehen, in den Datenstrom eingefügt. Falls $i = 4$ und $j = 2$ (wie in der Abbildung), werden zwei Parity-Bits nach jeweils vier Codesymbolen des Reed-Solomon Codes eingefügt. Das erste wird dann bitweise über die ungeraden Bitpositionen der i Codesymbole gebildet, während das zweite Parity-Bit über die geraden Bitpositionen der i Codesymbole gebildet wird. Der fehlererkennende Code mit $i = 4$ und $j = 2$ ist zur sicheren Erkennung einzelner Fehlerevents

mit einer Länge von bis zu drei Bits in der Lage. Die beiden Parameter (i, j) müssen für eine möglichst zuverlässige Fehlererkennung an die Fehlermuster des Detektors bzw. des Modulationsdecoders angepasst werden.

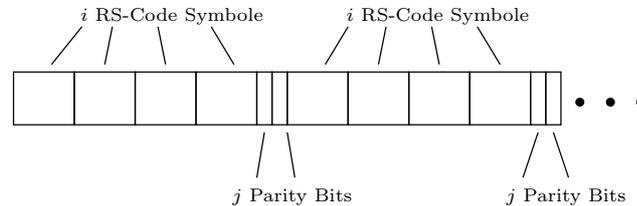


Abbildung 6.1: Einfügen eines inneren Parity-Check Codes in den Datenstrom: Nach jedem i . RS Codesymbol (hier: $i = 4$), werden j Parity-Check Bits (hier: $j = 2$) eingefügt. Falls eine der Parity-Check Bedingungen nicht erfüllt ist, wird das unzuverlässigste der $i = 4$ RS Codesymbole als Erasure markiert.

Der Ablauf der Decodierung ist in Abb. 6.2 zu sehen. Der innere EDC wird zuerst decodiert, um Detektionsfehler zu erkennen. Die Position der Fehler wird dann anhand der Soft-Outputs $L(c_i)$ des Detektors bestimmt. Der EDC berechnet daraus die Position der Erasures und reicht diese Information gemeinsam mit den m -Bit Codesymbolen r_k an den Reed-Solomon Decoder weiter. Dieser führt schließlich eine Error-/Erasure Korrektur durch.

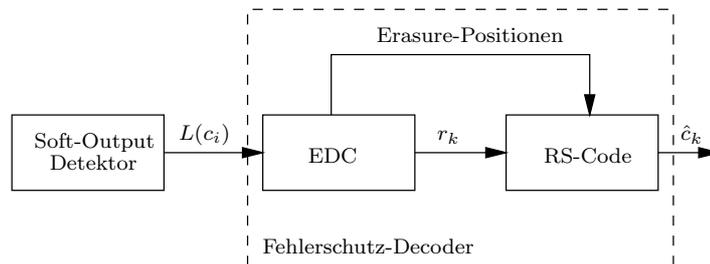


Abbildung 6.2: Ablauf der Decodierung des fehlererkennenden Codes (EDC) und des Error-/Erasure-korrigierenden Reed-Solomon Codes.

Der genaue Ablauf der Decodierung ist wie folgt [SSHS03]:

1. Die Zuverlässigkeitsinformation eines m -Bit RS-Codesymbols muss aus den Bit-zuverlässigkeiten der Detektorausgaben $L(c_i)$ berechnet werden. Sie wird auf die kleinste Bitzuverlässigkeit der m Bits innerhalb des Codesymbols gesetzt.
2. Daraufhin wird der innere EDC decodiert. Falls mindestens eine der j Parity-Check Gleichungen nicht erfüllt ist, wird mit den folgenden Schritten fortgefahren, ansonsten erfolgt direkt Schritt (6.).
3. Das unzuverlässigste der i RS-Codesymbole innerhalb eines EDC Codeworts wird als Erasure gekennzeichnet. Zu diesem Zweck wird zu dem Erasure-Locator-Polynom des äußeren Reed-Solomon Codes aus (6.14) eine Nullstelle an der Position α^{ji} hinzugefügt. Dabei muss beachtet werden, dass die maximal zulässige Anzahl ρ

von Erasures nicht überschritten wird. Des Weiteren ist es je nach Signal-Rausch-Verhältnis von Vorteil, nur eine kleinere Anzahl $\rho_{eff} < \rho$ zuzulassen, damit auch nicht erkannte Fehler später im Reed-Solomon Decoder korrigiert werden können.

4. Wie in Kap. 2.4.2 beschrieben, treten die Fehler auf dem magnetischen Kanal in Bündeln auf. Aus diesem Grund ist es unter Umständen sinnvoll, mehr als ein Erasure pro EDC Codewort zu setzen. Wenn sich die Zuverlässigkeitsinformationen der beiden unzuverlässigsten Symbole um weniger als t unterscheiden, dann werden beide Symbole als Erasure markiert.
5. Unter Umständen kann die Decodierung des EDC auch deshalb fehlschlagen, weil nur ein oder mehrere Parity-Bits fehlerhaft sind. In diesem Fall werden korrekte Codesymbole als Erasure gekennzeichnet. Die Anzahl solcher falsch gesetzter Erasures wird durch die Einführung eines weiteren Schwellwertes d minimiert: Wenn die geringste Zuverlässigkeit innerhalb des EDC Codewortes größer ist als d , wird kein Erasure gesetzt.
6. Abschließend erfolgt die Decodierung des Error-/Erasure-korrigierenden Reed-Solomon Codes.

Die Effizienz des in diesem Kapitel präsentierten Algorithmus hängt davon ab, wie akkurat der innere EDC arbeitet. Insgesamt müssen zur Beurteilung des Verfahrens drei Fälle unterschieden werden:

1. Es ist ein Fehler auf dem Kanal aufgetreten, der durch den EDC richtig erkannt und als Erasure markiert wurde. Zur Korrektur dieses Fehlers reicht ein redundantes Codesymbol aus. Dieses Ereignis trete ρ -mal auf.
2. Es ist ein Fehler auf dem Kanal aufgetreten, der jedoch nicht als Erasure markiert wurde. In diesem Fall werden wie bei einem herkömmlichen Reed-Solomon Code zwei redundante Codesymbole zur Korrektur benötigt. Dieses Ereignis trete ν -mal auf.
3. Ein korrektes Codesymbol wird vom EDC als Erasure markiert. Dieses Ereignis kann isoliert auftreten, wenn ausschließlich Parity-Bits fehlerhaft sind, oder in Kombination mit Fall (2.), wenn das falsche Codesymbol als Erasure gekennzeichnet wird. In diesem Fall wird ein redundantes Codesymbol benötigt, obwohl ein korrektes Codesymbol markiert wurde. Dieses Ereignis trete η -mal auf.

Wenn also in einem RS-Codewort $\nu + \rho$ Fehler aufgetreten sind, werden mit EDC $\rho + 2\nu + \eta$ redundante Symbole und ohne EDC $2\rho + 2\nu$ redundante Symbole zur Korrektur benötigt. Falls $\eta < \rho$, benötigt der Code mit EDC also weniger redundante Symbole als der Code ohne EDC. Insgesamt wird aber nur dann Redundanz eingespart, wenn unter Berücksichtigung der Coderate des EDC gilt, dass

$$(\rho + 2\nu + \eta) \cdot \frac{j + i \cdot m}{i \cdot m} < 2\rho + 2\nu. \quad (6.16)$$

Des Weiteren wurde bereits in Kap. 5.3 erläutert, wie der EDC als Modulationscode verwendet werden kann. In diesem Fall verschlechtert sich die effektive Coderate auf dem Kanal nicht und die Effizienz des Verfahrens hängt nur noch von $\eta < \rho$ ab.

6.5 Turbo Codes

Als erste Alternative zu Reed-Solomon Codes werden die 1993 von Berrou et al. präsentierten Turbo Codes [BGT93] betrachtet. Diese iterativen Codes haben seit ihrer Präsentation sehr viel Aufmerksamkeit erlangt, da sie eine dicht an der Kanalkapazität liegende Decodierleistung besitzen.

Im Gegensatz zu den in Kap. 6.4 vorgestellten Reed-Solomon Codes oder den LDPC Codes (Kap. 6.6) bestehen Turbo Codes aus einer Verkettung von zwei oder mehr Komponenten-Codes mit einem dazwischen liegenden Interleaver. Beim iterativen Decodierprozess der Codes wird zwischen den Komponenten-Codes Zuverlässigkeitsinformation ausgetauscht.

In den vergangenen Jahren waren die bei Turbo Codes verwendeten Konzepte Gegenstand intensiver Untersuchungen: Dabei wurde die Verwendung unterschiedlicher Komponenten-Codes oder Decodieralgorithmen betrachtet [RVH95, HOP96], die Form der Verkettung variiert [BM96a, BDMP98] oder der Decodieralgorithmus selbst untersucht [BM96a, PSC96, MMC98]. Auch im Bereich der magnetischen Datenspeicherung gab es intensive Untersuchungen mit Turbo Codes [MMH98, Rya98, RMM98, SFO⁺99, RMAAY00, AAM01]. Ein Problem von Turbo Codes ist allerdings, dass die Fehlerrate bei hohem SNR nur noch sehr flach abfällt (engl. Error Floor [PSC96]), so dass sie im direkten Vergleich nicht die Decodierleistung von Reed-Solomon Codes oder LDPC Codes erreichen [Che01, PS02].

Im Folgenden wird zunächst das hinter Turbo Codes stehende Prinzip erläutert. Dies umfasst die Konstruktion der Codes ebenso wie eine Beschreibung des iterativen Decodieralgorithmus. Danach erfolgt eine Beschreibung der verwendeten Encoder/Decoder am Beispiel der ursprünglichen Turbo Codes.

6.5.1 Das Turbo Prinzip

Turbo Codes erreichen ihre hervorragende Korrekturfähigkeit durch den gleichzeitigen Einsatz verschiedener bekannter Prinzipien der Codierungstheorie. In dieser Hinsicht besteht die Errungenschaft der Codes in der geschickten Kombination dieser Verfahren [BM96a]. Dies umfasst

- die Verwendung von zwei oder mehr Komponenten-Codes, die über einen Interleaver parallel miteinander verkettet sind,
- den Einsatz von rekursiven, systematischen Faltungscodes (engl. Recursive Systematic Convolutional (RSC) Codes) sowie

- die Verwendung eines iterativen Decodierprozesses mit Soft-Input/Soft-Output (SISO) MAP-Decodern und den Austausch von so genannter extrinsischer Information.

Der Aufbau des Turbo Encoders von Berrou et al. [BGT93] ist in Abb. 6.3 zu sehen. Die Eingangsdaten i_k werden parallel zu beiden RSC Encodern geführt, wobei sie vor dem zweiten Encoder zunächst einen Interleaver passieren. Wenn beide Encoder systematisch sind, lassen sich die Parity-Bits $p_k^{(1)}$ von Encoder 1 und $p_k^{(2)}$ von Encoder 2 getrennt von den Informationsbits i_k nach außen führen. Die Informationsbits des zweiten Encoders (die ja ohnehin nur eine verwürfelte Ausgabe des ersten Encoders darstellen) werden in diesem Fall verworfen. Ein Multiplexer erzeugt aus den Ausgaben beider Encoder dann einen codierten Datenstrom c_k mit der Coderate $\frac{1}{3}$. Im Falle der originalen Turbo Codes wird jedes zweite Parity-Bit beider Encoder punktiert (das heißt gestrichen) und so ein Code mit der Rate $\frac{1}{2}$ erzeugt.

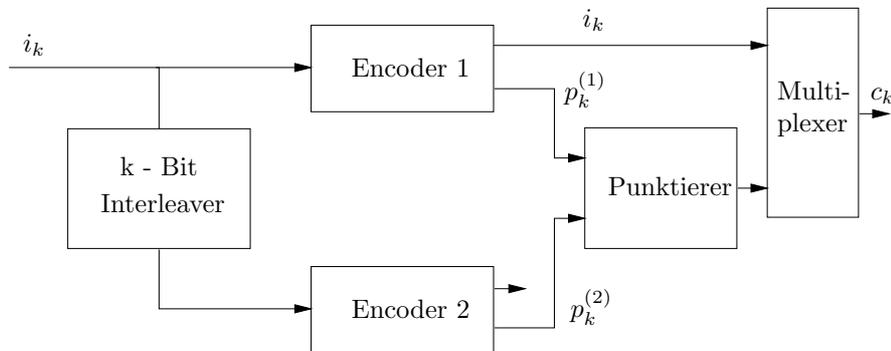


Abbildung 6.3: Aufbau eines Turbo Encoders nach Berrou et al.

Der dazugehörige Decoder ist in Abb. 6.4 zu sehen. Aus den empfangenen, verrauschten Symbolen werden Kanalzuverlässigkeiten $L_{ch}(r_k|c_k)$ generiert (vgl. (3.3)). Der aus Informationssymbolen und – gegebenenfalls punktierten – Parity-Symbolen bestehende Datenstrom muss im Decoder zunächst wieder in Informationssymbole $L_{ch}(r_k^{(i)}|c_k)$ (für beide Decoder identisch) und Parity-Symbole $L_{ch}^{(1)}(r_k^{(p)}|c_k)$ für Decoder 1 bzw. $L_{ch}^{(2)}(r_k^{(p)}|c_k)$ für Decoder 2 demultiplext werden. Die punktierten Parity-Symbole werden dabei auf den Wert Null gesetzt.

Die beiden Decoder werden so verkettet, dass der Ausgang des einen Decoders mit dem Eingang des anderen verbunden ist. Die Decodierung läuft iterativ ab, das heißt, sie wird so lange wiederholt, bis eine Abbruchbedingung erfüllt ist. Dabei muss der Interleaver auch im Decoder berücksichtigt werden: Bevor die Information vom ersten Decoder weiter gereicht wird, muss sie einen Interleaver passieren, während ein De-Interleaving auf der Feed-Back-Schleife vom zweiten zum ersten Decoder notwendig ist (Abb. 6.4).

Bei der Decodierung findet ein gegenseitiger Informationsaustausch statt. Dabei sind alle ausgetauschten Informationen Log-Likelihood Werte. Die Ein- und Ausgänge des Soft-In/Soft-Out Decoders 1 aus Abb. 6.4 sind in Abb. 6.5 zu sehen. Am Eingang liegt die Kanal-Zuverlässigkeitsinformation $L_{ch}(r_k|c_k)$ eines Soft-Output Modulationsdeco-

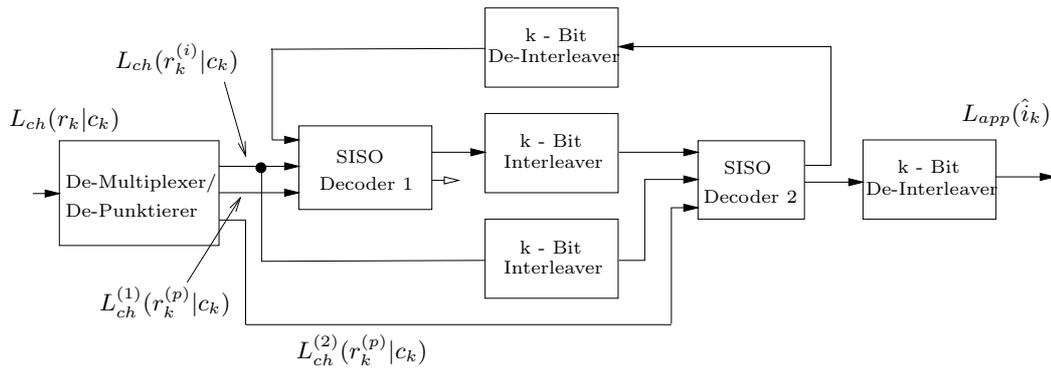


Abbildung 6.4: Aufbau eines Turbo Decoders nach Berrou et al.

ders oder Detektors für alle Codesymbole c_k an. Bei Verwendung von systematischen Codes wird der Datenstrom wieder in Informationssymbole $L_{ch}^{(1)}(r_k^{(i)}|c_k)$ und Parity-Symbole $L_{ch}^{(1)}(r_k^{(p)}|c_k)$ aufgeteilt. Sofern eine weitere a priori Information bekannt ist, zum Beispiel wenn für das Informationssymbol i_k eine Eins wahrscheinlicher als eine Null ist, kann dies dem Decoder in Form der a priori Zuverlässigkeit $L_a^{(1)}(i_k)$ für alle Informationssymbole i_k mitgeteilt werden.

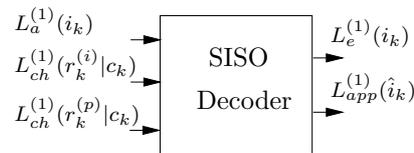


Abbildung 6.5: Ein- und Ausgänge eines Soft-In/Soft-Out (SISO) Decoders

Am Ausgang des Decoders wird die a posteriori Zuverlässigkeitsinformation $L_{app}^{(1)}(\hat{i}_k)$ für alle Informationssymbole i_k bestimmt, deren Vorzeichen die binäre Detektorausgabe \hat{i}_k festlegt. Gleichzeitig wird die so genannte extrinsische Information $L_e^{(1)}(i_k)$ berechnet. Diese wird ebenfalls für alle Informationssymbole i_k bestimmt. Sie ist Bestandteil von $L_{app}^{(1)}(\hat{i}_k)$, allerdings wird die Kanalzuverlässigkeit $L_{ch}^{(1)}(r_k|c_k)$ und die a priori Zuverlässigkeit $L_a^{(1)}(i_k)$ herausgerechnet. Somit drückt die extrinsische Information $L_e^{(1)}(i_k)$ eines Codebits i_k nur den Teil der Zuverlässigkeitsinformation aus, der über die Zusammenhänge des Codes und somit über alle anderen Codebits bestimmt wird.

Am Beispiel eines Codesymbols c_i eines einfachen Parity-Check Codes $\underline{c} = \{c_0, \dots, c_{n-1}\}$ lässt sich der Unterschied zwischen Kanalinformation und extrinsischer Information verdeutlichen: Die Kanalinformation entspricht dem empfangenen Wert c_i . Gleichzeitig lässt sich aber der Wert des Codesymbols c_i auch indirekt über alle anderen Codesymbole $\{c_j | j = 0, \dots, n-1; j \neq i\}$ und die Parity-Check Bedingung bestimmen (zum Beispiel: Es ist eine gerade Anzahl von Einsen pro Codewort gefordert).

Nur die extrinsische Information wird während der Decodierung zwischen den Decodern der Komponenten-Codes ausgetauscht. Dabei dient die extrinsische Information

des einen Decoders in der folgenden Iteration als a priori Information für den anderen Decoder.

Im Gegensatz zu den in Kap. 6.6 beschriebenen LDPC Codes erfolgt bei Turbo Codes keine Überprüfung auf ein gültiges Codewort. Statt dessen wird entweder eine vorgegebene Iterationszahl durchgeführt oder die Veränderung der extrinsischen Information überwacht [Moh93, WMK⁺00]. Wenn das Abbruchkriterium erreicht ist, wird schließlich $L_{app}(\hat{i}_k)$ als Decodierergebnis ausgegeben.

Das Turbo Prinzip ist nicht auf die hier beschriebene Variante von Berrou et al. [BGT93] beschränkt. Es sind Anordnungen mit mehr als zwei Komponenten-Codes ebenso denkbar, wie beispielsweise eine serielle Verkettung [BDMP98] oder der Einsatz unterschiedlicher Komponenten-Codes [HOP96].

6.5.2 Beschreibung der Encoder und Decoder

Die ursprünglichen Turbo Codes verwenden rekursive und systematische Faltungscodes als Komponenten-Codes. Diese lassen sich über ein digitales Infinite Impulse Response (IIR) Filter beschreiben. Die Verwendung rekursiver Codes ist für die Konvergenz des iterativen Decoders von Vorteil [tB01]. Im Gegensatz zu den bisher beschriebenen Block-Codes besitzen Faltungscodes keine Codewörter fester Länge. Stattdessen wird ein unendlicher Datenstrom durch den Encoder geleitet. Bei Turbo Codes ist jedoch für den iterativen Decodiervorgang eine feste Blocklänge erforderlich. Daher werden die Turbo Encoder nach der Blocklänge N durch das Einfügen weiterer Bits wieder in den Ausgangszustand gebracht (terminiert). Durch diese Maßnahme verhalten sich die Codes wie Block-Codes.

Encoder

In Abb. 6.6 ist der Encoder des von Berrou et al. [BGT93] verwendeten Faltungscodes zu sehen. Dieser ist aus binären Shift-Registern und binären modulo-2 Addierern aufgebaut. Die Anzahl der Shift-Register bestimmt die Gedächtnislänge des Codes. Die Gedächtnislänge des Codes bestimmt dessen Decodierfähigkeit, allerdings auch maßgeblich die Komplexität des Decoders. Der hier gezeigte Encoder hat eine Gedächtnislänge von $m = 4$.

Die Vorschrift, nach der ein Ausgang j aus dem Eingang i bzw. den Shift-Register-Ausgängen erzeugt wird, lässt sich für nicht rekursive Faltungscodes über ein Generator-Polynom vom Grad m beschreiben. Dieses wird im Frequenzbereich mit Hilfe des Delay-Operators D in der Form

$$G_{ij}(D) = \sum_{l=0}^m g_l \cdot D^l \quad (6.17)$$

angegeben. Der Encoder aus Abb. 6.6 erzeugt einen systematischen Code, da der Eingang 1 direkt und ohne einen Addierer oder ein Shift-Register zu passieren auf Ausgang 1

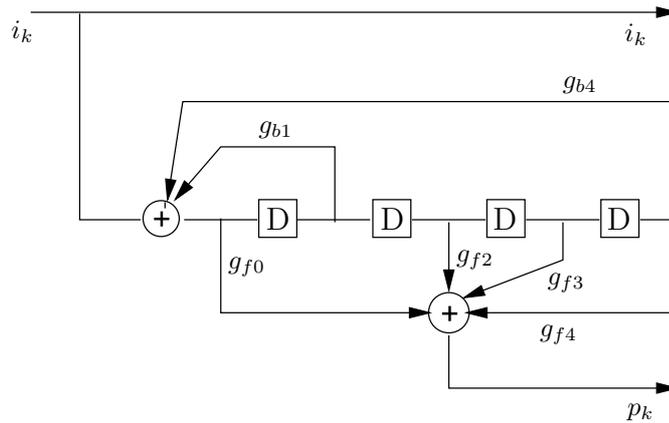


Abbildung 6.6: Encoder eines rekursiven und systematischen Faltungscodes

geführt wird. Das dazu gehörige Generatorpolynom G_{11} ergibt sich somit einfach zu $G_{11} = 1$.

Charakteristisch für die rekursiven Faltungscodes ist die Rückkopplungsschleife, die den Ausgang der Shift-Register wieder auf den Eingang zurückführt. Rekursive Faltungscodes besitzen ein Generator-Polynom, das sich aus einem Forward-Polynom $G_F(D)$, das die Verknüpfung in Vorwärtsrichtung beschreibt und einem Backward-Polynom $G_B(D)$, das die Struktur der Feed-Back Schleife beschreibt, zu

$$G_{ij}(D) = \frac{G_B(D)}{G_F(D)} \quad (6.18)$$

zusammensetzt. Für den Encoder aus Abb. 6.6 ergibt sich das Generatorpolynom G_{12} zu

$$G_{12}(D) = \frac{1 + D + D^4}{1 + D^2 + D^3 + D^4}. \quad (6.19)$$

Die Polynome werden häufig auch im Zeitbereich angegeben. Für (6.19) besitzt das Backward-Polynom dann die Koeffizienten $g_b = \{1, 1, 0, 0, 1\}$ und das Forward-Polynom die Koeffizienten $g_f = \{1, 0, 1, 1, 1\}$. Insbesondere bei großen Gedächtnislängen ist diese Schreibweise jedoch unübersichtlich, so dass häufig jeweils drei binäre Koeffizienten zu einer Oktalzahl zusammen gefasst werden. Es ergibt sich dann eine für den betrachteten Turbo Encoder übersichtliche Darstellung nach dem Muster $(g_b, g_f) = (31, 27)$. Im weiteren Verlauf dieser Arbeit wird auf die Schreibweise in Oktalform zurückgegriffen.

Decoder

Aus dem Inhalt der binären Shift-Register des Turbo Encoders kann ähnlich wie beim Partial Response Kanal (Kap. 4.2.1) ein Zustand definiert werden. Bei einer Gedächtnislänge von m sind 2^m Zustände möglich. Daraus kann ein Trellis aufgespannt werden,

der die Zustandsänderungen des Encoders in Abhängigkeit von den Eingangsdaten beschreibt. Aus diesem Grund kann zur Decodierung von Turbo Codes jeder Algorithmus verwendet werden, der sich an einem Trellis decodieren lässt, zum Beispiel der Soft-Output Viterbi-Algorithmus [HH89] (Kap. 4.2.2).

Die beste Decodierleistung erreichen Turbo Codes jedoch mit dem BCJR-Algorithmus [BCJR74], der eine symbolweise Maximum A Posteriori (MAP) Decodierung durchführt und die a posteriori Zuverlässigkeit $L_{app}(\hat{i}_k)$ nach der Vorschrift

$$L_{app}(\hat{i}_k) = \ln \frac{P(i_k = +1 | \underline{r})}{P(i_k = -1 | \underline{r})} \quad (6.20)$$

berechnet. Der Algorithmus arbeitet an einem Trellis fester Länge und erfordert daher eine Terminierung der Codes. In Abb. 6.7 ist ein Trellis zu einem Faltungs-Encoder mit einer Gedächtnislänge von $m = 2$ und somit $2^m = 4$ Zuständen zu sehen.

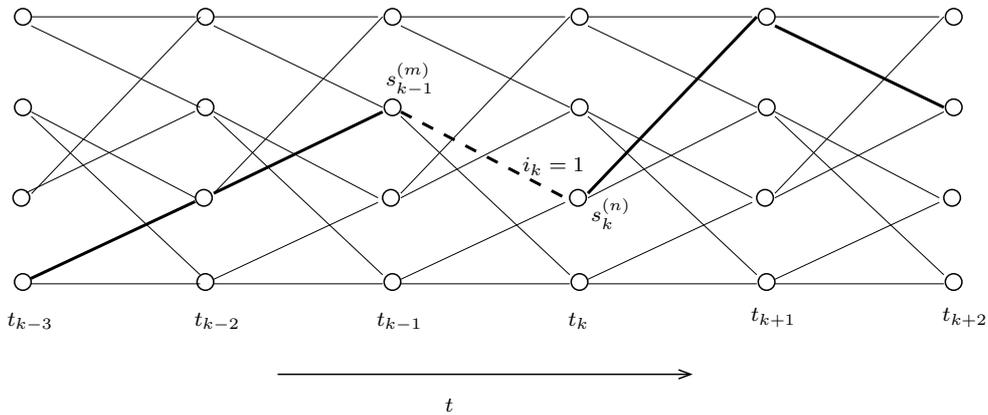


Abbildung 6.7: Berechnung der symbolweisen Zuverlässigkeitsinformation mit Hilfe des BCJR-Algorithmus

Jeder Weg durch den Trellis entspricht einem gültigen Codewort, wobei ein Codebit c_i einem Zustandsübergang im Trellis von Zustand $s_{k-1}^{(m)}$ in den Zustand $s_k^{(n)}$ entspricht. Zum Zeitpunkt t_{k-1} existieren insgesamt $2 \cdot 2^m$ Wege in einen anderen Zustand zum Zeitpunkt t_k , von denen aufgrund der binären Struktur des Codes 2^m Übergänge einem $i_k = +1$ und 2^m Übergänge einem $i_k = -1$ entsprechen. Damit lässt sich (6.20) über die Wahrscheinlichkeiten $P(s_{k-1}, s_k | \underline{r})$ eines Übergangs von Zustand s_{k-1} in Zustand s_k ausdrücken:

$$L_{app}(\hat{i}_k) = \ln \frac{\sum_{(s_{k-1}, s_k), i_k = +1} P(s_{k-1}, s_k | \underline{r})}{\sum_{(s_{k-1}, s_k), i_k = -1} P(s_{k-1}, s_k | \underline{r})} \quad (6.21)$$

Bahl et al. haben gezeigt, dass sich die Wahrscheinlichkeiten aus (6.21) in drei voneinander unabhängige Komponenten unterteilen lassen [BCJR74]:

- Die Vorwärtsrekursion $\alpha_{k-1}(s_{k-1})$ gibt die Wahrscheinlichkeit für den Zustand s_{k-1} zum Zeitpunkt t_{k-1} an. Zur Berechnung der $\alpha_{k-1}(s_{k-1})$ ist die Durchführung eines Viterbi-Algorithmus vom Ausgangszustand s_0 bis zum Zustand s_{k-1} notwendig.
- Die Pfadübergangswahrscheinlichkeit $\gamma_k(s_{k-1}, s_k)$ gibt die Wahrscheinlichkeit an, dass zum Zeitpunkt t_k vom Zustand s_{k-1} in den Zustand s_k verzweigt wird. $\gamma_k(s_{k-1}, s_k)$ kann analog zu (4.16) über die Kanalcharakteristik bestimmt werden.
- Die Rückwärtsrekursion $\beta_k(s_k)$ gibt die Wahrscheinlichkeit an, dass zum Zeitpunkt t_k der Zustand s_k angenommen wird. Zur Berechnung der $\beta_k(s_k)$ ist wiederum die Durchführung eines Viterbi-Algorithmus notwendig. Dieser startet jedoch im Endzustand s_N und läuft dann in entgegengesetzter Richtung bis in den Zustand s_k .

Die Zuverlässigkeitsinformation $L_{app}(\hat{i}_k)$ ergibt sich schließlich wie folgt:

$$L_{app}(\hat{i}_k) = \ln \frac{\sum_{(s_{k-1}, s_k), i_k=+1} \alpha_{k-1}(s_{k-1}) \cdot \gamma_k(s_{k-1}, s_k) \cdot \beta_k(s_k)}{\sum_{(s_{k-1}, s_k), i_k=-1} \alpha_{k-1}(s_{k-1}) \cdot \gamma_k(s_{k-1}, s_k) \cdot \beta_k(s_k)} \quad (6.22)$$

Für den MAP Decoder ist also die Durchführung zweier Viterbi-Algorithmen notwendig, die einen terminierten Trellis in entgegengesetzter Richtung durchlaufen. Zur Berechnung der Vorwärtsrekursion $\alpha_{i-1}(s_{i-1})$ und der Rückwärtsrekursion $\beta_i(s_i)$ müssen allerdings die Wahrscheinlichkeiten gemäß (4.16) berechnet werden – eine einfache Bestimmung der Pfadmetriken wie beim Viterbi-Algorithmus (Kap. 4.2.2) ist nicht ausreichend.

Aus diesem Grund sind neben dem BCJR-Algorithmus auch andere Decoder untersucht worden [RVH95]. Viel versprechende Ergebnisse wurden bei hohem SNR mit dem Max-Log-MAP Algorithmus erzielt, bei dem für die Berechnung der Zuverlässigkeitsinformation in (6.21) jeweils nur das Maximum aus beiden Summen verwendet wird.

6.5.3 Eigenschaften der iterativen Decodierung

Die außergewöhnlich gute Decodierleistung von Turbo Codes hat eine Vielzahl von Untersuchungen motiviert, die zum Verständnis des iterativen Decodiervorgangs beigetragen haben. Aus den Ergebnissen ergeben sich auch Rückschlüsse für die Anwendbarkeit von Turbo Codes auf dem magnetischen Kanal, so dass hier eine Übersicht erfolgt.

Wenn die erzielbare Bitfehlerrate (BER) eines Turbo Codes über dem Signal-Rausch-Verhältnis (SNR) aufgetragen wird, ergibt sich ein für iterative Decodierverfahren sehr charakteristischer Verlauf: Bei niedrigem SNR verbleiben die Kurven zunächst auf einem relativ konstanten Niveau, um dann sehr steil abzufallen. Bei höherem SNR flachen die Kurven allerdings wieder ab. Dieser Verlauf ist auch bei den Untersuchungen in dieser Arbeit in Kap. 7.5.1 zu beobachten.

Perez et al. [PSC96] haben den steilen Abfall der Kurve als Wasserfall-Region (engl. waterfall region) und die Abflachung als Error Floor bezeichnet. Sie erklären den ungewöhnlichen Verlauf mit den Distanzeigenschaften der Codes: Turbo Codes besitzen eine relativ geringe Minimaldistanz, die zu dem schlechten asymptotischen Verlauf bei hohem SNR führt. In dieser Hinsicht sind sie somit nach dem klassischen Verständnis der Codierungstheorie schlechte Codes. Umgekehrt hängt die hervorragende Leistung im Bereich der Wasserfall-Region nicht von der Minimaldistanz, sondern vom Distanzspektrum ab. Die Anzahl von Codewörtern mit einer geringen Hammingdistanz – also nicht unbedingt der Minimaldistanz – ist im Vergleich zu anderen Codes deutlich kleiner. In dieser Hinsicht wiederum sind Turbo Codes also hervorragende Codes.

Diese Eigenschaft erklärt sich aus der Struktur der Encoder [PSC96]: Die Verwendung von rekursiven Faltungscodes führt dazu, dass auch Informationswörter mit einem geringen Hamminggewicht zu Codewörtern mit einem hohen Hamminggewicht führen. Selbst wenn einer der Encoder ein Codewort mit einem geringen Hamminggewicht erzeugt, sorgt der Interleaver dafür, dass der zweite Encoder mit hoher Wahrscheinlichkeit ein Codewort mit einem großen Hamminggewicht erzeugt. Die geringe Minimaldistanz der Turbo Codes ist auf einige Sequenzen mit dem Hamming-Gewicht zwei zurückzuführen, aus denen Codewörter mit sehr kleinem Hamminggewicht erzeugt werden.

6.5.4 Turbo Codes auf dem magnetischen Kanal

Die hohe Decodierleistung von Turbo Codes bei moderatem SNR hat auch im Bereich der magnetischen Aufzeichnung Untersuchungen angestoßen. Ein Einsatz auf dem magnetischen Kanal erfordert jedoch einige Modifikationen an den Codes: Die hohen Datendichten und -übertragungsraten (siehe Kap. 3.1) sind nur unter Verwendung von fehlerkorrigierenden Codes mit hohen Coderaten erreichbar.

Aus diesem Grund müssen die Codes punktiert werden. Dies beeinflusst jedoch die Distanzeigenschaften des Codes, so dass sich sowohl die Decodierleistung in der Wasserfall-Region als auch die ohnehin nicht gute Minimaldistanz verschlechtern. Dieser Umstand ist insbesondere deshalb kritisch, weil auf dem magnetischen Kanal sehr niedrige Fehlerraten erzielt werden müssen (Kap. 3.1) und der Error-Floor sich hier viel stärker auswirkt, als zum Beispiel auf einem Funkkanal.

Dennoch wurden Turbo Codes in verschiedenen Varianten für den magnetischen Kanal vorgeschlagen. Als erstes wurde die klassische parallele Verkettung von Faltungscodes für den magnetischen Kanal untersucht [Rya98]. Die Codes wurden dazu auf unterschiedliche Coderaten von $\frac{4}{5}$ bis $\frac{64}{65}$ punktiert. Der erzielte Codiergewinn bei einer Bitfehlerrate von 10^{-6} lag zwischen 2 und 5 dB. Allerdings ist in allen Fällen ein mehr oder weniger ausgeprägter Error Floor beobachtet worden. Dieser verhindert, dass die bei Festplatten erforderliche Bitfehlerrate von mindestens 10^{-13} erreicht wird.

Bei einigen Untersuchungen wurde ein Kanalmodell verwendet, bei dem ein Soft-Output Detektor in die Iterationsschleifen einbezogen wird, indem die Zuverlässigkeitsinformation vom Ausgang des fehlerkorrigierenden Codes zum Detektor zurückgekoppelt wird (so genannte Turbo Equalization [DJB95]). Danach kann der Detektionsalgorithmus mit dieser zusätzlichen a priori Information wiederholt werden. Wie bereits in Kap. 6.1 erwähnt,

kann der magnetische Kanal als ein Code mit der Rate Eins angesehen werden, der zum Beispiel mit dem BCJR-Algorithmus [BCJR74] an einem Trellis decodiert wird. Die Einbeziehung des Soft-Output Detektors in die Iterationsschleifen führt zu einer Verbesserung von rund 0,5 dB [RMM98].

Eine weiterer entscheidender Aspekt für niedrige Fehlerraten ist ein zusätzlicher Kanal-Interleaver zwischen Detektor und Turbo Code, der dafür sorgt, dass die in Bündeln auftretenden Fehler des magnetischen Kanals verteilt werden [RMAAY00]. Somit treten sie aus Sicht des Decoders in Form von zufällig verteilten Einzelbitfehlern auf. Durch diese Maßnahme kann die Decodierleistung des Turbo Codes deutlich verbessert werden [RMAAY00].

Im Falle einer seriellen Verkettung kann Komplexität eingespart werden, wenn statt eines Turbo Codes nur ein rekursiver Faltungscode eingesetzt wird, der in Verbindung mit Turbo Equalization iterativ decodiert wird, allerdings führt dies gleichzeitig zu einer Verschlechterung der Decodierleistung [SFO⁺99]. Da der Error-Floor durch Codewörter mit geringem Hamminggewicht erzeugt wird, kann die Decodierleistung durch eine geeignete Vor-Codierung, wie zum Beispiel NRZI, erhöht werden, die kritische Sequenzen ausfiltert, ohne dass die Komplexität erhöht wird [MMN99].

6.6 Low-Density Parity-Check (LDPC) Codes

Als weitere Alternative zu Reed-Solomon Codes werden Low-Density Parity-Check (LDPC) Codes betrachtet. LDPC Codes gehören ebenfalls zur Klasse der linearen Block-Codes. Sie wurden bereits im Jahre 1962 von Gallager präsentiert [Gal62]. Die damalige geringe zur Verfügung stehende Rechenleistung hat jedoch dazu geführt, dass die Codes als zu komplex erachtet und dadurch jahrelang kaum beachtet wurden. Erst in den 1990er Jahren nach der Präsentation von Turbo Codes [BGT93] haben die Arbeiten von MacKay [Mac99] bzw. Richardson und Urbanke [RU98] gezeigt, dass LDPC Codes bei großen Codelängen (bis 10.000.000 Bit) und niedrigen Coderaten ($\frac{1}{3}$ bis $\frac{1}{2}$) eine außerordentlich gute Decodierleistung besitzen. Dies hat weitere Bemühungen motiviert, LDPC Codes auch für kürzere Codes und höhere Coderaten anzupassen [MD99], wodurch sie auch für den Einsatz auf dem magnetischen Kanal interessant wurden.

6.6.1 Erzeugung und Decodierung von LDPC Codes

Die Erzeugung und Decodierung eines LDPC Codes erfolgt mit Hilfe der Generatormatrix \mathcal{G} bzw. der Parity-Check Matrix \mathcal{H} . Die Parity-Check Matrix ist schwach mit Elementen aus einem Galoisfeld $GF(q)$ besetzt ($\approx 5\%$ der Elemente ungleich Null). Während die ursprünglich von Gallager eingeführten Codes binär waren (also $GF(2)$), haben MacKay et al. die Erweiterung auf $GF(q)$ eingeführt [MD98]. Die Generatormatrix wird aus der Parity-Check Matrix bestimmt. Dazu wird sie durch Gauß-Elimination zuerst in die Form

$$\mathcal{H} = [P^T | I] \quad (6.23)$$

mit der Einheitsmatrix I gebracht. Daraus kann dann die Generatormatrix in der Form

$$\mathcal{G} = [I|P] \quad (6.24)$$

direkt angegeben werden. Im Encoder wird ein Informationswort $\underline{i} = \{i_0, i_1, \dots, i_{k-1}\}$, bestehend aus k Informationssymbolen, durch die Multiplikation mit der $k \times n$ Matrix \mathcal{G} auf ein Codewort $\underline{c} = \{c_0, c_1, \dots, c_{n-1}\}$ mit

$$\underline{c} = \underline{i} \cdot \mathcal{G} \quad (6.25)$$

aus n Codesymbolen abgebildet. Auf der Decoderseite wird zunächst mit Hilfe von \mathcal{H} überprüft, ob die empfangene Sequenz $\underline{r} = \{r_0, r_1, \dots, r_{n-1}\}$ ein gültiges Codewort ist. Zu diesem Zweck wird durch Multiplikation von \underline{r} mit der $m \times n$ Matrix \mathcal{H} der Syndromvektor $\underline{z} = \{z_0, z_1, \dots, z_{m-1}\}$ mit

$$\underline{r} \cdot \mathcal{H}^T = \underline{z} \quad (6.26)$$

gebildet. Jede der m Zeilen von \mathcal{H} entspricht einer Parity-Check Gleichung. Falls \underline{r} einem gültigen Codewort entspricht, ist der Syndromvektor Null. Ist $\underline{z} \neq 0$, liegt in jedem Fall ein Übertragungsfehler vor. Der Decoder bestimmt dann das Codewort $\hat{\underline{c}} = \{\hat{c}_0, \dots, \hat{c}_{n-1} | \hat{c}_l \in \{0, 1\}\}$ mit der geringsten Hammingdistanz zu \underline{r} , das die Bedingung $\hat{\underline{c}} \cdot \mathcal{H}^T = 0$ erfüllt. Es wird also nur überprüft, ob die empfangene Sequenz einem Codewort entspricht. Somit ist bei einem LDPC Code eine Falschkorrektur möglich.

Zur Decodierung wird der so genannte Belief Propagation (BP) Algorithmus verwendet [Gal62]. Dieser bestimmt mit Hilfe eines iterativen Verfahrens das am wahrscheinlichsten übertragene Codewort $\hat{\underline{c}}$. Die Arbeitsweise des BP Algorithmus lässt sich am besten mit Hilfe des so genannten Tanner Graphen [Tan81] erläutern. Dieser zweiteilige Graph ist eine grafische Repräsentation der Parity Check Matrix \mathcal{H} . In Abb. 6.8 ist eine Parity-Check Matrix der Dimension 3×6 zu sehen. Die Matrix gehört zu einem LDPC Code der Coderate $\frac{1}{2}$ mit einer Codelänge $n = 6$, $k = 3$ Informationssymbolen und $m = 3$ Parity-Checks. Die Matrix hat ein Zeilengewicht W_z von 4, das heißt, jede Parity-Check Gleichung wird über 4 Bits gebildet. Das Spaltengewicht W_s der Matrix ist 2. Somit wird jedes Codesymbol von 2 Parity-Checks abgesichert. Eine Matrix mit einem konstanten Zeilen- und Spaltengewicht erzeugt einen regulären LDPC Code. Der zu dieser Matrix gehörende Tanner-Graph ist in Abb. 6.9 zu sehen.

$$\begin{array}{|c|c|c|c|c|c|} \hline 1 & 0 & 1 & 0 & 1 & 1 \\ \hline 0 & 1 & 1 & 1 & 0 & 1 \\ \hline 1 & 1 & 0 & 1 & 1 & 0 \\ \hline \end{array}$$

Abbildung 6.8: Beispiel einer 3×6 Parity-Check Matrix \mathcal{H} , die einen Code mit der Länge $n = 6$ und der Rate $\frac{1}{2}$ erzeugt.

Der Tanner-Graph besitzt auf der einen Seite n Bit-Knoten und auf der anderen Seite m Check-Knoten. Die Bit-Knoten entsprechen den n Symbolen eines LDPC Codewortes

und die Check-Knoten den m Parity-Check Gleichungen. Über die Kanten des Tanner-Graphen wird die Parity-Check Matrix \mathcal{H} gemäß Gleichung (6.26) wie folgt nachgebildet: Der Bit-Knoten c_i ist mit dem Check-Knoten z_j genau dann über eine Kante verbunden, wenn das Matrix-Element \mathcal{H}_{ij} Eins ist (Abb. 6.9).

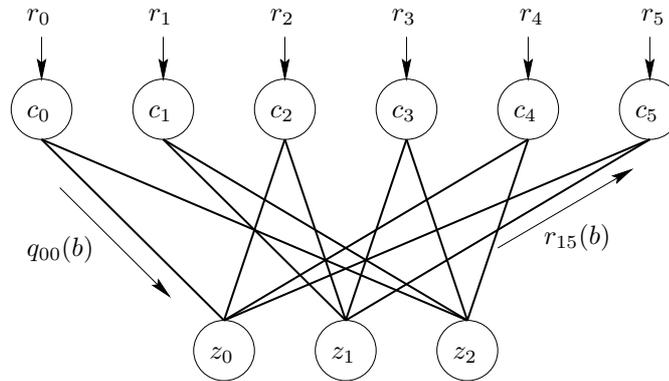


Abbildung 6.9: Darstellung der Parity-Check Matrix aus Abbildung 6.8 als zweiteiliger Tanner-Graph. Ein Bitknoten c_i ist mit einem Check-Knoten z_j genau dann verbunden, wenn das Matrixelement H_{ij} Eins ist. Während der Decodierung werden über diese Verbindungen Nachrichten in beiden Richtungen ausgetauscht, zum Beispiel $q_{00}(b)$ oder $r_{15}(b)$.

Bei der Decodierung wird mit Hilfe der Parity-Check Matrix das wahrscheinlichste Codewort \hat{c} bestimmt. Dazu werden sowohl die empfangene Sequenz \underline{r} , als auch die durch die Parity-Check Gleichungen vorgegebenen Zusammenhänge – und somit die Kanten im Tanner-Graphen – berücksichtigt. Gesucht ist die a posteriori Wahrscheinlichkeit $P(c_i = 1 | \underline{r}, S_i)$ bzw. $P(c_i = 0 | \underline{r}, S_i)$ für ein Codebit $c_i = 1$ bzw. $c_i = 0$ unter den Bedingungen, dass die Sequenz \underline{r} empfangen wurde und das Ereignis S_i eintritt. S_i tritt genau dann ein, wenn alle mit c_i verknüpften Parity-Check Gleichungen erfüllt sind.

Im Gegensatz zu einer wirklichen Maximum A Posteriori (MAP) Decodierung (Kap. 6.2) nähert der Belief-Propagation Algorithmus die gesuchten a posteriori Wahrscheinlichkeiten nur an. Dabei werden in einem iterativen Verfahren Nachrichten zwischen den Bitknoten und den Check-Knoten ausgetauscht. Die Nachricht $q_{ij}(b)$, die von Bitknoten c_i zu Check-Knoten z_j geschickt wird, kennzeichnet den Glauben (engl. Belief) von Bitknoten c_i , dass das dazu gehörige Bit c_i den Wert b hat – mit $b \in \{0, 1\}$ für binäre LDPC Codes (Abb. 6.9). Umgekehrt kennzeichnet die Nachricht $r_{ji}(b)$, die von Check-Knoten z_j zu Bitknoten c_i geschickt wird, den Glauben von Check-Knoten z_j , dass die Parity-Check Gleichung j erfüllt ist, wenn das Bit c_i den Wert b besitzt (Abb. 6.9). Im Folgenden wird der Ablauf des iterativen Algorithmus erläutert. Eine Iteration besteht aus einer Berechnung der r -Nachrichten (horizontaler Durchlauf) und einer anschließenden Berechnung der q -Nachrichten (vertikaler Durchlauf). Mit Hilfe einer Abbruchbedingung wird daraufhin überprüft, ob der Algorithmus bereits ein gültiges Codewort gefunden hat. Falls nicht und falls eine maximale Iterationszahl noch nicht erreicht wurde, startet eine weitere Decoder-Iteration.

Initialisierung

Der Algorithmus startet mit einer Initialisierung der q -Nachrichten. Dies kann zum Beispiel mittels einer Abschätzung der zu erwartenden Fehlerwahrscheinlichkeit auf dem Kanal erfolgen [MD99]. In diesem Fall werden alle Bitknoten mit dem gleichen Startwert initialisiert. Deutlich bessere Ergebnisse lassen sich jedoch erzielen, wenn jedem Bitknoten eine individuelle Fehlerwahrscheinlichkeit zugewiesen wird [PS02]. Zu diesem Zweck kann auf dem magnetischen Kanal zum Beispiel die Zuverlässigkeitsinformation eines Soft-Output Detektors (zum Beispiel aus Kap. 4.2.2 oder 4.5) verwendet werden.

Der horizontale Durchlauf

Beim horizontalen Durchlauf werden die r -Nachrichten für alle m Parity-Check Gleichungen, das heißt für alle m Zeilen der Parity-Check Matrix \mathcal{H} , getrennt berechnet. In Abb. 6.10 ist die j -te Zeile einer $m \times n$ Parity-Check Matrix als Ausschnitt des dazugehörigen Tanner-Graphen zu sehen. Die Nachricht $r_{ji}(b)$, die Check-Knoten z_j zu Bitknoten c_i schickt, wird aus den q -Nachrichten aller Bitknoten bestimmt, die mit Check-Knoten z_j verbunden sind – mit Ausnahme von Knoten c_i selbst. Durch das Ausklammern der Nachricht q_{ij} wird nur die Information an c_i geschickt, die der Knoten selbst nicht besitzt (so genannte extrinsische Information).

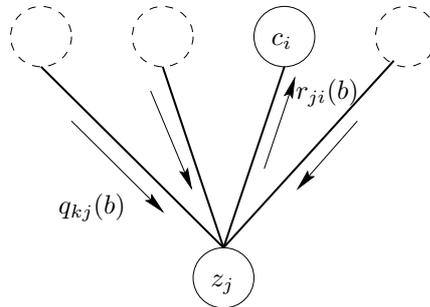


Abbildung 6.10: Ausschnitt aus einem Tanner-Graphen: Berechnung der $r_{ji}(b)$.

Die Menge der mit Check-Knoten z_j verbundenen Bitknoten c_k sei V_j . Wie bereits erwähnt, kennzeichnet die Nachricht $r_{ji}(b)$ den Glauben, dass die Check-Gleichung z_j erfüllt ist, falls $c_i = b$. Dies ist für binäre Codes und Even Parity genau dann erfüllt, wenn

- $c_i = 0$ und unter allen $c_k \in V_j \setminus i$ eine *gerade* Anzahl von Einsen ist oder wenn
- $c_i = 1$ und unter allen $c_k \in V_j \setminus i$ eine *ungerade* Anzahl von Einsen ist.

Formal ergibt sich daraus laut Gallager für binäre Codes [Gal62]

$$r_{ji}(0) = \frac{1}{2} + \frac{1}{2} \prod_{k \in V_j \setminus i} (1 - 2q_{kj}(1))$$

$$r_{ji}(1) = 1 - r_{ji}(0).$$
(6.27)

Dieser Schritt muss für jeden Check-Knoten z_j , für alle Bitknoten aus V_j und für alle möglichen Werte b von c_i durchgeführt werden. Im Falle von regulären LDPC Codes über $GF(q)$ ist also die Berechnung von $(m \times W_z \times q)$ r -Nachrichten pro Halbiteration erforderlich.

Der vertikale Durchlauf

Danach startet mit dem vertikalen Durchlauf die zweite Halbiteration. Dabei werden die q -Nachrichten für alle n Codesymbole getrennt berechnet. In Abb. 6.11 ist die zum Codesymbol c_i gehörende i -te Spalte einer $m \times n$ Parity-Check Matrix \mathcal{H} als Ausschnitt des dazu gehörigen Tanner-Graphen zu sehen. Die Nachricht $q_{ij}(b)$, die Bitknoten c_i zu Check-Knoten z_j schickt, wird aus den r -Nachrichten aller Check-Knoten bestimmt, die mit dem Bitknoten c_i verbunden sind – mit Ausnahme von Knoten z_j selbst. Zwischen den Knoten wird also wiederum nur extrinsische Information ausgetauscht.

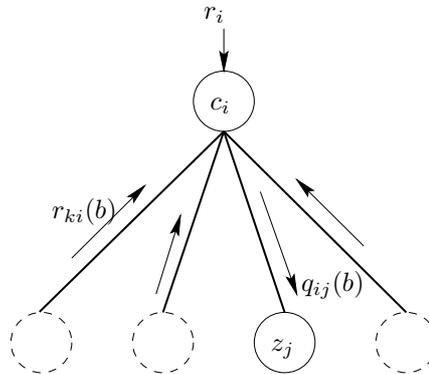


Abbildung 6.11: Ausschnitt aus einem Tanner-Graphen: Berechnung der $q_{ij}(b)$.

Die Menge der mit Bitknoten c_i verknüpften Check-Knoten z_k sei B_i . Wie bereits erwähnt, kennzeichnet die Nachricht $q_{ij}(b)$ den Glauben, dass $c_i = b$. Zur Bestimmung des Codesymbols c_i kann der Decoder sowohl den entsprechenden Empfangswert r_i als auch alle r -Nachrichten der Check-Knoten z_k aus $B_i \setminus j$ heranziehen. Aus dem Empfangswert r_i kann bei bekannter Kanalcharakteristik eine a priori Wahrscheinlichkeit $P(c_i = 1 | r_i) = P_i$ bestimmt werden [Pro95]. Formal ergibt sich laut Gallager für binäre Codes [Gal62]

$$q_{ij}(0) = K_{ij}(0) \cdot (1 - P_i) \cdot \prod_{k \in B_i \setminus j} r_{ki}(0)$$

$$q_{ij}(1) = K_{ij}(1) \cdot P_i \cdot \prod_{k \in B_i \setminus j} r_{ki}(1).$$
(6.28)

Dieser Schritt muss für jeden Bitknoten c_i und für alle Check-Knoten aus B_i und für alle möglichen Werte b von c_i durchgeführt werden. Die Konstanten K_{ij} werden so normiert, dass $q_{ij}(0) + q_{ij}(1) = 1$. Zusätzlich zu dieser Normierung müssen im Falle von regulären LDPC Codes über $GF(q)$ ($n \times W_s \times q$) q -Nachrichten pro Halbiteration berechnet werden.

Abbruchkriterium

Nach einem horizontalen und einem vertikalen Durchlauf wird für jeden Bitknoten c_i eine Nachricht Q_i berechnet, die im Gegensatz zu den q -Nachrichten aus (6.28) die r -Nachrichten *aller* Elemente aus B_i berücksichtigt. Dieser Schritt muss somit nicht für jede Kante des Tanner-Graphen, sondern nur für jeden Bitknoten c_i und für alle möglichen Werte b von c_i durchgeführt werden, also $n \times q$ mal für reguläre LDPC Codes über $GF(q)$. Es ergibt sich laut Gallager für binäre Codes [Gal62]

$$Q_i(0) = K_i(0) \cdot (1 - P_i) \cdot \prod_{k \in B_i} r_{ki}(0)$$

$$Q_i(1) = K_i(1) \cdot P_i \cdot \prod_{k \in B_i} r_{ki}(1).$$
(6.29)

Die Konstanten K_i werden wiederum so normiert, dass $Q_i(0) + Q_i(1) = 1$. Danach kann von den r - und q -Nachrichten wieder zu der binären Information zurückgekehrt werden, indem der Decoder eine Abschätzung \hat{c}_i für die Codebits c_i abgibt:

$$\hat{c}_i = \begin{cases} 1 & \text{falls } Q_i(1) > 0,5 \\ 0 & \text{sonst} \end{cases}$$
(6.30)

Abschließend wird mit Hilfe von (6.26) überprüft, ob das so erhaltene Decodierergebnis $\hat{\underline{c}}$ ein gültiges Codewort ist. Falls ja, das heißt $\hat{\underline{c}} \cdot \mathcal{H}^T = 0$, wird der Decodiervorgang abgebrochen. Falls nein, wird eine neue Iteration gestartet, sofern die maximale Iterationszahl noch nicht erreicht wurde. Die Q -Nachrichten aus (6.29) konvergieren für einen zyklusfreien Tanner-Graphen und mit steigender Iterationszahl gegen die a posteriori Wahrscheinlichkeiten für die Codebits c_i [Mac99].

6.6.2 Implementierungsaspekte

Der Belief-Propagation Algorithmus erfordert nur Multiplikationen und Additionen und somit beispielsweise keine aufwändigen Exponential- oder Logarithmus-Berechnungen. Allerdings sind zur Berechnung der r -, q - und Q -Nachrichten aus (6.27) bis (6.29) sehr viele Multiplikationen notwendig. Insbesondere bei sehr langen Codewörtern muss für die Berechnung der q -Nachrichten eine sehr lange Kette von Multiplikationen gebildet werden (6.28). Dies kann zu einer numerischen Instabilität führen, weil sich die Nachrichten $q_{ij}(b)$ für unterschiedliche Werte von b kaum unterscheiden. Sie kann allerdings durch eine Skalierung der Zuverlässigkeitsinformation teilweise behoben werden [PS02].

Der Log-BP Algorithmus [Gal62] behebt dieses Problem durch die Verwendung von Log-Likelihood Werten an Stelle von Wahrscheinlichkeiten. Bei der Initialisierung wird ein Log-Likelihood Wert $L(c_i)$ unter Verwendung der a priori Wahrscheinlichkeiten $P(c_i = 1 | r_i) = P_i$ und $P(c_i = 0 | r_i) = 1 - P_i$ so gebildet, dass

$$L(c_i) = \log \frac{P(c_i = 0 | r_i)}{P(c_i = 1 | r_i)}. \quad (6.31)$$

Analog dazu können auch aus den r -, q - und Q -Nachrichten aus (6.27) bis (6.29) Log-Likelihood Werte gebildet werden.

Mit dem Log-BP Algorithmus lässt sich der Decoder für praktische Implementierungen wesentlich vereinfachen, da nur noch Additionen und Table-Look-Ups notwendig sind [Gal62]. Für mobile Anwendungen ergibt sich daraus der Vorteil, dass die Decoderschaltung mit einer geringen Verlustleistung implementiert werden kann. Des Weiteren ist der Decoder auch numerisch stabiler als ein herkömmlicher BP-Decoder. Dabei werden ausschließlich für die Implementierung der Table-Look-Ups mathematische Näherungen verwendet. Davon abgesehen ist der Log-BP Algorithmus identisch mit dem herkömmlichen BP Algorithmus.

Ein weiterer Vorteil von LDPC Codes ergibt sich aus dem leicht parallelisierbaren Decodieralgorithmus. Die Berechnung der r - und der q -Nachrichten kann innerhalb einer Halbiteration für jeden Knoten unabhängig erfolgen. Dies ermöglicht parallele Hardware-Implementierungen, die bereits bei niedrigen Taktfrequenzen einen sehr hohen Datendurchsatz erreichen [HB01, BH02].

Darüber hinaus sind weitere Algorithmen entwickelt worden, um die Decodierkomplexität zu verringern [Wib96, RU98, Fos99]. Bei Hardware-Implementierungen wird jedoch in der Regel auf den Log-BP Algorithmus zurückgegriffen.

6.6.3 LDPC Codes auf dem magnetischen Kanal

MacKay und Davey [MD99] waren die ersten, die LDPC Codes auch mit kurzen Blocklängen und mit hohen Coderaten untersucht haben. Da auch unter diesen, für Festplatten günstigeren Bedingungen gute Ergebnisse erzielt werden konnten, sind LDPC Codes schließlich auch auf dem magnetischen Kanal intensiv untersucht worden [FFKM99, STC01, OM01, PS02, Ste02].

Die dabei erzielten Ergebnisse waren durchweg gut und lagen im Bereich der zuvor veröffentlichten Resultate mit Turbo Codes (siehe Kap. 6.5). LDPC Codes besitzen allerdings eine bessere Minimaldistanz als Turbo Codes, so dass in den Simulationen kein Error-Floor festgestellt werden konnte [STC01]. Auch neueste Ergebnisse, bei denen bis zu einer Fehlerrate von 10^{-10} simuliert wurde, haben keinen Error Floor gezeigt [HM03].

Die gute Leistung von LDPC Codes auf dem magnetischen Kanal ist auf einer Vielzahl unterschiedlicher Lesekanäle bestätigt worden. So konnten sie erfolgreich mit PRML-Detektoren mit $EPR4$ - [FFKM99] und ME^2PR4 -Target [STC01], mit Signalraumdetektoren [PS02, Ste02] sowie mit RLL- [OM01] und MTR-Codes [STC01] kombiniert werden. Bei sehr hohen Coderaten und einer Sektorgröße von 512 Bytes sind sie den Reed-Solomon Codes im direkten Vergleich jedoch unterlegen [PS02].

Der Einsatz von Turbo Equalization [DJB95] führt auch bei LDPC Codes zu einer Verbesserung von rund 0,5 dB [STC01], die sich allerdings auf Kanälen unter dem Einfluss von Transitionsrauschen in etwa halbiert [OM01]. Dieser Umstand ist kritisch, da der Anteil von Transitionsrauschen auf dem magnetischen Kanal mit einer steigenden Datendichte zunimmt [Str00].

Kapitel 7

Evaluation von Lesekanälen mit Zuverlässigkeitsinformation

In den vorherigen Kapiteln 4, 5 und 6 wurden für die verschiedenen Komponenten des magnetischen Kanals Methoden präsentiert, die die Verarbeitung von Zuverlässigkeitsinformation ermöglichen. Hier werden nun aus diesen Komponenten Lesekanäle zusammengesetzt und miteinander verglichen. Das Hauptziel ist der Entwurf eines Kanals mit einer geringeren Fehlerrate als bei einem Referenzkanal mit Reed-Solomon Codes und Hard-Decoding. Durch die Einführung von Soft-Decoding soll dieses Ziel ohne eine Erhöhung des Redundanzanteils im Datenstrom erzielt werden. Gleichzeitig soll der Anstieg der Hardware-Komplexität im Hinblick auf mobile Anwendungen möglichst gering sein.

Die Ergebnisse werden jeweils für Kanäle mit weißem Rauschen (Average White Gaussian Noise, AWGN) am Eingang des Equalizers und für Kanäle, die von Transitionsrauschen dominiert werden, vorgestellt. Des Weiteren werden sowohl unbeschränkte Kanäle betrachtet, die nur aus einem Detektor und einem Fehlerschutzcode bestehen, als auch modulationscodierte Kanäle.

7.1 Simulationsumgebung

Alle in diesem Kapitel veröffentlichten Ergebnisse basieren auf Simulationsdaten. Zu diesem Zweck wurde der Lesekanal-Simulator HdSim verwendet [DSS], mit dessen Hilfe das in Kap. 2.4.2 definierte Kanalmodell realisiert wurde. Für einen Simulationslauf wurden jeweils 50 Millionen Nutzdatenbits verwendet.

Für die Diskussion werden die erzielbaren Bitfehlerraten für unterschiedliche Signal-Rausch-Verhältnisse (SNR) ausgewertet. Dabei wird das Signal-Rausch-Verhältnis, je nach Lesekanal, zwischen 20 dB und 30 dB variiert. Falls angegeben, wird das Signal des Weiteren durch ein Transitionsrauschen (Jitter) mit der Varianz $\sigma^2 = 0,075 \cdot T$ überlagert.

Da in dieser Arbeit verschiedene Detektoren, Modulationscodes und fehlerkorrigierende

Codes zu einem Lesekanal kombiniert werden sollen, ergibt sich eine sehr große Anzahl unterschiedlicher Konstellationen. Aus diesem Grund sind einige der Kanalparameter nicht variiert worden. Beispielsweise ist die Datendichte bei allen Simulationen auf einen Wert von $PW_{50}/T = 3.0$ fixiert worden. Des Weiteren werden – wie bereits in Kapitel 2.4.2 beschrieben – weder Non-Linear Transition Shift (NLTS), noch Thermal Asperity (TA) berücksichtigt.

7.2 Betrachtete Komponenten

In dieser Arbeit soll ein Lesekanal entworfen werden, der eine Verbesserung der Signalzuverlässigkeit durch die Verwendung von Soft-Decoding erreicht. Neben den heute üblichen Standard-Komponenten werden daher verschiedene Methoden zur Integration von Zuverlässigkeitsinformation zur Verbesserung des Fehlerschutzcodes untersucht. Dabei kommen die im Folgenden aufgelisteten Komponenten zum Einsatz:

Detektoren

Auf dem magnetischen Kanal werden verbreitet Partial Response Maximum Likelihood (PRML) Detektoren eingesetzt. Aus diesem Grund werden sie auch in dieser Arbeit betrachtet. Signalraumdetektoren werden zum Vergleich herangezogen, da sie sich als eine einfach implementierbare und zuverlässige Alternative erwiesen haben [Str00].

1. Partial Response Maximum Likelihood (PRML) Detektor mit EPR4 sowie mit ME^2PR4 Target. Auf Soft-Decoding Kanälen wird der Soft-Output Viterbi Algorithmus zur Bestimmung der Zuverlässigkeitsinformation verwendet.
2. Signalraumdetektor (Noise-Whitening Signalspace Detector, WSSD) mit 3D-110 Target. Auf Soft-Decoding Kanälen wird der Soft-Output WSSD (S^3D) aus Kap. 4.5 eingesetzt. In der Regel berücksichtigt der Detektor den Einfluss der Fehlerfortpflanzung bei der Bestimmung der Soft-Outputs. Wenn dieser Effekt für eine weitere Vereinfachung der Detektorarchitektur vernachlässigt wird, dann wird explizit darauf hingewiesen.

Modulationscodes

Neben unbeschränkten Kanälen, die nur aus einem Detektor und einem fehlerkorrigierenden Code zusammengesetzt sind, werden die folgenden beiden Varianten mit einem zusätzlichen Modulationscode betrachtet:

1. Maximum Transition Run (MTR) Code [BM97b] der Rate $\frac{6}{7}$. Auf Soft-Decoding Kanälen wird der aus modifizierten logischen Gattern konstruierte Soft-In/Soft-Out Decoder aus Kap. 5.2.5 verwendet.

2. Alternativ zu einem Modulationscode wird ein Parity-Check Code mit der Vorschrift Odd-Parity getestet, mit dessen Hilfe ein Minimum an Timing-Information im Lesesignal erzeugt werden kann (Kap. 5.3).

Fehlerkorrigierende Codes

Die in dieser Arbeit untersuchten Lesekanäle sollen zu einer Verbesserung der bisher eingesetzten Varianten mit Reed-Solomon Codes in Verbindung mit Hard-Decoding eingesetzt werden. Insgesamt werden die folgenden Varianten betrachtet:

1. $RS(255, 240)$ Reed-Solomon Code mit 8-Bit Codesymbolen, 3-fachem Interleaving und den Parametern $n = 255$, $k = 240$ und $m = 8$.
2. Wie oben, allerdings wird ein Reed-Solomon Code eingesetzt, der Fehler und Erasures korrigieren kann. Die Erasure-Positionen werden mit Hilfe eines inneren Parity-Check Codes bestimmt, der gleichzeitig als Modulationscode dient (vgl. Kap. 6.4.3).
3. Low-Density Parity-Check (LDPC) Code mit einer Länge von 512 Informationsbytes und einer Coderate von 0,88 bzw. 0,94. Der Code wurde über das Galoisfeld $GF(4)$ gebildet.
4. Turbo Code mit dem Generatorpolynom $(31, 27)$ und einer Länge von 512 Informationsbytes, der auf die Coderate 0,88 punktiert wurde. Zwischen Modulationsdecoder und Fehlerschutzdecoder wird ein zusätzlicher Kanal-Interleaver eingesetzt [RMAAY00]. Dieser wandelt die Bündelfehler des magnetischen Kanals in zufällige Einzelbitfehler um.

Im Folgenden werden zunächst die Fehlermuster der eingesetzten Detektoren betrachtet. Danach erfolgt die Untersuchung verschiedener Lesekanäle.

7.3 Untersuchung der Fehlermuster verschiedener Detektoren

Partial Response Maximum Likelihood (PRML) und Signalraumdetektoren wurden bereits intensiv auf unbeschränkten und MTR-codierten Kanälen untersucht [Str00, SHS00]. Dabei hat sich gezeigt, dass die Signalraumdetektoren trotz einer geringeren Komplexität bei gleichem SNR niedrigere Fehlerraten erreichen. Der Vorsprung gegenüber PRML-Detektoren vergrößert sich sogar bei starkem Transitionsrauschen [Str00].

Allerdings wurde bei den bisherigen Untersuchungen [Str00] ausschließlich die Bitfehler-rate am Ausgang des Detektors untersucht. Entscheidend für die Zuverlässigkeit eines magnetischen Lesekanals ist jedoch die Fehlerrate am Ausgang des Fehlerschutzcodes. An dieser Stelle kann es jedoch zu deutlich anderen Resultaten kommen, da nicht mehr nur die gemittelte Fehlerrate, sondern auch die Verteilung der Fehler eine Rolle spielen. Produziert beispielsweise ein Detektor zwar eine hohe Bitfehler-rate, die allerdings

eine günstige Verteilung aufweist, so kann das Ergebnis am Ausgang der Fehlerkorrektur sehr gut sein. Umgekehrt kann ein Detektor, der seltene, aber sehr lange Bündelfehler erzeugt, ein schlechtes Ergebnis am Ausgang der Fehlerkorrektur erzielen, obwohl die Bitfehlerrate am Detektorausgang sehr niedrig ist. Für die Fehlerrate am Ausgang des Lesekanals ist also immer das Zusammenspiel aller Komponenten entscheidend.

Zunächst erfolgt an dieser Stelle jedoch eine Untersuchung der Fehlermuster der betrachteten Systeme. Für den Fehlerschutzdecoder ist die Bitfehlerrate und die Verteilung der Fehler am Eingang des Decoders relevant. Aus diesem Grund werden im Folgenden die Fehlermuster am Ausgang des Detektors (bei unbeschränkten Kanälen) bzw. des Modulationsdecoders (bei MTR-codierten Kanälen) betrachtet.

Dabei werden sowohl die verschiedenen in Kapitel 4 vorgestellten Detektoren, als auch der Einfluss von Transitionsrauschen untersucht. Alle Untersuchungen wurden durch die Simulation von 50 Millionen Nutzdatenbits bei einem Signal-Rausch-Verhältnis von 26 dB durchgeführt. Die in diesem Kapitel untersuchten Lesekanäle bestehen entweder nur aus einem Detektor und einem Fehlerschutzcode oder sie besitzen darüber hinaus einen Modulationscode. Daher wird auch der Einfluss des Modulationscodes auf die Fehlerraten und -verteilungen untersucht. Entscheidend für die erreichbare Fehlerrate am Ausgang des Fehlerschutzdecoders ist die Fehlerrate und -verteilung am Eingang der Fehlerkorrektur. Hier muss untersucht werden, ob die Verbesserung durch das Entfernen kritischer Sequenzen oder die Verschlechterung durch die Fehlerfortpflanzung im Modulationsdecoder überwiegt.

In Abb. 7.1 ist die Verteilung der Fehlerlängen bei Einsatz eines $EPR4$ -Detektors zu sehen. Auffallend ist, dass auf einem unbeschränkten Kanal Fehlerlängen von über 10 Bit auftreten. Diese Eigenschaft verschlechtert sich unter dem Einfluss von Transitionsrauschen noch einmal deutlich. Wird hingegen ein MTR-Code verwendet, treten nur noch Bündelfehler mit einer maximalen Länge von 10 Bit auf. Dafür ist die Anzahl kürzerer Fehlerbündel erhöht, was auf eine Fehlerfortpflanzung bei kurzen Fehler-Ereignissen hindeutet.

Bei Einsatz eines PRML-Detektors mit ME^2PR4 -Target (Abb. 7.2) ergibt sich ein deutlich günstigerer Verlauf. Bündelfehler sind praktisch auf eine Länge von 4 Bit beschränkt und die Fehlerstatistik wird von Einzelbit-Fehlern dominiert. Unter dem Einfluss von starkem Transitionsrauschen ergibt sich eine ähnliche Verteilung wie beim $EPR4$ -Detektor, allerdings auf einem deutlich niedrigeren Niveau. Auf einem MTR-codierten Kanal ist die Verteilung der des $EPR4$ -Detektors ähnlich: Lange Fehler-Ereignisse werden auf Kosten einer erhöhten Fehlerfortpflanzung bei kurzen Fehler-Ereignissen reduziert.

Ein völlig anderes Bild zeigt sich bei Verwendung eines Signalraumdetektors (Abb. 7.3). Dieser Detektor ist auf unbeschränkten Kanälen deutlich unempfindlicher gegenüber Transitionsrauschen – die langen Fehlerketten der PRML-Detektoren treten hier nicht auf. Beim Einsatz eines Modulationscodes tritt hier im Gegensatz zu den PRML-Detektoren keine Verbesserung ein. Die Häufigkeit kurzer, geradzähliger Fehlerbündel nimmt sogar zu. Offenbar wirkt sich an dieser Stelle die Kombination aus Fehlerfortpflanzung im Detektor und im Modulationsdecoder besonders negativ auf die Fehlerrate aus. Darüber hinaus produziert der Signalraumdetektor in sehr seltenen Fällen recht lange Fehler-Ereignisse [Pla01]. Diese setzen sich aus mehreren kurzen Fehlerbündeln zusammen. Aus

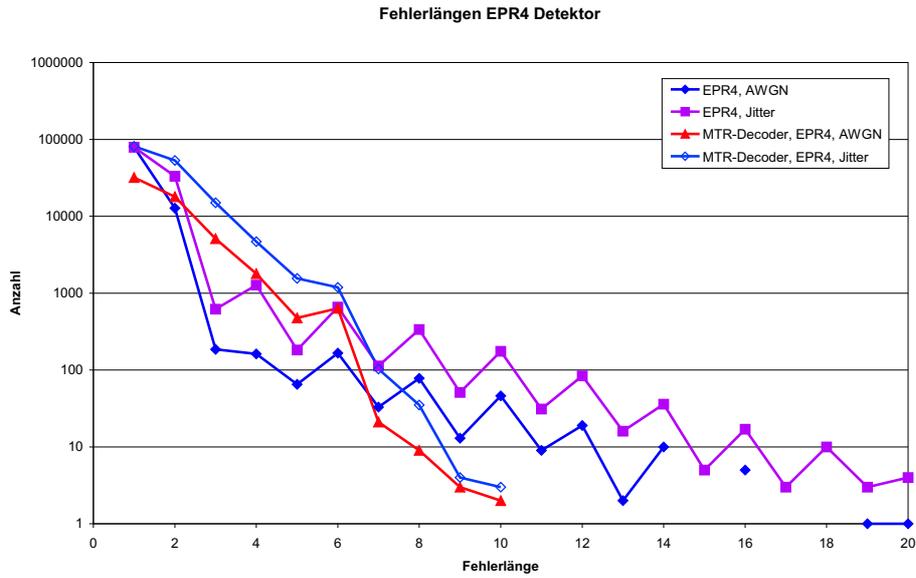


Abbildung 7.1: Verteilung der Fehlerlängen eines EPR4-PRML Detektors.

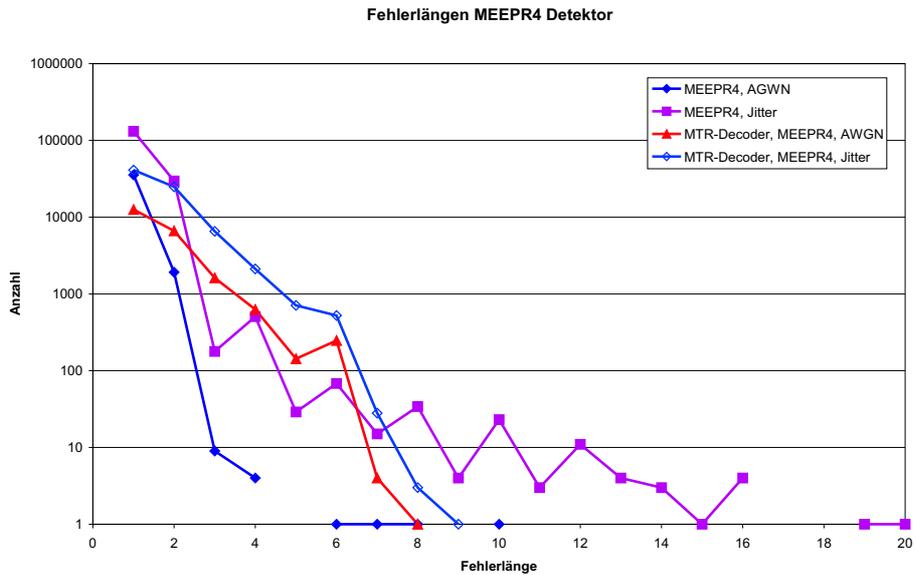


Abbildung 7.2: Verteilung der Fehlerlängen eines ME²PR4-PRML Detektors.

diesem Grund sind sie in Abb. 7.3 nicht zu erkennen.

Die aus den Diagrammen in Abb. 7.1 bis 7.3 zu erkennenden Eigenschaften lassen vermuten, dass für PRML-Detektoren Modulationscodes insbesondere bei starkem Transitionsrauschen unerlässlich sind. Ein Lesekanal, der – wie in Kap. 5.3 vorgeschlagen –

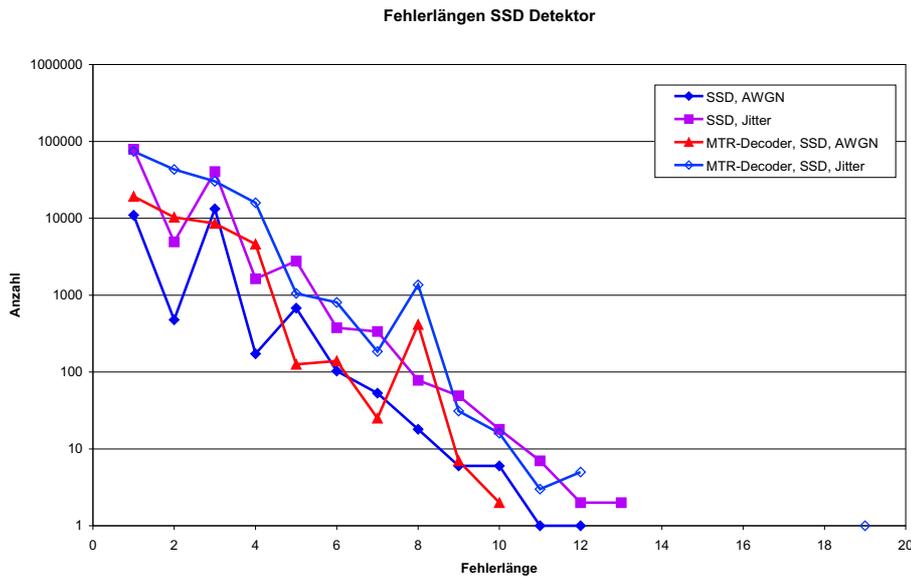


Abbildung 7.3: Verteilung der Fehlerlängen eines Signalraumdetektors.

einen einfachen Parity-Check Code als Modulationscode einsetzt, ist aus diesem Grund vermutlich nur bei Einsatz eines Signalraumdetektors zu empfehlen.

7.4 Lesekanal mit Reed-Solomon Codes und Hard-Decoding

Lesekanäle mit Reed-Solomon Codes haben sich bei Festplatten und anderen Datenspeichern als Standard etabliert. Aus diesem Grund soll an dieser Stelle ein solcher Kanal als Referenz für alle im Weiteren betrachteten Soft-Decoding Kanäle betrachtet werden. Dabei kommt der in Kap. 7.2 beschriebene Reed-Solomon Code zum Einsatz. Es werden sowohl PRML-Detektoren mit EPR4 und ME²PR4 Target, als auch ein Signalraumdetektor (WSSD Detektor) verwendet.

7.4.1 Unbeschränkte Kanäle

Zunächst wird die Leistung auf unbeschränkten Kanälen untersucht, die nur aus einem Detektor und einem Fehlerschutzcode bestehen. Der Hauptgrund für die Betrachtung dieser Kanäle ist die Untersuchung des Zusammenspiels zwischen diesen beiden Komponenten. Da in diesem Fall auf einen Modulationscode verzichtet wird, kann keinerlei Timing-Information im Lesesignal garantiert werden.

Die erreichbaren Bitfehlerraten am Ausgang des Fehlerschutzdecoders sind in Abb. 7.4 zu sehen. Zunächst wird von weißem Rauschen am Eingang des Equalizers ausgegangen. Der PRML-Detektor mit ME²PR4-Target erzielt die geringste Bitfehlerrate, allerdings mit nur einem geringen Vorsprung vor dem Signalraumdetektor. Dies ist erstaunlich, da

der Signalraumdetektor eine wesentlich einfachere Architektur besitzt. Der EPR4-PRML Detektor ist den beiden anderen Verfahren deutlich unterlegen.

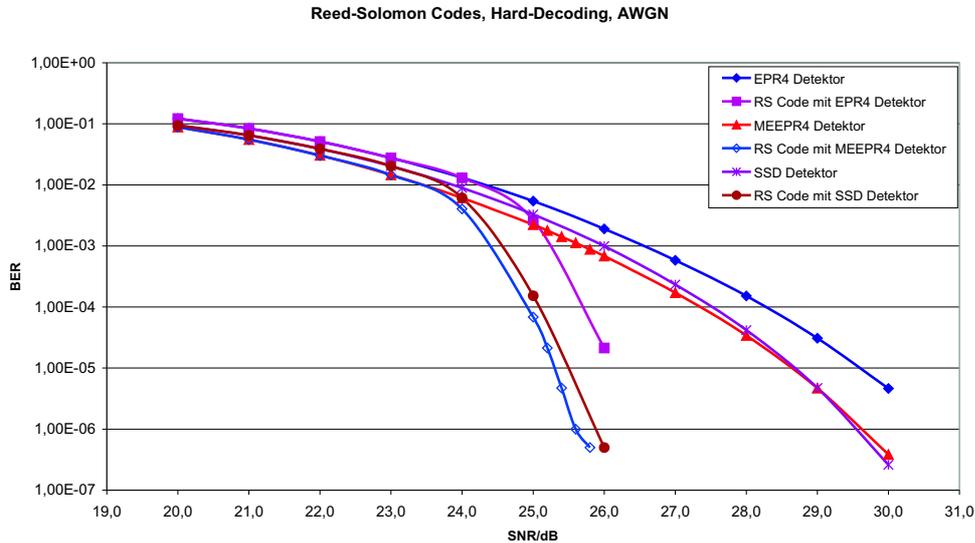


Abbildung 7.4: Fehlerraten verschiedener Lesekanäle unter Verwendung von Reed-Solomon Codes auf einem unbeschränkten Kanal unter dem Einfluss von weißem Rauschen.

Unter dem Einfluss von Transitionsrauschen mit der Varianz $\sigma^2 = 0,075 \cdot T$ ergibt sich ein sehr ähnliches Bild (Abb. 7.5). Wiederum erreicht der EPR4-Detektor nicht die niedrigen Fehlerraten der beiden anderen Detektionsverfahren. Im Vergleich zum AWGN-Kanal aus Abb. 7.4 verschlechtern sich die Ergebnisse um rund 2 dB.

7.4.2 Modulationscodierte Kanäle

Für die folgenden Untersuchungen wurden die Lesekanäle um einen Modulationscode erweitert. Hierbei wird ein MTR-Code mit der Rate $\frac{6}{7}$ eingesetzt. Die Analyse der Fehlermuster in Kap. 7.3 hat gezeigt, dass bei den PRML Detektoren und unter dem Einfluss von Transitionsrauschen die Anzahl langer Fehler-Ereignisse reduziert werden kann. Beim Signalraumdetektor trat diese Verbesserung hingegen nicht auf.

Diese Ergebnisse spiegeln sich auch in den erzielbaren Bitfehlerraten am Ausgang des Reed-Solomon Decoders wider. Dies führt dazu, dass unter dem Einfluss von AWGN der Abstand zwischen dem Kanal mit dem ME²PR4-PRML Detektor und dem Signalraumdetektor größer wird (Abb. 7.6). Der EPR4-Detektor bleibt das schlechteste Verfahren im Vergleich.

Noch deutlicher tritt der Vorteil eines Modulationscodes bei den PRML-Detektoren unter dem Einfluss von Transitionsrauschen auf. Hier liegt die Verschlechterung gegenüber dem AWGN-Kanal nur noch bei ca. 1 dB, während sich die Bitfehlerrate des Signalraumdetektors weiterhin um rund 2 dB verschlechtert (Abb. 7.7). Am Ausgang eines

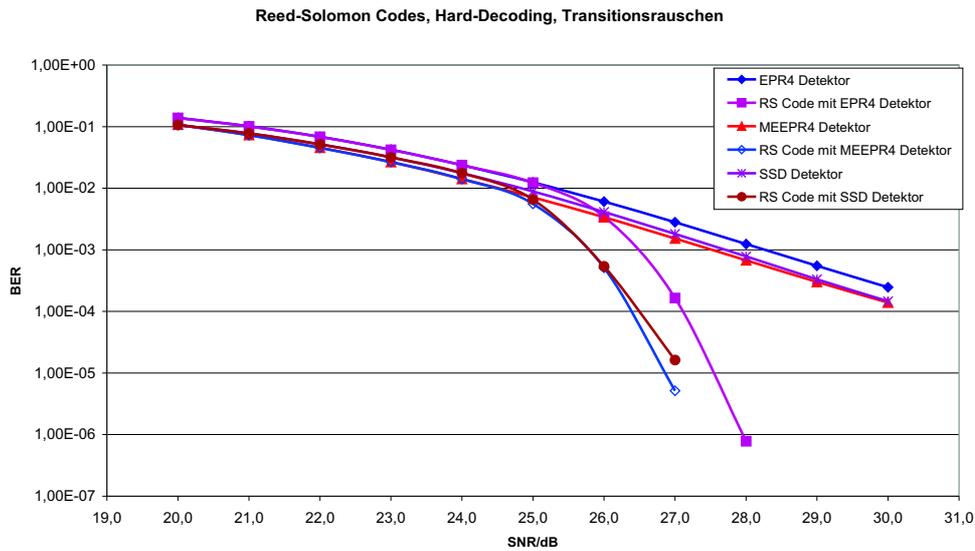


Abbildung 7.5: Fehlerraten verschiedener Lesekanäle unter Verwendung von Reed-Solomon Codes auf einem unbeschränkten Kanal unter dem Einfluss von Transitionsrauschen.

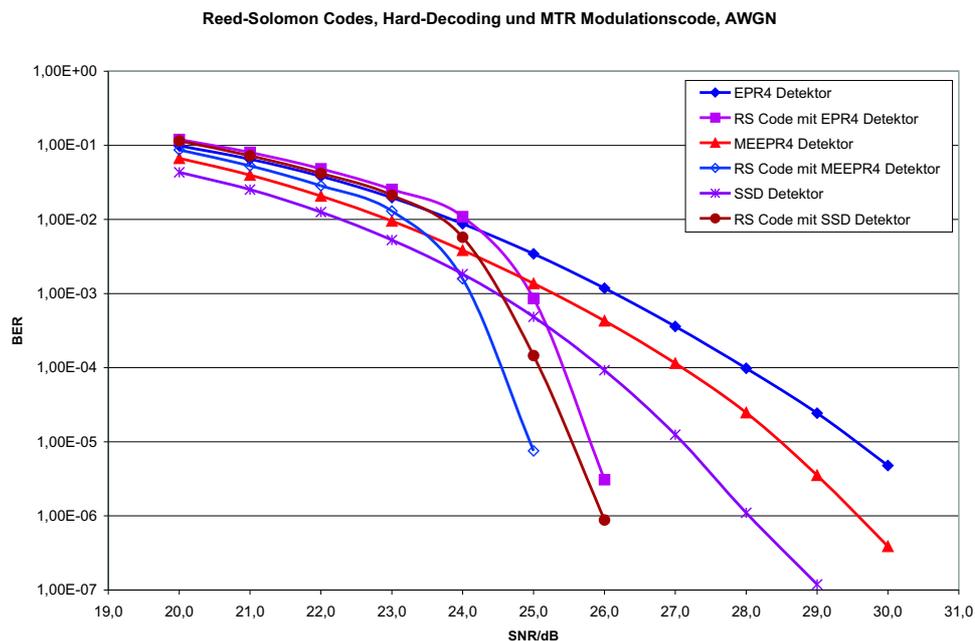


Abbildung 7.6: Fehlerraten verschiedener Lesekanäle unter Verwendung von Reed-Solomon Codes auf einem MTR-codierten Kanal unter dem Einfluss von weißem Rauschen.

Reed-Solomon Decoders zeigt sich somit, dass ein MTR-Code bei Verwendung eines Signalraumdetektors nicht zu niedrigeren Fehlerraten führt. Dies bestätigt die Resultate,

die am Ausgang des Detektors gemessen wurden [Str00].

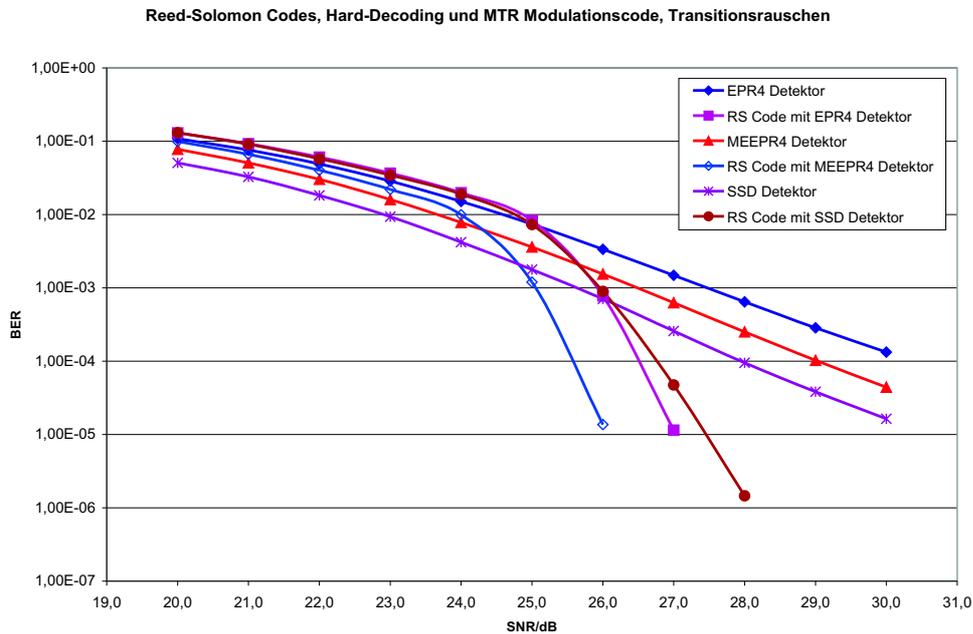


Abbildung 7.7: Fehlerraten verschiedener Lesekanäle unter Verwendung von Reed-Solomon Codes auf einem MTR-codierten Kanal unter dem Einfluss von Transitionsrauschen.

7.5 Lesekanäle mit Zuverlässigkeitsinformation

In diesem Kapitel sollen die betrachteten Lesekanäle für die Verarbeitung von Zuverlässigkeitsinformation modifiziert werden. Diese wird in den Detektoren individuell für jedes Bit generiert. Bei den PRML-Detektoren wird zu diesem Zweck der Soft-Output Viterbi-Algorithmus (SOVA) verwendet. Andere Verfahren, wie zum Beispiel der BCJR-Algorithmus, werden aus Gründen der Komplexität nicht berücksichtigt. Im Falle des Signalraumdetektors kommt das in Kap. 4.5 präsentierte Verfahren zum Einsatz.

7.5.1 Unbeschränkte Kanäle

Das Ziel bei der Verwendung von Zuverlässigkeitsinformation ist die Verbesserung der Fehlerkorrektur. Aus diesem Grund werden an dieser Stelle zunächst wiederum Kanäle betrachtet, die auf unbeschränkten Daten arbeiten. Das Augenmerk liegt dabei auf der Verbesserung, die sich durch die Verwendung von Zuverlässigkeitsinformation erzielen lässt.

Eine Voraussetzung für die Verarbeitung von Zuverlässigkeitsinformation ist ein fehlerkorrigierender Code, der diese Zusatzinformation gewinnbringend weiterverarbeiten kann. Dazu sind die bislang betrachteten Reed-Solomon Codes nicht in der Lage. Aus

diesem Grund werden zunächst nur Turbo Codes und Low-Density Parity-Check Codes betrachtet.

Verwendung von Turbo Codes

Turbo Codes waren im Jahre 1993 ein wichtiger Anstoß für viele Veröffentlichungen im Bereich Soft-Decoding, unter anderem für den magnetischen Kanal (siehe Kap. 6.5). Aus diesem Grund sollen sie auch hier als fehlerkorrigierende Codes in Betracht gezogen werden. Turbo Codes verwenden parallel verkettete terminierte Faltungscodes mit einer Coderate von $\frac{1}{3}$. Höhere Coderaten, wie sie für den magnetischen Kanal unerlässlich sind, können durch das Punktieren von Parity-Bits erreicht werden. Für die Simulationen in diesem Kapitel wurde der (31, 27)-Code aus dem ursprünglichen Paper von Berrou et al. verwendet [BGT93] und auf eine Coderate von 0,88 punktiert.

Die Ergebnisse sind in Abb. 7.8 für AWGN zu sehen. Die PRML-Detektoren erreichen eine sehr gute Decodierleistung. Bei Verwendung eines ME²PR4-PRML Detektors wird bereits bei einem SNR von weniger als 24 dB eine Fehlerrate von 10^{-5} erreicht. Der EPR4-Detektor liegt jetzt mehr als 1 dB schlechter. Es ist zu beachten, dass die Ergebnisse mit einer niedrigeren Coderate als bei den Reed-Solomon Codes in Kap. 7.4 erzielt wurden.

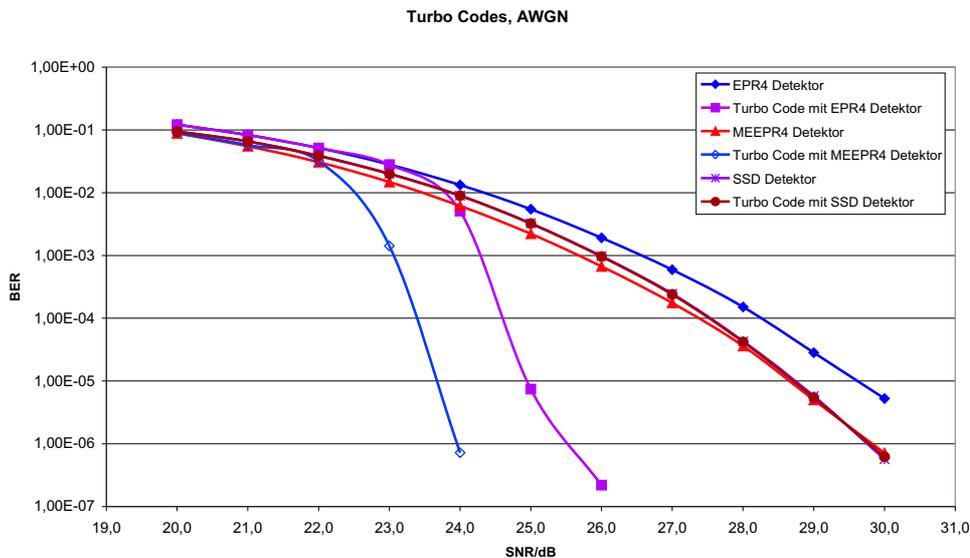


Abbildung 7.8: Fehlerraten verschiedener Lesekanäle unter Verwendung eines (31, 27) Turbo Codes auf einem unbeschränkten Kanal unter dem Einfluss von weißem Rauschen.

Unter dem Einfluss von Transitionsrauschen (Abb. 7.9) verschlechtern sich die Ergebnisse um rund 1 dB. Dieser Abfall ist geringer als bei der Verwendung von Reed-Solomon Codes. Gleichzeitig ist allerdings bei hohem SNR ein deutliches Abflachen der Kurven zu erkennen. Dieses als Error Floor bekannte Phänomen tritt also bei starkem Transitionsrauschen (Jitter) stärker in Erscheinung. Da mit steigender Datendichte der Jitter-Anteil

am Rauschspektrum stetig zunimmt, ist dieser Umstand für Festplatten besonders kritisch. Bei höheren Coderaten verstärkt sich der Einfluss des Error Floors [Che01, Pla01], so dass sich die Untersuchungen in dieser Arbeit auf Turbo Codes mit der Coderate 0,88 beschränken.

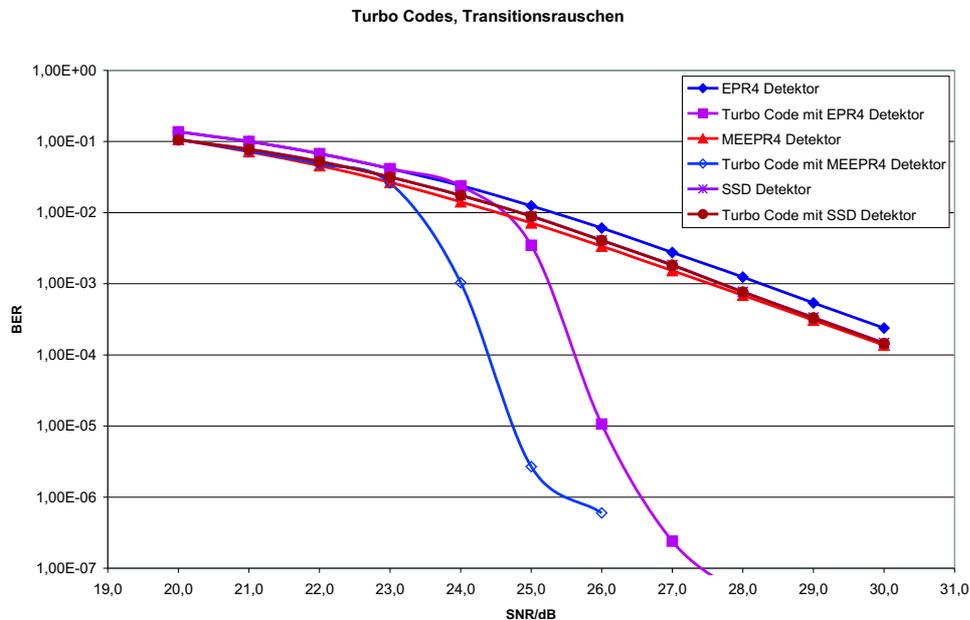


Abbildung 7.9: Fehlerraten verschiedener Lesekanäle unter Verwendung eines (31, 27) Turbo Codes auf einem unbeschränkten Kanal unter dem Einfluss von Transitionsrauschen.

Bei Verwendung eines Signalraumdetektors ergibt sich ein komplett anderes Bild: Der für zufällige Einzelbitfehler optimierte Turbo Code funktioniert bei der sehr starken Punktierung nicht mit dem zu Fehlerfortpflanzung neigenden Signalraumdetektor. In Kombination mit Turbo Codes ist also der Einsatz eines PRML-Detektors erforderlich. Dies bestätigt frühere Untersuchungen [Pla01], bei denen Lesekanäle mit Turbo Codes unter Verwendung eines Signalraumdetektors erst bei einer Blocklänge von 4096 Bytes eine höhere Zuverlässigkeit als Kanäle mit PRML-Detektoren erreicht haben. Sofern die Sektorgröße von Festplatten auf 512 Bytes beschränkt bleibt, ist diese Kombination also nicht zu empfehlen.

Verwendung von LDPC Codes

Erst nachdem die Turbo Codes eine große Aufmerksamkeit erlangt haben, wurden die eigentlich viel älteren Low-Density Parity-Check (LDPC) Codes [Gal62] wieder intensiv erforscht. In verschiedenen Veröffentlichungen wurde gezeigt, dass LDPC Codes eine ähnlich gute Decodierleistung wie Turbo Codes besitzen [Mac99]. Bei kurzen Blocklängen sind eher Turbo Codes im Vorteil, während LDPC-Codes bei sehr großen Blocklängen überlegen sind [RU98].

In dieser Arbeit werden LDPC-Codes mit zwei verschiedenen Coderaten untersucht. Ein Code der Rate 0,88 dient als Vergleich zu den Turbo Codes, während ein Code der Rate 0,94 mit den Reed-Solomon Codes aus Kap. 7.4 verglichen wird.

Die Ergebnisse für den LDPC Code mit der Coderate 0,88 sind in Abb. 7.10 für AWGN bzw. in Abb. 7.11 für starkes Transitionsrauschen zu sehen. Mit den PRML-Detektoren sind die erreichten Bitfehlerraten in beiden Fällen den Turbo Codes sehr ähnlich. Der große Vorteil der LDPC Codes liegt jedoch darin, dass auch unter dem Einfluss von Transitionsrauschen kein Error Floor zu erkennen ist. Der Abfall gegenüber den AWGN-Kurven liegt wie schon bei Turbo Codes im Bereich von nur 1 dB. Mit dem Signalraumdetektor ist die erreichte Bitfehlerrate wiederum deutlich schlechter, allerdings sind die Ergebnisse etwas besser als bei der Verwendung von Turbo Codes. Wie auch dort, erreicht ein LDPC Decoder bei Verwendung des Signalraumdetektors erst bei deutlich größeren Blocklängen seine volle Leistungsfähigkeit [Pla01].

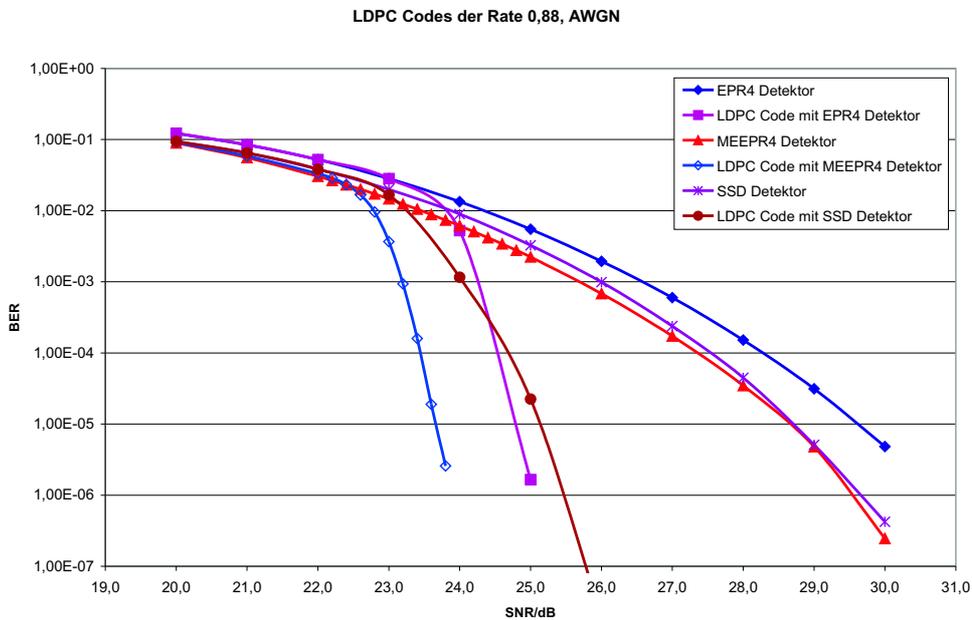


Abbildung 7.10: Fehlerraten verschiedener Lesekanäle unter Verwendung eines LDPC Codes mit der Coderate 0,88 auf einem unbeschränkten Kanal unter dem Einfluss von weißem Rauschen.

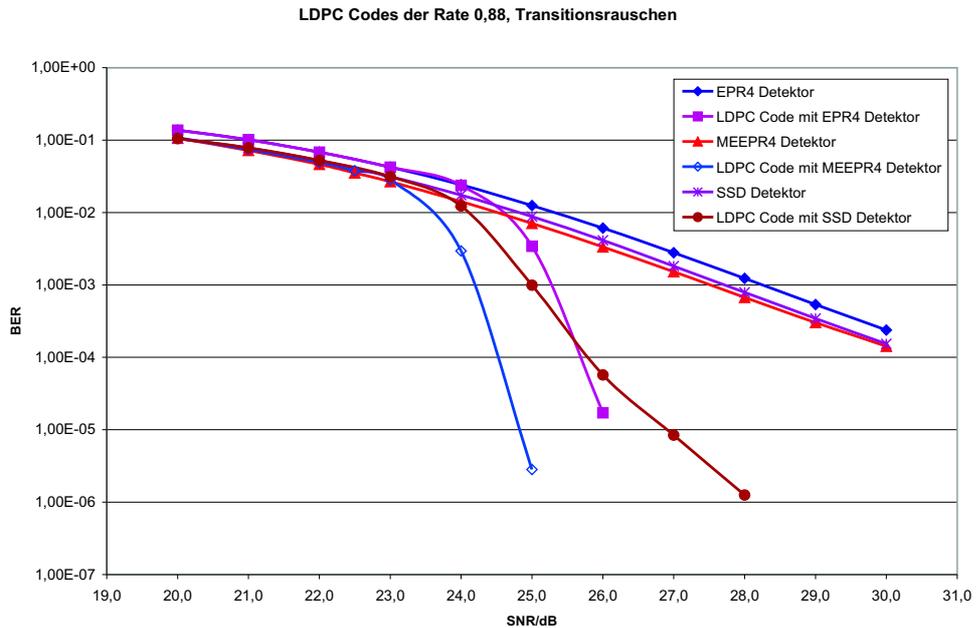


Abbildung 7.11: Fehlerraten verschiedener Lesekanäle unter Verwendung eines LDPC Codes mit der Coderate 0,88 auf einem unbeschränkten Kanal unter dem Einfluss von Transitionsrauschen.

Bei einem LDPC Code mit einer höheren Coderate von 0,94 vergrößert sich der Abstand zwischen den beiden PRML-Detektoren; der Signalraumdetektor führt auch hier zu den schlechtesten Ergebnissen (Abb. 7.12 für AWGN bzw. Abb. 7.13 für starkes Transitionsrauschen). Bei Verwendung eines EPR4-PRML Detektors zeigt sich zudem unter dem Einfluss von starkem Transitionsrauschen eine leichte Abflachung der Kurve bei hohem SNR. Aus diesem Grund ist bei sehr hochratigen LDPC Codes der Einsatz eines ME^2PR4 -PRML Detektors zu empfehlen, der dieses dem Error Floor ähnliche Verhalten nicht aufweist. Die Fehlerrate von Reed-Solomon Codes mit Hard-Decoding wird allerdings nicht ganz erreicht.

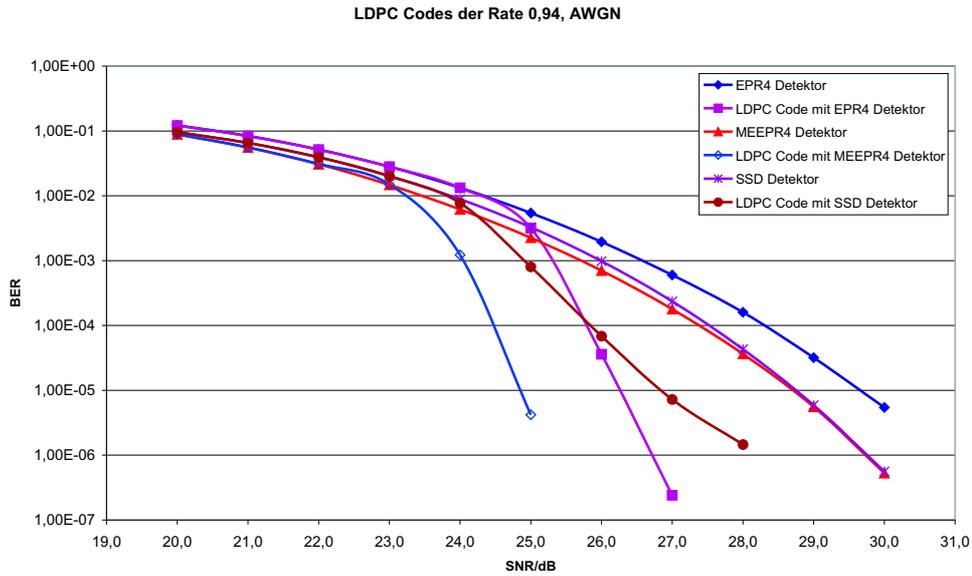


Abbildung 7.12: Fehlerraten verschiedener Lesekanäle unter Verwendung eines LDPC Codes mit der Coderate 0,94 auf einem unbeschränkten Kanal unter dem Einfluss von weißem Rauschen.

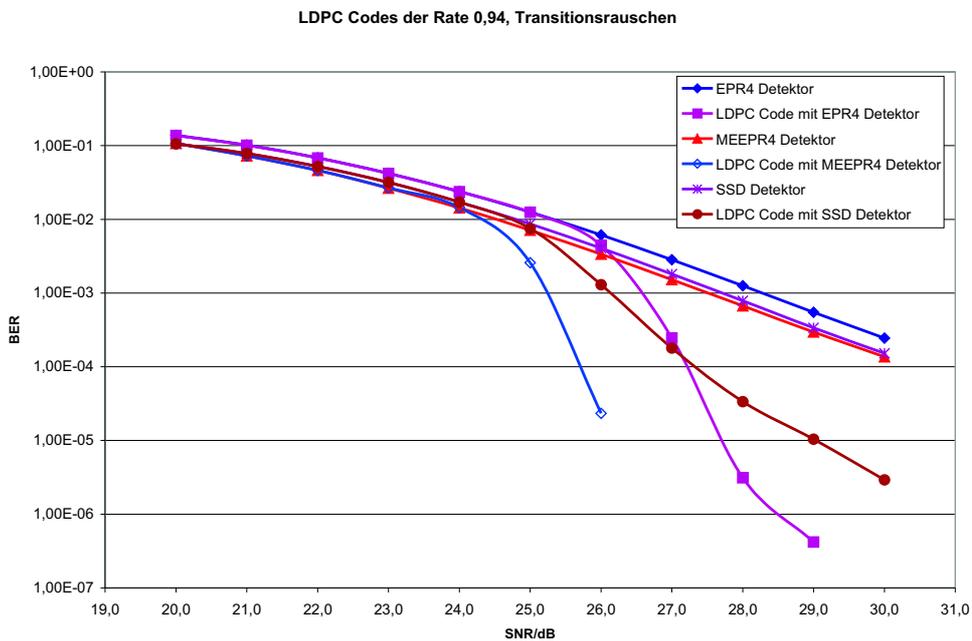


Abbildung 7.13: Fehlerraten verschiedener Lesekanäle unter Verwendung eines LDPC Codes mit der Coderate 0,94 auf einem unbeschränkten Kanal unter dem Einfluss von Transitionsrauschen.

7.5.2 Modulationscodierte Kanäle

Hier werden die Systeme mit LDPC bzw. Turbo Codes aus Kap. 7.5.1 um den Soft-In/Soft-Out Modulationsdecoder aus Kap. 5.2.5 erweitert. Ein Modulationscode ist auf dem magnetischen Kanal notwendig, weil sonst bei langen Nullfolgen der Takt nicht mehr aus dem Lesesignal zurückgewonnen werden kann.

Als Alternative wird der Error-/Erasure korrigierende Reed-Solomon Code aus Kap. 6.4.3 betrachtet. Dieser verwendet einen inneren Parity-Check Code sowohl zum Markieren von Erasures, als auch als einfachen Modulationscode. Neben der Bitfehlerrate wird die Effizienz des Erasure-markierenden Codes für PRML- und Signalraumdetektoren untersucht.

Verwendung von Turbo Codes

Die Verwendung eines MTR-Codes bringt bei PRML-Kanälen in Verbindung mit Turbo Codes nicht den bei Reed-Solomon Codes beobachteten Vorteil (Kap. 7.4). Stattdessen scheint die Fehlerfortpflanzung bei kurzen Fehlerbündeln eine größere Rolle zu spielen als die Reduktion sehr langer Fehler-Ereignisse. Der Error Floor tritt bei allen PRML-Detektoren bereits bei einem niedrigeren SNR auf. Dies ist sowohl unter dem Einfluss von AWGN (Abb. 7.14), als auch bei starkem Transitionsrauschen (Abb. 7.15) zu beobachten. Die Ergebnisse des Signalraumdetektors bleiben auf modulationscodierten Kanälen weit hinter denen der PRML-Detektoren zurück. Die weitere Verschlechterung der Ergebnisse durch die Verwendung eines Modulationscodes spricht ebenfalls gegen die Verwendung von Turbo Codes auf dem magnetischen Kanal.

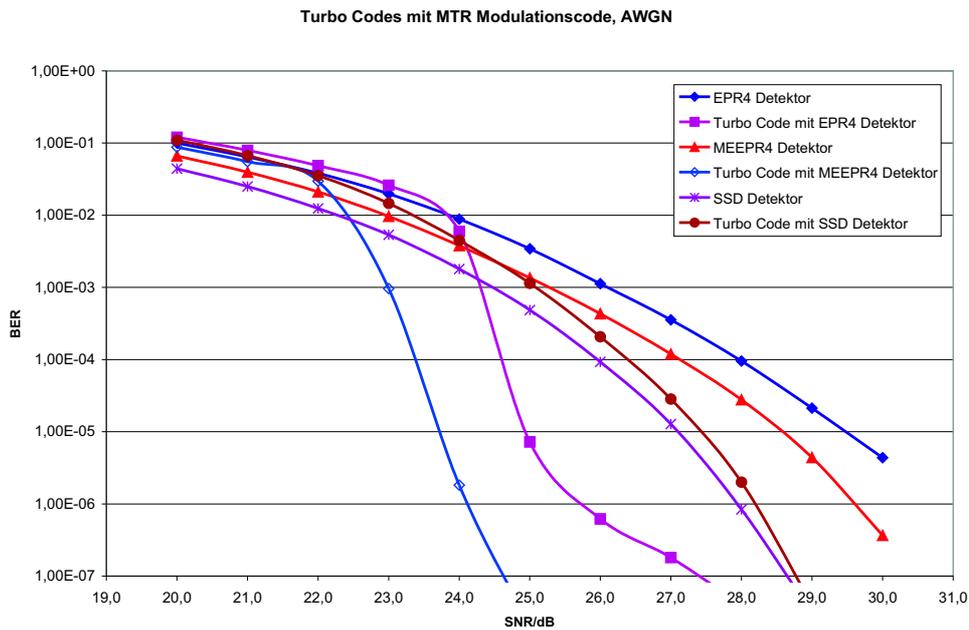


Abbildung 7.14: Fehlerraten verschiedener Lesekanäle unter Verwendung eines (31,27) Turbo Codes auf einem MTR-codierten Kanal unter dem Einfluss von weißem Rauschen.

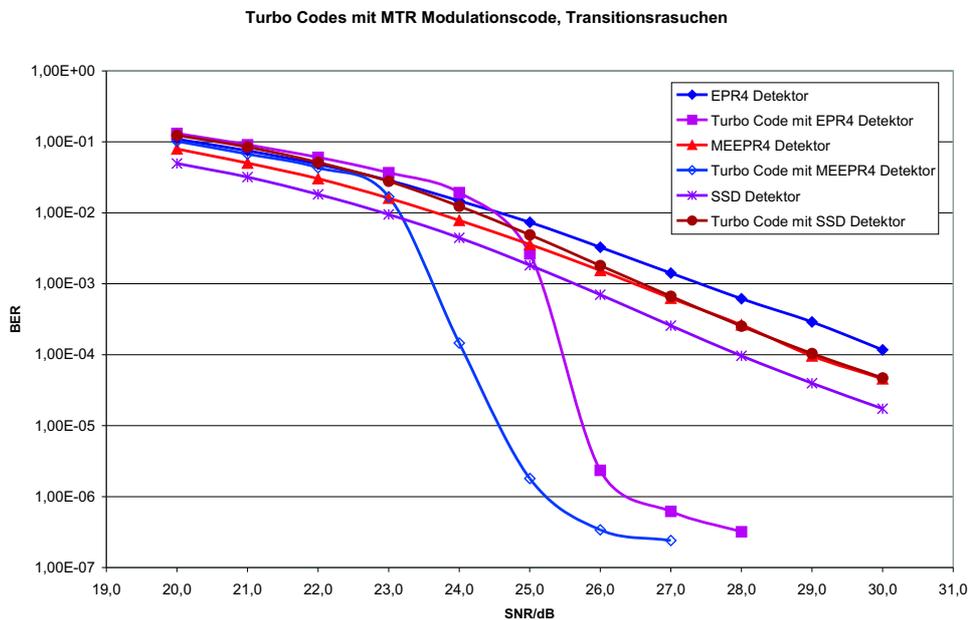


Abbildung 7.15: Fehlerraten verschiedener Lesekanäle unter Verwendung eines (31,27) Turbo Codes auf einem unbeschränkten Kanal unter dem Einfluss von Transitionsraschen.

Verwendung von LDPC-Codes

Die bei Turbo Codes beobachtete Verschlechterung der Performance durch einen MTR-Code tritt in Verbindung mit LDPC Codes nicht auf – die Ergebnisse verbleiben in etwa im gleichen Bereich wie auf einem unbeschränkten Kanal.

Bei Verwendung der LDPC-Codes mit einer Coderate von 0,88 (Abb. 7.16 für AWGN) fällt auf, dass sich die Resultate des Signalraumdetektors im Vergleich zu den PRML-Detektoren verbessern: Die Fehlerraten des EPR4-PRML Detektors und des Signalraumdetektors liegen sehr dicht beieinander, während der ME²PR4-PRML Detektor rund 1 dB besser abschneidet. Unter starkem Transitionsrauschen (Abb. 7.17) verschieben sich die Relationen wieder leicht zu Ungunsten des Signalraumdetektors, dessen Fehlerrate stärker ansteigt als bei den beiden PRML-Detektoren.

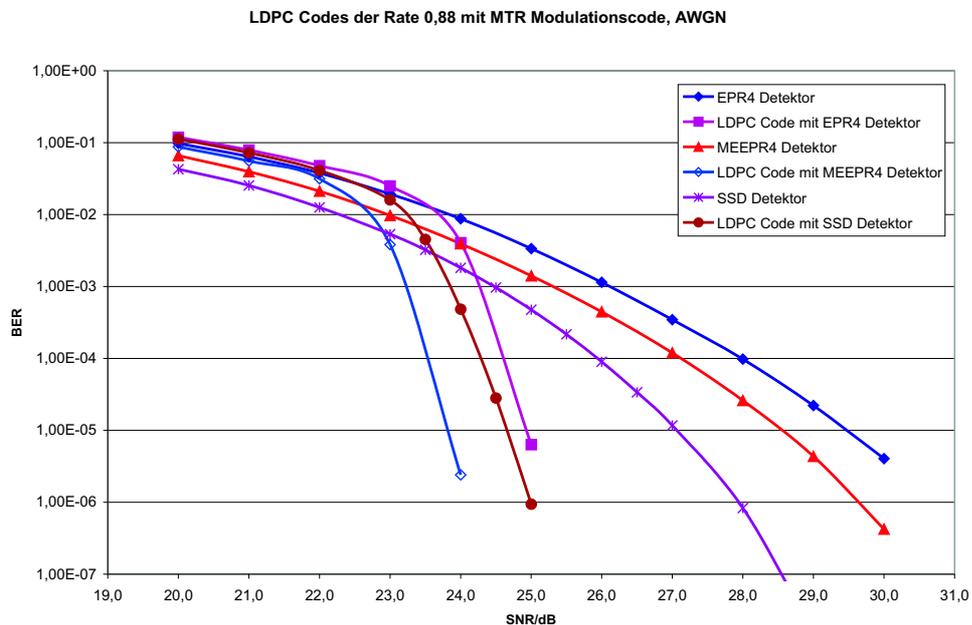


Abbildung 7.16: Fehlerraten verschiedener Lesekanäle unter Verwendung eines LDPC Codes mit der Coderate 0,88 auf einem MTR-codierten Kanal unter dem Einfluss von weißem Rauschen.

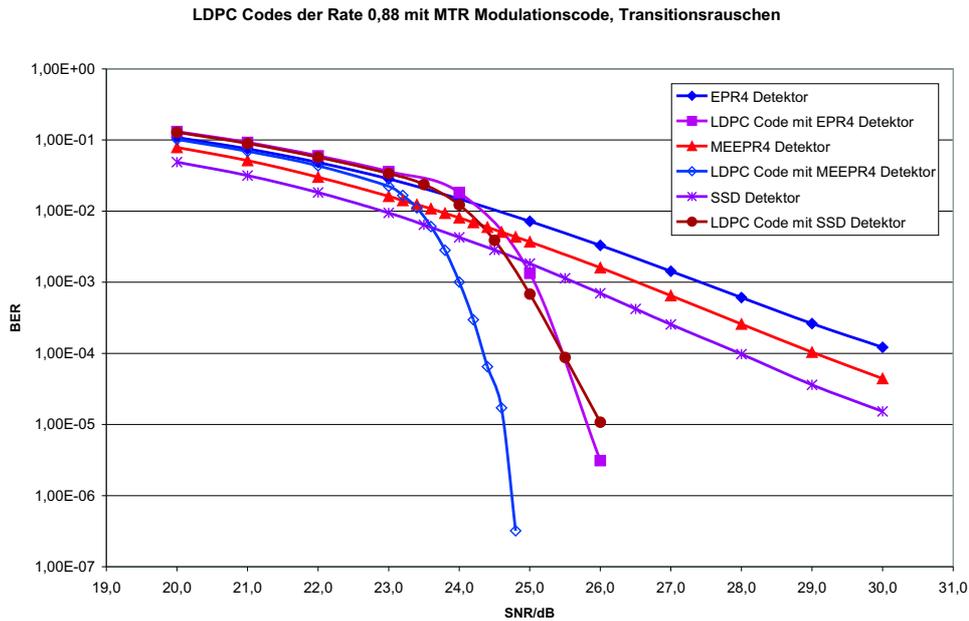


Abbildung 7.17: Fehlerraten verschiedener Lesekanäle unter Verwendung eines LDPC Codes mit der Coderate 0,88 auf einem MTR-codierten Kanal unter dem Einfluss von Transitionsrauschen.

Bei einer Coderate von 0,94 ändern sich die Relationen für den ME^2PR4 -PRML Detektor kaum (Abb. 7.18 für AWGN bzw. in Abb. 7.19 für starkes Transitionsrauschen). Im Vergleich zu Reed-Solomon Codes mit Hard-Decoding kann in etwa die gleiche Fehlerrate erreicht werden. Der EPR4-PRML Detektor profitiert bei Jitter vom Einsatz eines MTR-Codes (vgl. Abb. 7.13): Der Effekt der Abflachung der Kurve tritt hier nicht mehr auf. Allerdings ist die Decodierleistung jetzt fast 2 dB schlechter als beim ME^2PR4 -PRML Detektor.

Auch der Lesekanal mit dem Signalraumdetektor weist einen deutlich günstigeren Verlauf der Fehlerraten als bei unbeschränkten Daten auf. Allerdings ist trotzdem noch deutlich zu sehen, dass die Kombination von kurzen und hochratigen LDPC-Codes mit einem Signalraumdetektor nicht zu guten Ergebnissen führt.

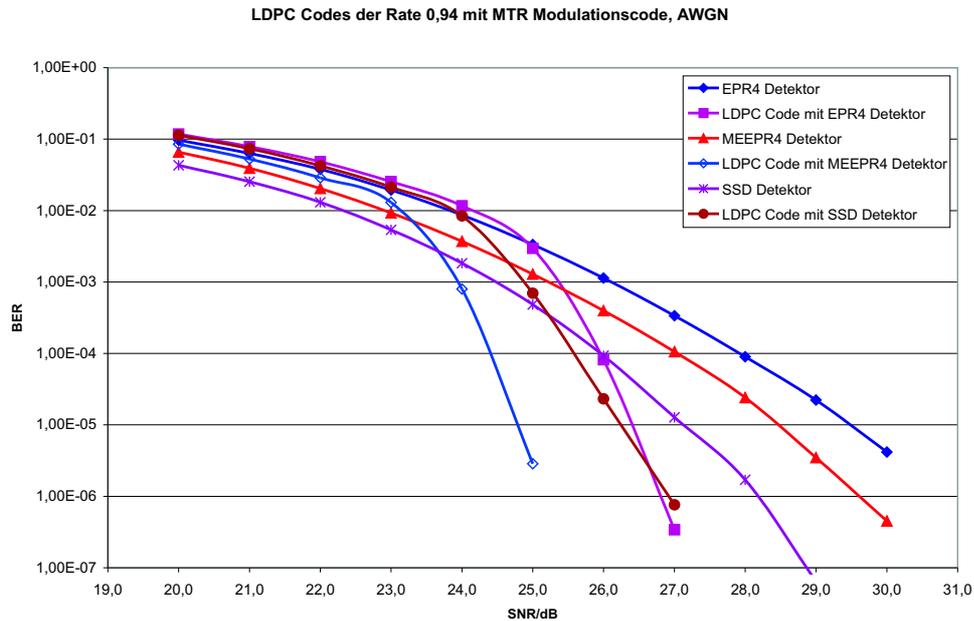


Abbildung 7.18: Fehlerraten verschiedener Lesekanäle unter Verwendung eines LDPC Codes mit der Coderate 0,94 auf einem MTR-codierten Kanal unter dem Einfluss von weißem Rauschen.

Verwendung von Erasure-korrigierenden Reed-Solomon Codes

In Kap. 6.4.3 ist ein Verfahren vorgestellt worden, das Reed-Solomon Codes die Verarbeitung von Zuverlässigkeitsinformation ermöglicht. Dabei werden mit Hilfe eines inneren Parity-Check Codes und der Zuverlässigkeitsinformation potentielle Fehlerpositionen als Erasures markiert. Dieser Parity-Check Code wird als Odd-Parity ausgelegt, wodurch gleichzeitig Timing-Information im Lesesignal garantiert wird (Kap. 5.3).

Die Effizienz dieses Parity-Check Codes soll im Folgenden für einen Signalraumdetektor und einen PRML-Detektor (hier mit EPR4-Target) untersucht werden. Die Korrekturfähigkeit des Reed-Solomon Codes wird durch zwei Größen beeinflusst:

1. Wenn ein Symbolfehler korrekt identifiziert werden kann, ist ein einziges Parity-symbol des Reed-Solomon Codes ausreichend zur Fehlerkorrektur. In diesem Fall kann also die Korrekturfähigkeit des Reed-Solomon Codes erhöht werden: Im Extremfall, das heißt bei einer Identifizierung aller Symbolfehler, kann die doppelte Anzahl von Fehlern korrigiert werden.

Daher wird in diesem Kapitel untersucht, welcher Prozentsatz der auf dem Kanal aufgetretenen Symbolfehler durch den Parity-Check Code identifiziert und somit als Erasure markiert werden kann. Dieser Anteil wird im Folgenden als *korrekte Erasures* bezeichnet.

2. Anders herum kann es natürlich auch passieren, dass ein Erasure gesetzt wird,

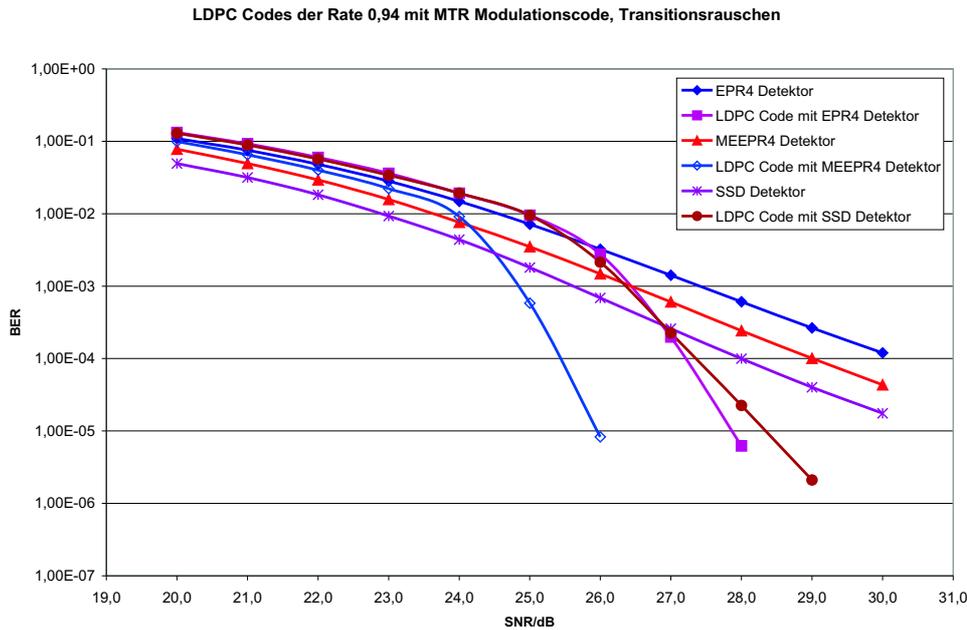


Abbildung 7.19: Fehlerraten verschiedener Lesekanäle unter Verwendung eines LDPC Codes mit der Coderate 0,94 auf einem MTR-codierten Kanal unter dem Einfluss von Transitionsrauschen.

obwohl das betreffende Symbol fehlerfrei ist. Dies geschieht zum Beispiel dann, wenn nur Parity-Bits fehlerhaft sind. In diesem Fall wird also ein Teil der Korrekturfähigkeit des Reed-Solomon Codes verschwendet.

Der Vorteil einer hohen Anzahl von korrekt als Erasure identifizierten Symbolfehlern kann also wieder aufgehoben werden, wenn auf der anderen Seite sehr viele der gesetzten Erasures fehlerfreie Codesymbole des Reed-Solomon Codes markieren.

Aus diesem Grund wird also des Weiteren der Prozentsatz der Erasures untersucht, der korrekte Codesymbole markiert. Dieser Anteil wird im Folgenden als *falsche Erasures* bezeichnet.

In Abb. 7.20 ist der Prozentsatz der korrekten Erasures für einen Signalraumdetektor und einen EPR4-PRML Detektor für verschiedene Signal-Rausch-Verhältnisse dargestellt. Es wurde ein (3,2)-Parity-Check Code eingesetzt, der nach jeweils drei Codesymbolen des Reed-Solomon Codes zwei Parity-Bits in den Datenstrom einfügt. Die Kurven sind getrennt für einen Kanal mit AWGN und einen Kanal mit starkem Transitionsrauschen (Jitter) dargestellt.

Bei starkem Rauschen ist die Effizienz des Parity-Check Codes sehr schlecht: Es werden bis zu einem SNR von 21 dB weniger als 20 Prozent der aufgetretenen Symbolfehler korrekt erkannt. Zwischen 22 dB und 26 dB steigt die Effizienz stark an, wobei sich bei Verwendung des Signalraumdetektors ein günstigerer Verlauf ergibt und die Kurven bereits 1 dB früher ansteigen. Bei hohem SNR nähern sich die Kurven der beiden De-

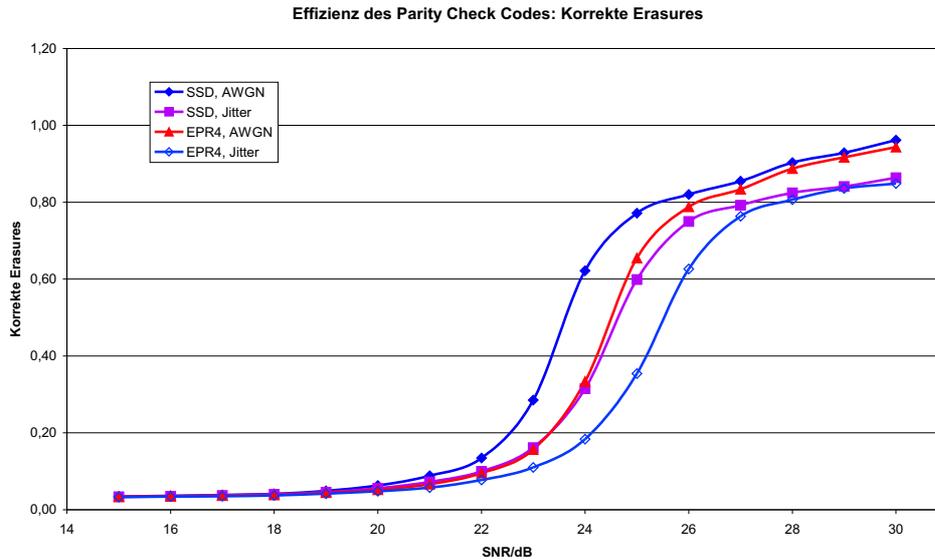


Abbildung 7.20: Prozentsatz der Symbolfehler, die mit Hilfe eines (3,2) Parity-Check Codes und durch die Zuverlässigkeitsinformation als Erasures markiert werden können.

tektoren wieder an. Starkes Transitionsrauschen verschlechtert die Ergebnisse bei beiden Detektoren um rund 1 dB und führt darüber hinaus zu einem Abfall der Effizienz des Parity-Check Codes von knapp 10 Prozentpunkten. Der Verlauf der Kurven für verschiedene Signal-Rausch-Verhältnisse zeigt aber, dass das Verfahren eher von AWGN beeinflusst wird als von Transitionsrauschen. Dieses Ergebnis ist gerade in Bezug auf künftige Festplatten wichtig, bei denen der Anteil von Transitionsrauschen am Rauschspektrum weiter zunehmen wird.

In Abb. 7.21 ist der Prozentsatz der falschen Erasures wiederum für einen Signalraumdetektor und einen EPR4-PRML Detektor für verschiedene Signal-Rausch-Verhältnisse dargestellt. Erstaunlicherweise ist der Anteil von falschen Erasures bei einem niedrigen SNR sehr gering und steigt mit steigendem SNR zunächst an. In diesem Bereich treten noch sehr viele Fehler gebündelt auf, die durch den Parity-Check Code nicht erfasst werden können. Mit steigendem SNR steigt die Anzahl der gesetzten Erasures stark an. Im Bereich unterhalb von 22 dB reicht die Qualität der Zuverlässigkeitsinformation aber noch nicht zu einer Identifikation fehlerhafter Codesymbole aus.

Erst darüber sinkt der Prozentsatz falscher Erasures wieder ab. Wiederum ergibt sich für den Signalraumdetektor ein günstigerer Verlauf, als für den EPR4-PRML Detektor. Insbesondere unter dem Einfluss von Transitionsrauschen fällt der Prozentsatz falscher Erasures deutlich schneller und stärker ab. Unter AWGN hingegen erreichen beide Detektoren bei hohem SNR einen Anteil falscher Erasures von weniger als 5 Prozent. Bei Jitter liegt der Anteil für den Signalraumdetektor nur unwesentlich höher, während beim EPR4-PRML Detektor auch bei einem SNR von 30 dB noch 10 Prozent aller Erasures korrekte Codesymbole markieren.

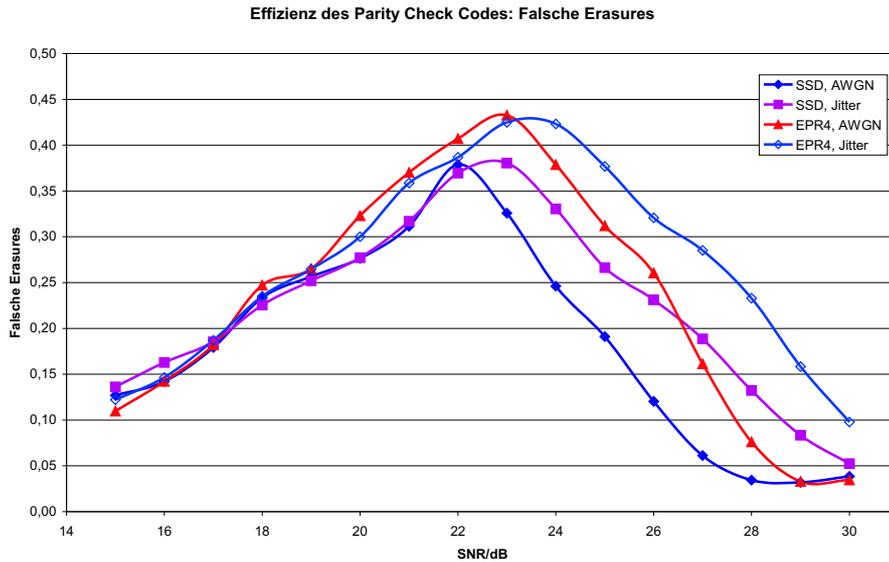


Abbildung 7.21: Prozentsatz der Erasures, die bei Verwendung eines (3,2) Parity-Check Codes ein korrektes Codesymbol markieren.

Im Folgenden wird untersucht, wie sich der innere Erasure-markierende Code auf die erzielbare Fehlerrate auswirkt (Abb. 7.22 für AWGN). Im Vergleich zu Reed-Solomon Codes mit Hard-Decoding (Abb. 7.4) ergibt sich ein deutlich steilerer Abfall der Fehlerraten bei steigendem SNR. Dabei ist zu beachten, dass der Parity-Check Code im Bereich des steilen Kurvenabfalls noch nicht sehr effizient arbeitet (vgl. Abb. 7.20 und 7.21). Der für die Praxis relevante Bereich liegt im Bereich von rund 28 bis 29 dB. Dort ist die Effizienz des Parity-Check Codes sehr gut, und das Optimum einer Verdopplung der Decodierleistung im Vergleich zu Hard-Decoding kann fast erreicht werden.

Die Relation der drei verschiedenen Detektionsprinzipien zueinander bleibt im Vergleich zum Hard-Decoding gleich: Der EPR4-Detektor ist weiterhin deutlich hinter dem Signalraumdetektor, der die Fehlerraten des ME^2PR4 Detektors allerdings nicht ganz erreicht. Unter dem Einfluss von Jitter verschieben sich die Relationen der drei verglichenen Kanäle zueinander kaum (Abb. 7.23). Die Kurve des Lesekanals mit dem Signalraumdetektor liegt noch etwas dichter an der Kurve des Kanals mit dem ME^2PR4 Detektor. Offenbar wirkt sich die geringere Anzahl falscher Erasures hier positiv auf das Ergebnis aus.

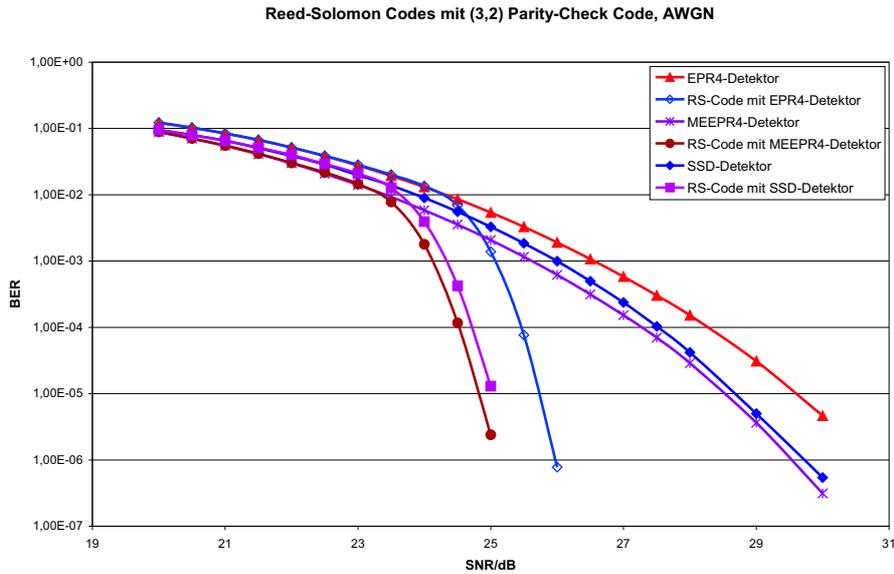


Abbildung 7.22: Fehlerraten verschiedener Lesekanäle unter Verwendung von Reed-Solomon Codes und eines (3,2) Parity-Check Codes unter dem Einfluss von weißem Rauschen.

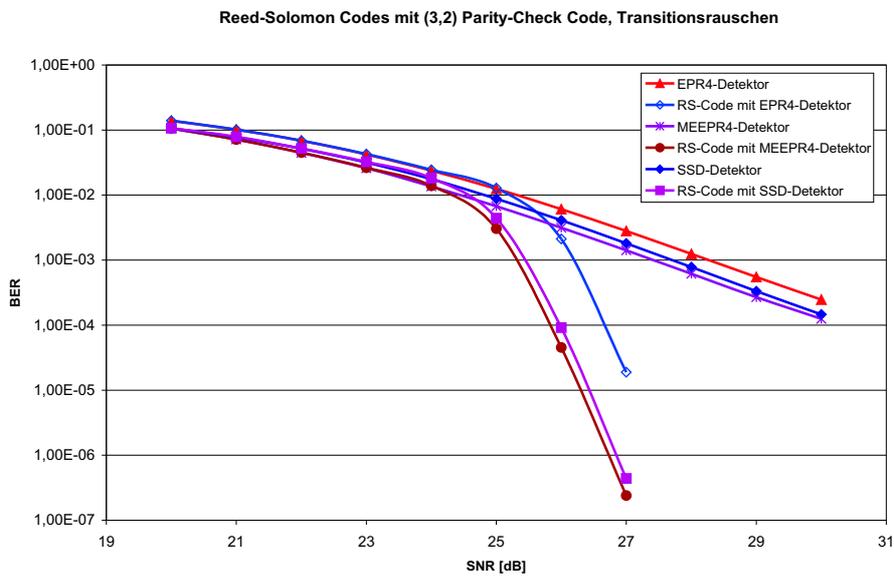


Abbildung 7.23: Fehlerraten verschiedener Lesekanäle unter Verwendung von Reed-Solomon Codes und eines (3,2) Parity-Check Codes unter dem Einfluss von Transitionsrauschen.

Wenn die Gesamt-Coderate des Kanals erhöht werden soll, kann beispielsweise nur noch ein einziges Parity-Bit über drei Reed-Solomon Codesymbole gebildet werden. Auf diese Weise ergibt sich ein (3,1)-Parity Check Code, der dann allerdings nur noch zur Er-

kennung von ungeradzahigen Fehler-Ereignissen in der Lage ist. Die mit einem solchen Code erzielbaren Fehlerraten sind in Abb. 7.24 für AWGN zu sehen. Die Decodierleistung der Kanäle mit PRML-Detektoren fällt stärker ab, als die des Signalraumdetektors, welcher jetzt die gleiche Fehlerrate wie ein ME²PR4-PRML Detektor ermöglicht. Dieser Umstand lässt sich anhand der Fehlerverteilung der Detektoren erklären: Während die PRML-Detektoren auch gehäuft geradzahlige Fehlerbündel verursachen, dominieren beim Signalraumdetektor ungerade Fehler-Ereignisse, die sich mit einem einzigen Parity-Bit erkennen lassen.

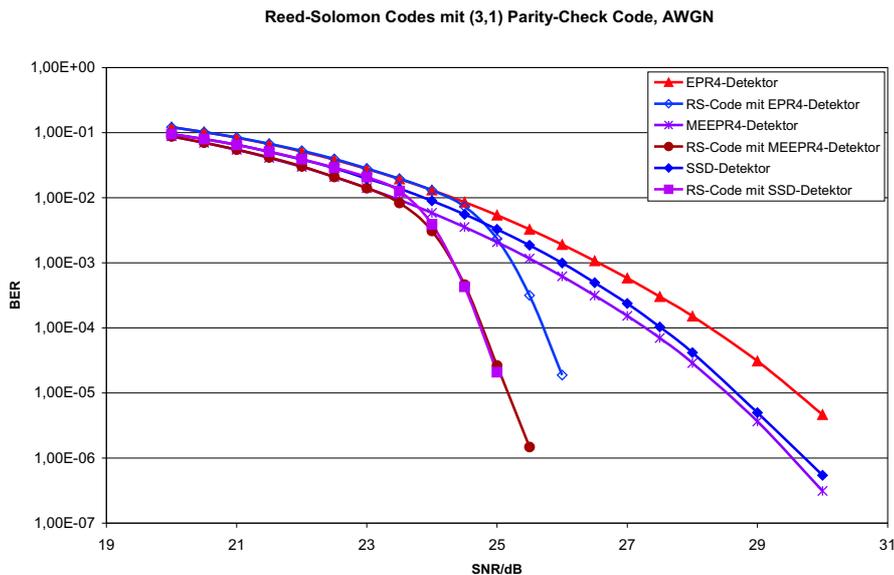


Abbildung 7.24: Fehlerraten verschiedener Lesekanäle unter Verwendung von Reed-Solomon Codes und eines (3,1) Parity-Check Codes unter dem Einfluss von weißem Rauschen.

7.5.3 Systemweite Optimierung

In Kap. 4.5.6 wurde das Konzept eines adaptiven Detektors als Multi-Layer Perzeptron (MLP) erläutert [DS03]. Die Verwendung eines adaptiven Detektors hat in der Praxis eine Reihe von Vorteilen. So kann sich der Detektor beispielsweise an Abweichungen durch Bauteiltoleranzen oder an eine nicht exakt bekannte Rauschkorrelation anpassen. Des Weiteren ist bei einer Änderung von Kanalparametern kein Re-Design der Detektorarchitektur notwendig [DSS02].

Neben diesen während des Designprozesses wichtigen Vorteilen ermöglicht ein adaptiver Detektor eine Anpassung an zeitvariante Kanalparameter, wie zum Beispiel Temperatureinflüsse, während des Detektorbetriebs. Zu diesem Zweck wurde neben dem offline-adaptiven Detektor, der nach einer initialen Adaptionsphase statisch betrieben wird, auch ein online-adaptiver Detektor präsentiert, bei dem die Adaption jederzeit wieder angestoßen werden kann [DS03].

Ein Nachteil des online-adaptiven Detektors ist eine potentielle Instabilität während des Betriebs. Der Adaptionsprozess kann zum Beispiel bei langen Fehler-Ereignissen immer wieder angestoßen werden, obwohl die Konstellation der Entscheidungsebenen sehr gut ist. Aus diesem Grund wird ein Kriterium benötigt, das den Adaptionsprozess erst dann wieder anstößt, wenn die Anzahl der Fehler einen gewissen Schwellwert überschritten hat. Die Zuverlässigkeitsinformation als einen solchen Indikator zu verwenden ist unter Umständen kritisch, da eine hohe Anzahl korrekter Bits mit geringer Zuverlässigkeit zu der erwähnten Instabilität führen kann.

Statt dessen kann der in Kap. 7.5.2 eingeführte Parity-Check Code zu diesem Zweck eingesetzt werden. Dieser kann bereits im Detektor decodiert werden, so dass direkt ein Fehler-Indikator zur Verfügung steht. Wenn die Anzahl der Decodierfehler innerhalb eines Zeitfensters einen definierten Schwellwert überschreitet, kann der Adaptionsprozess erneut gestartet werden. Durch eine geeignete Wahl des Schwellwertes kann sowohl eine zuverlässige Adaption an zeitvariante Parameter, als auch ein stabiler Detektorbetrieb erreicht werden.

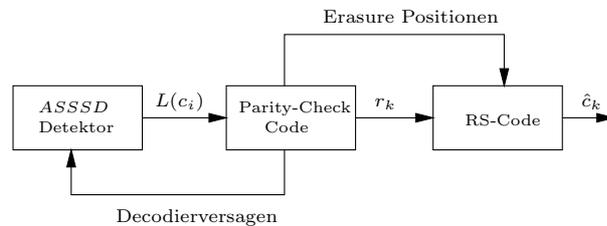


Abbildung 7.25: Systemweite Optimierung des Soft-Decoding Lesekanals: Der innere Parity-Check Code wird zum Einen zur Überwachung der Adaptionsgüte des Adaptiven Soft-Output Signalraumdetektors (ASSSD) verwendet und zum Anderen (in Verbindung mit des Soft-Outputs $L(c_i)$ des Detektors) zur Markierung von potentiell fehlerhaften RS-Code Symbolen r_k als Erasures.

Durch diese Maßnahme kann auf eine sehr effiziente Weise ein adaptiver Soft-Decoding Kanal konstruiert werden (Abb. 7.25). Ausgehend von einem Kanal aus einem Hard-Output Signalraumdetektor, einem MTR Modulationscode und einem Reed-Solomon Code, wird dem System zunächst im Detektor für die Bestimmung der Zuverlässigkeitsinformation und für die Erweiterung zum adaptiven Detektor zusätzliche Hardware hinzugefügt. Auf der anderen Seite stellt der Parity-Check Code eine systemübergreifende Vereinfachung dar, weil er an drei Stellen des Kanals eingesetzt werden kann:

- Im Detektor dient der Code als Indikator für schlecht adaptierte Entscheidungsebenen.
- Des Weiteren ersetzt er den MTR Modulationscode und garantiert ein Minimum an Timing-Information im Datenstrom.
- Zusätzlich wird der Code gemeinsam mit der Zuverlässigkeitsinformation des Detektors als Marker für Erasures im Reed-Solomon Decoder und somit zur Verbesserung der Fehlerkorrektur verwendet. An Stelle einer Verbesserung der Fehlerkorrektur bei gleich bleibender Redundanz kann alternativ auch die gleiche Zuverlässig-

keit bei weniger Redundanz und somit einer Vereinfachung der Decoderkomplexität erzielt werden. Im Fehlerschutzdecoder kann also zwischen einer Erhöhung der Zuverlässigkeit und einer Vereinfachung der Komplexität gewählt werden.

Ein derart optimierter Lesekanal bietet also eine Reihe von Vorteilen: Der adaptive Detektor erlaubt eine Anpassung an zeitvariante Kanal- und Bauteileparameter und ist dadurch in der Praxis besser einsetzbar. Gleichzeitig kann aufgrund der Verwendung von Zuverlässigkeitsinformation eine deutlich niedrigere Fehlerrate erzielt werden. Durch den Verzicht auf einen MTR-Modulationscode und durch die effiziente Auswertung der Zuverlässigkeitsinformation im Reed-Solomon Decoder wird der Anstieg der Komplexität kompensiert.

7.5.4 Diskussion und Zusammenfassung

Die Untersuchung verschiedener Lesekanäle in diesem Kapitel hat gezeigt, dass keine der eingesetzten Komponenten isoliert betrachtet als die beste Lösung für einen Festplatten-Lesekanal angesehen werden kann. Stattdessen muss immer der Kanal als Ganzes betrachtet werden. Auf diese Weise ergeben sich mehrere Konstellationen, die sehr gute Ergebnisse erzielen.

Im klassischen Fall von Hard-Decoding von Reed-Solomon Codes konnten auf unbeschränkten Kanälen sowohl mit einem ME²PR4-PRML Detektor, als auch mit einem Signalraumdetektor sehr gute Ergebnisse erzielt werden. Bemerkenswert ist dabei, dass mit Hilfe des deutlich einfacher zu implementierenden Signalraumkonzepts nahezu die gleiche Zuverlässigkeit erreichbar ist. Diese Eigenschaft prädestiniert den Signalraumdetektor als gute Low-Complexity Lösung.

Wenn zusätzlich ein MTR-Modulationscode eingesetzt wird, können nur die Kanäle mit PRML-Detektoren profitieren. Insbesondere bei starkem Transitionsrauschen werden lange Fehlerbündel verhindert. Bei einem Kanal mit Signalraumdetektor wird der Vorteil aufgrund der Reduktion sehr langer Fehlerbündel durch das verstärkte Auftreten kurzer Fehler-Ereignisse wieder aufgehoben.

Sowohl die PRML-Detektoren, als auch der in dieser Arbeit präsentierte Signalraumdetektor konnten in den Soft-Output Versionen zu einer Erhöhung der Signalzuverlässigkeit beitragen. Einzig die Kombination mit Turbo Codes kann in keinem der betrachteten Lesekanäle empfohlen werden. Die Codes haben sich als sehr empfindlich gegenüber Fehlerfortpflanzung und Transitionsrauschen erwiesen und zeigen dabei einen ausgeprägten Error Floor. Da der Fokus dieser Arbeit auf der Suche nach einem Soft-Decoding Kanal mit geringer Komplexität lag, wurde der Detektor nicht in die Iterationen des Turbo Codes einbezogen (Turbo Equalization [DJB95]). Die in der Literatur angegebene Verbesserung von rund 0,5 dB hätte aber ohnehin nicht zu grundlegend anderen Ergebnissen geführt.

In Verbindung mit Low-Density Parity-Check (LDPC) Codes hingegen ist das Phänomen des Error Floors nicht aufgetreten. Die Codes arbeiten insbesondere mit dem ME²PR4-PRML Detektor sehr gut zusammen. Selbst in Verbindung mit sehr hochratigen LDPC Codes kann mit diesem Detektor die Zuverlässigkeit von Reed-Solomon Codes erreicht

werden. Dies ist sowohl auf unbeschränkten Kanälen, als auch in Verbindung mit dem in dieser Arbeit präsentierten Soft-In/Soft-Out MTR-Modulationsdecoder der Fall. LDPC Codes erreichen mit den beiden anderen Detektoren jedoch nicht die gleiche hohe Zuverlässigkeit wie Reed-Solomon Codes. Ein weiterer Vorteil von LDPC-Codes ist die Tatsache, dass ihre Zuverlässigkeit unter Transitionsrauschen nicht so stark einbricht wie bei Reed-Solomon Codes.

Werden LDPC Codes mit einer leicht abgesenkten Coderate eingesetzt, verbessern sich die Resultate noch einmal deutlich. Da die Decodierleistung des Weiteren mit der Blocklänge der Codes ansteigt [RU98, PS02], könnten die Codes vor allem dann interessant werden, wenn die Festplattenindustrie zu größeren Sektoren als den heute üblichen 512 Byte übergeht. In Verbindung mit einem MTR-codierten ME²PR4-Kanal ist ein sehr leistungsfähiges Soft-Decoding System möglich, dessen Komplexität allerdings dann auch deutlich höher liegt. Der Fokus dieser Arbeit besteht jedoch nicht in der Suche nach einem optimalen System unter der Vernachlässigung der Komplexität, so dass dieser Ansatz nicht weiter verfolgt wurde. Außerdem müssten längere LDPC Codes auch mit Reed-Solomon Codes mit einer größeren Symbolgröße von zum Beispiel 12 Bit verglichen werden, die in dieser Arbeit aus Gründen der Hardware-Komplexität ebenfalls nicht betrachtet wurden.

Das in dieser Arbeit vorgestellte Verfahren zur Soft-Decodierung von Reed-Solomon Codes mit Hilfe eines Erasure-markierenden Parity-Check Codes kann hingegen zu einer sehr effizienten Steigerung der Zuverlässigkeit beitragen. Bei einem hohen SNR, das jedoch in einem für Festplatten realistischen Bereich liegt, kann das Optimum einer Verdopplung der Decodierleistung nahezu erreicht werden. Dieses System bietet mit einem Soft-Output Signalraumdetektor eine hervorragende Decodierleistung und kann zudem sehr effizient implementiert werden. Die Unempfindlichkeit dieses Kanals gegenüber Transitionsrauschen ist für den Einsatz in zukünftigen Lesekanälen mit einer höheren Datendichte ein weiterer wichtiger Vorteil. Die Fehlermuster des Signalraumdetektors erlauben im Gegensatz zu den PRML Detektoren bereits bei einem Parity-Code mit der Coderate $\frac{k}{k+1}$ einen zuverlässigen Betrieb, so dass auch der Verlust durch die Coderate gering gehalten werden kann.

Gleichzeitig dient der innere Parity-Check Code dabei nicht nur zur Markierung von Erasures, sondern darüber hinaus als einfacher Modulationscode. Somit kann ein Minimum an Timing-Information gewährleistet werden, das zur Taktrückgewinnung auf dem magnetischen Kanal unerlässlich ist. Der Verzicht auf einen MTR-Code ist insbesondere beim Signalraumdetektor weniger kritisch, da der Detektor im Gegensatz zu den PRML Detektoren kaum von diesem profitiert.

Wenn der Signalraumdetektor als adaptives Perzeptronen-Netzwerk aufgebaut wird, das eine automatische Anpassung an zeitvariante Kanalparameter ermöglicht, lässt sich der Parity-Check Code auch dort verwenden. Der Code wird dann bereits im Detektor decodiert und dient einem online-adaptiven Prozess als Indikator für eine schlechte Konstellation der Ebenen im Signalraum. In diesem Fall ist die Erhöhung der Hardware-Komplexität im Detektor jedoch größer als bei dem statischen Detektor.

Kapitel 8

Zusammenfassung und Ausblick

In dieser Arbeit wurden verschiedene Lesekanäle für Festplatten untersucht. Die betrachteten Kanäle sind für eine Anwendung in mobilen Festplatten vorgesehen. Aus diesem Grund lag ein Schwerpunkt der Betrachtung auf der Komplexität der eingesetzten Algorithmen und Schaltungen. Allerdings sind die Anforderungen in Bezug auf die garantierte Fehlerrate bei mobilen Festplatten genauso hoch wie etwa bei Desktop-Modellen. Da auch bei mobilen Festplatten ähnlich hohe Datendichten erreicht werden, ist auch hier eine sehr leistungsfähige Signalverarbeitung notwendig.

In Zukunft ist von weiter steigenden Datendichten auszugehen, so dass neue Innovationen erforderlich sind. Im Bereich der Signalverarbeitung haben insbesondere seit der Vorstellung der Turbo Codes [BGT93] viele Soft-Decoding Verfahren eine große Aufmerksamkeit erlangt. Diese ermöglichen sehr niedrige Fehlerraten durch die Generierung und Weiterverarbeitung einer so genannten Zuverlässigkeitsinformation für jedes Bit.

Soft-Decoding wurde auch schon für die Verwendung auf einem magnetischen Lesekanal in Betracht gezogen. Allerdings sind dazu aufwändige Modifikationen an allen signalverarbeitenden Komponenten des Lesekanals notwendig: Neben einem Detektor, der die Soft-Outputs berechnet, müssen auch der Modulationsdecoder und der Fehlerschutzdecoder zur Weiterverarbeitung von Zuverlässigkeitsinformation modifiziert werden. Des Weiteren sind die eingesetzten Decodieralgorithmen oftmals sehr komplex. Die Algorithmen verwenden häufig probabilistische Verfahren in Verbindung mit einem iterativ ablaufenden Decodierprozess. Sofern sich die Soft-Decoding Kanäle nicht effizient in Hardware implementieren lassen, sind sie insbesondere für den mobilen Einsatz nicht geeignet.

In dieser Arbeit wurden verschiedene Soft-Decoding Lesekanäle konstruiert und untersucht. Dazu mussten die einzelnen Komponenten teilweise angepasst oder um die Ausgabe/Verarbeitung von Zuverlässigkeitsinformation erweitert werden. Die Basis der Untersuchungen bildet ein Signalraumdetektor, der eine hohe Zuverlässigkeit mit einer effizienten Implementierung ermöglicht [BM98, SHS00]. Der Detektor wurde in dieser Arbeit um die Ausgabe von Zuverlässigkeitsinformation erweitert. Der dabei entstandene Soft-Output Signalspace Detector (S^3D) kann mit einem äußerst geringen Zusatzaufwand implementiert werden. Als Vergleich wurden die bei Festplatten häufig eingesetzten

PRML-Detektoren mit ME^2PR4 bzw. $EPR4$ Target herangezogen.

Basierend auf diesen Detektionsprinzipien wurden zwei verschiedene Konzepte für einen Soft-Decoding Kanal verfolgt. Zum einen wurden die vielfach im Bereich von Datenspeichern anzutreffenden Reed-Solomon Codes verwendet. Die Zuverlässigkeitsinformation wird in diesem Fall mit Hilfe eines zusätzlichen, inneren Parity-Check Codes ausgenutzt. Dieser Code wird vor dem Reed-Solomon Code decodiert und zur Identifikation von Fehlerpositionen verwendet. Wenn die Position fehlerhafter Codesymbole bekannt ist, kann die Redundanz des Reed-Solomon Codes durch Erasure-Decodierung effizienter ausgenutzt und dadurch die Fehlerrate weiter abgesenkt werden. Zusätzlich dient der Parity-Check Code als einfacher Modulationscode, der die erforderliche Timing-Information im Lesesignal gewährleistet.

Durch den in dieser Arbeit verfolgten Ansatz wird der Lesekanal systemübergreifend optimiert. Im Detektor müssen die Soft-Outputs bestimmt werden. Dies erfordert beim Signalraumdetektor jedoch keine aufwändigen Modifikationen. Des Weiteren muss der Reed-Solomon Decoder zur Verarbeitung von Erasures erweitert werden. Dazu ändert sich jedoch nur die Initialisierung des Fehlerstellenpolynoms, so dass auch hier kein großer Zusatzaufwand notwendig ist. Auf der anderen Seite kann der üblicherweise eingesetzte Modulationscode durch den sehr einfachen Parity-Check Code ersetzt werden. Somit wird für die Implementierung eines Soft-Decoding Kanals die Komplexität an einigen Stellen moderat erhöht und an anderen Stellen sogar verringert. Insgesamt wurde durch diese Maßnahme eine sehr effiziente Implementierung erreicht. Des Weiteren kann der Parity-Check Code bei Verwendung eines adaptiven Signalraumdetektors zusätzlich als Indikator für eine schlechte Adaption eingesetzt werden.

Das zweite in dieser Arbeit betrachtete Konzept basiert auf der direkten Verarbeitung von Zuverlässigkeitsinformation im Fehlerschutzdecoder. Zu diesem Zweck wurden sowohl Turbo Codes, als auch Low-Density Parity-Check (LDPC) Codes untersucht. Beide Codes mussten zuvor hinsichtlich der Codelänge und der Coderate an den magnetischen Kanal angepasst werden. Da beide Verfahren Soft-Information direkt verarbeiten, ist ein Parity-Check Code zur Identifikation von Fehlern hier nicht notwendig. Statt dessen wurde in dieser Arbeit ein Soft-In/Soft-Out Modulationsdecoder am Beispiel eines MTR-Codes konzipiert [SS04]. Dieser verwendet modifizierte logische Gatter, die die Zuverlässigkeitsinformation direkt weiterverarbeiten können. Somit ergibt sich die Struktur und die Komplexität des Soft-Decoders direkt aus dem Hard-Decoder. Das vorgestellte Prinzip ist für alle Modulationscodes anwendbar, die sich über ein System von Booleschen Gleichungen beschreiben lassen.

Bei der Evaluation der beiden Konzepte wurde deutlich, dass der Lesekanal der Festplatte als Ganzes betrachtet werden muss und eine allgemeingültige Aussage für Einzelkomponenten nicht möglich ist. So hängt es beispielweise nicht nur vom eingesetzten Detektor, sondern auch vom fehlerkorrigierenden Code ab, ob sich ein MTR-Code positiv auf die Fehlerrate auswirkt. Anders herum kann ein in Verbindung mit Reed-Solomon Codes überlegenes Detektionsprinzip nach dem Austausch des fehlerkorrigierenden Codes plötzlich zu sehr schlechten Ergebnissen führen.

Das Konzept eines systemübergreifend eingesetzten Parity-Check Codes führt insbesondere mit dem Signalraumdetektor zu sehr guten Ergebnissen. Die Fehlerverteilung

dieses Detektors erlaubt eine sehr zuverlässige Identifikation von Detektionsfehlern. Da der Signalraumdetektor auf unbeschränkten Kanälen deutlich robuster gegenüber Transitionsrauschen ist als die PRML-Detektoren, bietet sich die Anwendung auf künftigen magnetischen Kanälen an. In Verbindung mit Reed-Solomon Codes kann das Optimum einer Verdopplung der Korrekturfähigkeit durch Soft-Decoding fast erreicht werden. Ein Kanal mit ME²PR4-PRML Detektor bietet hier in etwa die gleiche Zuverlässigkeit, allerdings ist dazu ein erheblich höherer Hardware-Aufwand notwendig.

Werden hingegen LDPC Codes betrachtet, zeigen sich die PRML-Detektoren dem Signalraumkonzept überlegen. Offenbar wirkt sich hier die Fehlerfortpflanzung aufgrund der Feed-Back Struktur in der Detektorarchitektur negativ aus. Speziell der ME²PR4-PRML Detektor bietet hier im Vergleich sowohl auf unbeschränkten Kanälen, als auch in Verbindung mit dem Soft-In/Soft-Out MTR-Decoder sehr gute Ergebnisse, die sich auch unter dem Einfluss von Transitionsrauschen nur um rund 1 dB verschlechtern. Dennoch kann die Fehlerrate von Reed-Solomon Codes mit Hard-Decoding trotz deutlich höherem Aufwand nicht unterboten werden.

Beim Einsatz von Turbo Codes tritt in allen betrachteten Konstellationen ein Error Floor auf, der sich bei Transitionsrauschen sogar noch verstärkt. Die extrem niedrigen Fehlerraten, die auf dem magnetischen Kanal notwendig sind, erscheinen mit diesen Codes daher nicht erreichbar. Da der Anteil von Transitionsrauschen am Rauschspektrum mit der Datendichte weiter zunimmt [Str00], sind Turbo Codes auch in Zukunft für den magnetischen Kanal nicht zu empfehlen.

Somit bleiben Reed-Solomon Codes derzeit die erste Wahl für einen Festplatten-Lesekanal. Das in dieser Arbeit vorgestellte Konzept zur Soft-Decodierung mittels einer systemübergreifenden Optimierung bietet eine sehr einfache und wirkungsvolle Möglichkeit zur Erhöhung der Zuverlässigkeit. Insbesondere in Verbindung mit einem Signalraumdetektor ist ein solcher Kanal sehr effizient zu implementieren.

Der Einsatz von LDPC-Codes führt nur in Verbindung mit einem ME²PR4-PRML Detektor und dem in dieser Arbeit entstandenen Soft-In/Soft-Out Modulationsdecoder zu sehr guten Ergebnissen. Die Decodierleistung von Reed-Solomon Codes wird jedoch nicht unterboten. Bei größeren Sektoren und somit längeren Codes, wie sie von Festplattenherstellern derzeit diskutiert werden, könnten sich die Relationen jedoch zugunsten der LDPC Codes verschieben. Bei künftigen Untersuchungen müssten diese dann allerdings mit Reed-Solomon Codes verglichen werden, die eine größere Symbolgröße verwenden.

Literaturverzeichnis

- [AAM01] K. Anim-Appiah and S.W. McLaughlin. Turbo Codes Cascaded with High-Rate Block Codes for $(0, k)$ -Constrained Channels. *IEEE Journal on Selected Areas in Communications*, 19(4):677–685, April 2001.
- [ACH83] R. L. Adler, D. Coppersmith, and M. Hassner. Algorithms for Sliding Block Codes. *IEEE Trans. Inf. Th.*, 29(1):5–22, January 1983.
- [ANB⁺94] T. C. Anthony, S. L. Naberhuis, J. A. Brug, M. J. K. Bhattacharyya, L. T. Trans, V. W. Hesterman, and G. G. Lopatin. Dual-Stripe Magnetoresistive Heads for High-Density Recording. *IEEE Trans. Magn.*, 30(2):303–308, March 1994.
- [Ash97] K. G. Ashar. *Magnetic Disk Drive Technology: Heads, Media, Channel, Interfaces, and Integration*. IEEE Press, New York, 1997.
- [BCJR74] L.R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate. *IEEE Trans. Inf. Theory*, IT-20:284–287, Mar. 1974.
- [BDG79] G. Battail, M.C. Decouvelaere, and P. Godlewski. Replication Decoding. *IEEE Transactions on Information Theory*, 25:332–345, May 1979.
- [BDMP98] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara. Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding. *IEEE Transactions on Information Theory*, 44:909–926, 1998.
- [Ber93] C. Berrou. A Low-Complexity Soft Output Viterbi Decoder Architecture. In *Proc. of IEEE Int. Conf. on Communications '93*, pages 737–740, Geneva, Switzerland, 1993.
- [Ber94] H. N. Bertram. *Theory of Magnetic Recording*. Cambridge University Press, 1994.
- [Ber96] J. W. M. Bergmans. *Digital Baseband Transmission and Recording*. Kluwer Academic Publishers, Dordrecht, 1996.
- [BGM85] N. R. Belk, P. K. George, and G. S. Mowry. Noise in High Performance Thin-film Longitudinal Magnetic Recording Media. *IEEE Trans. Magn.*, 21(5):1350–1355, September 1985.

- [BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon Limit Error Correcting Coding and Decoding: Turbo-Codes. In *Proc. of IEEE Int. Conf. on Communications '93*, pages 1064–1070, Geneva, Switzerland, May 1993.
- [BH02] A.J. Blanksby and C.J. Howland. A 690-mW 1-Gb/s 1024-b, rate-1/2 Low-Density Parity-Check Code Decoder. *IEEE Journal of Solid-State Circuits*, 37(3):404–412, March 2002.
- [Bli81] W.G. Bliss. Circuitry for performing error correction calculations on base-band encoded data to eliminate error propagation. *IBM Tech. Discl. Bull.*, 23:4633–4634, 1981.
- [BM96a] S. Benedetto and G. Montorsi. Unveiling turbo codes: some results on parallel concatenated coding schemes. *IEEE Trans. Inform. Theory*, pages 409–428, 1996.
- [BM96b] B. Brickner and J. Moon. A High Dimensional Signal Space Implementation of FDTS/DF. *IEEE Trans. Magn.*, 32(5):3941–3943, September 1996.
- [BM97a] B. Brickner and J. Moon. Architectures for the Implementation of a Fixed Delay Tree Search Detector. *IEEE Trans. Magn.*, 33(2):1116–1124, March 1997.
- [BM97b] B. Brickner and J. Moon. Design of Efficient Rate $m/(m+1)$ MTR (2;k) Modulation Codes. Technical report, Communication and Data Storage Lab, University of Minnesota, 1997.
- [BM98] B. Brickner and J. Moon. Low Complexity Signal Space Detection for MTR Coded Channels. *IEEE Trans. Magn.*, 34(1):104–109, January 1998.
- [BMN83] R. A. Baugh, E. S. Murdock, and B. R. Natarajan. Measurement of noise in magnetic media. *IEEE Trans. Magn.*, 19(5):1722–1724, September 1983.
- [Bos99] M. Bossert. *Channel Coding for Telecommunications*. Wiley, 1999.
- [Bri98] B.J. Brickner. *Maximum Transition Run Coding and Pragmatic Signal Space Detection for Digital Magnetic Recording*. PhD thesis, University of Minnesota, 1998.
- [BvV91] P. G. Blanken and J. J. L. M. van Vlerken. Modeling of Electromagnetic Systems. *IEEE Trans. Magn.*, 27(6):4509–4515, November 1991.
- [CDH⁺92] R. D. Cideciyan, F. Dolivo, R. Hermann, W. Hirt, and W. Schott. A PRML System for Digital Magnetic Recording. *IEEE Journal on Selected Areas in Communications*, 10(1):38–56, January 1992.
- [Che01] Y. Chen. Application of Iterative BCJR Decoding for the Magnetic Recording Channel. Master's thesis, University of Dortmund, May 2001.
- [CHIW98] D.J. Costello, J. Hagenauer, H. Imai, and S.B. Wicker. Applications of Error Control Coding. *IEEE Transactions on Information Theory*, 44(6):2531–2560, October 1998.

- [Con98] T. Conway. A New Target Response with Parity Coding for High Density Magnetic Recoding Channels. *IEEE Transactions on Magnetics*, 34(4):2382–2386, July 1998.
- [DJB95] C. Douillard, M. Jezequel, and C. Berrou. Iterative Correction of Inter-symbol Interference: Turbo-Equalization. *European Transactions on Communications*, 6(5):507–511, 1995.
- [DS03] I. Dahm and S. Schmermbeck. A Fully Adaptive Signalspace Detector with Soft-Information (ASSSD). In *Proceedings of the International Conference on Communications (ICC)*, May 2003.
- [DSS] I. Dahm, S. Schmermbeck, and G. Stromberg. Hard Disk Simulation Environment, HdSim. Computer Engineering Institute, University Dortmund, <http://www-ds.e-technik.uni-dortmund.de/>.
- [DSS00] I. Dahm, G. Stromberg, and U. Schwiegelshohn. Leistungsarme Signalverarbeitung für Festplattenlesekanäle. In *IGT Workshop Mikroelektronik für die Informationstechnik*, volume 162, pages 269–274, 2000.
- [DSS02] I. Dahm, S. Schmermbeck, , and U. Schwiegelshohn. Using Reliability Information in Magnetic Recording Systems. In *Proceedings of the 3rd VIVA Colloquium*, March 2002.
- [FC98] J.L. Fan and A.R. Calderbank. A Modified Concatenated Coding Scheme with Application to Magnetic Data Storage. *IEEE Transactions on Information Theory*, 44(4):1565–1574, July 1998.
- [FFKM99] J.L. Fan, A. Friedmann, E. Kurtas, and S. McLaughlin. Low density parity check codes for magnetic recording. In *Proceedings of the Allerton Conference on Control, Communications and Computing*, 1999.
- [For66] G.D. Forney. Generalized Minimum Distance Decoding. *IEEE Trans. Inf. Theory*, IT-12(2):125–131, Apr. 1966.
- [For73] G. D. Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973.
- [Fos99] M. Fossorier. Reduced Complexity Iterative Decoding of Low-Density Parity-Check Codes Based on Belief Propagation. *IEEE Transaction on Communications*, pages 673–680, May 1999.
- [Gal62] R.G. Gallager. Low-Density Parity-Check Codes. *IRE Trans. Inform. Theory*, 8:21–28, Jan 1962.
- [GH96] E. Grochowski and R. F. Hoyt. Future Trends in Hard Disk Drives. *IEEE Trans. Magn.*, 32(3):1850–1854, May 1996.
- [GKP92] R.L. Galbraith, G.J. Kerwin, and J.M. Poss. Magneto-resistive head thermal asperity digital compression. *IEEE Transactions on Magnetics*, 28(5):2731–2732, 1992.

- [Gro] E. Grochowski. IBM Magnetic Hard Disk Drive Technology. Almaden Research Center, <http://www.storage.ibm.com/hdd/technolo/grochows/grocho01.htm>.
- [GZ61] D. Gorenstein and N. Zierler. A Class of Error-Correcting Codes in p^m . *Journal of the Society of Industrial and Applied Mathematics*, 9:207–214, 1961.
- [Hag99] J. Hagenauer. The Turbo Principle for Decoding of Concatenated Codes. In *Proceedings of the IEEE International Workshop on Concatenated Codes*, Ulm, Germany, 1999.
- [HB01] C. Howland and A. Blanksby. Parallel Decoding Architectures for Low Density Parity Check Codes. In *IEEE Int. Symposium on Circuits and Systems*, volume 4, pages 742–745, 2001.
- [Hee98] C. Heegard. Turbo-Coding for Magnetic Recoding. In *IEEE Information Theory Workshop*, pages 18–19, San Diego, CA, 1998.
- [HH89] J. Hagenauer and P. Höher. A Viterbi Algorithm with Soft-Decision Outputs and its Applications. In *Proc. GLOBECOM '89*, pages 47.1.1–47.1.7, Dallas, Texas, 1989.
- [HM03] J. Ha and S.W. McLaughlin. Optimal Puncturing of Irregular Low-Density Parity-Check Codes. In *Proceedings of the International Conference on Communications (ICC)*, May 2003.
- [HOP96] J. Hagenauer, E. Offer, and L. Papke. Iterative Decoding of Binary Block and Convolutional Codes. *IEEE Trans. Inf. Theory*, IT-42:429–445, March 1996.
- [HSW95] M. Hassner, U. Schwegelshohn, and S. Winograd. On-The-Fly Error Correction in Data Storage Channels. *IEEE Trans. Magn.*, 31(2):1149–1154, March 1995.
- [Hun71] R. P. Hunt. A magnetic readout transducer. *IEEE Trans. Magn.*, 7(1):150–154, 1971.
- [HW99] C. Heegard and S.B. Wicker. *Turbo Coding*. Kluwer Academic Publishers, 1999.
- [Kam97] N. Kamiya. On Acceptance Criterion for Efficient Successive Errors-and-Erasures Decoding of Reed-Solomon and BCH Codes. *IEEE Trans. Inf. Theory*, 43(5):1477–1488, 1997.
- [KV02] R. Koetter and A. Vardy. Soft Decoding of Reed Solomon Codes and Optimal Weight Assignment. In *Proc. Int. Conf. on Source and Channel Coding, Berlin*, January 2002.
- [KWM91] K.J. Knudson, J. Wolf, and L.B. Milstein. Producing Soft-Decision Information at the Output of a Class IV Partial Response Viterbi Detector. In *Proc. of IEEE Int. Conf. on Communications '91*, pages 820–824, 1991.

- [LC83] S. Lin and D.J. Costello. *Error Control Coding: Fundamentals and Applications*. Prentice-Hall, Englewood Cliffs, 1983.
- [Mac99] D.J.C. MacKay. Good Error-Correcting Codes based on very Sparse Matrices. *IEEE Transactions on Information Theory*, 45, 1999.
- [MB96] J. Moon and B. Brickner. Maximum Transition Run Codes for Data Storage Systems. *IEEE Trans. Magn.*, 32(5):3992–3994, September 1996.
- [MD95a] C.D. Mee and E.D. Daniel, editors. *Magnetic Recording Technology*. McGraw-Hill, 2nd edition edition, 1995.
- [MD95b] C.D. Mee and E.D. Daniel, editors. *Magnetic Storage Handbook*. McGraw-Hill, 2nd edition edition, 1995.
- [MD98] D.J.C. MacKay and M. Davey. Low Density Parity Check Codes over GF(q). *IEEE Communication Letters*, 2(6):165–167, June 1998.
- [MD99] D.J.C. MacKay and M. Davey. Evaluation of Gallager Codes for Short Block Length and High Rate Applications. In *Proceedings of IMA Workshop on Codes, Systems and Graphical Models 1999*, volume 123, 1999.
- [MJ98] J. Moon and T. Jeon. Sequence Detection for Binary ISI Channels Using Signal-Space Partitioning. *IEEE Trans. Communications*, 46(7):891–901, July 1998.
- [MMC98] R.J. McEliece, D.J.C. MacKay, and J-F. Cheng. Turbo Decoding as an Instance of Pearl’s Belief Propagation Algorithm. *IEEE Journal on Selected Areas in Commun.*, 16(2):140–152, 1998.
- [MMH98] L. McPheters, S.W. McLaughlin, and E.C. Hirsch. Turbo Codes for PR4 and EPR4 Magnetic Recording. In *1998 Conf. on Information Sciences and Systems*, pages 1778–1782, Princeton,NJ, 1998.
- [MMN99] L. McPheters, S. McLaughlin, and K. Narayanan. Precoded PRML, Serial Concatenation and Iterative (Turbo) Decoding for Magnetic Recording. In *Proceedings of the International Conference on Magnetism*, 1999.
- [Moh93] M. Moher. Decoding via Cross-Entropy Minimization. In *Proceedings of the IEEE Globecom Conference*, pages 809–813, December 1993.
- [NKRA96] M. Nasiri-Kenari, C.K. Rushforth, and A.D. Abbaszadeh. A Reduced-Complexity Sequence Detector with Soft Outputs for Partial-Response Channels. *IEEE Trans. Comm.*, 44(12):1616–1619, Dec. 1996.
- [OM01] T.R. Oenning and J. Moon. Low Density Parity Check Coding for Magnetic Recording Channels with Media Noise. In *Int. Conference on Communications (ICC)*, pages 2189–2193, 2001.
- [Pat91] A. M. Patel. A New Digital Signal Processing Channel for Data Storage Products. *IEEE Trans. on Magn.*, 27(6):4579 – 4584, November 1991.

- [Pla01] J. Platte. Anpassung von LDPC Codes für den magnetischen Kanal. Master's thesis, University of Dortmund, 2001.
- [Pro95] J.G. Proakis. *Digital Communications*. McGraw-Hill, New York, 3rd edition, 1995.
- [PS02] J. Platte and S. Schmermbeck. Anpassung von Low Density Parity Check (LDPC) Codes an den magnetischen Kanal. Technical report, Universität Dortmund, 2002.
- [PSC96] L.C. Perez, J. Seghers, and D.J. Costello. A Distance Spectrum Interpretation of Turbo-Codes. *IEEE Transactions on Information Theory*, 42(6):1698–1709, 1996.
- [RMAAY00] W. Ryan, S.W. McLaughlin, K. Anim-Appiah, and M. Yang. Turbo, LDPC, and RLL Codes in Magnetic Recording. In *Proceedings of 2nd International Symposium on Turbo-Codes and Related Topics*, September 2000.
- [RMM98] W. Ryan, L. McPheters, and S. McLaughlin. Combined Turbo Coding and Turbo Equalization for the PR4 Equalized Lorenzian Channel. In *Proc. Information Sciences and Systems*, 1998.
- [RS60] I. S. Reed and G. Solomon. Polynomial Codes over certain Finite Fields. *J Soc. Indust. Appl. Math.*, 8(2), June 1960.
- [RU98] T. Richardson and R. Urbanke. The Capacity of Low-Density Parity-Check Codes Under Message Passing Decoding. *IEEE Transactions on Information Theory*, 1998.
- [RVH95] P. Robertson, E. Vilebrun, and P. Höher. A Comparison of Optimal and Suboptimal MAP Decoding Algorithms Operating in the Log Domain. In *Proc. of IEEE Int. Conf. on Communications '95*, pages 1009–1013, Seattle, WA, Jun 1995.
- [Rya98] W. Ryan. Performance of High Rate Turbo-Codes on a PR4-Equalized Magnetic Recording Channel. In *Proc. 1998 Int. Conf. on Comm.*, pages 947–951, Atlanta,GA, 1998.
- [Saw98] E. Sawatzky. Thermal Asperities: MR Heads Face New Dangers. *Data Storage Magazine*, February 1998.
- [SFO⁺99] T. Souvignier, A. Friedmann, M. Öberg, P. Siegel, R. Swanson, and J. Wolf. Turbo Decoding for PR4: Parallel versus Serial Concatenation. *Proc. 1999 Int. Conf. Comm.*, 1999.
- [Sha48] C.E. Shannon. A Mathematical Theory of Communication. *Bell Systems Technical Journal*, 27:379–423 and 623–656, 1948.
- [SHS00] G. Stromberg, M. Hassner, and U. Schwiegelshohn. Signal Space Detection in Colored Noise. *IEEE Trans. Magn.*, 36(3):604–612, May 2000.

- [Sie85] P. H. Siegel. Recording Codes for Digital Magnetic Storage. *IEEE Trans. Magn.*, 21(5):1344–1349, September 1985.
- [SMO01] B. Steingrímsson, J. Moon, and T. Oenning. Signal Space Detection for DVD Optical Recording. *IEEE Transactions on Magnetics*, 37(2):670–675, March 2001.
- [SS04] S. Schmermbeck and G. Stromberg. Soft-In/Soft-Out Modulations Codes. Technical report, University of Dortmund, 2004.
- [SSHS03] S. Schmermbeck, G. Stromberg, M. Hassner, and U. Schwiegelshohn. Low-Complexity Signal Processing for ISI-Channels. In *Proceedings of the Globecom Conference 2003*, December 2003.
- [STC01] H. Song, R.M. Todd, and J.R. Cruz. Applications of Low-Density Parity-Check Codes to Magnetic Recording Channels. *IEEE Journal on Selected Areas in Communications*, 19(5):918–923, May 2001.
- [Ste81] L. D. Stevens. The Evolution of Magnetic Storage. *IBM Journal of Research and Development*, 25:663, 1981.
- [Ste02] B. Steingrímsson. Soft Signal Space Detection for the Lorentzian Magnetic Recording Channel. *IEEE Transactions on Magnetics*, 38(5), September 2002.
- [Str00] G. Stromberg. *Signal Space Detection with Application to Magnetic Recording*. PhD thesis, University of Dortmund, September 2000.
- [SW91] P. H. Siegel and J. K. Wolf. Modulation and Coding for Information Storage. *IEEE Communications Magazine*, 29(12):68–86, December 1991.
- [Tan81] R.M. Tanner. A Recursive Approach to Low Complexity Codes. *IEEE Transactions on Information Theory*, pages 533–547, September 1981.
- [Tar96] A. Taratorin. *Characterization of Magnetic Recording Systems*. Guzik Technical Enterprises, 1996.
- [TB00] D.A. Thompson and J.S. Best. The Future of Magnetic Data Storage Technology. *IBM J. Res. Develop.*, 44(3):311–322, May 2000.
- [tB01] S. ten Brink. Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes. *IEEE Transactions on Communications*, 49(10):1727–1737, October 2001.
- [Tex98] Texas Instruments, Storage Products Group. *SSI 32P4938A PR4ML Read Channel with 16/17 ENDEC w/Thermal Asperity Suppression*, rev 1 edition, March 1998.
- [TGV⁺02] M.J. Thul, F. Gilbert, T. Vogt, G. Kreiselmaier, and N. Wehn. A Scalable System Architecture for High-Throughput Turbo-Decoders. In *Proc. 2002 Workshop on Signal Processing Systems (SiPS 2002)*, pages 152–158, September 2002.

- [Tho75] D. Thompson. Magnetomotive Transducers in High-Density Magnetic Recording. In *AIP Conf. Proc.*, volume 24, pages 528–533, 1975.
- [TVGW02] M. Thul, T. Vogt, F. Gilbert, and N. Wehn. Evaluation of Algorithm Optimizations for Low-power Turbo-Decoder Implementations. In *Proc. 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3101–3104, May 2002.
- [TWR02] M. Thul, N. Wehn, and L. Rao. Enabling High-Speed Turbo-Decoding through Concurrent Interleaving. In *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 897–900, May 2002.
- [Vit67] A. J. Viterbi. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Trans. Inf. Th.*, 13(2):260–269, April 1967.
- [VP03] B. Vasic and K. Pedagani. A Runlength Limited Low-Density Parity-Check Coding Scheme. In *Proceedings of the International Conference on Communications (ICC)*, May 2003.
- [WB94] S.B. Wicker and V.K. Bhargava. *Reed-Solomon Codes and their Applications*. IEEE Press, 1994.
- [Wib96] N. Wiberg. *Codes and Decoding on General Graphs*. PhD thesis, Linköping University, 1996.
- [Wic95] S.B. Wicker. *Error Control Systems for Digital Communication and Storage*. Prentice Hall, Englewood Cliffs, New Jersey, 1995.
- [WLW01] A. Worm, H. Lamm, and N. Wehn. Design of Low-Power High-Speed Maximum a Posteriori Architectures. In *Proceedings of the Design, Automation and Test in Europe (DATE) Conference*, pages 258–265, March 2001.
- [WMK⁺00] A. Worm, H. Michel, G. Kreiselmaier, M. Thul, and N. Wehn. Advanced Implementation Issues of Turbo-Decoders. In *Proc. 2nd International Symposium on Turbo-Codes and Related Topics*, pages 351–354, September 2000.
- [WT99] S.X. Wang and A.M. Taratorin. *Magnetic Information Storage Technology*. Academic Press, 1999.
- [WWH90] R. W. Wood, M. Williams, and J. Hong. Considerations for High Data Rate Recording with Thin-Film-Heads. *IEEE Trans. Magn.*, 26(6):2654–2959, November 1990.