Parallele Prozessorarchitektur für die Raum-Zeit-Entzerrung in breitbandigen Funksystemen mit adaptiven Gruppenantennen

Von der Fakultät für Elektrotechnik und Informationstechnik der Rheinisch-Westfälischen Technischen Hochschule Aachen zur Erlangung des akademischen Grades eines Doktors der Ingenieurwissenschaften genehmigte Dissertation

vorgelegt von

Diplom-Ingenieur
Lars Brühl

aus Euskirchen

Berichter: Universitätsprofessor Dr.-Ing. Bernhard Rembold
Universitätsprofessor Dr.-Ing. Gerd Ascheid

Tag der mündlichen Prüfung: 8. Juli 2005

Berichte aus der Hochfrequenztechnik

Lars Brühl

Parallele Prozessorarchitektur für die Raum-Zeit-Entzerrung in breitbandigen Funksystemen mit adaptiven Gruppenantennen

D 82 (Diss. RWTH Aachen)

Shaker Verlag
Aachen 2005

Bibliografische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.ddb.de abrufbar.

Zugl.: Aachen, Techn. Hochsch., Diss., 2005

Printed in Germany.

ISBN 3-8322-4523-5
ISSN 0945-0793

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen
Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9
Internet: www.shaker.de • eMail: info@shaker.de

VORWORT

Diese Arbeit entstand im Rahmen meiner Tätigkeit am Institut für Hochfrequenztechnik der RWTH Aachen. Ich danke meinem Doktorvater, Herrn Prof. Dr.-Ing. B. Rembold, für die Betreuung auf dem Weg zur Promotion, die freie Entfaltungsmöglichkeit in die digitale Welt und die mir gewährten Einblicke in die unergründlichen Geheimnisse der Hochfrequenztechnik. Herrn Prof. Dr.-Ing. G. Ascheid danke ich für sein Interesse an meiner Arbeit und die Übernahme des Korreferates.

Das freundliche Klima am Institut hat mir die Arbeit erleichtert und zur Freude gemacht. Dafür möchte ich mich bei allen aktiven und ehemaligen Mitarbeitern bedanken. Besonderer Dank gilt meinen Projekt- und Stammtischkollegen Christoph Walke, Wilhelm Keusgen, Christoph Degen, Reza Alirezaei, Christos Oikonomopoulos und Frank Geschewski für den leidenschaftlichen Einsatz im Kampf gegen hartnäckige Zombies beim Bau des Demonstrators. Bei der Umsetzung des Parallelprozessors hat Daniel Borkowski zunächst in seiner Diplomarbeit und anschließend als Freund und Kollege wesentliche Beiträge geleistet. Herzlichen Dank für die hervorragende Teamarbeit und viel Erfolg für die Weiterentwicklung des Prozessors, ich weiß ihn in guten Händen.

Meinem Ziehvater Jann-Erik Dietert danke ich für die Unterstützung bei meinen ersten Schritten während der Diplomarbeit, meinen späteren Kollegen Wolfgang Fischer und Michael Wenzel dafür, dass ich vor lauter Arbeit auch das Leben nicht vergaß. Vladimir Blazek verdanke ich die herzliche Aufnahme in der Gruppe Optik und die unvergesslichen Reisen zur Posterkonferenz in Prag. Mein Dank gilt auch Dominik Hölscher, der guten Seele des Instituts, ohne den die Mühlen nicht mahlen würden, Olivier Koch für's tapfere Korrekturlesen und Guido Companie, Kai Wendlandt, Martin Bonczkowitz, Claude Boden, Luigi Lescarini und George Varghese, die mit ihren Diplomarbeiten wichtige Grundsteine für meine Arbeit gelegt haben.

Schließlich möchte ich mich auch bei meiner Familie bedanken, besonders bei meinen Eltern Anne und Klaus für die Aufopferung und Unterstützung auf meinem bisherigen Weg, meinem älteren Bruder Jens, der mir als Vorbild zu jeder Zeit mit Plan, Rat und Tat zur Seite stand, und nicht zuletzt bei meiner Oma Klara, die mich mit Ihrem Obulus und zahllosen Fahrdiensten durch's Studium gebracht hat.

Ulm, im September 2005 Lars Brühl

INHALTSVERZEICHNIS

KAPITEL 1

Einleitung

1.1 Motivation

Der Übergang von der reinen Sprachübertragung zu breitbandigen Multimedia- und Datendiensten führt zu Kapazitätsengpässen in den bestehenden Mobilfunksystemen. Treibende Kraft in der Entwicklung ist das Internet: die zunehmende Verbreitung von Festnetzanschlüssen mit immer höheren Datenraten schafft neue Möglichkeiten für den Anwender und neue Geschäftsfelder für die Industrie. Der Handel mit Musik- und Videodaten ist bereits heute ein profitabler Markt mit enormen Steigerungsraten. So haben die Musikdownloads in den USA, UK und Deutschland im Jahre 2004 um den Faktor 10 zugenommen [1]. Der tägliche Umgang mit großen Datenmengen im Festnetz wird auch in Zukunft die Durchsatzforderungen an mobile Datendienste weiter antreiben. Die Übertragungskapazität der bestehenden Mobilfunksysteme wird dazu nicht ausreichen. Ursprünglich für den Bürobereich ausgelegte WLAN-Systeme nach dem IEEE802.11 Standard [2] drängen bereits massiv in den Privatkundenbereich und decken mit hohen Datenraten im Nahbereich ab, was die flächendeckenden Mobilfunksysteme nicht leisten können. Die Industrie versucht derzeit mit dem neuen Standard IEEE802.16 [3] unter dem Namen WiMAX die entstandene Lücke zwischen Verfügbarkeit und Datendurchsatz zu schließen.

Für die hohen Übertragungsraten steht nur ein begrenztes Frequenzspektrum zur Verfügung. Außerdem ist von Seiten der Politik und der Öffentlichkeit zunehmend mit der Forderung nach einer Reduktion der Strahlenbelastung zu rechnen. Als Schlüsseltechnologie zum Erreichen der erforderlichen hohen spektralen Effizienz ist MIMO (*Multiple Input, Multiple Output*) meist in Verbindung mit Multi-Hop-Netzen und hoch adaptiven Medienzugriffsverfahren in praktisch allen künftigen Standards vorgesehen. MIMO erhöht die Übertragungseffizienz durch Ausnutzung der räumlichen Selektivität des Funkkanals mit Hilfe von Mehrantennensystemen. Die räumlich selektive Abstrahlung erlaubt mit Hilfe von Relais-Stationen (multihop) eine Verbesserung der Funkabdeckung. Entscheidend ist die flexible Adaption des Übertragungsverfahrens an die momentanen Eigenschaften des Funkkanals sowie die momentanen Dienstanforderungen.

Notwendige Voraussetzung für den Erfolg drahtloser Kommunikationssysteme sind Wirtschaftlichkeit sowie ein geringer Platz- und Energieverbrauch der Endgeräte und Zugangspunkte. Diese Faktoren werden schon heute von der digitalen Signalverarbeitung dominiert. MIMO-Verfahren in Verbindung mit hohen Datenraten erfordern im Vergleich zu bisherigen Systemen eine wesentlich höhere Rechenleistung. Verfügbare Signalprozessoren stoßen hier schnell an ihre Grenzen. Eine wesentliche Erhöhung der erzielbaren Taktraten ist aus heutiger Sicht nicht zu erwarten. Daher kann die erforderliche Rechenleistung nur über eine massive Parallelisierung der Verarbeitung erreicht werden. Während spezifische Hardware-Lösungen eine hohe Rechenkapazität mit geringem Leistungsverbrauch und in großen Stückzahlen auch niedrige Herstellungskosten bieten, genügen sie nicht mehr den zukünftigen Anforderungen einer hohen Flexibilität des Medienzugriffs.

Es besteht also ein dringender Bedarf an programmierbaren Signalprozessoren, die für die gegebene Anwendung eine effiziente und hoch parallele Verarbeitung ermöglichen[1]. Voraussetzung für die Wirtschaftlichkeit ist die fortschreitende Verbesserung der Technologie für integrierte Schaltungen, wie von Moore bereits 1965 vorhergesagt [4] und bis heute im wesentlichen gültig. Nur durch eine weitere Erhöhung der Gatterdichte kann die erforderliche Rechenleistung künftig auf einer wirtschaftlichen Chipgröße mit akzeptablem Energieverbrauch realisiert werden.

Zielsetzung dieser Arbeit ist die Entwicklung einer parallelen Prozessorarchitektur für die Raum-Zeit-Signalverarbeitung in breitbandigen MIMO-Systemen. Die Architektur wird auf die potentiellen Anforderungen zukünftiger Standards ausgelegt und mit einer exemplarischen Umsetzung in der praktischen Anwendung verifiziert. Aufbauend auf dem nun folgenden Stand der Technik wird die Zielsetzung anschließend weiter eingegrenzt.

1.2 Stand der Technik

Am aktuellen Stand der Forschung und Technik soll im Folgenden aufgezeigt werden, welche Verfahren und Algorithmen für zukünftige Kommunikationssysteme zur breitbandigen Datenübertragung in Frage kommen. Im Anschluss werden Aufbau und Leistungsfähigkeit aktueller Signalprozessoren betrachtet.

[1] Auf dem Markt der 3D-Grafik-Beschleuniger für Personal Computer hat sich eine ähnliche Entwicklung bereits vollzogen. Aktuelle GPUs (*graphic processor unit*) bieten inzwischen mit einer flexibel programmierbaren, parallelen Verarbeitung die zehnfache Rechenleistung im Vergleich zum Hauptprozessor bei ähnlichen Herstellungskosten und Energieverbrauch.

1.2.1 Breitbandige Übertragungsverfahren

Mehrantennensysteme ermöglichen durch Ausnutzung der räumlichen Selektivität im Funkkanal eine Erhöhung der informationstheoretischen Übertragungskapazität [5]. In der Anwendung zeigt sich ein Abtausch zwischen dem Diversitätsgewinn, der Trennbarkeit parallel übertragener Datenströme sowie der Unterdrückung räumlich gerichteter Interferenz [6, 7]. In einem System mit mehreren Antennen auf Sende- und Empfangsseite (*Multiple Input, Multiple Output*, MIMO) kann grundsätzlich zwischen einem Mehrteilnehmersystem mit räumlichem Mehrfachzugriff (*Space Division Multiple Access*, SDMA) und dem räumlichem Multiplex zwischen zwei Teilnehmern unterschieden werden.

Entscheidend ist die Kenntnis des Übertragungskanals sowie der Interferenzsituation auf Sende- und/oder Empfangsseite. Eine Kanalschätzung in einem Mehrantennenempfänger ermöglicht einen Diversitätsgewinn durch optimale Kombination (*maximum ratio combining*) sowie eine räumliche Entzerrung zur Trennung parallel gesendeter Datenströme. In einem Mehrantennensender ermöglicht die Kanalkenntnis mit einer räumlichen Vorverzerrung eine optimale Nutzung von Sendediversität sowie die gleichzeitige Übertragung paralleler Datenströme zu mehreren Empfängern. Mit Hilfe von Space-Time-Codes [8] kann räumliche Sendediversität sub-optimal auch ohne Kanalkenntnis genutzt werden. Für den räumlichen Multiplex mit Kanalkenntnis auf Sende- und Empfangsseite kann eine Kombination aus Entzerrung und Vorverzerrung angewendet werden. Das im Sinne der Kanalkapazität optimale Übertragungsverfahren ist dann der Multiplex über die Eigenmoden des Kanals mit einer Leistungsverteilung nach dem Waterfilling-Prinzip [9].

Eine Übertragung auf den Eigenmoden ist ohne kooperierende Signalverarbeitung in physikalisch getrennten Teilnehmerstationen für den räumlichen Mehrfachzugriff nicht möglich. Neben der räumlichen Signatur sind hier zudem auch die Verteilung der Sendeleistung sowie die Fairness bei unterschiedlichen Dienstgüteanforderungen zu berücksichtigen. Die Suche nach der optimalen Übertragungsstrategie, besonders für den Downlink, ist immer noch Gegenstand der aktuellen Forschung [10, 11].

Für die rein räumliche Filterung oder auch Strahlformung wurde bereits in den 90er Jahren der Begriff der intelligenten Antennen (*smart antennas*) geprägt. Die *Intelligenz* steckt dabei in der Adaption der Richtwirkung einer Gruppenantenne, z.B. auf Grundlage geschätzter Ausbreitungsrichtungen. Diese Verfahren sind meist auf eine stark gerichtete Ausbreitung beschränkt.

Bei großen Bandbreiten und einer hohen räumlichen Dispersion, wie z.B. in Picozellen oder Innenräumen, ist eine Kombination der räumlichen Filterung mit der

zeitlichen Entzerrung der Intersymbolinterferenz erforderlich. Man spricht dann auch von der Raum-Zeit-Signalverarbeitung (*space-time signal processing*). Dieser Ansatz ist in kommerziellen Systemen bisher kaum zu finden, wird aber in praktisch allen neuen Standards der Breitbandkommunikation berücksichtigt und von der Literatur seit Jahren erschöpfend behandelt. Eine gute und etwas tiefergreifende Übersicht ist z.B. in [12] oder [13] zu finden.

In Breitbandzugangsnetzen ist das Transfervolumen vom Zugangspunkt zum Endgerät (Downlink) in der Regel deutlich größer als in der Gegenrichtung (Uplink). Ein Zeitduplex-Betrieb (*time division duplex*, TDD) ermöglicht einen asymmetrischen Datenverkehr, der flexibel an momentane Dienstanforderungen angepasst werden kann. Im Hinblick auf eine räumliche Vorverzerrung kann im Zeitduplex-Betrieb zudem die Reziprozität des Übertragungskanals genutzt werden. Die am Empfänger gewonnene Kanalkenntnis kann dann auch im Sender verwendet werden und erübrigt so die Übertragung von Feedback-Daten. Voraussetzung ist eine geeignete Kalibrierung der Sende- und Empfangszüge [14, 15]. Für den Zeitraum zwischen der Kanalschätzung am Empfänger und der Übertragung der vorverzerrten Daten muss der Kanal zudem ausreichend konstant bleiben.

Alle bestehenden und in der Entwicklung befindlichen Standards zur Übertragung hoher Datenraten (z.B. xDSL, IEEE802.11x, IEEE802.16x, DVBT [16] oder DAB [17]) basieren auf OFDM (*Orthogonal Frequency Division Multiplex*). OFDM erlaubt eine aufwandsgünstige Entzerrung frequenzselektiver Kanäle und bietet zudem eine hohe Flexibilität in einer adaptiven Nutzung des Spektrums, z.B. über die unterträgerspezifische Wahl von Modulationsstufe und Sendeleistung. Nachteile sind die erhöhten Anforderungen an die Sendeverstärker aufgrund einer größeren Signaldynamik sowie der Verlust an spektraler Effizienz durch zyklische Wiederholungen zwischen den OFDM-Symbolen.

Das in OFDM angewendete Prinzip der Frequenzbereichsentzerrung kann auch für die Einträgermodulation genutzt werden [18]. Voraussetzung ist wie bei OFDM die Übertragung einer zyklischen Erweiterung jedes Datenblockes. Ein grundsätzlicher Unterschied gegenüber OFDM ist die bei der Einträgerübertragung implizite Frequenzdiversität. Schmalbandige Fadingeinbrüche sorgen in OFDM-Systemen für den Ausfall eines oder mehrerer Unterträger. Dies muss durch eine geeignete Kanalcodierung aufgefangen werden; man spricht dann auch von *coded* OFDM oder kurz COFDM. In IEEE802.16 sind Einträger- und Mehrträgermodulation als Übertragungsmodi vorgesehen, für den UMTS Downlink wird eine OFDM Erweiterung diskutiert [19]. Die Frequenzbereichsentzerrung erlaubt den universellen Einsatz einer einheitlichen Signalverarbeitung für beide Modulationsarten.

In Mobilfunksystemen, wie IS95 oder UMTS-FDD, erfolgt der Mehrfachzugriff über eine Spreizung mit teilnehmerspezifischen Spreizsequenzen (*code division multiple access*, CDMA). Zur Versorgung großer Nutzerzahlen pro Zelle setzen diese, auch W-CDMA (*wideband* CDMA) genannten Systeme [20], einen hohen Spreizfaktor von bis zu 512 ein. Die Spreizsequenzen, bzw. die überlagerten Scrambling-Sequenzen, sind auf ihre Auto- und Kreuzkorrelationsfunktionen hin optimiert und ermöglichen so eine aufwandseffiziente Unterdrückung von Intersymbol- und Mehrfachzugriffsinterferenz durch ein kohärentes Aufaddieren der signifikanten Pfade nach dem Rake-Prinzip.

Für die breitbandige Versorgung kleiner Zellen wird CDMA eher in Kombination mit OFDM (*multi-carrier* CDMA, MC-CDMA) oder mit TDMA (TD/CDMA) angewendet. Hier werden orthogonale Spreizsequenzen mit kleineren Spreizfaktoren ($1 \ldots 16$) eingesetzt. Da die Orthogonalität in einem frequenzselektiven Kanal gestört wird und die Korrelationseigenschaften für einen Rake-Empfänger nicht ausreichen, ist am Empfänger eine aufwändigere, adaptive Unterdrückung der Intersymbol- und Mehrfachzugriffsinterferenz erforderlich. Diese sogenannte Mehrteilnehmerdetektion (*multi-user detection* oder auch *joint detection*) weist eine hohe Analogie zum räumlichen Multiplex bzw. Mehrfachzugriff auf. Eine kurze Übersicht der verschiedenen Lösungsansätze folgt im nächsten Unterkapitel.

Bei MC-CDMA erfolgt die Spreizung über den Frequenzbereich [21]. Die Datensymbole werden mit unterschiedlicher Gewichtung redundant auf mehreren Unterträgern übertragen. In TD/CDMA erfolgt dagegen eine sequentielle Übertragung der mit den Datensymbolen gewichteten Spreizsequenz. Mathematisch sind beide Spreizverfahren identisch und unterscheiden sich nur durch eine Fourier-Transformation der effektiv angewendeten Spreizsequenzen [22]. Der Ansatz einer universellen Frequenzbereichsentzerrung für Ein- und Mehrträgermodulation lässt sich damit auf CDMA erweitern [23]. Während sich TD/CDMA im UMTS-TDD Standard bereits durchgesetzt hat und umgesetzt wird, ist bisher kein System bekannt, das MC-CDMA tatsächlich nutzt.

1.2.2 Raum-Zeit-Signalverarbeitung

Die Raum-Zeit-Signalverarbeitung umfasst grundsätzlich die gesamte physikalische Schicht für den Datentransfer über einen frequenzselektiven MIMO-Kanal. Der Schwerpunkt dieser Arbeit liegt in der kombinierten Unterdrückung der Intersymbolinterferenz und der Interferenz eines nicht-orthogonalen Mehrfachzugriffs (SDMA und/oder CDMA) oder Multiplex. Die Algorithmen basieren im wesentlichen auf den ursprünglich für CDMA entwickelten Verfahren der Mehrteilnehmerdetektion [24].

Es folgt hier eine kurze Übersicht der wichtigsten Detektionsverfahren.

Der optimale MIMO Empfänger ist der Maximum-Likelihood-Detektor. Da der Rechenaufwand exponentiell mit der Teilnehmeranzahl wächst, muss meist auf suboptimale Verfahren zurückgegriffen werden. Diese gliedern sich grundsätzlich in linear und nichtlinear.

Die linearen Verfahren beschränken sich auf eine gewichtete Linearkombination der Empfangswerte. Die Gewichtung erfolgt entweder nach dem *zero forcing* (ZF) Ansatz über eine Inversion der Übertragungsmatrix oder nach dem Optimierungskriterium des minimalen quadratischen Fehlers (MMSE, *minimum mean square error*). Der MMSE-Ansatz berücksichtigt mit geringfügig höherem Aufwand die aktuelle Interferenzsituation und ist daher dem ZF überlegen. Für die Einträgermodulation erfolgt die Verarbeitung üblicherweise blockweise (*block linear equalizer*, BLE) auf Grundlage einer Matrixinversion [25]. Zur Minimierung des Rechenaufwandes kann die Matrix mit einem *overlap-save* Ansatz in der Dimension reduziert und über eine Transformation in den Frequenzbereich block-diagonalisiert werden [26]. Letzteres führt zur Frequenzbereichsentzerrung, die auch unmittelbar für die Mehrträgermodulation angewendet werden kann. Alternativ zur Matrixinversion kann die MMSE-Lösung auch iterativ nach der Theorie der adaptiven Filter gefunden werden [27]. Eine iterative Lösung ist vor allem zum Nachführen (*tracking*) eines zeitvarianten Kanals interessant.

Die nichtlinearen Verfahren erhöhen die Leistungsfähigkeit durch Rückführung bereits getroffener Entscheidungen. Unterschieden wird zum einen zwischen paralleler und sukzessiver Rückführung, zum anderen zwischen harten und weichen Entscheidungen. Bei der parallelen Entscheidungsrückführung wird der gesamte Detektionsprozess in mehreren Iterationen wiederholt. Beispiele sind mit abnehmender Leistungsfähigkeit und Komplexität: *unbiased minimum variance equalizer* [28], *soft cholesky block decision feedback equalizer* [29] und *recurrent neural network* [30]. Bei der sukzessiven Entscheidungsrückführung werden die Datenströme nacheinander detektiert, wobei jeweils die Interferenz der bereits entschiedenen Datenströme vom Eingangssignal abgezogen wird. Der VBLAST-Algorithmus iteriert in der optimalen Reihenfolge [31]. Mit einer sub-optimalen Reihenfolge kann der Aufwand über eine QR-Zerlegung deutlich reduziert werden [32, 33]. Bis auf die Entscheidung bei der schrittweisen Lösung des Gleichungssystems entspricht der Lösungsweg dem des linearen Ansatzes.

Schließlich ermöglicht das Turbo-Prinzip durch Einbeziehung der Decodierung in den Iterationsprozess der Entzerrung eine weitere Verbesserung. Die nichtlinearen Verfahren erreichen so zum Teil annähernd die Leistungsfähigkeit des Maximum-

Likelihood-Detektors, erfordern im Vergleich zu den linearen Verfahren meist jedoch auch einen wesentlich höheren Rechenaufwand. Für die Einträgermodulation mit Frequenzbereichsentzerrung ist jede Art der Entscheidungsrückführung problematisch, da für die Entscheidung in den Zeitbereich und zurück in den Frequenzbereich transformiert werden muss. Dies erhöht den Rechenaufwand und ist kritisch aufgrund der Latenzzeit der Fourier-Transformation.

1.2.3 Digitale Signalprozessoren

Der Markt für digitale Signalprozessoren (DSPs) wird im High-End Bereich von *Texas Instruments* (TI) und *Analog Devices* (ADI) dominiert. Die Rechenleistung einer Festkomma-Arithmetik wird meist in MAC-Operationen (MAC, *multiply accumulate*) angegeben. Ein MAC umfasst eine Multiplikation gefolgt von einer Addition. Bei einer Fließkomma-Arithmetik werden Addition und Multiplikation dagegen gleichwertig als FLOPS (*floating point operations per second*) gezählt. TI bietet zur Zeit mit der C64x-Reihe bis zu 4 $^{\mathrm{GMAC}}/_{\mathrm{s}}$ für eine 16 bit-Festkomma-Arithmetik bei einer Taktrate von 1 GHz. Die C67x-Reihe verfügt über eine Fließkomma-Arithmetik mit bis zu 1,3 GFLOPS bei 225 MHz. ADI erreicht mit dem TigerSHARC bei 600 MHz Taktfrequenz in einem Chip entsprechend 4,8 $^{\mathrm{GMAC}}/_{\mathrm{s}}$ oder 3,6 GFLOPS.

Während sich die Prozessoren im Detail deutlich unterscheiden, sind eine Reihe von Gemeinsamkeiten zu beobachten:

- VLIW-Architektur (*very large instruction words*) mit mehreren, getrennt programmierbaren Funktionseinheiten. Die Verteilung der Operationen auf die Funktionseinheiten kann so schon bei der Programmierung durch den Compiler optimiert werden.
- SIMD-Funktionalität (*single instruction, multiple data*) mit zwei identischen Datenpfaden
- Aufteilung der 32 bit-Register in zwei 16 bit- bzw. vier 8 bit-Worte ermöglicht mit einem 32 bit-Multiplizierer mehrere parallele Multiplikationen je Datenpfad
- Integrierte Hardware-Beschleuniger für die Codierung und Decodierung in der Kommunikationstechnik

Die I/O Bandbreite beträgt bei dem C6x etwa 17 $^{\mathrm{Gbit}}/_{\mathrm{s}}$. Der TigerSHARC erreicht mit vier *LinkPorts* zur Vernetzung mehrerer DSPs bereits 32 $^{\mathrm{Gbit}}/_{\mathrm{s}}$. Die Leistungsaufnahme liegt für beide Familien bei etwa zwei Watt. In typischen Anwendungen der Signalverarbeitung, wie z.B. FIR-Filter oder Fourier-Transformation, wird die theoretisch verfügbare Rechenleistung nahezu voll ausgeschöpft. In komplexeren

Algorithmen mit einer gegenseitigen Abhängigkeit der Daten ist es aufgrund der Latenzzeit in den Pipelines oft schwer, die Parallelität effizient zu nutzen.

Neben den klassischen Signalprozessoren mit VLIW-Architektur drängen auch neue, hoch parallele Prozessoren auf den Markt, so z.B. der *Cell*, entwickelt in einer Kooperation von IBM, Sony und Toshiba, sowie der *Linedancer* der Firma ASPEX.

Die Linedancer-Technologie wird u.a. von Philips für Anwendungen in der Kommunikationstechnik vermarktet. Eine große Anzahl sehr einfacher Prozessorelemente sind hier über ein rekonfigurierbares Netzwerk verschaltet. Die Prozessorelemente führen Operationen mit einer assoziativen Addressierung bit-seriell aus. Ein integrierter Controller konfiguriert zur Laufzeit die Netzwerkstruktur und generiert die Befehlsworte. Der Prozessor von Philips ist mit 4096 Prozessorelementen spezifiziert und erreicht mit 300 MHz Taktrate 6,1 $^{\mathrm{GMAC}}/_{\mathrm{s}}$. Leistungsverbrauch und Chipfläche sind nicht spezifiziert.

Der Cell wurde erst kürzlich mit ersten Spezifikationen angekündigt [34] und soll Ende 2005 verfügbar sein. Die erste Generation verfügt über acht parallele Prozessorelemente und erreicht nach ersten Labortests eine Taktrate von über 4 GHz. Die Prozessorelemente sind unabhängig programmierbar und verarbeiten jeweils bis zu vier Fließkomma-MAC-Operationen pro Taktzyklus. Daraus ergibt sich eine theoretische Rechenleistung von 256 GFLOPS. Bei der Programmierung werden abgeschlossene Aufgaben in so genannten *Software Cells* zusammengefasst, die dann zur Laufzeit auf die verfügbaren Prozessorelemente verteilt werden. Die Rechenleistung kann so über die Anzahl an Prozessorelementen unabhängig vom Programmcode skaliert werden.

Die Entwicklung der x86 Prozessoren von Intel und AMD zeigt in der letzten Zeit nur noch marginale Erhöhungen der Taktrate. Intel hat bei 4 GHz bereits massiv mit thermischen Problemen zu kämpfen. Die Road-Maps beider Hersteller zeigen einen deutlichen Trend zu Mehrprozessorsystemen, mit zunächst zwei, später dann zehn oder mehr Prozessoren auf einem Chip.

Schließlich bieten auch FPGAs (*field programmable gate arrays*) inzwischen eine leistungsfähige Plattform für die parallele Signalverarbeitung. Entscheidend für den Wandel von der Steuerlogik zur System-on-Chip-Plattform war neben der permanenten Verbesserung der Integrationsdichte die Einführung dedizierter, d.h. hart verdrahteter Komponenten, wie z.B. Multiplizierer, Speicherblöcke oder ganzer CPU-Kerne. Mit der Virtex-4 Architektur verspricht Xilinx auf einem Chip bis zu 256 $^{\mathrm{GMAC}}/_{\mathrm{s}}$ in 18 bit-Festkomma-Arithmetik. Aufgrund hoher Kosten werden FPGAs aus dem High-End-Bereich meist nur in Forschung und Entwicklung eingesetzt.

Ein speziell für die Raum-Zeit-Verarbeitung in MIMO-Systemen optimierter Pro-

zessor ist auf dem Markt bisher nicht verfügbar. In der Forschung wurden bereits einige echtzeitfähige MIMO-Demonstratoren entwickelt, die aber meist auf DSPs [35] bzw. einer Kombination aus DSPs und FPGAs [36, 37] basieren. Feste Strukturen wie z.B. Korrelatoren, FFTs oder die Matrix-Vektormultiplikation zur Gewichtung der Datenströme werden üblicherweise im FPGA umgesetzt, die Algorithmen zur Berechnung der Gewichtsfaktoren aus Gründen der Flexibilität mit Fließkomma-Arithmetik in Signalprozessoren. Die integrierten Lösungen in [38, 39, 40, 41] sind nicht programmierbar und in der Rechenleistung auf einen geringen Datendurchsatz beschränkt.

1.3 Bewertung

In der Motivation wurde der Bedarf an leistungsfähigen Prozessoren für die breitbandige Kommunikation mit höchster spektraler Effizienz dargestellt. Die Entwicklung paralleler, programmierbarer Rechner-Architekturen muss sich auf den zukünftigen Markt ausrichten, da momentan weder die Anwendung in den bestehenden Standards noch eine witschaftliche Realisierbarkeit mit der verfügbaren Technologie gegeben sind. Mehrantennensysteme mit einer Raum-Zeit-Signalverarbeitung sind der Schlüssel zur Steigerung der Effizienz. Bestehende Systeme zeigen, dass OFDM das wohl geeignetste Modulationsverfahren zum Erreichen hoher Bandbreiten ist. Die Einträgerübertragung mit Frequenzbereichsentzerrung ist eine vielversprechende Alternative mit nahezu identischer Signalverarbeitung. Es kann außerdem von einem Zeit-Duplex ausgegangen werden. Während Spreizverfahren keine primäre Bedeutung haben, können sie Aufgrund der Analogie zum räumlichen Mehrfachzugriff mit berücksichtigt werden.

Von den beschriebenen Algorithmen stehen für diese Arbeit die linearen Verfahren im Vordergrund. Die sukzessive Entscheidungsrückführung mit optimaler Reihenfolge erfordert einen wesentlich höheren Aufwand. Bei der parallelen Entscheidungsrückführung, bzw. bei den Turbo-Verfahren ist der Gesamtaufwand proportional zum Durchsatz, hohe Bandbreiten werden damit schwer zu realisieren sein. Die linearen Verfahren ermöglichen dagegen eine Trennung zwischen Gewichtung und Gewichtsberechnung. Bei einer geringen Mobilität kann so der Aufwand reduziert werden, da die Gewichtsberechnung nur den zeitlichen Veränderungen des Kanals folgen muss. Die explizite Berechnung der Gewichtsmatrix hat zudem eine hohe Bedeutung für die Vorverzerrung, da hier eine Entscheidungsrückführung nicht mehr möglich ist. Die sukzessive Entscheidungsrückführung mit einer sub-optimalen Reihenfolge wird als mögliche Erweiterung angesehen, die zumindest für eine Mehr-

trägermodulation mit geringen Modifikationen umgesetzt werden kann.

An einer Bespielkonfiguration wird nun der notwendige Rechenaufwand abgeschätzt und mit der Leistungsfähigkeit bestehender Prozessoren verglichen. Betrachtet wird ein MIMO System mit acht Sendern, acht Empfängern und einer Symbolrate von 25 MHz. Die räumliche Gewichtung zur Trennung der gesendeten Datenströme im komplexen Basisband erfordert $8 \times 8 \times 4 = 256$ MAC-Operationen pro Symboldauer, d.h. eine Rechenleistung von $6.4\,{}^{\mathrm{GMAC}}/_{\mathrm{s}}$ bzw. zwei TigerSHARCs. Die Raum-Zeit-Entzerrung eines frequenzselektiven MIMO-Kanals über eine lineare Filterung erfordert im Zeitbereich $6.4\,{}^{\mathrm{GMAC}}/_{\mathrm{s}}$ pro Filterkoeffizient. Für eine Transformation in den Frequenzbereich und zurück muss eine Fourier-Transformation mit einem kontinuierlichen Durchsatz von 400 MSPS durchgeführt werden (MSPS, *mega samples per second*). Ein TigerSHARC erzielt einen Durchsatz von etwa 256 MSPS. Im Frequenzbereich ist die Filterung nach dem Prinzip der schnellen Faltung dann über eine einfache Gewichtung möglich. Für die Raum-Zeit-Entzerrung im Frequenzbereich sind also insgesamt vier TigerSHARCs erforderlich.

Zum einen zeigt dieses einfache Beispiel die deutliche Reduktion des Rechenaufwandes durch die Transformation in den Frequenzbereich. Schon ab zwei Filterkoeffizienten ist die Frequenzbereichsentzerrung aufwandsgünstiger als eine Filterung im Zeitbereich. Dies ist eine Besonderheit in MIMO-Systemen: während die Anzahl der Multiplikationen für die Faltung quadratisch mit der Matrixdimension steigt, nimmt der zusätzliche Aufwand der Fourier-Transformationen nur linear zu.

Zum anderen macht das Beispiel deutlich, dass der momentan leistungsfähigste Signalprozessor bei weitem nicht ausreicht: die Gewichtung allein erfordert bereits vier Prozessoren. Der Hauptaufwand ist jedoch für die Berechnung der Gewichte zu erwarten. Die Entwicklung der x86 Prozessoren zeigt, dass die Taktraten nicht wesentlich gesteigert werden können. Eine Parallelisierung der Verarbeitung ist also zwingend erforderlich.

1.4 Ziel der Arbeit

Ziel der Arbeit ist die Entwicklung einer effizienten und zugleich flexibel programmierbaren Prozessorarchitektur für die Raum-Zeit-Signalverarbeitung im Frequenzbereich. Die Architektur zielt auf eine Anwendung in zukünftigen Standards und richtet sich nach aktuellen Entwicklungen in der Forschung.

Für die Anwendungen gelten die folgenden Voraussetzungen:

- Hohe Datenraten von mehreren hundert Megabit pro Sekunde
- Der Übertragungskanal ist aufgrund der hohen Bandbreite frequenzselektiv
- Paketbasierte Übertragung mit Aufwärts- und Abwärtsstrecke im Zeitduplex
- Einheitliche, zyklisch erweiterte Blockstruktur für die effiziente Verarbeitung im Frequenzbereich
- Verwendung von Einträger- und Mehrträgermodulation
- Die räumlichen Selektivität des Funkkanals wird durch Mehrantennensysteme für den Mehrfachzugriff bzw. Multiplex genutzt
- Zur Trennung der Teilnehmerdatenströme werden auch die Spreizverfahren MC-CDMA bzw. TD/CDMA eingesetzt
- Räumlich-zeitliche Entzerrung durch eine lineare Mehrteilnehmerdetektion für den nicht-orthogonalen Mehrfachzugriff
- Nutzung von Kanalkenntnis im Sender für eine räumlich-zeitliche Vorverzerrung

Die Entwicklung der Prozessorarchitektur erfolgt unter den folgenden Anforderungen:

- Programmierbarkeit für eine flexible Adaption des Übertragungsverfahrens und der Systemparameter an die Kanaleigenschaften und Dienstgüteanforderungen
- Parallelisierung der Verarbeitung für eine Rechenleistung über $10\,{}^{\mathrm{GMAC}}/_{\mathrm{s}}$
- hohe Effizienz im Ressourcenverbrauch im Sinne der Wirtschaftlichkeit durch eine anwendungsspezifische Optimierung der Architektur
- Auslegung der Rechenwerke anhand der anwendungsspezifischen Anforderungen an Zahlendarstellung (Fest- oder Fließkomma-Arithmetik) und Wortbreiten

Für die anwendungsspezifische Optimierung der Architektur ist zunächst eine genaue Betrachtung der Anwendung erforderlich. Im Anschluss wird ein Konzept für die Architektur im Rahmen der Voraussetzungen allgemeingültig entwickelt und qualitativ bewertet. Anhand einer konkreten exemplarischen Konfiguration (Demonstrator) wird das Konzept dann realisiert und quantitativ bewertet. Neben prinzipiellen Konzepten für die Realisierung der Raum-Zeit-Signalverarbeitung ist eine realistische Abschätzung des erforderlichen Aufwandes zu erwarten. Die Auslegung der Festkomma-Arithmetik auf die betrachteten Algorithmen liefert grundsätzliche Aussagen über Anforderungen und Machbarkeit, die auch auf eine DSP-Implementierung übertragbar sind.

1.5 Gliederung

Kapitel 2 legt die Grundlage für die Anwendung über die Definition und Modellierung der Übertragung. Besonderen Wert wird auf eine einheitliche Beschreibung für die Ein- und Mehrträgermodulation unter zusätzlicher Berücksichtigung von Spreizverfahren gelegt. Die Einführung eines einfach parametriebaren, stochastischen Kanalmodells erlaubt später eine Dimensionierung der Wortbreiten im Prozessor unter definierten, praxisrelevanten Bedingungen.

Auf Basis des Übertragungsmodells wird in **Kapitel 3** das Optimierungsproblem einer linearen Detektion nach dem Kriterium des minimalen mittleren quadratischen Fehlers formuliert. Im Anschluss werden mögliche Lösungsansätze für das Problem vorgestellt und auf die zugrunde liegenden Operationen hin untersucht. Sämtliche Ansätze lassen sich auf eine feststehende Abfolge komplexwertiger Vektorprodukte und die Normierung durch einen reellen Divisor zurückführen. Die Anzahl der jeweils benötigten Operationen wird abgeschätzt und tabellarisch gegenübergestellt.

Kapitel 4 definiert zunächst den Einsatz des Prozessors für verschiedene Betriebszustände eines Zugangspunktes. Mit einer einheitlichen Schnittstelle kann der Prozessor für die Raum-Zeit-Verarbeitung zum Senden und Empfangen mit Ein- und Mehrträgermodulation verwendet werden. Die Aufstellung möglicher Ansätze zur Parallelisierung führt zu der gewählten Architektur: die parallele Ausführung komplexer MAC-Operationen für eine sequentielle Berechnung der Vektorprodukte, jeweils parallel für unabhängige Datensätze mit einer gemeinsamen Ansteuerung. Es folgt eine detaillierte Beschreibung der Architektur und der anwendungsspezifischen Auslegung der Rechenwerke, des lokalen Speichers und der Steuereinheit.

Für zwei numerisch kritische Anwendungsbeispiele (Matrixinversion und RLS-Algorithmus) wird in **Kapitel 5** der Einfluss der Zahlendarstellung im Rechenwerk in Abhängigkeit der Wortbreiten untersucht. Die bitgenaue Modellierung einer Festkomma- und einer Fließkomma-Arithmetik erlaubt über die Simulation der Übertagung mit dem stochastichen Kanalmodell eine vergleichende Bewertung anhand des Bitfehlerverhältnisses.

Kapitel 6 beschreibt schließlich die Umsetzung der vorgeschlagenen Architektur auf einer leistungsfähigen FPGA-Plattform als Bestandteil eines kompletten MIMO-Demonstrators. Die Umsetzung für eine praxisnahe Anwendung erlaubt eine aussagekräftige Bewertung der erzielbaren Leistungsfähigkeit sowie der dazu erforderlichen Ressourcen.

KAPITEL 2

Systemmodell

Aufbauend auf den Vorgaben der Zielsetzung folgt nun die Definition des Systemmodells. Zunächst wird in Kapitel 2.1 das zugrunde liegende Kommunikationssystem und die verwendete Notation eingeführt. Die Beschreibung der Datenübertragung in Kapitel 2.2 führt zu einem Systemmodell, das den Zusammenhang zwischen den gesendeten Daten und dem Empfangssignal in einer einfachen Matrixgleichung wiedergibt. Für die Bewertung der Zahlendarstellung (Kapitel 5) wird schließlich in Kapitel 2.3 ein stochastiches Kanalmodell eingeführt.

Das Systemmodell bildet die Grundlage für die lineare Raum-Zeit-Entzerrung, die eine Lösung der Matrixgleichung und damit eine Schätzung der gesendeten Daten ermöglicht.

2.1 Definition des Szenarios und der Notation

Im Folgenden wird ein Mehrteilnehmersystem nach Bild 2.1 betrachtet: Ein zentraler Zugangspunkt versorgt mehrere Teilnehmerstationen. Die Übertragung von den Teilnehmerstationen zum Zugangspunkt wird als Uplink, die Gegenrichtung als Downlink bezeichnet. Der Zugangspunkt verfügt über M Antennen, die Teilnehmerstationen jeweils über eine oder auch mehrere Antennen.

Der Mehrfachzugriff der Teilnehmer auf den Kanal erfolgt durch eine Trennung über die Zeit, über Codes sowie über die räumliche Selektivität (TD/CD/SDMA). Die Übertragung zweier Datenströme in unterschiedlichen Zeitschlitzen oder Frequenzbändern ist ohne gegenseitige Beeinflussung bzw. orthogonal möglich. Für die Auslegung des Prozessors ist vor allem der nicht-orthogonale Mehrfachzugriff über die Spreizung und die räumliche Selektivität von Interesse.

In einem Zeitschlitz sei eine Untermenge von K Teilnehmerstationsantennen am nicht-orthogonalen Mehrfachzugriff beteiligt, d.h. sie werden zur gleichen Zeit im gleichen Frequenzband betrieben. Die übrigen Teilnehmer werden unabhängig davon in anderen Zeitschlitzen bedient. Es wird angenommen, dass Teilnehmerstationen mit mehreren Antennen einen einfachen räumlichen Multiplex anwenden ohne

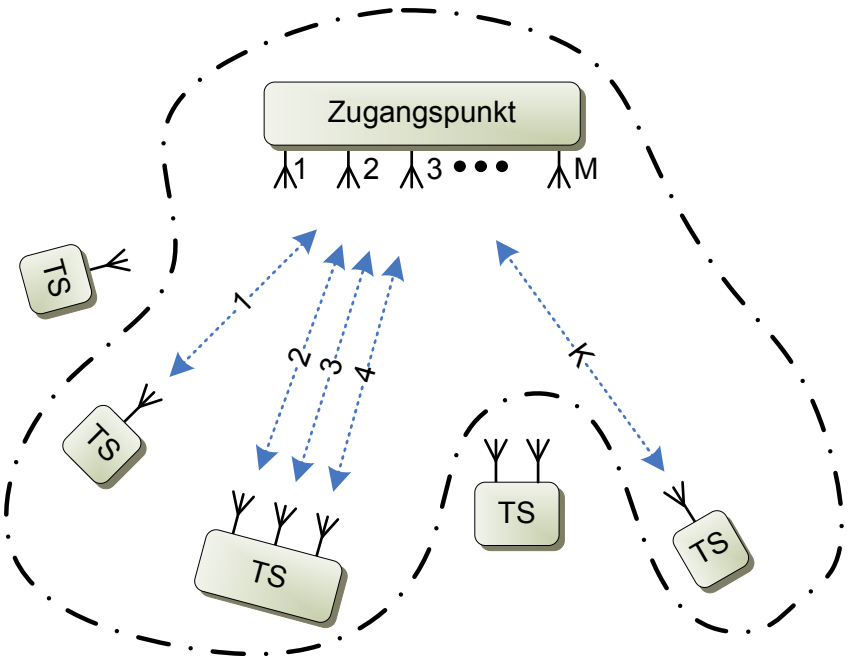


Bild 2.1: Mehrteilnehmersystem mit M Antennen am Zugangspunkt und K logischen Verbindungen zu einer Untermenge an Teilnehmerstationen (TS) über einen nichtorthogonalen Mehrfachzugriff. Die übrigen Teilnehmer werden unabhängig davon in anderen Zeitschlitzen bedient.

eine zusätzliche Verschaltung der Signale in Sender und Empfänger, z.B. über Space-Time-Codes. Damit werden in Uplink und Downlink jeweils K unabhängige Datenströme übertragen, wobei jeder Datenstrom genau einer Teilnehmerstationsantenne zugeordnet ist.

Der Begriff *Antenne* kann in diesem Zusammenhang unterschiedlich interpretiert werden: eine physikalische Antenne kann über mehrere Tore verfügen, ein Antennentor kann wiederum von mehreren physikalischen Antennen gespeist werden. Entscheidend für die Signalverarbeitung ist nur die Anzahl an parallelen Sende/Empfangsmodulen. Es wird hier angenommen, dass zu jeder Antenne genau ein Sende/Empfangsmodul gehört.

Die Übertragung wird in Zeitschlitze aufgeteilt. In jedem Zeitschlitz wird eine variable Anzahl von T Datenblöcken einer einheitlichen Länge übertragen. Ein Datenblock besteht jeweils aus $N = QU$ Chips, dabei ist Q der Spreizfaktor und U die Anzahl der übertragenen QAM-Symbole. Die Größen Q und U seien variabel (mit $Q = 1,2,4,8,\ldots$), N sei fest. Die Teilnehmerkonstellation und die Übertragungsparameter bleiben innerhalb eines Zeitschlitzes konstant, können sich aber zwischen den Zeitschlitzen beliebig ändern. Uplink und Downlink werden im Zeitduplex zu unterschiedlichen Zeitpunkten auf der gleichen Frequenz betrieben.

Die Übertragungsrichtung kann sich innerhalb eines Zeitschlitzes ändern.

Zur Übersicht werden die wichtigsten Übertragungsparameter mit den in dieser Arbeit verwendeten Symbolen noch einmal zusammengefasst (Eine vollständige Auflistung ist im Symbolverzeichnis zu finden):

- M : Anzahl der Antennen am Zugangspunkt
- K : Anzahl der gleichzeitig aktiven Antennen der Teilnehmerstationen
- Q : Spreizfaktor
- U : Anzahl der in einem Datenblock übertragenen Symbole
- N : Anzahl an Unterträgern bzw. Chips pro Datenblock
- T : Anzahl der Datenblöcke in einem Zeitschlitz

Alle skalaren Größen werden kursiv gedruckt. Für ganzzahlige Parameter werden Großbuchstaben verwendet (M), eine Indizierung innerhalb des gültigen Parameterraumes erfolgt dann durch den entsprechenden Kleinbuchstaben, z.B. die Antenne m mit $m = 1 \dots M$. Vektoren und Matrizen werden fett gedruckt: Kleinbuchstaben für Vektoren ($\mathbf{x}$) und Großbuchstaben für Matrizen ($\mathbf{A}$). Folgen werden in altdeutsch gedruckt und immer mit Index angegeben, z.B. $\mathfrak{a}(n)$. Zwischen komplexen und reellen Größen wird nicht unterschieden.

Komplex konjugiert wird mit $(.)^*$, transponiert mit $(.)^T$ dargestellt. Die Kombination aus beiden ist die Hermitesche $(.)^H$. Die lineare Faltung von $\mathfrak{a}(n)$ und $\mathfrak{b}(n)$ wird mit $\mathfrak{a}(n) * \mathfrak{b}(n)$, die zyklische Faltung mit $\mathfrak{a}(n) \circledast \mathfrak{b}(n)$ beschrieben. Die diskrete Fourier-Transformation wird durch $\mathcal{F}_N\{\mathfrak{x}(n)\}$ dargestellt, wobei der Index (hier N) die Länge der Transformation angibt; $\mathcal{F}_N^{-1}\{\mathbf{x}\}$ entspricht der inversen Fourier-Transformation. Zur Unterscheidung werden Signale im Zeitbereich immer als Folgen und im diskreten Frequenzbereich immer als Vektoren bzw. Matrizen angegeben. Zwischen Folgen und Vektoren mit dem gleichen Buchstaben besteht immer der Zusammenhang der Fourier-Transformation, z.B. $\mathbf{x} = \mathcal{F}_N\{\mathfrak{x}(n)\}$, wobei Folgen einer Länge kleiner N mit Nullen aufgefüllt werden. Das Symbol $\mathbf{I}_K$ stellt die Einheitsmatrix dar, wobei der Index K die Dimension angibt. Die Notation $\lfloor a \rfloor$ bedeutet den nächst kleineren ganzzahligen Wert von a.

Mit der Notation $[\mathbf{A}]_{z,0:s}$ werden Teile aus Matrizen oder Vektoren selektiert, hier ein Vektor von Spalte 0 bis s in Zeile z. Die Dimension der Matrix wird entsprechend durch $[Z \times S]$ gekennzeichnet. Über $\mathbf{A}^{(n)}$ wird die n-te Matrix $\mathbf{A}$ aus einer Menge gleichartiger Matrizen indiziert. Während $\mathrm{diag}(\mathbf{x})$ eine Diagonalmatrix aus den Elementen des Vektors $\mathbf{x}$ beschreibt, entspricht $\mathrm{diag}(\mathbf{A})$ der Hauptdiagonalen der Matrix $\mathbf{A}$. Das Symbol $\otimes$ bezeichnet das Kroneckerprodukt zweier Matrizen.

2.2 Modellierung der Übertragung

In diesem Kapitel wird das Systemmodell für eine zyklisch erweiterte Blockübertragung in dem oben definierten Mehrteilnehmersystem eingeführt. Die Motivation für die zyklische Erweiterung ist eine Umwandlung der linearen Faltung des Kanals in eine zyklische Faltung für eine aufwandseffiziente Signalverarbeitung im Frequenzbereich. Wichtige Randbedingung im Sender ist die Bandbegrenzung durch eine definierte spektrale Formung und das Vermeiden von Signalsprüngen in aufeinander folgenden Blöcken. Im Gegensatz zur klassischen OFDM-Praxis [42] wird ein neuer Ansatz über die lineare Faltung mit einem Pulsformfilter im Zeitbereich vorgeschlagen. Der Ansatz erlaubt eine einheitliche Modellierung für Einträger- und Mehrträgerübertragung.

Das Übertragungsmodell wird zunächst für die Einträgermodulation mit einem Teilnehmer erst im Zeitbereich und dann im Frequenzbereich formuliert. Die Beschreibung im Frequenzbereich wird dann um eine alternative Mehrträgermodulation erweitert. Anschließend folgt der Übergang zum Vektorkanal durch die Einführung der Spreizung und eines Mehrtorempfängers am Zugangspunkt. Schließlich wird das Modell auf den nicht-orthogonalen Mehrfachzugriff erweitert.

2.2.1 Einträgermodulation im diskreten Zeitbereich

Zunächst wird eine Einträgerübertragung von einem Sender zu einem Empfänger betrachtet, d.h. $M = K = 1$. Im Sender werden N komplexwertige Chips zu einer Folge $\mathfrak{d}_c(n)$ mit $n = 0 \ldots N-1$ zusammengefasst. Die Folge wird um die Länge L zyklisch erweitert, d.h. die letzten L Chips werden wiederholt und vor der Folge eingefügt:

$$\mathfrak{s}(n) = \begin{cases} \mathfrak{d}_c(n+N-L) & \text{für} \quad n = 0 \ldots L-1 \\ \mathfrak{d}_c(n-L) & \quad\quad\; n = L \ldots N+L-1 \end{cases} \tag{2.1}$$

Auf diese Weise werden nun T zyklisch erweiterte Folgen $\mathfrak{s}^{(t)}(n)$ mit $t = 0 \ldots T-1$ erzeugt, zusammengesetzt und dann $\ddot{U}$-fach abgetastet durch Einfügen von $(\ddot{U}-1)$ Nullen zwischen den einzelnen Chips:

$$\mathfrak{s}_{\ddot{u}}(n_{\ddot{u}}) = \begin{cases} \mathfrak{s}^{(t)}(n) & \text{für} \quad n_{\ddot{u}} = (n + t(N+L))\ddot{U} \\ 0 & \text{sonst} \end{cases} \tag{2.2}$$

Die resultierende Folge $\mathfrak{s}_{\ddot{u}}(n_{\ddot{u}})$ wird anschließend mit einem interpolierenden Pulsformfilter mit der Impulsantwort $\mathfrak{g}(n_{\ddot{u}})$ linear gefaltet:

$$\mathfrak{p}_s(n_{\ddot{u}}) = \mathfrak{g}(n_{\ddot{u}}) * \mathfrak{s}_{\ddot{u}}(n_{\ddot{u}}) \tag{2.3}$$

Als Pulsformfilter wird üblicherweise ein *root-raised-cosine*-Filter [43] verwendet, das über den roll-off Faktor r sowie die Filterlänge W_f spezifiziert wird. Die Folge $\mathfrak{p}_s(n_{\ddot{u}})$ wird nun über einen D/A-Wandler in ein analoges Signal umgewandelt. Die Pulsformung dient der Bandbegrenzung des Sendesignals. Die Überabtastung bestimmt den Abstand zwischen den spektralen Wiederholungen am Ausgang des D/A Wandlers und damit die Flankensteilheit der analogen Filter. Die Taktrate des Wandlers sei f_s, damit ergibt sich eine Chiprate von

$$f_{\text{chip}} = \frac{f_s}{\ddot{U}}. \tag{2.4}$$

Das Systemmodell wird in Kapitel 2.2.4 um eine Spreizung erweitert, daher wird hier bereits allgemeingültig der dort übliche Begriff Chiprate verwendet. Die Blockrate, d.h. die Rate mit der die Datenblöcke übertragen werden, beträgt

$$f_b = \frac{f_{\text{chip}}}{N + L}. \tag{2.5}$$

Die Dauer einer Taktperiode T_s, die Chipdauer T_{chip} und die Blockdauer T_b entsprechen den Kehrwerten der jeweiligen Raten.

Das Signal wird im Sender auf die Trägerfrequenz hochgemischt, verstärkt, über die Antenne abgestrahlt und schließlich über den Kanal übertragen. Im Empfänger wird das Empfangssignal verstärkt und zurück ins Basisband gemischt. Die Abtastung mit einem A/D-Wandler führt zu der zeitdiskreten Empfangsfolge $\mathfrak{p}_e(n_{\ddot{u}})$. Die Taktrate der Abtastung sei ebenfalls gleich f_s.

Der Zusammenhang zwischen dem Sendesignal $\mathfrak{p}_s(n_{\ddot{u}})$ und dem Empfangssignal $\mathfrak{p}_e(n_{\ddot{u}})$ kann über eine lineare Faltung mit der Impulsantwort $\mathfrak{h}_t(n_{\ddot{u}})$ des zeitdiskreten äquivalenten Tiefpasssystems [43] des Kanals beschrieben werden:

$$\mathfrak{p}_e(n_{\ddot{u}}) = \mathfrak{h}_t(n_{\ddot{u}}) * \mathfrak{p}_s(n_{\ddot{u}}) = \mathfrak{h}_t(n_{\ddot{u}}) * \mathfrak{g}(n_{\ddot{u}}) * \mathfrak{s}_{\ddot{u}}(n_{\ddot{u}}) \tag{2.6}$$

Dabei wird hier zunächst ein zeitdiskreter Übertragungskanal vorausgesetzt; im Kanalmodell in Kapitel 2.3 wird dann auch der Fall eines zeitkontinuierlichen Kanals berücksichtigt. Vorausgesetzt werden zunächst weiterhin ein stückweise statischer Kanal für die Dauer eines Zeitschlitzes sowie eine perfekte Synchronisation im Empfänger bezüglich der Trägerfrequenz und der Abtastung. Die Darstellung über das äquivalente Tiefpasssystem führt zu einer komplexwertigen Impulsantwort. Die Impulsantwort sei zeitlich begrenzt auf W_h Chiplängen. Wie in Gleichung (2.6) zu

erkennen ist, können die Impulsantworten von Kanal und Pulsformfilter auch zusammengefasst werden zu einer resultierenden Folge der Länge $W = W_h + W_f - 1$ Chiplängen. Die Filterlänge W_f ist dabei ebenfalls auf die Chiplänge zu normieren. Im Empfänger wird die Folge $\mathfrak{p}_e(n_{\ddot{u}})$ wieder in T einzelne Blöcke $\mathfrak{p}_e^{(t)}(n_{\ddot{u}})$ zerlegt, wobei jeweils die zyklische Erweiterung in jedem Datenblock verworfen wird:

$$\mathfrak{p}_e^{(t)}(n_{\ddot{u}}) = \mathfrak{p}_e\left(n_{\ddot{u}} + L + t\,(L+N)\right) \quad \text{für} \quad \begin{array}{rcl} n_{\ddot{u}} & = & 0 \ldots \ddot{U}N-1 \\ t & = & 0 \ldots T-1 \end{array} \tag{2.7}$$

Unter der Voraussetzung einer mit

$$L \geq W - 1 = W_h + W_f - 2 \tag{2.8}$$

ausreichend dimensionierten Länge der zyklischen Erweiterung entspricht jeder Datenblock einer zyklischen Faltung des entsprechenden Sendeblocks mit dem Pulsformfilter und der Kanalimpulsantwort. Durch eine zyklische Faltung der Datenblöcke mit einem Tiefpassfilter sollen nun störende Anteile außerhalb des Nutzbandes unterdrückt werden. Im Folgenden wird hierzu das gleiche Filter $\mathfrak{g}(n_{\ddot{u}})$ verwendet, dass im Sender zur Pulsformung eingesetzt wird. Damit ergibt sich insgesamt:

$$\mathfrak{x}_{\ddot{u}}^{(t)}(n_{\ddot{u}}) = \mathfrak{g}(n_{\ddot{u}}) \circledast \mathfrak{p}_e^{(t)}(n_{\ddot{u}}) = \mathfrak{g}(n_{\ddot{u}}) \circledast \mathfrak{h}_t(n_{\ddot{u}}) \circledast \mathfrak{g}(n_{\ddot{u}}) \circledast \mathfrak{d}_{\ddot{u}}^{(t)}(n_{\ddot{u}}), \tag{2.9}$$

wobei $\mathfrak{d}_{\ddot{u}}^{(t)}(n_{\ddot{u}})$ der t-ten überabgetasteten Chipfolge entspricht. Nach der Filterung wird das Empfangssignal mit der Chiprate abgetastet:

$$\mathfrak{x}^{(t)}(n) = \mathfrak{x}_{\ddot{u}}^{(t)}(\ddot{U}n) \qquad \text{mit} \qquad n = 0 \ldots N-1 \tag{2.10}$$

Bei der Abtastung kommt es zu Aliasing, d.h. zu einer Überlappung im Frequenzbereich. Eine ideale Entzerrung ist daher im allgemeinen bei Chiprate nicht mehr möglich (siehe auch nächstes Kapitel). Daher wird häufig zunächst nur auf doppelte Chiprate abgetastet. Erst die Anwendung eines kanalangepassten Filters (*channel matched filter*) erlaubt eine Reduktion auf Chiprate ohne jeglichen Informationsverlust (*sufficient statistics*). Aufgrund des deutlich geringeren Aufwandes für die nachfolgende Signalverarbeitung wird im Folgenden dennoch eine Entzerrung bei Chiprate angenommen.

2.2.2 Transformation in den Frequenzbereich

Die Verwendung einer zyklischen Erweiterung führt zum Übergang von einer linearen zu einer zyklischen Faltung mit dem Kanal und erlaubt damit eine sehr einfache Dar-

stellung der Übertragung im diskreten Frequenzbereich. Die Beschreibung erfolgt jeweils für einen zyklisch erweiterten Datenblock, der Index t zur Kennzeichnung der einzelnen Blöcke innerhalb eines Zeitschlitzes entfällt im Folgenden. Für die Darstellung im Frequenzbereich wird eine vektorielle Schreibweise mit Spaltenvektoren eingeführt.

Ausgangspunkt ist der Chipvektor $\mathbf{d}_\mathrm{c}$, ein Spaltenvektor mit N Elementen. Für die Einträgerübertragung entspricht $\mathbf{d}_\mathrm{c}$ der diskreten Fourier-Transformierten der Chipfolge $\mathfrak{d}_c(n)$:

$$\mathbf{d}_\mathrm{c} = \mathcal{F}_N\{\mathfrak{d}_c(n)\} \tag{2.11}$$

Die $\ddot{U}$-fache Abtastung in Gleichung (2.2) entspricht im Frequenzbereich einer $\ddot{U}$-fachen Wiederholung des Vektors $\mathbf{d}_\mathrm{c}$. Für die einzelnen Elemente des überabgetasteten Chipvektors $\mathbf{d}_\mathrm{ü}$ gilt:

$$[\mathbf{d}_\mathrm{ü}]_{n+\ddot{u}N} = [\mathbf{d}_\mathrm{c}]_n \qquad \text{für} \qquad \ddot{u} = 0 \ldots \ddot{U} - 1 \tag{2.12}$$

Unter den oben getroffenen Voraussetzungen kann nun direkt Gleichung (2.9) angewendet werden um die gesamte Übertragung zu modellieren. Die zyklische Faltung im Zeitbereich entspricht dabei einer elementweisen Multiplikation im Frequenzbereich. Die Impulsantworten von Pulsformfilter $\mathfrak{g}(n_\ddot{u})$ bzw. Kanal $\mathfrak{h}_t(n_\ddot{u})$ werden mit Nullen auf $\ddot{U}N$ Elemente aufgefüllt, in den diskreten Frequenzbereich transformiert und bilden so die beiden Spaltenvektoren $\mathbf{g} = \mathcal{F}_{\ddot{U}N}\{\mathfrak{g}(n_\ddot{u})\}$ und $\mathbf{h}_\mathrm{t} = \mathcal{F}_{\ddot{U}N}\{\mathfrak{h}_t(n_\ddot{u})\}$. Für den überabgetasteten Empfangsvektor $\mathbf{x}_\mathrm{ü}$ am Ausgang des Empfangsfilters gilt dann:

$$\mathbf{x}_\mathrm{ü} = \operatorname{diag}(\mathbf{g}) \operatorname{diag}(\mathbf{h}_\mathrm{t}) \operatorname{diag}(\mathbf{g})\, \mathbf{d}_\mathrm{ü} \tag{2.13}$$

Bei der Abtastung mit der Chiprate (entsprechend Gleichung (2.10)) kommt es zu einer $\ddot{U}$-fachen Überlagerung der im Sender wiederholten Spektralanteile. Für die einzelnen Elemente des resultierenden Vektors gilt:

$$[\mathbf{x}_\mathrm{c}]_n = \sum_{\ddot{u}=0}^{\ddot{U}-1} [\mathbf{x}_\mathrm{ü}]_{n+\ddot{u}N} = [\mathbf{d}_\mathrm{c}]_n \underbrace{\sum_{\ddot{u}=0}^{\ddot{U}-1} [\mathbf{g}]_{n+\ddot{u}N} [\mathbf{h}_\mathrm{t}]_{n+\ddot{u}N} [\mathbf{g}]_{n+\ddot{u}N}}_{[\mathbf{h}_\mathrm{eff}]_n} \tag{2.14}$$

Der Anteil des Chipvektors kann nach Gleichung (2.12) aus der Summe herausgezogen werden. Die übrigen Anteile der Summe werden zu effektiven Kanalkoeffizienten zusammengefasst. Der Empfangsvektor berechnet sich dann einfach aus der elementweisen Multiplikation von Sendevektor und effektivem Kanalvektor:

$$\mathbf{x}_\mathrm{c} = \mathrm{diag}(\mathbf{h}_\mathrm{eff})\,\mathbf{d}_\mathrm{c} \tag{2.15}$$

An der Zusammensetzung der effektiven Kanalkoeffizienten in Gleichung (2.14) wird das im letzten Kapitel bereits angesprochene Problem einer Entzerrung bei Chiprate deutlich: durch eine ungünstige Überlagerung kann es zu einer Auslöschung von Spektralanteilen kommen ($[\mathbf{h}_\mathrm{eff}]_n \approx 0$), obwohl die tatsächlichen Kanalkoeffizienten von Null verschieden sind ($[\mathbf{h}_\mathrm{t}]_{n+\ddot{u}N} \neq 0$ für $\ddot{u} = 0 \ldots \ddot{U}-1$). Die Gewichtung mit einem kanalangepassten Filter vor der Unterabtastung führt dagegen zu einer optimalen Überlagerung (*maximum ratio combining*) der Summanden. Die Überabtastung im Sender entspricht einer Frequenzdiversität, die bei einer Entzerrung mit Chiprate nicht genutzt wird. Die theoretisch erzielbare Frequenzdiversität ist dabei abhängig von der Flankensteilheit von Pulsform- und Empfangsfilter: steile Flanken unterdrücken die wiederholten Spektralanteile sehr stark, eine Nutzung der Diversität ist dann kaum möglich, der Gewinn einer überabgetasteten Entzerrung gegenüber einer Entzerrung bei Chiprate ist dann gering.

Gleichung (2.15) stellt eine einfache Beschreibung der Übertragung für einen Teilnehmer dar. Das Modell geht im Sinne einer aufwandseffizienten Entzerrung im Empfänger von einer Abtastung auf Chiprate aus. Der dabei wichtige Einfluss der überabgetasteten Pulsform- und Empfangsfilter wird in den effektiven Kanalkoeffizienten vollständig berücksichtigt. Voraussetzung ist die zyklische Erweiterung mit der Bedingung in Gleichung (2.8) sowie die vollständige Synchronisation von Sender und Empfänger. Während hier zunächst noch ein zeitdiskreter und invarianter Kanal angenommen wurde, folgt in Kapitel 2.3 eine allgemeinere Modellierung des Kanals.

2.2.3 Anwendung auf Mehrträgermodulation

Das oben beschriebene Übertragungsmodell wird nun auf eine Mehrträgermodulation mit N Unterträgern angewendet. In der Art der Überabtastung und der linearen Faltung zur Pulsformung unterscheidet sich der hier gewählte Ansatz von der klassischen OFDM Übertragungstechnik [42]. Die einheitliche Modellierung erlaubt allgemeingültige Ansätze für die Frequenzbereichsentzerrung. Für die Realisierung hybrider Systeme mit Ein- und Mehrträgermodulation können gemeinsame Komponenten für die Signalverarbeitung verwendet werden. Im Folgenden werden die Unterschiede zur sonst üblichen Praxis erläutert und die Schnittstellen zu dem bestehenden Modell definiert.

In einem OFDM-System werden einzelne Datenblöcke mit mehreren zueinander or-

thogonalen Unterträgern übertragen. Jeder Unterträger wird in Amplitude und Phase moduliert und trägt damit die Information eines QAM-Symbols. Für die Realisierung dieser Modulation wird üblicherweise eine inverse Fourier-Transformation auf den Datenvektor mit den QAM-Symbolen angewendet.

Werden die äußeren Unterträger nicht moduliert, also mit einer Amplitude von Null gewichtet, so ist das Ausgangssignal bandbegrenzt. Ein harter Übergang zwischen aufeinanderfolgenden Datenblöcken stört die Bandbegrenzung durch Phasen- und Amplitudensprünge. Um das zu vermeiden, wird oft eine Fensterung verwendet: dabei werden Anfang und Ende eines jeden Datenblockes multiplikativ mit Rampenfunktionen gewichtet. Die Rampen aufeinanderfolgender Datenblöcke überlagern sich und sorgen für ein langsames Ein- bzw. Ausblenden der Blöcke. Der Bereich für die Fensterung verkürzt die nutzbare Länge der zyklischen Erweiterung.

Ein Root-Raised-Cosine-Filter mit einem Roll-Off-Faktor von 0 ermöglicht auch für den hier vorgeschlagenen Ansatz eine unverzerrte Übertragung ohne spektrale Wiederholung der Unterträger. Die lineare Faltung mit dem Pulsformfilter bewirkt im Überlappungsbereich zweier Datenblöcke einen bandbegrenzten Übergang und ist daher mit der Fensterung zu vergleichen. Wie in Gleichung (2.8) zu erkennen ist, verkürzt der Einflußbereich des Pulsformfilters ebenfalls die wirksame Länge der zyklischen Erweiterung. Der Übergang zwischen den Blöcken wird im Empfänger verworfen und hat daher keinen direkten Einfluss auf das Systemmodell. Die Pulsformung hat dagegen einen merklichen Einfluss auf die effektiven Kanalkoeffizienten. Durch Wahl des Roll-Off-Faktors zu 0 kann mit dem einheitlichen Systemmodell aber jederzeit die klassische OFDM-Übertragungstechnik nachgebildet werden.

Der Chipvektor $\mathbf{d}_c$ setzt sich für die Mehrträgermodulation unmittelbar aus den N zu übertragenden Chips zusammen. Der Rest des Modells kann unverändert von der Einträgermodulation übernommen werden.

2.2.4 Erweiterung auf Vektorkanäle

Die Verwendung mehrerer Empfangsantennen und die Einführung einer Spreizung bewirken eine mehrfache Übertragung von Frequenzbereichsdatensymbolen über unterschiedliche Subkanäle. Aus skalaren Kanalkoeffizienten werden Spaltenvektoren. Die Spreizung bedeutet sowohl für Einträger- als auch für Mehrträgerverfahren eine Wiederholung des Spektrums. Der Spreizcode dient einer Vorgewichtung der Unterträger, die nur für den Mehrfachzugriff im Codebereich bei flachen Kanälen notwendig ist. Bei nur einem Nutzer ($K = 1$) kann ein Spreiz- bzw. Antennengewinn am Zugangspunkt genutzt werden um den Signal-zu-Stör-Abstand zu vergrößern.

Außerdem können Schwundeffekte durch Diversität bekämpft werden. Vorraussetzung ist eine frequenzselektive bzw. räumlich selektive Übertragung. Im Mehrnutzerfall ($K > 1$) wird die Mehrfachübertragung teilweise zur Trennung der überlagerten Datenströme genutzt.

Zunächst wird eine Spreizung für die Einträgermodulation betrachtet. Der Spreizfaktor sei Q. Gegeben seien eine Symbolfolge $\mathfrak{d}_s(u)$ mit $U = N/Q$ QAM-Symbolen sowie eine Spreizcodefolge $\mathfrak{c}(n)$ der Länge Q. Die Spreizung entspricht einer Q-fachen Überabtastung von $\mathfrak{d}_s(u)$ und einer anschließenden Faltung mit $\mathfrak{c}(n)$ und führt zu der bereits eingeführten Chipfolge $\mathfrak{d}_c(n)$. Analog zur überabgetasteten Pulsformung entspricht dies im Frequenzbereich einer spektralen Wiederholung des Frequenzbereichssymbolvektors $\mathbf{d}_\mathrm{s} = \mathcal{F}_U\{\mathfrak{d}_s(u)\}$ und einer elementweisen Multiplikation mit $\mathbf{c} = \mathcal{F}_N\{\mathfrak{c}(n)\}$. Dies kann in Matrixschreibweise folgendermaßen dargestellt werden:

$$\mathbf{d}_\mathrm{c} = \mathrm{diag}(\mathbf{c})\,\mathbf{Z}\mathbf{d}_\mathrm{s}, \tag{2.16}$$

wobei $\mathbf{Z}$ eine Wiederholungsmatrix ist, die aus Q übereinandergesetzten Einheitsmatrizen $\mathbf{I}_U$ besteht:

$$\mathbf{Z} = (\mathbf{I}_U^{(0)}, \mathbf{I}_U^{(1)}, \ldots, \mathbf{I}_U^{(Q-1)})^T. \tag{2.17}$$

Jedes Element des Frequenzbereichssymbolvektors wird damit auf Q über das Gesamtspektrum verteilten Unterträgern parallel übertragen. Für die Spreizcodefolge $\mathfrak{c}(n)$ werden häufig Hadamard-Codes eingesetzt. Sie haben nur für den Mehrfachzugriff eine Bedeutung und werden daher erst im nächsten Kapitel näher erläutert.

Für die Mehrträgerübertragung gibt es verschiedene Ansätze für eine Spreizung. In dieser Arbeit wird zwischen einer Zeitbereichs- und einer Frequenzbereichsspreizung unterschieden. Ausgangspunkt ist der Frequenzbereichssymbolvektor $\mathbf{d}_\mathrm{s}$ bestehend aus U QAM-Symbolen. Bei der Zeitbereichsspreizung wird $\mathbf{d}_\mathrm{s}$ direkt in den Zeitbereich transformiert und dort analog zur Einträgerübertragung durch Überabtastung und Faltung gespreizt. Im Frequenzbereich betrachtet wird $\mathbf{d}_\mathrm{s}$ also wiederholt und mit der Fourier-Transformierten $\mathbf{c}$ der Codefolge gewichtet. Für die Einträger- und Mehrträgermodulation mit Zeitbereichsspreizung gilt also für den Codevektor $\mathbf{c}_\mathrm{td}$ mit dem Index td für *time domain*:

$$\mathbf{c}_\mathrm{td} = \frac{1}{\sqrt{Q}}\mathcal{F}_N\{\mathfrak{c}(n)\}. \tag{2.18}$$

Bei der Frequenzbereichsspreizung wird $\mathbf{d}_\mathrm{s}$ ebenfalls wiederholt, dann aber elementweise direkt im Frequenzbereich mit der Codefolge gewichtet. Für die Elemente des Codevektors $\mathbf{c}_\mathrm{fd}$ mit dem Index fd für *frequency domain* gilt:

$$[\mathbf{c}_{\mathrm{fd}}]_{u+qU} = \mathfrak{c}(q) \qquad \text{mit} \qquad u = 0 \ldots U-1 \quad \text{und} \quad q = 0 \ldots Q-1 \tag{2.19}$$

Eine Frequenzbereichsspreizung ist bei Einträgermodulation theoretisch zwar möglich, in der Implementierung aus Aufwandsgründen aber nicht sinnvoll. Mit der Anpassung des Codevektors beschreibt Gleichung (2.16) allgemeingültig alle betrachteten Spreizverfahren. In [44] werden die Spreizverfahren gegenübergestellt. Die Zeitbereichsspreizung hat den Vorteil einer geringeren Signaldynamik, aber den Nachteil einer ungleichmäßigen Leistungsverteilung auf den Unterträgern. Bei der Frequenzbereichsspreizung kann über eine Permutation der Unterträger die Teilnehmertrennung verbessert werden, allerdings nur auf Kosten der Frequenzdiversität.

Das Systemmodell aus Gleichung (2.15) wird nun mit Gleichung (2.16) erweitert:

$$\mathbf{x}_{\mathrm{c}} = \operatorname{diag}(\mathbf{h}_{\mathrm{eff}})\, \mathbf{d}_{\mathrm{c}} = \operatorname{diag}(\mathbf{h}_{\mathrm{eff}})\, \operatorname{diag}(\mathbf{c})\, \mathbf{Z}\mathbf{d}_{\mathrm{s}}. \tag{2.20}$$

Die ungespreizte Übertragung ist für $Q = 1$ mit $\mathbf{d}_{\mathrm{c}} = \mathbf{d}_{\mathrm{s}}$ als Sonderfall in dem erweiterten Modell enthalten.

Nun wird die Übertragung zu einem Zugangspunkt mit M Antennen betrachtet. Das bestehende Modell wird mit unterschiedlichen Kanalimpulsantworten auf jede einzelne Antenne angewendet. Die Übertragung vom Sender zur Antenne m sei durch den effektiven Kanalvektor $\mathbf{h}_{\mathrm{eff}}^{(m)}$ gegeben und führt zu dem Empfangsvektor $\mathbf{x}_{\mathrm{c}}^{(m)}$. Die Gleichungen aller Antennen werden in Matrixnotation zusammengefasst. Zur Vereinfachung der Notation wird die Übertragung getrennt für jedes Frequenzbereichsdatensymbol $d^{(u)} = [\mathbf{d}_{\mathrm{s}}]_u$ betrachtet:

$$\mathbf{x}^{(u)} = \underbrace{\mathbf{H}_{\mathrm{eff}}^{(u)} \mathbf{c}^{(u)}}_{\mathbf{a}^{(u)}} d^{(u)}, \tag{2.21}$$

wobei $\mathbf{x}^{(u)}$ und $\mathbf{c}^{(u)}$ Vektoren der Länge MQ sind und $\mathbf{H}_{\mathrm{eff}}^{(u)}$ eine Blockdiagonalmatrix mit Q Blöcken der Größe $MQ \times 1$ ist. Für die einzelnen Vektor- bzw. Matrixelemente gilt:

$$\left[\mathbf{x}^{(u)}\right]_{m+qM} = \left[\mathbf{x}_{\mathrm{c}}^{(m)}\right]_{u+qU} \tag{2.22}$$

$$\left[\mathbf{H}_{\mathrm{eff}}^{(u)}\right]_{m+qM,q} = \left[\mathbf{h}_{\mathrm{eff}}^{(m)}\right]_{u+qU} \tag{2.23}$$

$$\left[\mathbf{c}_{\mathrm{td}}^{(u)}\right]_{m+q} = [\mathbf{c}_{\mathrm{td}}]_{u+qU} \tag{2.24}$$

$$\left[\mathbf{c}_{\mathrm{fd}}^{(u)}\right]_{m+q} = [\mathbf{c}_{\mathrm{fd}}]_{u+qU} \triangleq \mathfrak{c}(q) \tag{2.25}$$

Die Beiträge der verschiedenen Antennen werden jeweils untereinander zusammengefasst. Aus dem gesamten Spektrum wird die jeweils zu $d^{(u)}$ zugehörige Unterträgergruppe selektiert (entspricht der u-ten Spalte der Wiederholungsmatrix $\mathbf{Z}$). Die Spreizung ist für alle Antennen identisch. Bei der Frequenzbereichsspreizung ist der Spreizcodevektor für alle u gleich, bei der Zeitbereichsspreizung jedoch unterschiedlich. Die Orthogonalität verschiedener Codes bleibt jedoch in allen Teilsystemen erhalten.

Wie in Gleichung (2.21) angedeutet, kann das Produkt aus Kanalmatrix und Codevektor zu einem Übertragungsvektor $\mathbf{a}^{(u)}$ zusammengefasst werden. Dieser Vektor beschreibt für eine Unterträgergruppe den Vektorkanal der Dimension MQ zwischen dem Sender einer Teilnehmerstation und dem Mehrtorempfänger des Zugangspunktes.

2.2.5 Erweiterung auf Mehrfachzugriff

Über einen Mehrfachzugriff im Zeitbereich (TDMA) und im Frequenzbereich (FDMA) können mehrere Datenströme völlig unabhängig d.h. orthogonal voneinander übertragen werden. Bei CDMA werden dagegen mehrere Datenströme zur gleichen Zeit im selben Frequenzband übertragen, jedoch mit Redundanz. So wird üblicherweise das Spektrum der Datenströme durch die Spreizung mehrfach auf verschiedenen Frequenzen übertragen (siehe vorhergehendes Kapitel). Das Prinzip kann aber auch auf eine zeitlich redundante Übertragung angewendet werden.

Die Verwendung orthogonaler Spreizcodes im Sender ermöglicht dann bei frequenzflachen bzw. zeitlich konstanten Übertragungskanälen eine sehr einfache und ideale Trennung im Empfänger. Vorteil gegenüber der unabhängigen Übertragung ist eine Verbesserung des Störabstandes durch den Spreizgewinn. Bei frequenzselektiven bzw. zeitlich veränderlichen Übertragungskanälen geht die Orthogonalität verloren. Im Empfänger muss neben der Entzerrung nun auch eine adaptive Trennung der Datenströme vorgenommen werden. Dies bedeutet einen erheblichen Mehraufwand. Vorteilhaft gegenüber der unabhängigen Übertragung ist die Diversität der redundant übertragenen Daten.

Betrachtet man den Mehrfachzugriff im Raumbereich ergeben sich grundsätzliche Unterschiede. Zunächst ist eine räumlich unabhängige Übertragung nur unter besonderen Voraussetzungen möglich (z.B. orthogonale Polarisation bei Einwegeübertragung), im Allgemeinen wird man aber immer eine Überlagerung der Datenströme vorfinden. Die Verwendung mehrerer Empfangsantennen liefert die zur Trennung der Datenströme notwendige Selektivität ohne Redundanz im Sender und

ermöglicht damit eine tatsächliche Erhöhung der Übertragungskapazität. Da die Redundanz erst im Kanal hinzugefügt wird, können keine orthogonalen Codes eingesetzt werden. In räumlich flache Kanälen, z.B. bei ausgeprägten Sichtverbindungen aus der gleichen Richtung, ist daher keine Trennung möglich.

Im Rahmen dieser Arbeit wird eine Kombination aus Spreizung und räumlicher Trennung für den Mehrfachzugriff eingesetzt. Im vorhergehenden Kapitel wurde ein Vektorkanal mit MQ Gleichungen eingeführt. Dieser ermöglicht einen Mehrfachzugriff mit $K \leq MQ$ gleichzeitig im selben Frequenzband übertragenen Datenströmen, die von unterschiedlichen Teilnehmern aber auch von unterschiedlichen Sendeantennen der Teilnehmer kommen können. Die Überlagerung der Datenströme am Empfänger entspricht einer Summation über die teilnehmerspezifischen Datensymbole $d^{(u,k)}$ und Übertragungsvektoren $\mathbf{a}^{(u,k)}$:

$$\mathbf{x}^{(\mathrm{u})} = \sum_{k=0}^{K-1} \mathbf{a}^{(u,k)} d^{(u,k)}. \tag{2.26}$$

Die Übertragungsvektoren aller Teilnehmer werden zu der System- bzw. Übertragungsmatrix $\mathbf{A}^{(u)}$ zusammengefasst, die Datensymbole zu dem Symbolvektor $\mathbf{d}^{(u)}$:

$$\mathbf{A}^{(u)} = \left(\mathbf{a}^{(u,0)}, \mathbf{a}^{(u,1)}, \ldots, \mathbf{a}^{(u,K-1)}\right) \tag{2.27}$$

$$\mathbf{d}^{(u)} = \left(d^{(u,0)}, d^{(u,1)}, \ldots, d^{(u,K-1)}\right)^T. \tag{2.28}$$

Damit kann die Summe in (2.26) in eine Matrixnotation überführt werden:

$$\mathbf{x}^{(\mathrm{u})} = \mathbf{A}^{(u)} \mathbf{d}^{(u)} \tag{2.29}$$

Analog zur Definition des Übertragungsvektors in (2.21) kann auch die Systemmatrix in ein Produkt aus Kanalmatrix und Codematrix zerlegt werden:

$$\mathbf{A}^{(u)} = \mathbf{H}_{\mathrm{eff}}^{(u)} \mathbf{C}^{(u)} \tag{2.30}$$

Die Kanalmatrix ist nun eine Blockdiagonalmatrix mit Q Blöcken der Dimension $M \times K$. Die Codematrix setzt sich aus Q übereinander gestapelten Diagonalmatrizen zusammen. Für die Zuordnung der von Null verschiedenen Elemente gilt:

$$\left[\mathbf{H}_{\text{eff}}^{(u)}\right]_{m+qM,k+qK} = \left[\mathbf{h}_{\text{eff}}^{(m,k)}\right]_{u+qU} \tag{2.31}$$

$$\left[\mathbf{C}_{\text{td}}^{(u)}\right]_{k+qK,k} = \left[\mathbf{c}_{\text{td}}^{(k)}\right]_{u+qU} \tag{2.32}$$

$$\left[\mathbf{C}_{\text{fd}}^{(u)}\right]_{k+qK,k} = \left[\mathbf{c}_{\text{fd}}^{(k)}\right]_{u+qU} \stackrel{\wedge}{=} \mathfrak{c}(q) \tag{2.33}$$

Die Matrixgleichung in (2.29) entspricht einem Gleichungssystem mit K unbekannten Datensymbolen und MQ Gleichungen. Zur Lösung des Gleichungssystems steht im Empfänger nur der Beobachtungsvektor mit den MQ Empfangssymbolen zur Verfügung, die KMQ Übertragungskoeffizienten sind zunächst noch unbekannt. Die Beobachtung ist in der Regel durch Rauschen gestört. Gleichung (2.29) wird daher mit dem Rauschvektor $\mathbf{n}^{(u)}$ erweitert:

$$\mathbf{x}^{(\mathrm{u})} = \mathbf{A}^{(u)}\mathbf{d}^{(u)} + \mathbf{n}^{(u)} \tag{2.34}$$

Am Empfängereingang wird ein additives weißes gaußsches Rauschen (AWGN) angenommen. Die Rauschsignale an den verschiedenen Toren seien unkorreliert. Durch die Fouriertransformation wird $\mathbf{n}^{(u)}$ zu einem komplex-normalverteilten Zufallsvektor mit unkorrelierten Elementen. Die Rauschleistung ist am Eingang konstant über der Frequenz, wird aber durch das Empfangsfilter gefärbt. Ein root-raised-cosine-Filter führt zu einer Leistungsverteilung entsprechend dem Quadrat der Filtercharakteristik. Die Unterabtastung auf Chiprate bewirkt eine Addition der Leistung überlappender Spektralanteile. Das Quadrat der Filtercharakteristik entspricht aber gerade einer Nyquistflanke, d.h. die Überlappung führt zu einer konstanten Leistung über das gesamte Spektrum.

Die Rauschleistung pro Empfangsantenne und pro Unterträger sei σ_n. Die Leistungsregelung der Teilnehmerstationen stellt am Empfängereingang den Sollwert für das mittlere Signal-zu-Stör-Verhältnis $\widehat{SNR}$ ein. Der Störabstand ist definiert als Verhältnis der mittleren teilnehmerspezifischen Empfangsleistung σ_e pro Antenne und Unterträger zur Rauschleistung σ_n. Mit der mittleren Leistung σ_d der Sendesymbole gilt:

$$\widehat{SNR} = \frac{\sigma_e}{\sigma_n} = \frac{\sigma_d}{\sigma_n}\frac{1}{KMN}\sum_{u=0}^{U-1} E\left\{\|\mathbf{A}^{(u)}\|_F^2\right\} \tag{2.35}$$

Das AWGN-Modell beschreibt thermisches Rauschen. In zellularen Systemen ist auch die Interferenz zwischen benachbarten Zellen zu berücksichtigen. Eine räumlich gerichtete Interferenz durch wenige Störer kann mit geeigneten Verfahren in einem

Mehrtorempfänger sehr effektiv bekämpft werden [45]. Auf diese Option wird in Kapitel 3.3 noch kurz eingegangen.

2.3 Stochastische Modellierung des Kanals

In Kapitel 5 wird der Einfluss der Zahlendarstellung mit begrenzten Wortlängen auf das Systemverhalten untersucht. Die dazu durchgeführten Simulationen beruhen auf einem stochastischen Kanalmodell mit korrelierten Zufallsvariablen. Eine allgemeine Übersicht zur Kanalmodellierung in MIMO Systemen wird in [46] gegeben, das hier verwendete Prinzip z.B. in [47] angewandt.

Für jeden Teilnehmer $k = 0 \ldots K-1$, für jede Antenne des Zugangspunktes $m = 0 \ldots M-1$ und für jeden Datenblock $t = 0 \ldots T-1$ eines Zeitschlitzes wird jeweils eine Impulsantwort $\mathfrak{h}_{iid}^{(m,k,t)}$ definiert. Jede Impulsantwort besteht aus N Taps, wobei jeder Tap eine komplex-normalverteilte Zufallsvariable ist. Die Taps seien mittelwertfrei, statistisch unabhängig und haben die gleiche normierte Varianz von eins, daher auch die Bezeichnung *iid* für *independent identically distributed*. Die komplexe Normalverteilung der Taps, bzw. die Rayleighverteilung ihrer Amplitude, gilt unter der Voraussetzung einer Überlagerung sehr vieler Ausbreitungswege mit gleicher Verzögerung (zentraler Grenzwertsatz).

Die diskrete Fouriertransformation aller Kanalimpulsantworten führt zu den Kanalvektoren

$$\mathbf{h}_{\text{iid}}^{(m,k,t)} = \mathcal{F}_N \left\{ \mathfrak{h}_{iid}^{(m,k,t)} \right\} \tag{2.36}$$

mit identischen statistischen Eigenschaften, d.h. die Zufallswerte können direkt im Frequenzbereich generiert werden. Alle Übertragungsfunktionen werden in einem Vektor zusammengefasst:

$$[\mathbf{h}_{\text{iid}}]_{m+M(k+K(n+Nt))} = \left[\mathbf{h}_{\text{iid}}^{(m,k,t)} \right]_n . \tag{2.37}$$

Die Multiplikation mit der Koeffizientenmatrix $\mathbf{C}_{\text{corr}}$ ergibt den korrelierten Zufallsvektor

$$\mathbf{h}_{\text{corr}} = \mathbf{C}_{\text{corr}} \mathbf{h}_{\text{iid}}. \tag{2.38}$$

Die vollständige Korrelation von jedem Kanalkoeffizienten zu allen anderen wird durch die Korrelationsmatrix $\mathbf{R}_{\text{hh}}$ beschrieben:

$$\mathbf{R}_{\mathrm{hh}} = \mathrm{E}\{\mathbf{h}_{\mathrm{corr}}\mathbf{h}_{\mathrm{corr}}^{H}\} = \mathbf{C}_{\mathrm{corr}} \underbrace{\mathrm{E}\{\mathbf{h}_{\mathrm{iid}}\mathbf{h}_{\mathrm{iid}}^{H}\}}_{\mathbf{I}_{\mathrm{MKNT}}} \mathbf{C}_{\mathrm{corr}}^{H} = \mathbf{C}_{\mathrm{corr}}\mathbf{C}_{\mathrm{corr}}^{H}. \tag{2.39}$$

Die gesamte Korrelationsmatrix wird zerlegt in:

- eine räumliche Korrelationsmatrix am Zugangspunkt $\mathbf{R}_{\mathrm{m}}$
- eine räumliche Korrelationsmatrix an den Teilnehmerstationen $\mathbf{R}_{\mathrm{k}}$
- eine Frequenzkorrelationsmatrix $\mathbf{R}_{\mathrm{n}}$
- eine Zeitkorrelationsmatrix $\mathbf{R}_{\mathrm{t}}$.

Die Korrelationsmatrizen gelten unabhängig voneinander jeweils für das gesamte System, also z.B. eine einheitliche Frequenzkorrelation für alle Teilnehmer an allen Zugangspunktantennen zu allen Zeitpunkten. Unter dieser Voraussetzung entspricht $\mathbf{R}_{\mathrm{hh}}$ dem Kroneckerprodukt der einzelnen Matrizen:

$$\mathbf{R}_{\mathrm{hh}} = \mathbf{R}_{\mathrm{t}} \otimes \mathbf{R}_{\mathrm{n}} \otimes \mathbf{R}_{\mathrm{k}} \otimes \mathbf{R}_{\mathrm{m}} \tag{2.40}$$

Für die räumliche Korrelation wird ein regelmäßiger Antennenabstand angenommen. Der Korrelationskoeffizient ist dann exponentiell abhängig von der Differenz der Antennenindizes:

$$[\mathbf{R}_{\mathrm{m}}]_{z,s} = \gamma_m^{|z-s|} \quad \text{mit} \quad z,s = 0 \ldots M-1 \tag{2.41}$$

$$[\mathbf{R}_{\mathrm{k}}]_{z,s} = \gamma_k^{|z-s|} \quad \text{mit} \quad z,s = 0 \ldots K-1, \tag{2.42}$$

wobei γ_m und γ_k der Korrelation benachbarter Antennen am Zugangspunkt bzw. den Teilnehmerstationen entsprechen. Für ein Mehrteilnehmersystem mit Einzelantennen an unabhängigen Teilnahmerstationen ist γ_k zu Null zu wählen.

Für die Frequenzkorrelation wird ein Modell mit exponentiell abfallender Leistungsverteilung der Impulsantworten zugrundegelegt. Wird ein zeitkontinuierlicher statistisch unabhängiger Zufallsprozess mit einem Exponentialimpuls (Zeitkonstante τ) gewichtet, so führt dies zwischen zwei Spektrallinien mit dem Abstand Δf zu einer Korrelation von:

$$\gamma(\Delta f) = \frac{1}{1 - j\pi\tau\Delta f} \tag{2.43}$$

Für eine zweifache Überabtastung kann damit eine diskrete Frequenzkorrelationsmatrix $\mathbf{R}_{2\mathrm{n}}$ konstruiert werden:

$$[\mathbf{R}_{2\mathrm{n}}]_{z,s} = \frac{1}{1 - j\pi\frac{\tau}{T_c}\frac{|z-s|}{N}} \qquad \text{mit} \qquad z,s = 0 \ldots 2N-1 \tag{2.44}$$

Der Übergang zu den effektiven Kanalkoeffizienten durch die Unterabtastung im Empfänger wird schließlich durch Multiplikation mit der Wiederholungsmatrix $\mathbf{Z}_2 = (\mathbf{I}_\mathrm{N}, \mathbf{I}_\mathrm{N})^T$ und der Diagonalmatrix $\mathbf{G}$ beschrieben:

$$\mathbf{R}_\mathrm{n} = \frac{N}{\mathrm{spur}(\mathbf{Z}_2^H \mathbf{G}^H \mathbf{R}_{2\mathrm{n}} \mathbf{G} \mathbf{Z}_2)} \mathbf{Z}_2^H \mathbf{G}^H \mathbf{R}_{2\mathrm{n}} \mathbf{G} \mathbf{Z}_2 \tag{2.45}$$

Die Diagonale von $\mathbf{G}$ enthält das Produkt der Übertragungsfaktoren von Pulsform- und Empfangsfilter von $-f_c \ldots f_c$. Die Matrix $\mathbf{R}_\mathrm{n}$ beschreibt die Korrelation der effektiven Kanalkoeffizienten für einen zeitkontinuierlichen Kanal. Das ideale, exponentiell abfallende Leistungsspektrum des Kanals wird mit den Impulsantworten von Pulsform- und Empfangsfilter verzerrt. An den Bandgrenzen kommt es zu einem Abfall der mittleren Empfangsleistung. Mit dem Vorfaktor wird die Gesamtleistung auf N normiert.

Anstatt der normierten Zeitkonstante $\frac{\tau}{T_c}$ in Gleichung (2.44) wird im Folgenden die Dämpfung α pro Chipdauer in Dezibel verwendet. Der Zusammenhang lautet:

$$\frac{\tau}{T_c} = \frac{20}{\alpha \ln(10)} \tag{2.46}$$

Ein zeitveränderlicher Übertragungskanal stört mit Dopplerverschiebungen die Orthogonalität der Unterträger. Mit einem Unterträgerabstand größer der zehnfachen maximalen Dopplerfrequenz $f_{d,max}$ kann dieser Effekt vernachlässigt werden. Die zeitliche Veränderung wird stückweise konstant modelliert: keine Veränderung während der Übertragung eines Datenblockes, sprunghafte Änderungen zwischen den Datenblöcken. Für das Dopplerspektrum wird das weit verbreitete Modell von Jakes [48] verwendet. Die Korrelation zwischen zwei Zeitpunkten ist hier durch die Besselfunktion erster Ordnung J_0 gegeben. Damit lässt sich eine diskrete Korrelationsmatrix konstruieren, die die Korrelation zwischen den Kanälen verschiedener Datenblöcke beschreibt:

$$[\mathbf{R}_\mathrm{t}]_{z,s} = J_0\Big(\underbrace{2\pi\frac{f_{d,max}}{f_b}}_{\beta_{d,max}}(z-s)\Big) \qquad \text{mit} \qquad z,s = 0 \ldots T-1 \tag{2.47}$$

Das Phasenmaß $\beta_{d,max}$ entspricht der maximalen Phasendrehung auf einem Ausbreitungsweg für die Dauer eines Datenblockes. Der Zusammenhang zur Geschwindigkeit v der bewegten Teilnehmer lautet:

$$\beta_{d,max} = 2\pi \frac{v}{c} \frac{f_t}{f_b}, \tag{2.48}$$

mit der Lichtgeschwindigkeit c und der Trägerfrequenz f_t.

Nach Festlegung aller Korrelationsmatrizen werden die entsprechenden Koeffizientenmatrizen über eine Matrixwurzel berechent. Die Operation der Matrixwurzel ist für symmetrische Matrizen eindeutig. Realisierungen der Zufallsvariable $\mathbf{h}_{\text{iid}}$ werden ausgewürfelt und durch Multiplikation mit $\mathbf{C}_{\text{corr}}$ korreliert. Die jeweiligen effektiven Kanalkoeffizienten werden dann entsprechend der Zusammensetzung aus Gleichung (2.37) dem korrelieren Zufallsvektor entnommen.

Die Normierung der Korrelationsmatrizen führt zu einer mittleren Empfangsleistung σ_e pro Teilnehmer, Empfangsantenne und Unterträger von

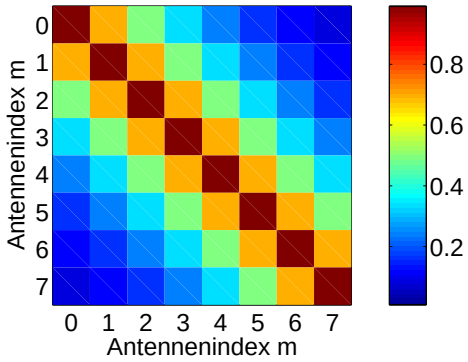
$$\sigma_e = \frac{\sigma_d}{KMN} \sum_{u=0}^{U-1} E\left\{\|\mathbf{A}^{(u)}\|_F^2\right\} = \frac{\sigma_d}{KMNT} \operatorname{spur}(\mathbf{R}_{\text{hh}}) = \sigma_d. \tag{2.49}$$

Die in den Gleichungen (2.18) und (2.19) definierten Spreizsequenzen sind so normiert, dass sie keinen Einfluss auf die mittlere Empfangsleistung haben.

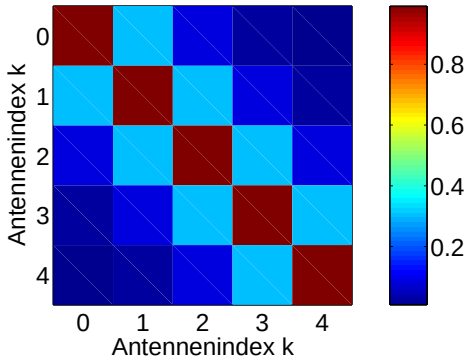
In Bild 2.2 sind die vier Korrelationsmatrizen für ein Beispiel dargestellt. Die räumliche Korrelation nimmt erwartungsgemäß mit zunehmendem Antennenabstand ab. Der Abfall in der Mitte der Hauptdiagonalen von $\mathbf{R}_{\text{n}}$ gibt die bereits erläuterte Reduktion der mittleren Leistung an den Bandgrenzen aufgrund der Unterabtastung wieder. Bei einer zeitlich begrenzten Impulsantwort ist der zyklische Frequenzgang zudem stetig, daher die erneute Zunahme der Korrelation zu den Ecken hin. Die zeitliche Korrelation zeigt dagegen eine konstante Leistung und konvergiert mit großen Abständen zu Null.

2.4 Fazit

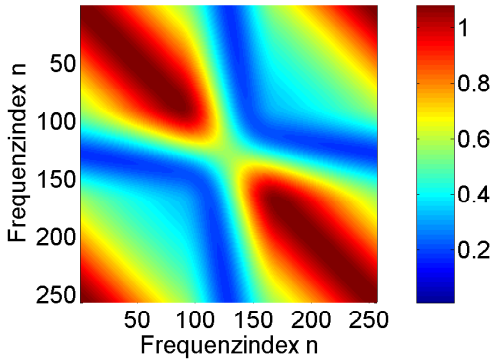
Das Systemmodell beschreibt für den Uplink eines Mehrnutzersystems den Zusammenhang zwischen den gesendeten Datensymbolen der Teilnehmerstationen und des Empfangssignals am Zugangspunkt. Die Beschreibung erfolgt direkt im Frequenzbereich, jeweils getrennt für zusammenhängende Unterträgergruppen. Das Modell gilt universell für die Ein- und Mehrträgermodulation. Anders als sonst üblich wird dazu auch bei der Mehrträgermodulation im Zeitbereich überabgetastet und zur Spektralformung kontinuierlich mit einem Pulsformfilter gefaltet. Durch Wahl eines idealen Tiefpasses ist auch der klassische Ansatz im Modell enthalten. Während das Modell im Empfänger eine Abtastung im Symboltakt voraussetzt, werden die Effekte



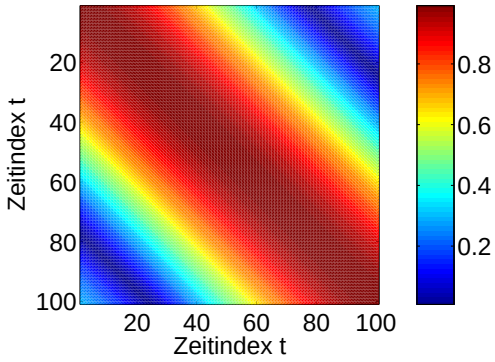
(a) räumliche Korrelation $\mathbf{R}_{\mathrm{m}}$ am Zugangspunkt für $\gamma_m = 0.7$



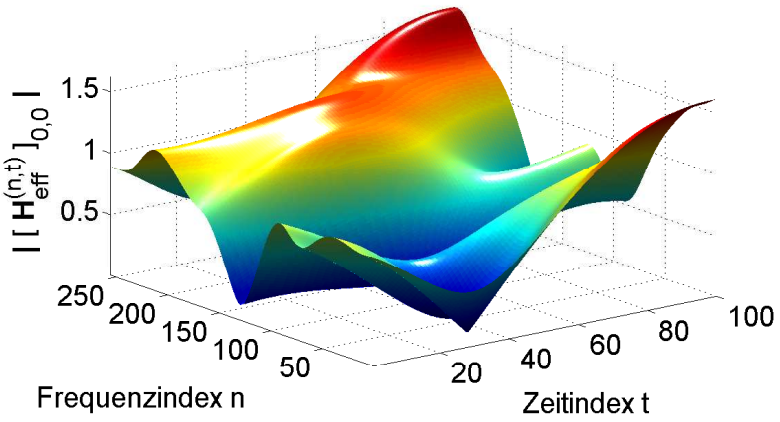
(b) räumliche Korrelation $\mathbf{R}_{\mathrm{k}}$ an den Antennen der Teilnehmerstationen für $\gamma_k = 0.3$



(c) Korrelation über die Frequenz $\mathbf{R}_{\mathrm{n}}$ für $\alpha = 6\,\mathrm{dB}$ und $r = 0{,}33$



(d) Korrelation über die Zeit $\mathbf{R}_{\mathrm{t}}$ für $\beta_{d,max} = 0{,}005$



(e) Beispiel einer Übertragungsfunktion über die Frequenz n und die Zeit t für m,$k = 0$

Bild 2.2: Darstellung der vier Korrelationsmatrizen für $M = 8$, $K = 5$, $N = 256$ und $T = 100$ sowie eine exemplarische Zufallsrealisierung des resultierenden Kanals. Die Korrelationsmatrix für den gesamten Zufallsprozess entspricht dem Kronecker-Produkt der vier Einzelmatrizen.

der Überabtastung und Pulsformfilter über effektive Kanalkoeffizienten vollständig abgebildet. Die berücksichtigten Spreizverfahren MC-CDMA und TD/CDMA unterscheiden sich nur durch eine Fourier-Transformation der Spreizsequenzen.

Für die Systemsimulationen in Kapitel 5 wurde ein einfach parametrierbares, stochastisches Kanalmodell eingeführt. Das Modell basiert auf korrelierten, komplexnormalverteilten Zufallsvariablen nach dem weit verbreiteten WSSUS-Prinzip (*wide sense stationary uncorrelated scattering*). Einfache, parametrierbare Modellannahmen legen die räumliche Korrelation zwischen den Antennen von Zugangspunkt und Teilnehmerstationen sowie die Korrelation über die Frequenz und die Zeit in getrennten Korrelationsmatrizen fest. Die Verknüpfung der vier Korrelationsmatrizen über ein Kronecker-Produkt beschreibt dann die Korrelation sämtlicher Kanalkoeffizienten.

Die einstellbaren Parameter des Kanalmodells sind zusammengefasst:

- die räumliche Korrelation benachbarter Antennen am Zugangspunkt γ_m und an den Teilnehmerstationen γ_k
- das auf die Chiplänge normierte Dämpfungsmaß α für die exponentiell abfallende mittlere Leistung der Impulsantwort
- die Übertragungsfunktionen von Pulsform und Empfangsfilter (bei Raised-Cosine-Filtern gegeben durch den Rolloff-Faktor r)
- das Phasenmaß $\beta_{d,max}$ mit der auf die Blockrate normierten maximalen Dopplerfrequenz zur Parametrierung des Jakes-Spektrums

KAPITEL 3

Lineare Frequenzbereichsentzerrung

Auf Grundlage des Systemmodells kann nun in Kapitel 3.1 das eigentliche Optimierungsproblem definiert werden. In Kapitel 3.2 werden dann Lösungsvorschläge auf Basis einer Kanalschätzung vorgestellt. Alternativ kann die Lösung über adaptive Filter durch die iterative Minimierung des Fehlers erfolgen. In Kapitel 3.3 werden dazu die zwei grundlegenden Algorithmen LMS und RLS eingeführt und mögliche Anwendungen beschrieben. Für die räumlich-zeitliche Vorverzerrung ergibt sich ein ähnliches, keinesfalls aber gleiches Optimierungsproblem, das in Kapitel 3.5 diskutiert wird. Schließlich folgt in Kapitel 3.6 eine tabellarische Aufstellung und Bewertung des Berechnungsaufwandes der vorgestellten Verfahren.

Ziel des Kapitels ist die Identifikation typischer Algorithmen für die lineare Frequenzbereichsentzerrung und ihrer Eigenschaften zur anwendungsspezifischen Optimierung des Parallelprozessors.

3.1 Optimierungsproblem

Ausgangspunkt für das Optimierungsproblem im Empfänger des Zugangspunktes ist das Übertragungsmodell nach Gleichung (2.34). Die Entkopplung der Unterträger im Frequenzbereich erlaubt eine getrennte Lösung des Problems für jeden Symbolvektor $\mathbf{d}^{(u)}$. Der Index u wird zur Vereinfachung im Folgenden weggelassen, d.h.:

$$\mathbf{x} = \mathbf{A}\mathbf{d} + \mathbf{n} \tag{3.1}$$

Zunächst wird der stationäre Fall betrachtet, d.h. die Systemmatrix $\mathbf{A}$ sei konstant. Ziel der Entzerrung ist eine Schätzung $\hat{\mathbf{d}}$ des Sendesymbolvektors $\mathbf{d}$ auf Grundlage des beobachteten Empfangsvektors $\mathbf{x}$. Wie in Kapitel 1.3 erläutert werden hier nur lineare Verfahren betrachtet, bei denen die Schätzwerte durch eine Linearkombination der empfangen Symbole berechnet werden. Die Linearkombination entspricht einer Multiplikation mit der Gewichtsmatrix $\mathbf{W}$:

$$\hat{\mathbf{d}} = \mathbf{W}\mathbf{x} \tag{3.2}$$

Zur Optimierung der Gewichtsfaktoren wird ein Fehlervektor $\mathbf{e}$ als Differenz zwischen Sollwert und Schätzwert definiert:

$$\mathbf{e} = \mathbf{d} - \hat{\mathbf{d}} \tag{3.3}$$

Die Minimierung des mittleren quadratischen Fehlers (*minimum mean square error*, MMSE) dient im Folgenden als Optimierungskriterium. Es wird eine entsprechende Fehlerfunktion $J(\mathbf{W})$ definiert:

$$J(\mathbf{W}) = E\left\{\mathbf{e}^H\mathbf{e}\right\} = E\left\{\|\mathbf{e}\|^2\right\} \tag{3.4}$$

Die Fehlerfunktion ist eine skalare Funktion in 2. Ordnung abhängig von den QMK Elementen der Gewichtsmatrix; sie ist konkav und besitzt genau ein globales Minimum. Die Suche nach diesem Minimum führt zu der im Sinne des mittleren quadratischen Fehlers optimalen Gewichtsmatrix $\mathbf{W}_{\mathrm{MMSE}}$:

$$\mathbf{W}_{\mathrm{MMSE}} = \arg\min_{\mathbf{W}} J(\mathbf{W}) = \arg\min_{\mathbf{W}} E\left\{\|\mathbf{d} - \mathbf{W}\mathbf{x}\|^2\right\} \tag{3.5}$$

Die Lösung zu diesem Problem wird durch die Wiener-Gleichung beschrieben:

$$\mathbf{W}_{\mathrm{MMSE}} = \mathbf{R}_{\mathrm{dx}}\mathbf{R}_{\mathrm{xx}}^{-1} \tag{3.6}$$

Das Einsetzen der Übertragungsgleichung (3.1) führt zu folgenden Korrelationsmatrizen:

$$\mathbf{R}_{\mathrm{dx}} = E\left\{\mathbf{d}\mathbf{x}^H\right\} = \mathbf{R}_{\mathrm{dd}}\mathbf{A}^H + \mathbf{R}_{\mathrm{dn}} \tag{3.7}$$

$$\mathbf{R}_{\mathrm{xx}} = E\left\{\mathbf{x}\mathbf{x}^H\right\} = \mathbf{A}\mathbf{R}_{\mathrm{dd}}\mathbf{A}^H + \mathbf{R}_{\mathrm{nn}} \tag{3.8}$$

Da die Datensymbole nicht mit dem Rauschen korreliert sind, kann $\mathbf{R}_{\mathrm{dn}}$ zu Null gesetzt werden. Für die optimale Gewichtsmatrix folgt [49]:

$$\mathbf{W}_{\mathrm{MMSE}} = \mathbf{R}_{\mathrm{dd}}\mathbf{A}^H\left(\mathbf{A}\mathbf{R}_{\mathrm{dd}}\mathbf{A}^H + \mathbf{R}_{\mathrm{nn}}\right)^{-1} = \left(\mathbf{A}^H\mathbf{R}_{\mathrm{nn}}^{-1}\mathbf{A} + \mathbf{R}_{\mathrm{dd}}^{-1}\right)^{-1}\mathbf{A}^H\mathbf{R}_{\mathrm{nn}}^{-1} \tag{3.9}$$

Unter der Voraussetzung, dass die Datensymbole der Teilnehmer und die Elemente des Rauschvektors untereinander jeweils unkorreliert sind und jeweils die gleiche mittlere Leistung (σ_d bzw. σ_n) haben, gilt:

$$\mathbf{R}_{\mathrm{dd}} = \sigma_d\mathbf{I}_{\mathrm{K}} \qquad \text{und} \qquad \mathbf{R}_{\mathrm{nn}} = \sigma_n\mathbf{I}_{\mathrm{MQ}} \tag{3.10}$$

Die allgemeingültige MMSE-Lösung nach Gleichung (3.9) vereinfacht sich zu:

$$\mathbf{W}_{\mathrm{MMSE}} = \left(\mathbf{A}^H\mathbf{A} + \varepsilon\mathbf{I}_{\mathrm{K}}\right)^{-1}\mathbf{A}^H = \mathbf{\Delta}^{-1}\mathbf{A}^H \qquad \text{mit} \qquad \epsilon = \frac{\sigma_n}{\sigma_d} \tag{3.11}$$

Rauschleistung und Symbolleistung sind über das SNR und die Skalierung der Systemmatrix verknüpft. Der allgemeine Zusammenhang ist in Gleichung (2.35) zu finden. Für die im stochastischen Kanalmodell gewählte Normierung aus Gleichung (2.49) gilt speziell:

$$\epsilon = \frac{1}{\widehat{SNR}} \tag{3.12}$$

Die Singulärwertzerlegung der Systemmatrix ermöglicht wesentliche Erkenntnisse über die Lösbarkeit des Optimierungsproblems:

$$\mathbf{A} = \mathbf{U}_{\mathrm{A}}\mathbf{\Sigma}_{\mathrm{A}}\mathbf{V}_{\mathrm{A}}^H \tag{3.13}$$

Die K Singulärwerte in $\mathbf{\Sigma}_{\mathrm{A}}$ sind nach abnehmendem Betrag sortiert. Die ersten K Singulärvektoren in $\mathbf{U}_{\mathrm{A}}$ bilden dann den Signalunterraum $\mathbf{U}_{\mathrm{As}}$, die übrigen $MQ-K$ Vektoren den Rauschunterraum $\mathbf{U}_{\mathrm{An}}$:

$$\sigma_A^{(1)} \geq \sigma_A^{(2)} \geq \ldots \geq \sigma_A^{(K)} \qquad \text{und} \qquad \mathbf{U}_{\mathrm{A}} = [\ \underbrace{\mathbf{U}_{\mathrm{As}}}_{[MQ\times K]}\ ,\ \underbrace{\mathbf{U}_{\mathrm{An}}}_{[MQ\times MQ-K]}\ ] \tag{3.14}$$

Es wird nun der Zusammenhang zur Eigenwertzerlegung der Korrelationsmatrix $\mathbf{R}_{\mathrm{xx}}$ und der Δ-Matrix aus Gleichung (3.11) betrachtet:

$$\mathbf{R}_{\mathrm{xx}} = \mathbf{U}_{\mathrm{R}}\mathbf{\Lambda}_{\mathrm{R}}\mathbf{U}_{\mathrm{R}}^H \qquad \text{und} \qquad \mathbf{\Delta} = \mathbf{U}_{\Delta}\mathbf{\Lambda}_{\Delta}\mathbf{U}_{\Delta}^H \tag{3.15}$$

Die Eigenwerte λ_R und λ_Δ auf den Diagonalen der Matrizen $\mathbf{\Lambda}_{\mathrm{R}}$ bzw. $\mathbf{\Lambda}_{\Delta}$ seien ebenfalls nach abnehmendem Betrag sortiert. Die ersten K Spaltenvektoren von $\mathbf{U}_{\mathrm{R}}$ sind bis auf das Vorzeichen identisch zu den Singulärvektoren in $\mathbf{U}_{\mathrm{As}}$, d.h. sie spannen den Signalunterraum auf. Für die Eigenwerte gilt:

$$\lambda_R^{(k)} = \begin{cases} \lambda_\Delta^{(k)} = {\sigma_A^{(k)}}^2 + \frac{\sigma_d}{\sigma_n} & \text{für} \quad k = 0..K-1 \\ \frac{\sigma_d}{\sigma_n} & \qquad\quad k = K..MQ-1 \end{cases} \tag{3.16}$$

Für symmetrische Matrizen sind Eigenwertzerlegung und Singulärwertzerlegung identisch. Die Kondition in der 2-Norm κ_2 einer Matrix ist definiert als Quotient zwischen größtem und kleinstem Singulärwert bzw. den entsprechenden Eigenwerten für symmetrische Matrizen.

Für die Kondition der betrachteten Matrizen gilt also:

$$\kappa_2(\mathbf{A}) = \frac{\sigma_A^{(1)}}{\sigma_A^{(K)}} \tag{3.17}$$

$$\kappa_2(\mathbf{R}_{\mathrm{xx}}) = 1 + \frac{\sigma_d}{\sigma_n}{\sigma_A^{(1)}}^2 \tag{3.18}$$

$$\kappa_2(\mathbf{\Delta}) = \frac{1 + \frac{\sigma_d}{\sigma_n}{\sigma_A^{(1)}}^2}{1 + \frac{\sigma_d}{\sigma_n}{\sigma_A^{(K)}}^2} \tag{3.19}$$

Der minimale Fehler im Optimum der Fehlerfunktion ergibt sich nach [49] allgemein zu:

$$J(\mathbf{W}_{\mathrm{MMSE}}) = \mathrm{spur}\left(\mathbf{R}_{\mathrm{dd}} - \mathbf{R}_{\mathrm{dx}}\mathbf{R}_{\mathrm{xx}}^{-1}\mathbf{R}_{\mathrm{dx}}^{H}\right) \tag{3.20}$$

Für den Spezialfall (3.10) gilt:

$$J(\mathbf{W}_{\mathrm{MMSE}}) = \sigma_d\,(K - \mathrm{spur}(\mathbf{W}_{\mathrm{MMSE}}\mathbf{A})) \tag{3.21}$$

Die MMSE-Lösung geht für kleine SNR in das *matched filter* (MF), für hohe SNR in den *zero forcer* (ZF) über. Das MF maximiert die Empfangsleistung ohne Rücksicht auf die Mehrfachzugriffsinterferenz. Der ZF erzwingt dagegen die Auslöschung der Mehrfachzugriffsinterferenz ohne das Rauschen zu berücksichtigen. Die MMSE-Lösung liefert immer den optimalen Abtausch zwischen den beiden Ansätzen und minimiert die Gesamtstörung durch Rauschen und Mehrfachzugriff.

3.2 Lösung über Kanalschätzung

Bei Kenntnis der momentanen Systemmatrix sowie der Autokorrelationsmatrizen von Sende- und Rauschvektor kann die Wienerlösung über (3.9) direkt berechnet werden. Im Folgenden wird der unkorrelierte Fall (3.10) mit der speziellen Wienerlösung nach (3.11) betrachtet. Das Verhältnis $\frac{\sigma_n}{\sigma_d}$ kann über den Pegel des thermischen Rauschens sowie die aktuelle Einstellung der Leistungsregelung als bekannt vorausgesetzt werden. Eine mögliche Färbung des Rauschens durch das Empfangsfilter ist ebenfalls bekannt und muss ggf. berücksichtigt werden.

Im Folgenden werden Verfahren für die Schätzung der Systemmatrix und Berechnungsmethoden für die anschließende Inversion sowie die Lösung des Problems über eine Zerlegung der Matrix vorgestellt. Schließlich wird eine Interpolation über die Frequenz zur Reduktion des Berechnungsaufwandes vorgeschlagen.

3.2.1 Orthogonale Kanalschätzung

Die Lösung des MMSE-Problems über eine Matrixinversion erfordert eine vollständige Schätzung der Kanalmatrix, d.h. aller Kanalkoeffizienten des Vektorkanals für jeden im aktuellen Zeitschlitz aktiven Teilnehmer. Die Verwendung von Trainingssequenzen reduziert den nutzbaren Datendurchsatz, ist in der Regel aber effektiver als blinde Verfahren. Durch eine geeignete Wahl der Trainingssequenzen können die Kanäle der verschiedenen Teilnehmer unabhängig voneinander, d.h. ohne Mehrfachzugriffsinterferenz, geschätzt werden. Alternativ besteht die Möglichkeit bei voller Interferenz den Kanal über iterative Verfahren zu schätzen (siehe Kapitel 3.3.4).

Das für UTRA-TDD vorgesehene Verfahren mit optimierten, zyklischen Trainingssequenzen [50] erlaubt die Schätzung der Kanalimpulsantworten mehrerer Teilnehmer mit minimaler Mehrfachzugriffsinterferenz. Die Transformation der geschätzten Impulsantworten in den Frequenzbereich ist jedoch recht aufwändig. Für die Entzerrung im Frequenzbereich ist es effizienter, auch die Kanalschätzung direkt im Frequenzbereich durchzuführen.

Wenn zu einem Zeitpunkt auf jedem Unterträger jeweils nur ein Teilnehmer ein Trainingssymbol sendet, können die Kanalkoeffizienten im Empfänger über die Division der Empfangswerte durch die entsprechenden Trainingssymbole geschätzt werden. Für die Schätzung von NK Kanalkoeffizienten werden dann K Blöcke mit Trainingssymbolen benötigt.

Aufgrund der Korrelation zwischen den Unterträgern reicht es aus, einige gleichmäßig über das Spektrum verteilte Stützstellen zu schätzen. Die Zwischenwerte können dann interpoliert werden (siehe dazu auch Kapitel 3.4). Die Interpolation wird in klassischen OFDM-Systemen eingesetzt, um mit wenigen Trainingssymbolen die vollständige Kanalkenntnis für die auf den übrigen Unterträgern übertragenen Daten zu erlangen. Die Mittelung der Schätzung über mehrere Blöcke verringert den Schätzfehler bei zu geringem Störabstand. Eine zweidimensionale Interpolation über Frequenz und Zeit ermöglicht die Anpassung an zeitveränderliche Kanäle.

Das Prinzip kann unmittelbar für die orthogonale Kanalschätzung mehrerer Nutzer angewendet werden, indem die Teilnehmer ihre Trainingssymbole auf nicht überlappenden Kammspektren senden. Die Anzahl der benötigten Trainingssymbole kann so um den Interpolationsfaktor reduziert werden. In einem Mehrantennenempfänger erfolgt die Schätzung parallel für jede Empfangsantenne.

Bei der Wahl der Trainingssequenzen ist das Verhältnis zwischen Spitzenleistung und mittlerer Leistung (Crestfaktor oder PAPR, *peak-to-average-power-ratio*) zu beachten. Der Crestfaktor muss möglichst klein gehalten werden, um einen ausrei-

chenden Störabstand und damit eine zuverlässige Kanalschätzung zu gewährleisten. Bei weißem Rauschen ist es sinnvoll, die Amplitude aller Trainingssymbole gleich groß zu wählen. Wenn neben der Amplitude auch die Phase der Trainingssymbole gleich ist, so führt das zum maximal möglichen Crestfaktor. Die Phasen der Trainingssymbole müssen also auf einen minimalen Crestfaktor hin optimiert werden. Eine Zufallssequenz mit gleichverteilter Phase bringt bereits eine wesentliche Verbesserung.

Die geschätzten Kanalkoeffizienten müssen entsprechend Gleichung (2.30) noch mit den Spreizsequenzen gewichtet werden um die Systemmatrix zu erhalten. Eine Spreizung der Trainingsequenzen im Sender würde dies erübrigen, führt aber zu einer Verlängerung der effektiven Kanallänge und erschwert damit die Interpolation. Für die Frequenzbereichsspreizung ist die Länge der Spreizcodes im Zeitbereich unbegrenzt, eine Interpolation ist dann nicht mehr möglich.

3.2.2 Lösung durch Inversion der Systemmatrix

Die direkte Berechnung der MMSE-Lösung erfordert eine Inversion der Matrix $\mathbf{\Delta}$ der Dimension $[K \times K]$. Die Matrix ist symmetrisch, positiv definit und kann über eine Cholesky- oder LDL-Zerlegung in Dreiecksmatrizen zerlegt werden:

$$\mathbf{\Delta} = \underbrace{\mathbf{G}\mathbf{G}^H}_{\text{Cholesky-Zerlegung}} = \underbrace{\mathbf{L}\mathbf{D}\mathbf{L}^H}_{\text{LDL-Zerlegung}} \tag{3.22}$$

Dabei sind $\mathbf{G}$ und $\mathbf{L}$ untere Dreiecksmatrizen, $\mathbf{D}$ ist eine Diagonalmatrix. Beide Zerlegungen sind eng verwandt und unterscheiden sich nur in ihrer Normierung: im Gegensatz zur Cholesky-Matrix $\mathbf{G}$ ist die Hauptdiagonale der L-Matrix normiert auf den Wert Eins. Die Normierungsfaktoren bilden die D-Matrix. Der Zusammenhang zur Cholesky-Matrix ist gegeben durch:

$$\mathbf{G} = \mathbf{L}\sqrt{\mathbf{D}} \qquad \Rightarrow \qquad \mathrm{diag}(\mathbf{G}) = \sqrt{\mathbf{D}} \tag{3.23}$$

Die Cholesky-Zerlegung ist numerisch sehr stabil, alle Einträge der Cholesky-Matrix sind im Betrag beschränkt durch die Hauptdiagonalelemente der Δ-Matrix. Auf die L-Matrix trifft dies zunächst nicht zu. Bei der Inversion der Cholesky-Matrix im Anschluss an die Zerlegung verschwindet aber die Beschränkung und damit auch der Vorteil gegenüber der LDL-Zerlegung.

In der Berechnung zeigt die LDL-Zerlegung wesentliche Vorteile. So wird für die Cholesky-Zerlegung die Wurzeloperation benötigt, für die LDL-Zerlegung nicht. Die auf den Wert Eins normierte Hauptdiagonale der L-Matrix erlaubt eine Inversion

LDL-Zerlegung der Matrix $\boldsymbol{\Delta}$:

$$
\begin{aligned}
&\textit{for} \quad s = 0 \; \textit{to} \; K-1 \quad \textit{do} \\
&\quad \left[\mathbf{D}^{-1}\right]_{s,s} = \frac{1}{[\boldsymbol{\Delta}]_{s,s} - [\mathbf{L}]_{s,0:s-1} \, [\mathbf{V}]^{H}_{s,0:s-1}} \\
&\quad \textit{for} \quad z = s+1 \; \textit{to} \; K-1 \quad \textit{do} \\
&\qquad [\mathbf{V}]_{z,s} = [\boldsymbol{\Delta}]_{z,s} - [\mathbf{L}]_{z,0:s-1} \, [\mathbf{V}]^{H}_{s,0:s-1} \\
&\qquad [\mathbf{L}]_{z,s} = [\mathbf{V}]_{z,s} \left[\mathbf{D}^{-1}\right]_{s,s} \\
&\quad \textit{end} \\
&\textit{end}
\end{aligned}
$$

Bild 3.1: Berechnungsvorschrift für die LDL-Zerlegung der Matrix $\boldsymbol{\Delta}$. Die Berechnung erfolgt spaltenweise. Die V-Matrix wird nur temporär benötigt und kann die Δ-Matrix überschreiben. Die D-Matrix wird direkt invertiert und dient dann zur Normierung der L-Matrix.

ohne jegliche Divisionen. Für die Cholesky-Matrix ist dies nur auf Umwegen und dann nur mit zusätzlichem Aufwand möglich. Aus diesen Gründen wird für die folgenden Betrachtungen die LDL-Zerlegung gewählt. Die zugehörige Berechnungsvorschrift ist in Bild 3.1 aufgelistet.

Die Berechnung erfolgt spaltenweise und aufgrund der Symmetrie nur innerhalb der unteren Dreiecksmatrix von $\boldsymbol{\Delta}$. Die Matrix $\mathbf{V}$ wird nur temporär zum Speichern von Zwischenergebnissen benötigt und darf die Δ-Matrix überschreiben. Die Elemente der Matrizen $\mathbf{D}$ und $\mathbf{V}$ werden über die Differenz zwischen einem Skalar und dem Skalarprodukt zweier Vektoren berechnet. Die L-Matrix folgt aus der V-Matrix durch spaltenweise Normierung mit dem entsprechenden Eintrag in der D-Matrix. Im Hinblick auf die folgende Inversion wird die D-Matrix unmittelbar invertiert. Die Normierung der L-Matrix kann dann durch Multiplikation mit den invertierten Werten durchgeführt werden. Dies ermöglicht eine effektivere Berechnung durch Einsparung von Divisionen, die in der Regel mit hohem Aufwand und großer Latenzzeit verbunden sind.

Nach der Zerlegung kann die Inverse $\mathbf{L}^{-1}$ der L-Matrix berechnet werden. Die entsprechende Berechnungsvorschrift ist in Bild 3.2 zu finden. Die Inverse ist ebenfalls eine untere Dreiecksmatrix mit Eins-Elementen auf der Hautdiagonalen. Berechnet werden nur die Elemente unterhalb der Hauptdiagonalen wiederum über ein vektorielles Skalarprodukt mit anschließender Addition. Diese Operation ist ausschlagge-

***Inversion der Matrix* L:**

for $d = 1$ *to* $K-1$ *do*

  for $s = 0$ *to* $K-1-d$ *do*

$$\left[\mathbf{L}^{-1}\right]_{d+s,s} = -\left(\left[\mathbf{L}\right]_{d+s,s} + \left[\mathbf{L}\right]_{d+s,s+1:d+s-1}\left[\mathbf{L}^{-1}\right]_{s+1:d+s-1,s}\right)$$

  end

end

Bild 3.2: Berechnungsvorschrift für die Inversion der unteren Dreiecksmatrix $\mathbf{L}$. Die zu eins normierte Hauptdiagonale bleibt erhalten, für die erste Nebendiagonale muss nur das Vorzeichen umgekehrt werden. Divisionen sind nicht erforderlich.

bend für die Auslegung der Rechenwerke des Prozessors. Wie bereits erwähnt, wird für die Inversion keine Division benötigt. Die Elemente werden in einer Reihenfolge entlang der Nebendiagonalen berechnet. Dieser Ansatz lässt den maximalen Spielraum für Latenzzeiten in der rekursiven Berechnung.

Die gewünschte MMSE-Lösung kann nun über eine Kette von Matrixmultiplikationen berechnet werden:

$$\mathbf{W}_{\mathrm{MMSE}} = \mathbf{\Delta}^{-1}\mathbf{A}^H = \mathbf{L}^{-1^H}\mathbf{D}^{-1}\mathbf{L}^{-1}\mathbf{A}^H \tag{3.24}$$

In der Berechnungsvorschrift sollte die Struktur der Matrizen berücksichtigt und triviale Multiplikationen vermieden werden, z.B. mit den Null-Elementen der Matrizen $\mathbf{L}^{-1}$ und $\mathbf{D}^{-1}$ sowie mit den Eins-Elementen auf der Hauptdiagonalen von $\mathbf{L}^{-1}$. Auf eine detaillierte Angabe der Berechnungsvorschriften wird hier verzichtet.

3.2.3 Lösung durch Zerlegung der Systemmatrix

Die Matrixinversion kann durch ein iteratives Lösen von Gleichungssystemen umgangen werden. Voraussetzung ist eine geeignete Zerlegung der Systemmatrix bzw. der Δ-Matrix. Der Lösungsweg wird hier am Beispiel einer LDL-Zerlegung der Δ-Matrix skizziert.

Ausgangspunkt ist die lineare Gewichtung aus (3.2) mit der speziellen MMSE-Lösung aus (3.11):

$$\hat{\mathbf{d}} = \mathbf{\Delta}^{-1}\mathbf{A}^H\mathbf{x} \tag{3.25}$$

Vorwärts-/Rückwärtseinsetzen auf Basis der LDL-Zerlegung:

- *Vorwärtseinsetzen*

$$\begin{aligned} &\textit{for} \quad z = 0 : K-1 \\ &\qquad [\mathbf{y}]_z = [\tilde{\mathbf{x}}]_z - [\mathbf{L}]_{z,0:z-1}[\mathbf{y}]_{0:z-1} \\ &\textit{end} \end{aligned}$$

- *Rückwärtseinsetzen*

$$\begin{aligned} &\textit{for} \quad z = K-1 : 0 \\ &\qquad [\hat{\mathbf{d}}]_z = [\mathbf{D}^{-1}]_{z,z}[\mathbf{y}]_z - [\mathbf{L}^H]_{z,z+1:K-1}[\hat{\mathbf{d}}]_{z+1:K-1} \\ &\textit{end} \end{aligned}$$

Bild 3.3: Berechnungsvorschrift zur Schätzung des Symbolvektors durch Vorwärts- und Rückwärtseinsetzen auf Basis einer LDL-Zerlegung der Matrix Δ.

Durch Multiplikation mit $\mathbf{\Delta}$ und Einsetzen der LDL-Zerlegung folgt

$$\mathbf{L}\underbrace{\mathbf{D}\mathbf{L}^H\hat{\mathbf{d}}}_{\mathbf{y}} = \underbrace{\mathbf{A}^H\mathbf{x}}_{\tilde{\mathbf{x}}} \tag{3.26}$$

Die Multipliktion des Empfangsvektors mit $\mathbf{A}^H$ entspricht einer kanalangepassten Filterung (*channel matched filter*). Durch Vorwärtseinsetzen kann dann schrittweise $\mathbf{y}$ und anschließend nach Multiplikation mit $\mathbf{D}^{-1}$ durch Rückwärtseinsetzen auch $\hat{\mathbf{d}}$ geschätzt werden. Die Berechnungsvorschrift ist in Bild 3.3 angegeben.

Wenn nur wenige Datenblöcke mit einer Kanalschätzung zu entzerren sind, reduziert dieser Ansatz den Rechenaufwand im Vergleich zur Matrixinversion. Weiterhin können im letzten Berechnungsschritt statt der berechneten Werte die bereits entschiedenen Symbole eingesetzt werden. Dies entspricht einer sukzessiven Entscheidungsrückführung mit zufälliger Reihenfolge und führt zu einer Verbesserung der Schätzung. Über eine QR-Zerlegung kann dann aufwandseffizient die Reihenfolge optimiert und die Schätzung nochmals verbessert werden [32, 33]. Entscheidend für die Prozessorarchitektur ist die Feststellung, dass auch dieser Ansatz im Wesentlichen über Skalarprodukte mit einer anschließenden Addition zu berechnen ist.

3.3 Iterative Lösungsansätze

Die Wienerlösung nach (3.6) kann, neben der im vorigen Kapitel beschriebenen direkten Berechnung, auch iterativ berechnet werden. Es wird zwischen zwei grundsätzlich verschiedenen Ansätzen unterschieden: stochastische Gradientensuchverfahren und deterministische Verfahren zur Minimierung des quadratischen Fehlers (*LS, least squares*).

Die Gradientensuchverfahren lösen das Optimierungsproblem aus (3.5) durch eine schrittweise Minimierung der Fehlerfunktion in Richtung des negativen Gradienten. Zur Berechnung des Gradienten muss die Statistik der Signale bekannt sein. Eine grobe Näherung durch den Momentangradienten führt zu dem weit verbreiteten LMS-Algorithmus (*least mean squares*), der im Mittel der Richtung des Gradienten folgt und zur Lösung führt.

Die LS-Verfahren lösen das Schätzproblem (3.1) mit minimalem quadratischen Fehler unter Berücksichtigung aller bisher übertragenen Daten. Die Lösung ist deterministisch und kann iterativ mit jeder neuen Beobachtung aktualisiert werden. Mit zunehmender Anzahl an Beobachtungen konvergiert die LS-Lösung dann zu der Wienerlösung. Die iterative Berechnung erfordert in jedem Schritt die Inversion der deterministischen Korrelationsmatrix. Der RLS-Algorithmus (*recursive least squares*) umgeht die Inversion durch eine iterative Aktualisierung der Inversen auf Grundlage des Matrixinversions-Lemmas. In der Konvergenzgeschwindigkeit ist der RLS dem LMS deutlich überlegen. Der Aufwand nimmt für den RLS jedoch quadratisch mit der Filterordnung zu, der LMS nur linear.

Bisher stand die Suche nach der optimalen Wienerlösung im Vordergrund, d.h. die Konvergenz auf einen stationären Zustand. In einem zeitvarianten Kanal verändert sich laufend die Form der Fehlerfunktion und damit auch die Lage des Optimums. Neben der Konvergenzgeschwindigkeit ist also auch das Nachführverhalten (*tracking*) der Algorithmen entscheidend. Hier liegt der eigentliche Vorteil der iterativen Verfahren gegenüber der direkten Berechnung über Kanalschätzung und Inversion: sie verfügen über ein Gedächtnis und führen kontinuierliche Veränderungen mit relativ geringem Aufwand nach.

Aufgrund der stark vereinfachten Modellierung kann der LMS-Algorithmus den Änderungen aber nur relativ langsam folgen. Eine veränderliche Übertragungsmatrix widerspricht dem LS-Modell. Der klassische RLS ist mit einem unbegrenzten Gedächtnis zum Nachführen nicht geeignet. Erst die Beschränkung des Gedächtnisses über einen Vergessensfaktor ermöglicht eine laufende Anpassung. Die Lösung kann aber immer nur eine Näherung des Optimums sein.

Über die Systembeschreibung mit Zustandsräumen (*steady state theory*, [51]) kann die zeitliche Veränderlichkeit des Kanals explizit berücksichtigt werden. Das Kalman-Filter ermöglicht die iterative Schätzung des Zustandsraumes mit minimalem quadratischen Fehler. Die Zustandsräume erlauben eine sehr allgemeine Beschreibung des Problems. Die Wiener-Theorie und das LS-Problem können mit Zustandsräumen beschrieben werden; LMS und RLS sind damit Spezialfälle des Kalman-Filters.

Ein weiterer Vorteil der iterativen Verfahren ist die implizite Interferenzunterdrückung gerichteter Störquellen. Eine gerichtete Störung, z.B. aus einer Nachbarzelle, kann als Korrelationsmatrix $\mathbf{R}_{\mathrm{nn}} \neq \sigma_n \mathbf{I}_{\mathrm{MQ}}$ aufgefasst werden. Die iterativen Verfahren streben immer die allgemeine Wienerlösung nach (3.11) an, die unter Berücksichtigung dieser Matrix den Fehler minimiert. Für die direkte Lösung über die Matrixinversion erfordert der Übergang zur allgemeinen Wienerlösung dagegen eine explizite Schätzung der Korrelationsmatrix sowie einen deutlich höheren Rechenaufwand.

Es folgt eine detailliertere Betrachtung der beiden grundlegenden Algorithmen LMS und RLS. Zur Initialisierung wird dann eine Iteration über die Frequenz vorgeschlagen. Schließlich wird kurz die Anwendung der iterativen Verfahren für eine alternative Kanalschätzung erläutert.

3.3.1 Der LMS-Algorithmus

Der LMS-Algorithmus (Bild 3.4) benötigt für die Aktualisierung pro Gewichtsfaktor nur eine Multiplikation und eine Addition. Vom Aufwand her ist er damit kaum zu unterbieten. Die Konvergenzgeschwindigkeit ist jedoch sehr gering auf Grund der groben Approximation des Gradienten sowie der fehlenden Dekorrelation des Eingangsvektors und einer sub-optimalen Schrittweitenanpassung.

LMS-Algorithmus:

$$\begin{aligned} \hat{\mathbf{d}}^{(l)} &= \mathbf{W}_{\mathrm{LMS}}^{(l-1)} \mathbf{x}^{(l)} \\ \mathbf{e}^{(l)} &= \mathbf{d}^{(l)} - \hat{\mathbf{d}}^{(l)} \\ \mathbf{W}_{\mathrm{LMS}}^{(l)} &= \mathbf{W}_{\mathrm{LMS}}^{(l-1)} + \mu \mathbf{e}^{(l)} {\mathbf{x}^{(l)}}^{H} \end{aligned}$$

Bild 3.4: Iterative Berechnungsvorschrift für den LMS-Algorithmus mit der Schrittweite μ.

Die Näherung durch den Momentangradienten führt zu Korrekturen in zufällige

Richtungen, die erst im Mittelwert in Richtung des Gradienten zeigen. Noch kritischer ist die Abhängigkeit von den Eigenwerten λ_R der Korrelationsmatrix $\mathbf{R}_{\mathrm{xx}}$ aus Gleichung (3.16). Die Eigenwerte bestimmen die Krümmung der Fehlerfunktion in ihren Hauptachsen. Eine Dekorrelation des Eingangsvektors dreht das Koordinatensystem in Richtung der Hauptachsen. Die Schrittweite kann dann getrennt für jede Hauptachse optimal an die Krümmung angepasst werden. Beim LMS wird, genau wie beim Gradienten-Verfahren, auf die Dekorrelation verzichtet. Eine getrennte Anpassung der Schrittweite ist nicht mehr möglich. Die einheitliche Schrittweite wird durch die stärkste Krümmung, d.h. den Kehrwert des größten Eigenwertes, nach oben beschränkt. Die Zeitkonstante der Lernkurve ist dann proportional zum Verhältnis zwischen größtem und kleinstem Eigenwert, d.h. zur Kondition $\kappa_2(\mathbf{R}_{\mathrm{xx}})$ der Korrelationsmatrix. In der vorliegenden Anwendung hängt die Kondition von der momentanen räumlichen Trennbarkeit ab und ist damit starken Schwankungen ausgesetzt. Die geringe Konvergenzgeschwindigkeit und die starke Abhängigkeit von der Eigenwertverteilung sind die Hauptgründe, die gegen den Einsatz des LMS für die geplante Anwendung sprechen. Solange der LMS den Anforderungen genügt, ist er natürlich aufgrund der geringen Komplexität die erste Wahl.

3.3.2 Der RLS-Algorithmus

Der RLS führt eine Dekorrelation mit Schrittweitenanpassung durch und ist daher dem LMS in der Konvergenz deutlich überlegen. Das Gedächtnis des RLS muss zum Nachführen schneller Veränderungen mit einem exponentiellen Vergessensfaktor $\beta = 0 \ldots 1$ auf ein begrenztes Zeitfenster beschränkt werden. Da das Gedächtnis mit zunehmendem β länger wird, ist der Begriff Vergessensfaktor zwar irreführend, in der Literatur dennoch etabliert. Das modifizierte deterministische Optimierungsproblem lautet:

$$\mathbf{W}_{\mathrm{RLS}}^{(L)} = \arg\min_{\mathbf{W}^{(L)}} \sum_{l=1}^{L} \beta^{L-l} \|\mathbf{d}^{(l)} - \mathbf{W}^{(L)}\mathbf{x}^{(l)}\|^2 \tag{3.27}$$

Im stationären Fall mit $\beta = 1$ und $L \to \infty$ entspricht die resultierende Gewichtsmatrix der Wienerlösung. Das Problem kann für den Zeitpunkt L iterativ über die Berechnungsvorschrift in Bild 3.5 gelöst werden.

Gleichung (3.28) entspricht der linearen a priori Schätzung mit dem Gewichtsvektor der vorangegangenen Iteration. Gleichung (3.29) setzt die Kenntnis des Sendevektors $\mathbf{d}$ durch Trainingssequenzen oder Entscheidungsrückführung voraus und berechnet den a priori Schätzfehler. In (3.30) wird der Empfangsvektor durch Gewichtung mit der Matrix $\mathbf{P}^{(l-1)}$ dekorreliert und dann in (3.32) mit dem skalaren Faktor aus (3.31)

RLS-Algorithmus:

$$\hat{\mathbf{d}}^{(l)} = \mathbf{W}_{\mathrm{RLS}}^{(l-1)}\mathbf{x}^{(l)} \tag{3.28}$$

$$\mathbf{e}^{(l)} = \mathbf{d}^{(l)} - \hat{\mathbf{d}}^{(l)} \tag{3.29}$$

$$\mathbf{k}^{(l)} = \mathbf{P}^{(l-1)}\mathbf{x}^{(l)} \tag{3.30}$$

$$n^{(l)} = \beta + \mathbf{x}^{(l)H}\mathbf{k}^{(l)} \tag{3.31}$$

$$\mathbf{k}_{\mathrm{n}}^{(l)} = \frac{1}{n^{(l)}}\mathbf{k}^{(l)} \tag{3.32}$$

$$\mathbf{P}^{(l)} = \frac{1}{\beta}\left(\mathbf{P}^{(l-1)} - \mathbf{k}^{(l)}\mathbf{k}_{\mathrm{n}}^{(l)H}\right) \tag{3.33}$$

$$\mathbf{W}_{\mathrm{RLS}}^{(l)} = \mathbf{W}_{\mathrm{RLS}}^{(l-1)} + \mathbf{e}^{(l)}\mathbf{k}_{\mathrm{n}}^{(l)H} \tag{3.34}$$

Bild 3.5: Iterative Berechnungsvorschrift für den RLS-Algorithmus mit exponentiellem Vergessensfaktor β

normiert. Die Matrix $\mathbf{P}^{(l-1)}$ entspricht der inversen deterministischen Korrelationsmatrix des Eingangsvektors; der resultierende Vektor $\mathbf{k}_{\mathrm{n}}$ wird aufgrund der Analogie zum Kalman-Filter auch als Kalman-Gain bezeichnet. Mit ihm können schließlich in (3.33) und (3.34) die Schätzwerte für die Korrelationsmatrix und die Gewichtsmatrix aktualisiert werden. Die Matrizen $\mathbf{P}^{(0)}$ und $\mathbf{W}_{\mathrm{RLS}}^{(0)}$ müssen als Initialisierung vorgegeben werden. Während alle übrigen Berechnungsschritte wieder in Skalarprodukte zerlegt werden können, ist für die Aktualisierung ein äußeres Vektorprodukt erforderlich.

Der RLS ist kein iterativer Suchalgorithmus. Im Sinne des Kriteriums in Gleichung (3.27) ist $\mathbf{W}_{\mathrm{RLS}}^{(L)}$ zu jedem Zeitpunkt optimal. Zu Beginn eines Zeitschlitzes, d.h. für kleine L, wird die Lösung noch durch die Initialisierung dominiert. Im stationären Fall mit einer Filterordnung von M wird nach etwa $2M$ Iterationen die Wienerlösung erreicht. Der nicht-stationäre Fall verletzt das zugrunde liegende Modell, das Optimum kann nur noch mit einem Restfehler angenähert werden.

Der Berechnungsaufwand wird durch die Matrixoperationen in (3.28), (3.30), (3.33) und (3.34) dominiert. Für den klassischen Einnutzerfall stellt die quadratische Abhängigkeit von der Filterordnung zur Aktualisierung der inversen Korrelationsmatrix eine drastische Erhöhung des Aufwandes gegenüber dem LMS dar, vor allem bei großer Filterordnung. Durch die Frequenzbereichstransformation in der vorliegenden Anwendung bleibt die Filterordnung mit MQ aber relativ klein. Es müssen außerdem K Filter parallel realisiert werden, wobei im Grenzfall $K = MQ$ ist. Da

die Korrelationsmatrix auf dem Eingangsvektor beruht, ist sie identisch für alle Teilnehmer und muss nur einmal berechnet werden. Der Aufwandsunterschied zwischen LMS und RLS ist nur noch gering, die deutlich höhere Konvergenzgeschwindigkeit rechtfertigt die Verwendung des RLS.

Wenn $K < MQ$, ist der Algorithmus dennoch ineffizient und für große Q möglicherweise nicht mehr zu realisieren. In der Literatur sind verschiedene Erweiterungen unter der Bezeichnung *Fast-RLS* zu finden, die den Aufwand auf eine lineare Abhängigkeit ($\approx 7MQ$) reduzieren. Sie setzen jedoch eine Struktur voraus, die nur für FIR-Filter im Zeitbereich [52] oder, mit einem etwas allgemeineren Ansatz, für orthonormale IIR-Filter [53] gegeben ist. Bei der vorliegenden Frequenzbereichsentzerrung wird der Signalunterraum der Dimension MQ immer nur durch K Eigenvektoren aufgespannt. Diese Eigenschaft kann über eine Kahunen-Loéve-Transformation zur Aufwandsreduktion bei $K < MQ$ genutzt werden. In Anhang A.1 wird dazu der KLT-RLS-Algorithmus eingeführt.

Die Anpassung des statischen RLS an einen zeitvarianten Kanal mit dem Vergessensfaktor ist nicht optimal. Die Beschreibung über Zustandsräume ermöglicht eine genauere Modellierung des zeitlichen Verlaufes. Das Kalman-Filter bietet damit eine iterative Lösung mit einem besseren Nachführverhalten. Die Analogie zum klassischen RLS sowie ein entsprechender Vorschlag zur Verbesserung des Nachführverhaltens über ein auto-regressives (AR-) Modell 2. Ordnung werden in Anhang A.2 erläutert.

3.3.3 Initialisierung durch Adaption über die Frequenz

Der Berechnungsaufwand für die iterativen Verfahren wird durch die Blockverarbeitung im Frequenzbereich wesentlich reduziert. Die Suche nach dem Optimum und das Nachführen eines zeitlich veränderlichen Kanals ist jedoch nur von Block zu Block möglich. Eine lange Trainingsphase von mehreren Blöcken für die Initialisierung zu Beginn eines Zeitschlitzes verringert die Effizienz der Übertragung. Über eine Matrixinversion auf Grundlage einer orthogonalen Kanalschätzung kann die Initialisierung beschleunigt werden. Der Aufwand für die Inversion mit geringer Latenzzeit ist allerdings sehr hoch. Eine Adaption über die Frequenz bietet eine mögliche Alternative für eine schnelle Initialisierung mit geringem Aufwand.

Die Anwendung der oben vorgestellten iterativen Verfahren von Unterträger zu Unterträger führt nach einigen Iterationen zu den gewünschten Gewichts- bzw. Kanalkoeffizienten. Die kontinuierlichen Änderungen über die Frequenz werden nachgeführt. Die Filterkoeffizienten der ersten, noch nicht konvergierten Unterträger,

müssen in einem zweiten Schritt berechnet werden. Die zyklische Erweiterung der Frequenzcharakteristik ist für das hier betrachtete Übertragungsmodell stetig, d.h. eine Adaption vom letzten auf den ersten Unterträger ist möglich. Alternativ kann eine Adaption in umgekehrter Richtung vom letzten zum ersten Unterträger durchgeführt werden.

Wenn die Unterschiede von Unterträger zu Unterträger zu groß werden, ist das Nachführen schwierig und die Konvergenz wird nur mit großem Restfehler erreicht. Für die Systemidentifikation wird die Veränderlichkeit über die Frequenz von der effektiven Kanallänge bestimmt. Bei der Entzerrung ist die Länge der Impulsantwort des inversen Systems entscheidend. Diese hängt unter anderem von der Korrelation der Antennen ab und kann wesentlich länger werden. Bei einer hohen Korrelation und hohem SNR ist mit einer starken Fluktuation der Gewichtskoeffizienten zu rechnen. Ein Nachführen wird dann kaum noch möglich sein.

Die Berechnungsvorschriften der iterativen Verfahren können für die Adaption über die Frequenz unmittelbar übernommen werden. Der Index l bezeichnet nun den aktuellen Unterträger. Bei der Adaption in umgekehrter Richtung müssen die a priori Schätzungen durch $l+1$ bzw. die Prädiktion des Kalmanfilters durch $l-1$ gekennzeichnet werden.

3.3.4 Iterative Kanalschätzung

Alternativ zu der orthogonalen Kanalschätzung aus Kapitel 3.2.1 können die in diesem Kapitel vorgestellten Verfahren auch für eine iterative Kanalschätzung eingesetzt werden. Ziel ist die Nachbildung des unbekannten Kanals mit minimalem quadratischen Fehler:

$$\mathbf{A}_{\mathrm{MMSE}} = \arg\min_{\mathbf{A}} J(\mathbf{A}) = \arg\min_{\mathbf{A}} \mathbf{E}\left\{\|\mathbf{x} - \mathbf{A}\mathbf{d}\|^2\right\} \tag{3.35}$$

In der Literatur wird diese Anwendung auch als Systemidentifikation bezeichnet [54]. Im Vergleich zur Entzerrung ergeben sich einige Unterschiede: Es handelt sich hier um MQ parallele Filter der Dimension K. Die Korrelationsmatrix des Eingangssignals ist nun identisch zu $\mathbf{R}_{\mathrm{dd}} = \sigma_d \mathbf{I}_{\mathrm{K}}$. Die Kondition ist gleich eins, d.h. die Nachteile des LMS bezüglich Dekorrelation und Schrittweitenanpassung sind unkritisch. Für den RLS kann die Aktualisierung der inversen Korrelationsmatrix entfallen. Die Matrix hat vollen Rang, d.h. die KLT bringt hier keinen Vorteil. Schließlich kann auch hier das Kalman-Filter mit einem AR-Modell 2. Ordnung das Nachführverhalten verbessern. Mit den Parametern Schrittweite, Vergessensfaktor

bzw. σ_w ist ein Abtausch zwischen der Robustheit gegenüber dem Rauschen und dem Konvergenz- bzw. Nachführverhalten zu treffen.

Die Iteration für die Kanalschätzung kann sowohl über die Zeit als auch über die Frequenz angewendet werden. Für die Wahl der Trainingssequenzen gelten die gleichen Überlegungen wie für die orthogonale Kanalschätzung in Kapitel 3.2.1. Eine Iteration über die Zeit erfordert mit $\mathbf{R}_{\mathrm{dd}} = \sigma_d \mathbf{I}_{\mathrm{K}}$ jedoch unkorrelierte Trainingssymbole, was mit einer festen Sequenz nicht erreicht werden kann. Mehrere Pseudo-Zufallsfolgen oder die zyklische Verschiebung einer Sequenz sollten hier eingesetzt werden.

3.4 Interpolation über die Frequenz

Die Systemmatrizen benachbarter Unterträger und aufeinanderfolgender Datenblöcke sind in der Regel stark korreliert, d.h. sie sind einander sehr ähnlich. Für die Kanalschätzung wurde in Kapitel 3.2.1 eine Interpolation vorgeschlagen, um die Anzahl der Trainingssymbole zu verringern. Das gleiche Prinzip kann auf die Gewichtsmatrizen angewendet werden, um die Anzahl der Matrixinversionen und damit den Rechenaufwand zu verringern. Im Folgenden wird nur die Interpolation über die Frequenz betrachtet, da eine Interpolation über die Zeit sehr speicheraufwändig ist.

Die Interpolation entspricht einer Tiefpassfilterung eines mit Nullen aufgefüllten und damit überabgetasteten Signals. Für die Interpolation im Frequenzbereich muss das Nyquist-Kriterium im Zeitbereich angewendet werden. Die Länge der Impulsantwort bestimmt die Bandbreite des Interpolationsfilters und damit auch den Interpolationsfaktor r_i bzw. die Anzahl der benötigten Stützstellen N_i. Für die Kanalschätzung mit perfekter Interpolation gilt:

$$r_i = \frac{N}{N_i} \overset{!}{\leq} \left\lfloor \frac{N}{W_{eff}} \right\rfloor = \left\lfloor \frac{N}{W_h + 2W_f - 2} \right\rfloor \tag{3.36}$$

Die maximal zulässige Länge des physikalischen Kanals ist durch die Länge der zyklischen Erweiterung begrenzt. Diese wird aus Effizienzgründen meist weniger als 20% der Blocklänge ausmachen. Der effektive Kanal ist durch Pulsform- und Empfangsfilter etwas länger. Der Frequenzbereich ist damit in der Regel etwa vierfach überabgetastet, d.h. mit einem Viertel der vorhandenen Unterträger kann die Kanalimpulsantwort vollständig beschrieben werden. Für eine Interpolation deutlich unterhalb der Nyquistgrenze ($r_i < 4$) kann der Aufwand durch sub-optimale Interpolationsfilter reduziert werden.

Die Impulsantwort des inversen Kanals ist in der Regel länger. Theoretisch ist sie

auf die gesamte Blocklänge ausgedehnt, eine Interpolation ist nur näherungsweise möglich. In der Praxis kann auch hier mit einem Interpolationsfaktor von 2 bis 4 gerechnet werden, d.h. nur ein Viertel bis die Hälfte der Matrixinversionen müssen tatsächlich berechnet werden. Der Aufwand für die Inversion nimmt proportional mit dem Interpolationsfaktor ab, der Aufwand für die Interpolation nimmt jedoch zu, je näher der Interpolationsfaktor an die Nyquistgrenze rückt.

Die Gewichtung mit einer Spreizsequenz verlängert den effektiven Kanal. Bei der Frequenzbereichsspreizung entspricht die Länge der Spreizsequenz der Blocklänge, eine Interpolation sowohl der Systemmatrizen als auch der Gewichtsmatrizen ist dann nicht mehr möglich. Eine Spreizung mit permutierten Unterträgern ist ebenfalls ungünstig für eine Interpolation, da der Spreizfaktor den minimalen Abstand der Stützstellen einschränkt.

Die Interpolation wird durch zyklische Faltung mit einem Interpolationsfilter umgesetzt und kann dann auch als Skalarprodukt zweier Vektoren aufgefasst werden. In einem zyklischen System ist auch eine ideale Interpolation mit endlichem Filteraufwand möglich. Für eine effiziente Realisierung ist jedoch eine Reduktion der Filterlänge anzustreben. Die Anforderungen sind abhängig von dem gewählten Interpolationsfaktor, je mehr Stützstellen bekannt sind, desto einfacher kann das Interpolationsfilter dimensioniert werden. Eine Dreiecksfunktion als Interpolationsfilter bewirkt eine lineare Interpolation, im Grenzfall mit nur drei Koeffizienten für $r_i = 2$.

3.5 Vorverzerrung mit Kanalkenntnis am Sender

Auf der Abwärtsstrecke kann die räumliche Selektivität im Sender des Zugangspunktes durch eine Vorgewichtung der Sendesignale genutzt werden. Entscheidend für die Wahl des Verfahrens ist die verfügbare Kenntnis über den Kanal der Abwärtsstrecke. Ohne jegliche Kanalkenntnis erlauben Space-Time-Codes eine effektive Nutzung der Sendediversität. Bei Kenntnis der Richtung kann eine adaptive Strahlformung die Hauptkeule auf den Empfänger richten und somit immer den vollen Antennengewinn nutzen. Die zusätzliche Steuerung der Nullstellen der Richtcharakteristik erlaubt die entkoppelte Übertragung für einen räumlichen Mehrfachzugriff oder Multiplex. Für eine frequenzselektive Übertragung mit räumlichem Mehrfachzugriff reichen die verfügbaren Freiheitsgrade einer rein räumlichen Gewichtung meist nicht mehr aus. Der Übergang zu einer frequenzselektiven Gewichtung ermöglicht dann neben der Formung der Richtwirkung zugleich eine Formung des Frequenzganges. Ziel ist, genau wie im Empfänger der Aufwärtsstrecke, eine Kompensation der räumlichen und zeitlichen Verzerrung auf dem Kanal. Der Ansatz wird daher auch als räumlich-

zeitliche Vorverzerrung bezeichnet.

Voraussetzung ist die vollständige Kenntnis des Kanals im Sender. Ein Feedback der in den Teilnehmern geschätzten Kanalinformation an die Basisstation ist aufgrund der großen Anzahl an Koeffizienten ineffizient. Bei einer Übertragung im Zeit-Duplex erlaubt die Reziprozität des Kanals die Verwendung der Kanalinformation aus der Aufwärtsstrecke. Das Gesamtsystem aus Kanal, Antennen, Sendern und Empfängern ist im allgemeinen zunächst nicht reziprok. Die Reziprozität wird annähernd erreicht durch eine Impendanzanpassung zwischen Antenne und Transceiver sowie eine Kalibrierung der Übertragungsfaktoren von Sendern und Empfängern [14, 15].

Das Systemmodell der Abwärtsstrecke kann äquivalent zur Aufwärtstrecke im Frequenzbereich für jede Unterträgergruppe definiert werden:

$$\hat{\mathbf{d}} = \mathbf{C}^T \left(\mathbf{H}^T \mathbf{V} \mathbf{d} + \mathbf{n} \right) = (\mathbf{H}\mathbf{C})^T \mathbf{V} \mathbf{d} + \mathbf{C}^T \mathbf{n} = \mathbf{A}^T \mathbf{V} \mathbf{d} + \tilde{\mathbf{n}} \tag{3.37}$$

Der Frequenzbereichsdatenvektor $\mathbf{d}$ mit K Einträgen wird mit der Gewichtsmatrix $\mathbf{V}$ vorgewichtet. Der resultierende Sendevektor der Länge MQ wird über den Kanal der Abwärtsstrecke übertragen. Unter Annahme der Reziprozität des effektiven Kanals entspricht dies der Multiplikation mit der transponierten Kanalmatrix $\mathbf{H}^T$ aus der Aufwärtsstrecke. Am Empfängereingang der Teilnehmerstationen wird additives weißes gaußsches Rauschen ($\mathbf{n}$) hinzugefügt. Die Signalverarbeitung im Empfänger besteht lediglich aus einer Entspreizung durch Multiplikation des Empfangssignals mit der transponierten Spreizmatrix $\mathbf{C}$. Das Produkt aus Spreiz- und Kanalmatrix entspricht nach (2.30) der transponierten Systemmatrix der Aufwärtsstrecke. Unkorreliertes Rauschen am Eingang des Empfängers bleibt auch durch die Entspreizung unkorreliert, die Rauschleistung nimmt jedoch um den Faktor Q zu.

Die volle Reziprozität wird in der Praxis nicht zu erreichen sein. Bei einer schwachen Reziprozität bleiben Phase, Amplitude und Laufzeit der Empfangssignale undefiniert. Durch eine entsprechende Synchronisation im Empfänger können diese Effekte kompensiert werden.

Die Gewichtsmatrix $\mathbf{V}$ soll nun so gewählt werden, dass der Empfangsvektor $\hat{\mathbf{d}}$ eine möglichst gute Schätzung des Sendevektors $\mathbf{d}$ darstellt. Die Minimierung des mittleren quadratischen Fehlers führt zu der Pseudoinversen der Systemmatrix und entspricht damit der ZF-Lösung. Die Empfänger sehen dann einen reinen AWGN-Kanal. Die Unterdrückung von Schwundeffekten und Mehrfachzugriffsinterferenz erfordert jedoch eine hohe Signaldynamik am Sender und damit einen erhöhten Aufwand für die Leistungsverstärker, die DA-Wandler und das Auflösungsvermögen der digitalen Signalverarbeitung.

Die Normierung der Sendeleistung auf einen konstanten Wert führt bei immer noch idealer Unterdrückung der Mehrfachzugriffsinterferenz zu einem Schwundverhalten aus Sicht der Empfänger. Unter Berücksichtigung eines konstanten Rauschpegels am Empfängereingang ist diese Lösung aber nicht mehr optimal im Sinne des MMSE. Die Minimierung des MSE unter der Randbedingung einer konstanten Sendeleistung σ_s (PCMMSE, *power constrained MMSE*) führt zu einer Lösung, die sehr ähnlich zur MMSE-Lösung der Aufwärtsstrecke ist [55]:

$$\mathbf{V}_{\mathrm{PCMMSE}} = \alpha_V \tilde{\mathbf{V}}^T \tag{3.38}$$

$$\text{mit} \quad \tilde{\mathbf{V}} = \left(\mathbf{A}^H \mathbf{A} + K \frac{\sigma_{\tilde{n}}}{\sigma_s} \mathbf{I}_{\mathrm{K}} \right)^{-1} \mathbf{A}^H \tag{3.39}$$

$$\text{und} \quad \alpha_V = \sqrt{\frac{\sigma_s}{\sigma_d \,\mathrm{spur}\left(\tilde{\mathbf{V}}^H \tilde{\mathbf{V}} \right)}} \tag{3.40}$$

Die Matrix $\tilde{\mathbf{V}}$ entspricht der Definition für $\mathbf{W}_{\mathrm{MMSE}}$ der Aufwärtsstrecke mit dem Unterschied, dass das reziproke SNR hier durch das Verhältnis aus effektiver Rauschleistung am Empfänger $K\sigma_{\tilde{n}} = \mathrm{spur}\left(E\left\{\tilde{\mathbf{n}}\tilde{\mathbf{n}}^H\right\}\right)$ und der gewünschten Sendeleistung ersetzt wird. Der skalare Vorfaktor α normiert den resultierenden Sendevektor auf die geforderte Sendeleistung. Bei einer geeigneten Wahl der Signalpegel, d.h. $K\frac{\sigma_{\tilde{n}}}{\sigma_s} = \frac{\sigma_n}{\sigma_d}$, kann die Gewichtsmatrix aus der Aufwärtsstrecke unmittelbar für die Vorverzerrung wieder verwendet werden und muss anschließend nur noch transponiert und mit α_V normiert werden.

Für die Einträgerübertragung muss die Sendeleistung auf allen Unterträgergruppen gleich gewählt werden, um einen Empfang ohne Entzerrung zu ermöglichen. Bei der Mehrträgerübertragung kann die Sendeleistung für jede Unterträgergruppe getrennt eingestellt werden. Hier sind mehrere Strategien möglich, so z.B. *water filling*, das den Gruppen mit der besten Kondition die meiste Leistung zuspricht, um diese mit hoher Modulationsstufe effizient zu betreiben. Der Normierungsfaktor muss für Ein- und Mehrträgersysteme im Empfänger kompensiert werden, was bei der schwachen Reziprozität durch die Amplitudensynchronisation aber bereits abgedeckt wird.

Stark unterschiedliche Pfaddämpfungen der Teilnehmer werden in der Aufwärtsstrecke üblicherweise durch eine Leistungsregelung ausgeglichen. In der Schätzung der effektiven Kanalkoeffizienten ist die Ausbreitungsdämpfung damit nicht enthalten. Wenn die Einstellung der Leistungsregelung aller Teilnehmer im Zugangspunkt bekannt ist, können die Spalten der Vorverzerrungsmatrix entsprechend gewichtet werden. Jeder Teilnehmer wird dann entsprechend seiner Pfaddämpfung

versorgt. Die unterschiedlichen Pegel stellen jedoch hohe Anforderungen an den Dynamikbereich des Senders und die Interferenzunterdrückung. Kleine Fehler aufgrund der nicht idealen Reziprozität oder einer zeitlichen Veränderung führen zu starken Störungen. Das Problem ist aus klassischen CDMA Systemen unter dem Begriff *near-far* bekannt. Genau wie dort üblich wird hier die Pfaddämpfung in der Abwärtsstrecke nicht berücksichtigt. Die Sendeleistung am Zugangspunkt wird entsprechend dem Teilnehmer mit der größten Pfaddämpfung geregelt. Die übrigen Teilnehmer werden mit mehr Leistung als eigentlich notwendig überversorgt.

3.6 Aufwandsabschätzung

Der Berechnungsaufwand für die oben vorgestellten Algorithmen soll nun abgeschätzt und tabellarisch gegenübergestellt werden. Der Aufwand wird in CMACs und DIVs angegeben. Eine MAC-Operation (*multiply accumulate*) entspricht der Multiplikation zweier Koeffizienten gefolgt von der Addition des Resultates auf einen Akkumulator. Signalprozessoren erlauben oft eine parallele Berechnung von Multiplikation und Addition in einer Pipeline. MACs bieten dann eine zuverlässigere Abschätzung als die allgemeineren Definitionen *FLOPS* (*floating point operations per second*) oder MIPS (*mega instructions per second*).

Da die Signalverarbeitung im Frequenzbereich komplexwertig ist, werden hier CMACs, d.h. komplexe (*complex*) MACs, gezählt. Ein CMAC besteht aus 4 reellen MACs. Die Akkumulation wird hier etwas allgemeiner als Addition mit einem beliebigen dritten Koeffizienten aufgefasst. Bei der Aufwandsberechnung wurden mehrfache Divisionen mit dem gleichen Teiler (z.b. Normierung eines Vektors) durch die einmalige Berechnung des Kehrwertes und eine anschließende mehrfache Multiplikation ersetzt. Dieser Ansatz reduziert Aufwand und Latenzzeit, bietet jedoch eine geringere Auflösung bei einer Festkomma-Arithmetik. Die betrachteten Algorithmen lassen eine Beschränkung der Kehrwertbildung auf positive reele Zahlen zu. Die entsprechende Operation wird als DIV bezeichnet.

In Tabelle 3.1 wird die Definition der für die Aufwandsabschätzung erforderlichen Parameter wiederholt und zwei Beispielkonfigurationen BSP1 und BSP2 spezifiziert. In Tabelle 3.2 wird dann der Aufwand in CMACs und DIVs pro Unterträger in Abhängigkeit der Parameter angegeben. Zur Veranschaulichung werden für die beiden Beispielkonfigurationen die CMACs explizit berechnet.

Grundsätzlich ist zu beobachten, dass mit Erhöhung des Spreizfaktors der Aufwand entweder gleich bleibt oder abnimmt. Ausnahmen bilden nur der RLS und das Kalman-Filter mit einen Term proportional zu Q. Über die KLT Transformation

Beschreibung	**Variable**	**BSP1**	**BSP2**
Antennen am Zugangspunkt	M	8	4
Antennen der Teilnehmer	K	7	3
Spreizfaktor	Q	1	
Interpolationsfaktor	r_i	4	
Tapanzahl des Interpolationsfilters	L_i	32	

Tabelle 3.1: Konfiguration zweier Beispielkonfigurationen BSP1 und BSP2

wird diese Abhängigkeit kompensiert. In BSP1 mit einem Spreizfaktor $Q = 4$ reduziert die KLT-Transformation die CMACs für den RLS von 648 auf 222, in BSP2 entsprechend von 164 auf 47.

Bei der Matrixinversion ist zu beachten, dass in beiden Beispielen etwa 85% der CMACs für Matrixmultiplikationen und nur 15% für die Zerlegung bzw. die Inversion der L-Matrix anfallen. Der Aufwand für die explizite Berechnung der Wienerlösung lohnt sich in BSP1 erst dann, wenn sie in mindestens 13 Datenblöcken zur Entzerrung eingesetzt wird. Für eine geringere Wirkungsdauer ist die Lösung über Vorwärts/Rückwärts-Einsetzen effizienter. In BSP2 liegt die Schwelle bei 9 Datenblöcken.

Die Interpolation zwischen den Kammspektren der Trainingssymbole für eine vollständige Kanalschätzung ist vom Aufwand her im Vergleich zur Matrixinversion nicht zu vernachlässigen. Die Beschränkung der Inversion auf einzelne Stützstellen mit einer anschließenden Interpolation verspricht für größere Matrixdimensionen eine deutliche Aufwandsreduktion. Für einen Interpolationsfaktor von 4 geht so der Gesamtaufwand für Kanalschätzung und Inversion in BSP1 von 1280 auf 768 CMACs zurück, für BSP2 nur noch von 176 auf 140 CMACs.

Für einen zeitvarianten Kanal zeigt der Vergleich zur iterativen Entzerrung, dass mit dem RLS keine Reduktion des Aufwandes zu erwarten ist. Mit dem Aufwand des RLS kann bei kontinuierlicher Entzerrung über Vorwärts/Rückwärts-Einsetzen in BSP1 etwa alle 5 Blöcke eine neue Systemmatrix geschätzt und zerlegt werden, in BSP2 schon alle 3 Blöcke. Eine entsprechend häufige Kanalschätzung über Trainingssequenzen geht auf Kosten der Übertragungseffizienz. Eine durchgängige, entscheidungsrückgeführte Kanalschätzung mit dem LMS in Kombination mit Vorwärts/Rückwärts-Einsetzen stellt hier einen möglichen Kompromiss dar. Mit dem Aufwand des RLS kann auf diese Weise in BSP1 alle 7 Blöcke die Matrixzerlegung aktualisiert werden, in BSP2 schon in jedem zweiten Block. Das Kalman-Filter mit dem auto-regressiven Modell 2. Ordnung ist im Vergleich deutlich aufwendiger.

Algorithmus	**CMACs**			**DIVs**
	Allgemein	**BSP1**	**BSP2**	
Gewichtung	MK	56	12	-
Matrixinversion				
Berechnung $\mathbf{\Delta}$	$\frac{1}{2}M(K^2+K)$	224	24	-
LDL-Zerlegung	$\frac{1}{6Q}(K^3+3K^2-4K)$	77	7	$\frac{K}{Q}$
Inversion von $\mathbf{L}$	$\frac{1}{6Q}(K^3-3K^2+8K-6)$	41	3	-
Berechnung $\mathbf{\Delta}^{-1}$	$\frac{1}{6Q}(K^3+6K^2-7K)$	98	10	-
Multiplikation mit $\mathbf{A}^H$	MK^2	392	36	-
Matrixinversion gesamt	$\frac{1}{2Q}(K^3+(3MQ+2)K^2+(MQ-1)K-2)$	832	80	$\frac{K}{Q}$
Vorwärts-/Rückwärts-einsetzen				
Berechnung von $\mathbf{L}$ und $\mathbf{D}^{-1}$	$\frac{1}{6Q}(K^3+3(MQ+1)K^2+(3QM-4)K)$	301	31	$\frac{K}{Q}$
Matched Filter und Einsetzen	$K(\frac{K-1}{Q}+M)$	98	18	-
Interpolation				
vollst. Kanalschätzung	$\frac{1}{r_i}ML_iK$	448	96	-
Kanalschätzung an $\frac{N}{r_i}$ Stützstellen	$\frac{1}{r_i^2}ML_iK$	112	24	-
alle Gewichtsfaktoren mit $\frac{N}{r_i}$ Stützstellen	$\frac{1}{r_i}ML_iK$	448	96	-
Iter. Kanalschätzung				
LMS	$2MK+M$	120	28	-
RLS (mit $\mathbf{P}=c\mathbf{I}_K$)	$2MK+2M+\frac{K}{Q}$	135	35	$\frac{1}{Q}$
Kalman (AR2)	$4MK+6\frac{K^2}{Q}+6\frac{K}{Q}$	560	120	$\frac{1}{Q}$
Iter. Entzerrung				
LMS	$2MK+\frac{K}{Q}$	119	27	-
RLS	$2MK+2M^2Q+3M$	264	68	$\frac{1}{Q}$
KLT-RLS	$3MK+4\frac{K^2}{Q}+3\frac{K}{Q}$	385	81	$\frac{1}{Q}$
Kalman (AR2)	$4MK+6M^2Q+6M$	656	168	$\frac{1}{Q}$
KLT-Kalman (AR2)	$5MK+10\frac{K^2}{Q}+6\frac{K}{Q}$	812	168	$\frac{1}{Q}$

Tabelle 3.2: Berechnungsaufwand in komplexwertigen MAC-Operationen (CMACs) und Kehrwertberechnungen (DIVs) pro Unterträger

3.7 Fazit

In diesem Kapitel wurde das Optimierungsproblem für die lineare Frequenzbereichsentzerrung nach dem Kriterium des minimalen quadratischen Fehlers dargestellt. Die Vorschläge zur Lösung des Problems gliedern sich in die explizite Berechnung der Wiener-Gleichung über eine Matrixinversion, das sukzessive Lösen über eine Zerlegung der Systemmatrix sowie die iterative Minimierung des Fehlers über eine Entscheidungsrückführung. Die ersten beiden Ansätze erfordern eine Schätzung der Systemmatrix. Für den betrachteten Mehrnutzerfall wurden eine orthogonale Schätzung in Verbindung mit einer Interpolation sowie die iterative Schätzung bei voller Mehrfachzugriffsinterferenz vorgeschlagen. Die räumlich-zeitliche Vorverzerrung ist ein ähnliches Optimierungsproblem, das mit leichten Modifikationen gelöst und unter einschränkenden Randbedingungen sogar auf die gleiche Lösung zurückzuführen ist.

Für die anwendungsspezifische Optimierung der Prozessorarchitektur im folgenden Kapitel sind die zur Verarbeitung der Algorithmen benötigten Operationen entscheidend. Die Multiplikation, Zerlegung und Inversion von Matrizen lassen sich in Skalarprodukte mit einer anschließenden skalaren Addition zerlegen. Die iterativen Verfahren erfordern zur Aktualisierung von Matrizen außerdem die Summation eines äußeren Vektorproduktes mit der ursprünglichen Matrix. Mit diesen beiden grundlegenden Vektoroperationen lassen sich alle betrachteten Algorithmen vollständig beschreiben. Die Algorithmen enthalten zudem keinerlei Entscheidungen in Abhängigkeit der Daten.

Die abschließende Aufstellung des Berechnungsaufwandes bestätigt den Bedarf an leistungsfähigen parallelen Prozessoren. Für ein MIMO-System mit 8 Sende- und 7 Empfangsantennen liegt der Aufwand für Kanalschätzung und Entzerrung in der Größenordnung von 1000 MACs pro Unterträger bzw. pro Chipdauer. Für eine Bandbreite von 25 MHz entspricht das einer Rechenleistung von 25 $^{\mathrm{GMAC}}/_{\mathrm{s}}$. Für ein kleineres System mit 4 Sende- und 3 Empfangsantennen liegt der Aufwand um einen Faktor 4 bis 10 darunter.

Der Aufwandsvergleich der verschiedenen Ansätze zeigt die Bedeutung einer flexiblen Umsetzung in Software. Die vorgestellten Verfahren sind in ihrer Bedeutung für eine potentielle Anwendung auf einem Parallelprozessor sicherlich als gleichwertig zu erachten. Es ist kein Verfahren zu identifizieren, das vom Aufwand/Nutzen-Verhältnis her dominiert. Die Kombination verschiedener Ansätze für Kanalschätzung und Entzerrung bietet zahlreiche Möglichkeiten, mit zum Teil ähnlichem Rechenaufwand aber sicherlich merklichen Unterschieden in der

Adaptionsfähigkeit, der Robustheit und der Genauigkeit. Der Software-Ansatz ermöglicht eine flexible Anpassung des Verfahrens an unterschiedliche Anforderungen und Randbedingungen.

KAPITEL 4

Anwendungsspezifischer Parallelprozessor

In Kapitel 2 wurde die Übertragung für SD/CDMA-Systeme mit Ein- und Mehrträgermodulation definiert. Der nicht-orthogonale Mehrfachzugriff in diesen Systemen erfordert eine adaptive Unterdrückung der gegenseitigen Störung. Die in Kapitel 3 vorgestellten Algorithmen lösen das Problem durch eine lineare, räumlich-zeitliche Entzerrung mit minimalem quadratischen Fehler. Alle vorgestellten Verfahren arbeiten im Frequenzbereich auf parallelen Unterträgergruppen.

In diesem Kapitel wird nun eine Hardware-Architektur vorgestellt, die eine räumlich-zeitliche Entzerrung bzw. Vorverzerrung in Echtzeit ermöglicht. In Kapitel 4.1 wird zunächst die Signalverarbeitung für den Zugangspunkt definiert und in Funktionsgruppen unterteilt. Anschließend werden Schnittstellen und Komponenten zusammengefasst, um einen einheitlichen Datenfluss in verschiedenen Betriebszuständen zu ermöglichen.

In Kapitel 4.2 werden dann vier mögliche Ansätze zur Parallelisierung der eigentlichen Raum-Zeit-Verarbeitung diskutiert. Der schließlich gewählte Ansatz nutzt die Unabhängigkeit der Unterträger im Frequenzbereich für eine parallele Verarbeitung auf mehreren Rechenwerken. Eine konkrete Hardware-Architektur für den Parallelprozessor wird dann in Kapitel 4.3 ausführlich beschrieben. Die einzelnen Rechenwerke arbeiten skalar und können entsprechend flexibel eingesetzt werden. Die Auslegung der Rechenwerke speziell auf die in Kapitel 3 betrachteten Algorithmen führt zu einer effizienten Nutzung der Ressourcen.

4.1 Definition der Betriebszustände und Schnittstellen

Es wird ein Mehrnutzersystem entsprechend der Definition aus Kapitel 2.1 vorausgesetzt. Der Zugangspunkt führt im Empfänger eine räumlich-zeitliche Entzerrung und im Sender eine räumlich-zeitliche Vorverzerrung (siehe Kapitel 3.5) durch. Die Teilnehmerstationen können dann entweder ganz ohne oder, bei fehlender oder suboptimaler Vorverzerrung, mit einem klassischen Entzerrer realisiert werden. Im Folgenden wird daher nur die Signalverarbeitung des Zugangspunktes betrachtet.

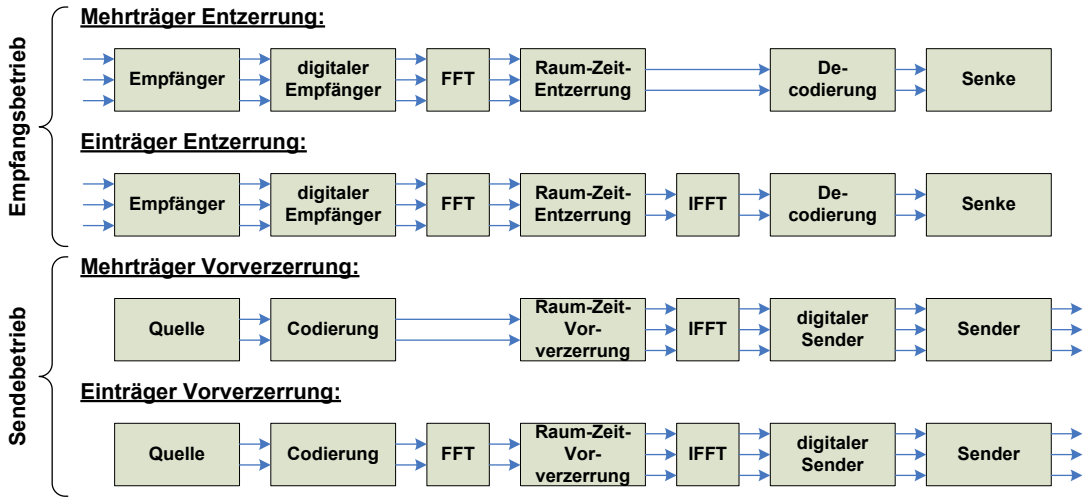


Bild 4.1: Verarbeitungsketten verschiedener Betriebszustände im Zugangspunkt, hier für $M = 3$ und $K = 2$

Die Verarbeitung der physikalischen Schicht wird in die folgenden Funktionsgruppen unterteilt: digitaler Sender und Empfänger, Codierer und Decodierer, die Fourier-Transformationen sowie die Raum-Zeit-Verarbeitung zur Entzerrung und Vorverzerrung im Frequenzbereich. Weiterhin werden vier Betriebszustände für die Signalverarbeitung definiert:

- Mehrträger-Entzerrung
- Einträger-Entzerrung
- Mehrträger-Vorverzerrung
- Einträger-Vorverzerrung

Die Verarbeitungsketten der einzelnen Betriebszustände sind in Bild 4.1 dargestellt. Eine reine Software-Implementierung aller Funktionsgruppen auf einem oder mehreren Signalprozessoren ist aus heutiger Sicht für hohe Datenraten von mehreren hundert Mbit pro Sekunde kaum realisierbar. Die Verteilung der Daten von einer Zentrale auf verschiedene Hardware-Beschleuniger ist mit hohen Datenraten ebenfalls problematisch. Sinnvoller ist die Verarbeitung in einer Pipeline entsprechend dem Datenfluss der Verarbeitungsketten in Bild 4.1. Der Wechsel zwischen den vier Betriebszuständen erfolgt dann entweder durch eine Anpassung der Software bzw. eine Rekonfiguration der einzelnen Verarbeitungsstufen oder durch ein Umlenken des Datenflusses zwischen Instanzen mit fester Funktionalität. Aktuelle FPGAs benötigen für die Rekonfiguration je nach Größe $10-50$ms. Für das Umschalten zwischen Ein- und Mehrträgermodulation wäre dies eventuell noch akzeptabel, für den Wechsel zwischen Sende- und Empfangsbetrieb hingegen zu langsam. Im Hinblick auf

einen schaltbaren Datenfluss durch gemeinsam genutzte Hardware-Komponenten werden zunächst die einzelnen Funktionsgruppen näher betrachtet.

Digitaler Sender und Empfänger umfassen Pulsform- bzw. Empfangsfilter, die Frequenzumsetzung bei einer Zwischenfrequenzabtastung und auch die Synchronisation von Trägerfrequenz und Chiprate. Das universelle Systemmodell aus Kapitel 2.2 erlaubt eine Umsetzung der digitalen Sender und Empfänger unabhängig vom Modulationsverfahren. Aufgrund der Überabtastung kommt es zu einem hohen Datendurchsatz, der I/O-Bandbreite und Rechenleistung verfügbarer Signalprozessoren leicht übersteigen kann. Die regelmäßige Struktur der Filter erlaubt eine effektive, parallele Umsetzung in ASICs. Auf dem Markt verfügbare Bausteine, z.B. von Analog Devices oder GRAYCHIP (inzwischen Texas Instruments), bieten zudem eine gewisse Flexibilität der Filterstrukturen sowie eine Konfiguration der Koeffizienten.

Zum Codierer und Decodierer wird im Folgenden neben der eigentlichen Kanalcodierung auch die Modulation der Bitströme auf ein QAM-Symbolalphabet bzw. die Entscheidung und Demodulation gezählt. Die Verarbeitung findet im Wesentlichen auf Bit-Ebene statt und kann auf Signalprozessoren mit 16 oder 32 bit Wortbreite nur ineffizient realisiert werden. Daher bietet sich auch hier eher eine Hardware-Implementierung an. Codierung und Decodierung sind prinzipiell ebenfalls unabhängig vom Modulationsverfahren. Die spezifischen Übertragungseigenschaften von Ein- und Mehrträgermodulation erfordern in der Regel jedoch unterschiedliche Parametersätze, z.B. für das Interleaving oder die Coderate. Eine Konfigurierbarkeit der Parameter wird für die Anpassung an die Kanaleigenschaften (Linkadaption) aber ohnehin gefordert.

Fourier-Transformationen können recht effizient auf Signalprozessoren umgesetzt werden. Da in allen Betriebszuständen ein kontinuierlicher Datenstrom transformiert wird, können ASICs auch hier ohne Einschränkungen in der Flexibilität eingesetzt werden. Signalprozessoren sind üblicherweise auf Butterfly-Operationen optimiert. In ASICs kann mit alternativen Realisierungsansätzen [56, 57], der Ressourcenverbrauch weiter reduziert werden. Der FFT-Prozessor sollte in der Blocklänge, der Transformationsrichtung (FFT/IFFT) sowie in der Skalierung der Festkomma-Darstellung zwischen den einzelnen Stufen konfigurierbar sein.

Die Raum-Zeit-Verarbeitung ist weitgehend unabhängig vom Modulationsverfahren. Gewichtung und Gewichtsberechnung sind bis auf die Matrixdimension nahezu identisch für Entzerrung und Vorverzerrung. Die Verarbeitung in einem Prozessor erlaubt einen gemeinsamen Zugriff auf lokal gespeicherte Gewichts- oder Kanalkoeffizienten zur Ausnutzung der Reziprozität.

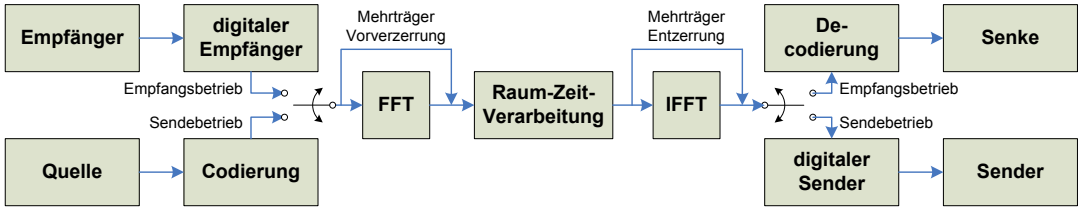


Bild 4.2: Einheitliches Systemkonzept für alle vier Betriebszustände mit mehrfach verwendeten Hardware-Komponenten. Die Raum-Zeit-Verarbeitung hat jeweils genau eine Schnittstelle zu den beiden FFT-Prozessoren.

Diese Eigenschaften ermöglichen ein einheitliches Systemkonzept für alle vier Betriebszustände (siehe Bild 4.2). Die Verarbeitungsketten von Sende- und Empfangsbetrieb werden auf eine gemeinsame Instanz für die Raum-Zeit-Verarbeitung zur Entzerrung und Vorverzerrung geschaltet. Der Wechsel zwischen Ein- und Mehrträgermodulation erfolgt durch die Änderung einzelner Parameter oder eine Rekonfiguration der äußeren Hardware-Komponenten. Für eine Mehrträgermodulation muss außerdem je nach Übertragungsrichtung entweder die FFT oder IFFT überbrückt werden. Damit sind Umgebung und Schnittstellen für die Raum-Zeit-Verarbeitung definiert. Im folgenden Kapitel werden Methoden für eine parallele Realisierung dieser Komponente diskutiert.

4.2 Methoden der Parallelisierung

Die in Kapitel 3 vorgestellten Algorithmen für eine lineare Raum-Zeit-Entzerrung stellen hohe Anforderungen an die Rechenleistung der Signalverarbeitung. Alleine die lineare Gewichtung benötigt bereits $4MK$ MACs pro Chipdauer. Mit einer Chiprate von $f_{\text{chip}} = 25\,^{\text{Mchip}}/_{\text{s}}$ und $K = M = 8$ Teilnehmern und Antennen ergibt das $6.4\,^{\text{GMAC}}/_{\text{s}}$. Für ein Nachführen der Gewichtsmatrix in jedem Datenblock braucht der RLS-Algorithmus bei diesen Spezifikationen und $Q = 1$ insgesamt $28\,^{\text{GMAC}}/_{\text{s}}$. Diese Rechenleistung ist nur durch eine parallele Verarbeitung zu erreichen.

Zunächst werden die für die Parallelisierung wesentlichen Eigenschaften der betrachteten Algorithmen zusammengefasst. Speziell bei den linearen Verfahren kann die Verarbeitung aufgeteilt werden in die Gewichtung und die Gewichtsberechnung. Die Echtzeitbedingung ist für die Gewichtung durch den Datendurchsatz, d.h. die Chiprate gegeben. Die Gewichtsberechnung muss der geforderten Anpassungsfähigkeit an kontinuierliche oder sprunghafte Kanaländerungen genügen. Die Transformation in den Frequenzbereich zerlegt eine große Aufgabe in eine Reihe kleinerer, jeweils identischer Teilaufgaben, die unabhängig voneinander gelöst werden können. Die

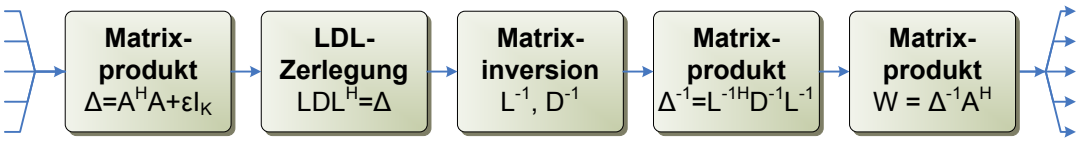


Bild 4.3: Parallele Verarbeitung von Teilaufgaben in einer Pipeline am Beispiel der speziellen Wiener-Lösung nach Gleichung (3.11). Verschiedene Datensätze können sequentiell durch die Pipeline geschoben werden.

iterative Adaption und die Interpolation über die Frequenz bilden eine Ausnahme, da sie gerade die Ähnlichkeit benachbarter Probleme ausnutzen. Die Teilaufgaben lassen sich wiederum in mehrere, aufeinander aufbauende Verarbeitungsschritte zerlegen. Dabei handelt es sich im wesentlichen um die folgenden Matrixoperationen: LDL-Zerlegung, Inversion einer Dreiecksmatrix, Matrix-Matrix- und Matrix-Vektor-Multiplikationen sowie innere und äußere Vektorprodukte.

Anzahl der Teilaufgaben und Größe der Matrizen bzw. Vektoren sind abhängig vom Spreizfaktor Q, der Teilnehmerzahl K sowie der Antennenanzahl M. Während M in der Regel fest vorgegeben ist, können sich die Parameter Q und K während des Betriebes ändern. Die Matrixoperationen lassen sich zerlegen in Vektorprodukte und skalare Divisionen. Die Länge der Vektoren variiert und liegt zwischen 1 und MQ. Die Vektorprodukte setzen sich vorwiegend aus komplexwertigen MAC-Operationen (CMACs) zusammen.

Auf Grundlage dieser spezifischen Eigenschaften werden nun vier mögliche Ansätze zur Parallelisierung der Verarbeitung diskutiert:

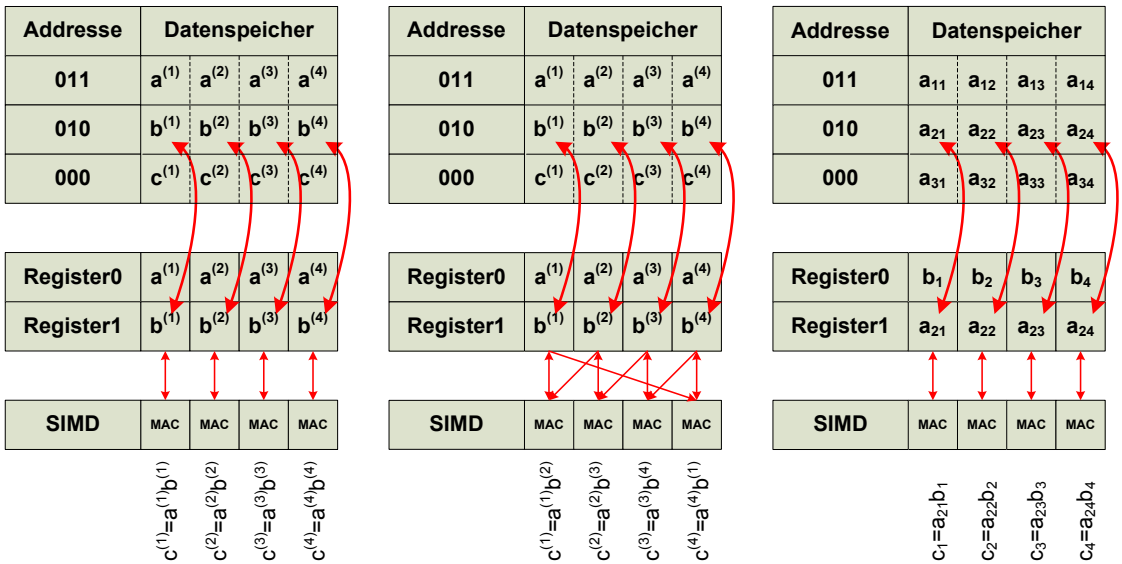
- **Parallelisierung der Verarbeitungsschritte in einer Pipeline:** Die einzelnen Verarbeitungsschritte der Algorithmen können als getrennte Teilaufgaben aufgefasst werden. Die wiederholte Anwendung der Verarbeitungsschritte auf mehrere Unterträgergruppen oder Datenblöcke ermöglicht eine parallele Verarbeitung der Teilaufgaben in einer Pipeline (siehe Bild 4.3). Die Verarbeitungsdauer der einzelnen Stufen muss den Echtzeitanforderungen genügen. Für einen effizienten Betrieb der Pipeline sollte die Rechenleistung jeder Stufe an den Aufwand der jeweiligen Teilaufgabe angepasst sein. Diese spezifische Festlegung erschwert einen Wechsel zwischen unterschiedlichen Algorithmen (z.B. zwischen Inversion und Adaption). Der erreichbare Grad der Parallelisierung ist durch die Anzahl unabhängig lösbarer Teilaufgaben stark eingeschränkt. Der Austausch kompletter Matrizen erfordert zudem einen hohen Datendurchsatz zwischen den Stufen der Pipeline.

  Die Aufteilung der Funktionsgruppen auf unterschiedliche Hardware-Kom-

ponenten im vorhergehenden Kapitel basiert bereits auf diesem Prinzip: das Empfangsfilter, die FFT, die Raum-Zeit-Verarbeitung und die Decodierung bilden eine Pipeline. Die einzelnen Stufen können auf die spezifischen Eigenschaften der jeweiligen Teilaufgaben (z.B. Operationen auf Bit-Ebene für die Decodierung) optimiert und parallel verarbeitet werden.
Für die Raum-Zeit-Verarbeitung ist diese Optimierung nicht unbedingt erforderlich, da alle Teilaufgaben auf den gleichen Vektoroperationen beruhen. Eine Ausnahme stellt die Aufteilung in Gewichtung und Gewichtsberechnung dar. Aufgrund der unterschiedlichen Echtzeitanforderungen und Komplexität werden für Demonstrationssysteme diese Teilaufgaben häufig auf einer hybriden Plattform aus FPGAs und DSPs umgesetzt. FPGAs erlauben eine parallele Umsetzung der Gewichtung mit einem hohen, konstant bleibenden Datendurchsatz. DSPs bieten dagegen eine schnelle und einfache Umsetzung verschiedener Algorithmen auf einer Fließkomma-Arithmetik in einer Hochsprache wie C++. Die Gewichtung ist sehr einfach zu parallelisieren, die Gewichtsberechnung hingegen nur im Rahmen der Möglichkeiten des Signalprozessors.

- **Parallele Verarbeitung unabhängiger Datensätze:** Die Entkopplung der Unterträgergruppen im Frequenzbereich erlaubt eine parallele Verarbeitung nach dem SIMD-Prinzip. Nach der Klassifizierung von Flynn [58] bedeutet SIMD (*single instruction, multiple data*) die parallele Ausführung einer Operation auf mehreren Datensätzen. Moderne Signalprozessoren (TigerSHARC, TMS320C64x) und auch General-Purpose-Prozessoren (Pentium4 mit SSE, PowerPC mit AltiVec) ermöglichen bis zu 16 parallele Multiplikationen nach dem SIMD-Prinzip.
  Die hohe Leistungsfähigkeit paralleler Rechenwerke mit dem geringen Aufwand für nur eine gemeinsame Steuereinheit verspricht eine effiziente Hardware. Ob diese Effizienz auch in der Anwendung genutzt werden kann, hängt jedoch stark von der Art des Zugriffs auf den Hauptspeicher ab. Die parallele Berechnung nach dem SIMD-Ansatz stellt hohe Anforderungen an den Datendurchsatz zwischen Prozessor und Speicher. Der parallele Zugriff auf alle Koeffizienten in einem Taktzyklus erfordert eine große Wortbreite von Datenspeicher und Registern, z.B. durch Parallelschaltung mehrerer Speicherbänke. Für die Verarbeitung unabhängiger Datensätze reicht eine direkte Verbindung zwischen den einzelnen Speicherbänken und den Rechenwerken bzw. Registern aus (siehe Bild 4.4(a)).
  In der vorliegenden Anwendung wird der Grad der Parallelisierung mit diesem Ansatz durch die minimale Anzahl der Unterträgergruppen $U_{\min} = \frac{N}{Q_{\max}}$

beschränkt. Die Interpolation bzw. Adaption über die Frequenz führt zu einer Abhängigkeit benachbarter Unterträgergruppen. Für diesen Fall wird ein Schaltnetzwerk für einen flexibleren Zugiff der Rechenwerke auf die einzelnen Registerbereiche erforderlich (Bild 4.4(b)).



(a) Direkte Verbindung zu den Rechenwerken für unabhängige Datensätze

(b) Schaltnetzwerk zum Austausch benachbarter Daten für abhängige Datensätze

(c) Berechnung von Vektorprodukten hier nur für Zeilenvektoren einer Matrix möglich

Bild 4.4: Parallele Speicherzugriffe für einen SIMD-Prozessor

- **Parallele Berechnung von Vektorprodukten und Matrixoperationen:** Aufgrund der Matrixnotation der Algorithmen kann der SIMD-Ansatz auch für eine parallele Berechnung der einzelnen Vektorprodukte eingesetzt werden. Die Vektorprodukte sind jedoch meist Bestandteil von Matrixoperationen. Ein paralleler Zugriff auf Matrixelemente ist nur eingeschränkt möglich. Bei der Speicherzuordnung in Bild 4.4(c) ist z.B. ein paralleler Zugriff auf Zeilenvektoren möglich, Spaltenvektoren können dagegen nur sequentiell ausgelesen werden.
  Die Einschränkungen durch den Zugriff auf einen zentralen Speicher können mit systolischen Netzen vermieden werden. Der Begriff taucht zum ersten Mal bei Kung und Leiserson [59] auf und benennt ein Feld von vernetzten Prozessorelementen, das wie ein Blutkreislauf von Daten durchströmt wird. Mit jedem Taktzyklus (Puls) werden Eingangsdaten und Zwischenergebnisse auf vorgegebenen Verbindungen zwischen benachbarten Elementen ausgetauscht. Die Prozessorelemente führen jeweils eine vorgegebene Operation aus. Die Anordnung und die Vernetzung der Elemente legt dabei die Funktion des Feldes,

also den Algorithmus, fest. Die Eingangsdaten werden meist am Rand des Feldes in einer vorgegebenen Reihenfolge in das Feld geschoben, die Resultate können an anderen Elementen nach einer gewissen Latenzzeit entnommen werden. Die Berechnung aufeinander folgender Datensätze kann sich überlappen (Pipeline-Betrieb). Der Datentransfer zum Hauptspeicher wird entlastet, da alle Zwischenergebnisse im Feld gehalten und die Eingangsdaten meist gleichmäßig über die Berechnungsdauer eingespeist werden. Da nicht alle Resultate aller Prozessorelemente zum Ergebnis beitragen, ist die Effizienz nicht immer optimal. Die Festlegung auf eine Topologie lässt zudem wenig Freiraum für unterschiedliche Algorithmen. Eine optimierte Umsetzung der Cholesky-Zerlegung ist z.B. in [60] zu finden.

Beide Ansätze für die Parallelisierung der Matrixoperationen, SIMD und systolische Felder, verlieren an Effizienz bei einer variablen Vektorlänge bzw. Matrixgröße. Bei einer Auslegung auf den Worst-Case ($MQ_{\max}$ bzw. $K_{\max}$) wird bei kleineren Dimensionen nur ein Teil der vorhandenen Rechenleistung genutzt. Auf Dreiecksmatrizen beruhende Algorithmen (LDL-Zerlegung und Inversion) können unabhängig von den Systemparametern auf einer SIMD-Architektur nicht mit voller Effizienz implementiert werden.

- **Parallele Berechnung von CMACs und Division:** Auf unterster Ebene beruhen alle Algorithmen auf komplexen MAC-Operationen und Divisionen. Bei Signalprozessoren ist es gängig, Multiplikation und Addition einer MAC-Operation parallel in einer Pipeline zu berechnen. Dieser Ansatz kann sehr einfach auf die jeweils vier Multiplikationen und Additionen einer CMAC-Operation erweitert werden. Mit einer parallelen Auslegung der Multiplizierer und Addierer kann dann ein CMAC pro Taktzyklus berechnet werden.

  Divisionen werden üblicherweise über eine iterative Näherung (z.B. Newton-Rapson-Verfahren) auf Multiplikationen und Additionen reduziert und in der arithmetischen Einheit berechnet [61]. Ein auf CMACs optimiertes Rechenwerk ist für die Divisionsalgorithmen nicht ideal geeignet. Eine Hardware-Realisierung der Division liefert eine höhere Effizienz und kann parallel zur Berechnung der CMACs betrieben werden. Im Hinblick auf eine gleichmäßige Auslastung muss der Durchsatz, d.h. die Anzahl der Zyklen pro Division, an die Anforderungen der Algorithmen angepasst werden.

Eine Kombination aus dem zweiten und vierten Ansatz bietet einen hohen Grad an Parallelisierung ohne Einschränkungen für eine flexible und effiziente Progammierung für unterschiedliche Algorithmen und Übertragungsparameter. Im folgenden Kapitel wird eine auf diesem Ansatz aufbauende Hardware-Architektur vorgeschla-

gen. Die beiden anderen Ansätze werden aufgrund der genannten Einschränkungen nicht weiter betrachtet.

4.3 Hardware-Architektur

Ein SIMD-Prozessor ermöglicht eine integrierte Realisierung der Raum-Zeit-Entzerrung über die parallele Verarbeitung mehrerer Unterträgergruppen. Im Gegensatz zu einer direkten Hardware-Umsetzung bietet ein programmierbarer Prozessor die Möglichkeit, verschiedene Algorithmen in Abhängigkeit der momentanen Kanaleigenschaften und Dienstanforderungen einzusetzen. Diese Flexibilität hat jedoch ihren Preis: der Software-Ansatz benötigt bei gleicher Leistungsfähigkeit in der Regel eine größere Chipfläche und im Betrieb mehr Leistung als eine spezifische Hardware-Umsetzung.

In diesem Kapitel wird eine anwendungsspezifische Architektur für den SIMD-Prozessor vorgestellt. Ziel ist eine hohe, mit einer Hardware-Umsetzung vergleichbaren Effizienz, speziell für die in Kapitel 3 vorgestellten Algorithmen. Die Effizienz eines Prozessors entspricht dem Verhältnis zwischen der erzielbaren Rechenleistung in einer Anwendung und dem dazu nötigen Aufwand, hier z.B. Chipgröße bzw. Stromverbrauch. Wartezyklen aufgrund von Ressourcenkonflikten sowie zusätzliche Operationen zur Steuerung des Programmablaufes, wie z.B. der Adressberechnung, beschränken die Effizienz einer Software-Realisierung. Im Folgenden wird die Effizienz daher auf die Auslastung der vorhandenen Ressourcen in der laufenden Anwendung bezogen. Ziel ist demnach ein Prozessor, dessen Rechenwerke ununterbrochen an Operationen arbeiten, die unmittelbar zum Ergebnis beitragen. Eine weitere Optimierung im Hinblick auf Chipgröße, Taktrate und Stromverbrauch ist nur unter Berücksichtigung der Chiptechnologie möglich und geht über den Rahmen dieser Arbeit hinaus.

In Kapitel 4.3.1 wird zunächst das Gesamtkonzept des SIMD-Prozessors vorgestellt. Es folgt eine Beschreibung der einzelnen Komponenten der Prozessorelemente: Rechenwerk (Kapitel 4.3.2) und Speichereinheit (Kapitel 4.3.3). Anschließend wird eine Steuereinheit für die Konfiguration der Prozessorelemente auf Basis von Vektorbefehlen vorgeschlagen. Alle Komponenten werden an dieser Stelle qualitativ beschrieben. Eine Quantifizierung des Konzeptes ist am Beispiel der FPGA-Umsetzung für den Demonstrator in Kapitel 6 zu finden.

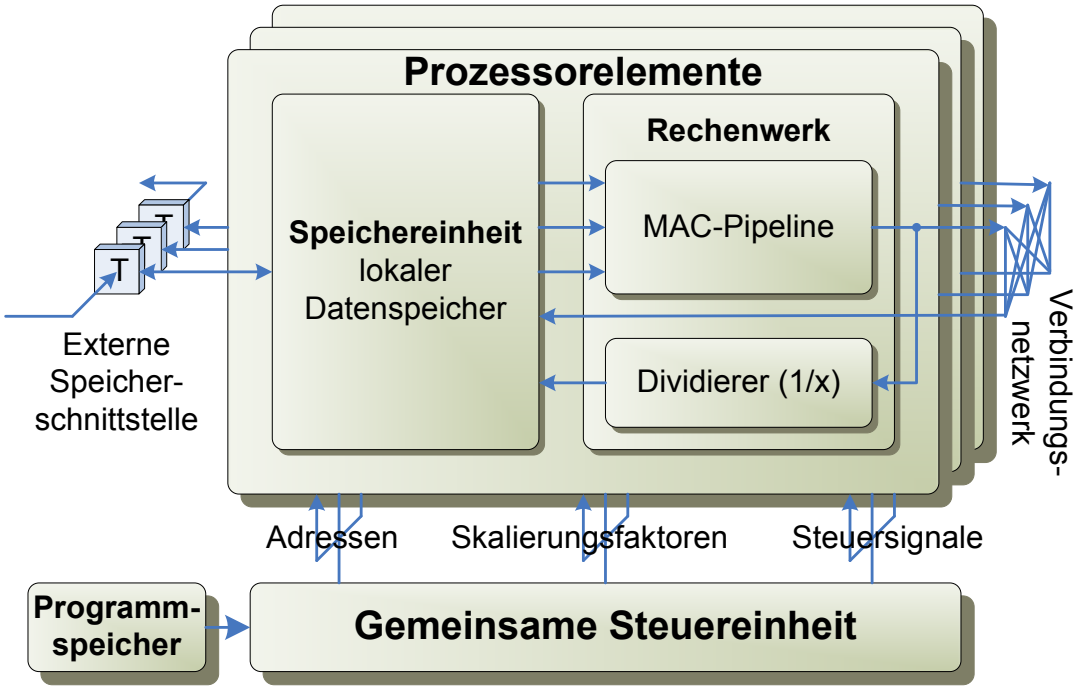


Bild 4.5: Übersicht der Prozessorarchitektur. Nach dem SIMD-Prinzip werden mehrere identische Prozessorelemente von einer gemeinsamen Steuereinheit konfiguriert. Zusammenhängende Unterträgergruppen werden auf die Prozessorelemente verteilt und parallel verarbeitet. Speichereinheit und Rechenwerk sind auf die Ausführung von Vektorprodukten optimiert.

4.3.1 SIMD-Konzept

Der SIMD-Prozessor, dargestellt in Bild 4.5, verfügt über P Prozessorelemente (PE), bestehend jeweils aus einem Rechenwerk und einer Speichereinheit. Zum Rechenwerk gehören eine MAC-Pipeline für die parallele Berechnung von CMAC-Operationen sowie ein Dividierer zur Berechnung eines reelen Kehrwertes. Die MAC-Pipeline liest ihre Eingangsdaten direkt aus dem lokalen Datenspeicher der Speichereinheit. Die sonst übliche Verwendung von Registerbänken zwischen Speicher und Rechenwerk ist für die überwiegende Berechnung von Vektorprodukten nicht sinnvoll und mit einem schnellen internen Speicher auch nicht nötig. Die Ausgangsdaten der MAC-Pipeline können über einen Ringbus mit benachbarten PE zyklisch ausgetauscht werden. Dies ermöglicht die Ausnutzung der Abhängigkeit benachbarter Unterträgergruppen, z.B. bei der Adaption über die Frequenz oder der Interpolation. Für beide Ansätze reicht eine Verbindung zum jeweils nächsten PE, wie Bild 4.4(b) skizziert. Der Dividierer bezieht seine Eingangsdaten vom Ausgang des Verbindungsnetzwerkes. Die Ergebnisse von MAC-Pipeline und Dividierer werden an die Speichereinheit zurückgegeben und dort im lokalen Datenspeicher abgelegt.

Für den Datenaustausch mit der externen Peripherie ist eine asynchrone Speicherschnittstelle vorgesehen. Zu den angeschlossenen FFT-Prozessoren wird ein kontinuierlicher Strom von Eingangs- und Ausgangsdaten angenommen, der durch eine Registerkette (*daisy chain*) geleitet wird. An jedem Glied der Kette ist jeweils ein PE angeschlossen und kann dort Eingangsdaten entnehmen und Ausgangsdaten einspeisen. Die Registerkette entspricht damit einem Schieberegister zur seriell/parallel Wandlung der Daten. Externe Speicherschnittstelle und Rechenwerk greifen unabhängig voneinander auf zwei getrennte Hälften eines Puffers zu. Nach der Verarbeitung eines Datenblockes können die Speicherinhalte dann ausgetauscht werden (*switch buffer*).

Die PE werden über eine gemeinsame Steuereinheit konfiguriert. Die Steuereinheit liest über einen Cache Befehlsworte aus einem externen Programmspeicher. Aus den Befehlen werden die Lese- und Schreibadressen, Skalierungsfaktoren sowie Steuersignale für die Konfiguration der Rechenwerke und des Datenflusses generiert und an die PE verteilt.

4.3.2 Rechenwerk

Das Rechenwerk besteht aus zwei Teilen: einer MAC-Pipeline und einem Dividierer. Die MAC-Pipeline dient der Berechnung von inneren, äußeren und elementweisen Vektorprodukten im komplexen Zahlenraum. Entsprechend den Vorüberlegungen in Kapitel 4.2 erfolgt die Verarbeitung der Vektoren seriell, die Ausführung der CMAC-Operationen dagegen parallel. Für das innere Produkt zweier Vektoren werden nacheinander die Produkte der einzelnen Elemente berechnet und akkumuliert. Das Ergebnis ist skalar und entspricht dem Akkumulatorinhalt nach der letzten Iteration. Die komplexe Multiplikation und die Akkumulation werden in der Pipeline gleichzeitig ausgeführt und zu einer CMAC-Operation zusammengefasst.

Die Initialisierung des Akkumulators mit einem Datenwort ermöglicht die Addition eines Skalars zu dem Vektorprodukt mit den ohnehin vorhandenen Ressourcen. Diese Operation der Form $d = c + \mathbf{ab}$ wird z.B. für die Matrixinversion benötigt. Zum Laden des Datenwortes wird ein zusätzlicher Port zum Speicher eingeführt.

Die resultierende Pipeline ist in Bild 4.6 dargestellt. Über drei parallele Eingänge werden komplexe Datenworte aus dem lokalen Speicher gelesen. Vier Multiplizierer und zwei Addierer bilden einen komplexen Multiplizierer für die Koeffizienten A und B. Das Ergebnis kann mit einem programmierbaren Faktor skaliert werden bevor es auf den Akkumulator gegeben wird. Die Skalierung wird über bitweises Verschieben des Binärwortes realisiert, d.h. der Skalierungsfaktor ist eingeschränkt

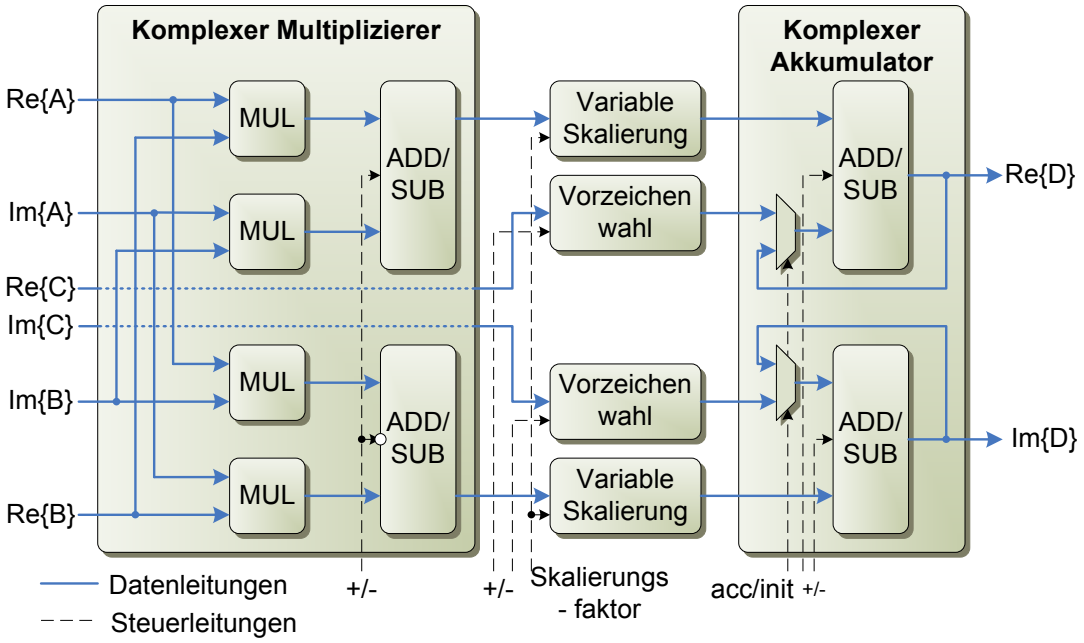


Bild 4.6: Die MAC-Pipeline dient der sequentiellen Berechnung von Vektorprodukten. Statt der Akkumulation kann auch eine Addition mit einem dritten Koeffizienten gewählt werden. Steuersignale erlauben alle möglichen Kombinationen von Vorzeichen und Konjugationen. Die Produkte können vor der Akkumulation skaliert werden.

auf ganzzahlige Zweierpotenzen. Der Akkumulator besteht jeweils für Real- und Imaginärteil aus einem Addierer und einem Multiplexer, der entweder den aktuellen Wert des Akkumulators oder den Koeffizienten C für die Addition mit dem aktuellen Produkt auswählt.

Für das elementweise Produkt oder auch Hadamard-Produkt zweier Vektoren entfällt die Akkumulation, die berechneten Produkte werden einzeln im Speicher abgelegt. Das äußere Vektorprodukt lässt sich zerlegen in mehrere Produkte eines Vektors mit einem Skalar. Für beide Produkte kann der zusätzliche Port (C) und der Akkumulator zur Addition des Produktes mit einem Vektor bzw. einer Matrix verwendet werden. Diese Operation wird bei den adaptiven Filtern zur Aktualisierung der Gewichte eingesetzt.

In allen vier Addierern kann über ein Steuersignal zwischen Addition und Subtraktion gewählt werden. Die Addierer des komplexen Multiplizierers werden wechselseitig geschaltet, die des Akkumulators getrennt voneinander. Außerdem können Real- und Imaginärteil des dritten Koeffizienten C jeweils über eine 2er-Komplementbildung negiert werden. Es ergeben sich fünf Steuerleitungen, über die alle möglichen Kombinationen an Vorzeichen und Konjugationen der Operation $\pm c^{(*)} \pm a^{(*)} b^{(*)}$ gewählt werden können.

Die MAC-Einheit ist damit in der Lage, ein beliebiges Vektorprodukt zu berechnen und auf ein Skalar, einen Vektor bzw. eine Matrix zu addieren. Vorzeichen und Konjugation der Koeffizienten können frei gewählt werden. Über die Pipeline-Struktur kann eine CMAC-Operation pro Taktzyklus berechnet werden. Die Gesamtanzahl an Taktzyklen für ein inneres oder elementweises Vektorprodukt entspricht der Vektorlänge und dem Produkt der Vektorlängen für das äußere Vektorprodukt.

Die betrachteten Algorithmen benötigen neben der Berechnung von Vektorprodukten auch eine Normierung von Vektoren mit reellen Faktoren. Eine sequentielle Division aller Vektorelemente durch den Normierungsfaktor ist sehr aufwändig. Effizienter ist die einmalige Kehrwertberechnung des Normierungsfaktors mit anschließender Multiplikation des Kehrwertes mit den Vektorelementen. Die Gewichtung des Vektors mit dem Kehrwert kann in der MAC-Pipeline ausgeführt werden. Für die Berechnung des Kehrwertes wird ein in Hardware ausgelegter Dividierer verwendet.

Der Dividierer hat also die Aufgabe den Kehrwert eines reellen Faktors zu berechnen. Die Division wird im Vergleich zu den MAC-Operationen nur selten benötigt. Eine parallele Realisierung des Dividierers für einen Kehrwert pro Taktzyklus hat einen hohen Ressourcenverbrauch und ist damit bei sporadischer Nutzung ineffizient. Die Alternative ist eine serielle Realisierung mit mehreren Taktzyklen pro Kehrwertberechnung. Die Erfahrungen bei der Realisierung haben jedoch gezeigt, dass bei gleichem Durchsatz ein paralleler Dividierer wesentlich weniger Ressourcen beansprucht als mehrere serielle Dividierer. Aus diesem Grund werden im Folgenden parallele Dividierer verwendet, die jeweils von einer Untergruppe mit P_d Prozessorelementen genutzt werden. Die resultierende Architektur ist in Bild 4.7 dargestellt.

Die Eingangswerte für die Kehrwertbildung aller P_d Prozessorelemente werden parallel in eine Registerkette geschrieben. Durch serielles Auslesen der Kette werden die Eingangswerte nacheinander dem eigentlichen Dividierer zugeführt und dort verarbeitet. Die berechneten Kehrwerte können mit einem variablen Faktor skaliert werden (siehe MAC-Pipeline). Der variable Skalierer speist dann seriell eine zweite Registerkette. Wenn alle P_d Kehrwerte berechnet und skaliert sind, können die jeweiligen Resultate der PE parallel aus dieser Kette entnommen werden. Mit einem Kehrwert pro Taktzyklus kann die Gesamtanordnung alle P_d Zyklen mit neuen Eingangswerten gespeist werden.

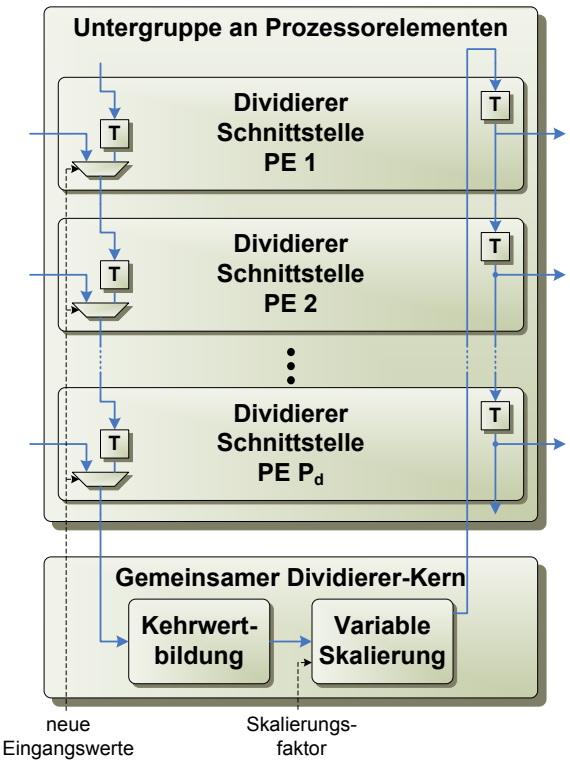


Bild 4.7: Dividierer zur Berechnung positiver, reelwertiger Kehrwerte. Über die parallel/seriell Wandlung am Eingang und Ausgang wird ein paralleler Dividierer sequentiell von mehreren Prozessorelementen genutzt. Die berechneten Kehrwerte können anschließend skaliert werden.

4.3.3 Speichereinheit

Die Speichereinheit (siehe Bild 4.8) umfasst den lokalen Datenspeicher einer PE, ein Schaltnetzwerk für den selektiven Zugriff auf mehrere Speicherbänke sowie die Schnittstelle zu der externen Peripherie. Der lokale Datenspeicher enthält konstante, vordefinierte Datensequenzen, wie z.B. Spreizcodes oder Trainingssequenzen und speichert temporäre Zwischenergebnisse des Rechenwerkes. Die MAC-Pipeline benötigt drei Ports zum parallelen Lesen der Koeffizienten A, B und C. Das parallele Schreiben der Ergebnisse von MAC-Pipeline und Dividierer erfordert zwei weitere Ports. Mit fünf parallelen Ports könnte das Rechenwerk ohne Zugriffskonflikte auf den Speicher zugreifen.

Der Ressourcenverbrauch eines Speichers nimmt jedoch mit der Portanzahl stark zu. Sinnvoll realisierbar sind meist nur Speicherbänke mit einem oder zwei Ports (*Single-* bzw. *Dual-Port-RAM*), die in jedem Taktzklus entweder zum Lesen oder Schreiben genutzt werden können. In der Speichereinheit werden zwei parallele Bänke (Bank X und Bank Y) mit jeweils zwei Ports als lokaler Datenspeicher eingesetzt. Für die

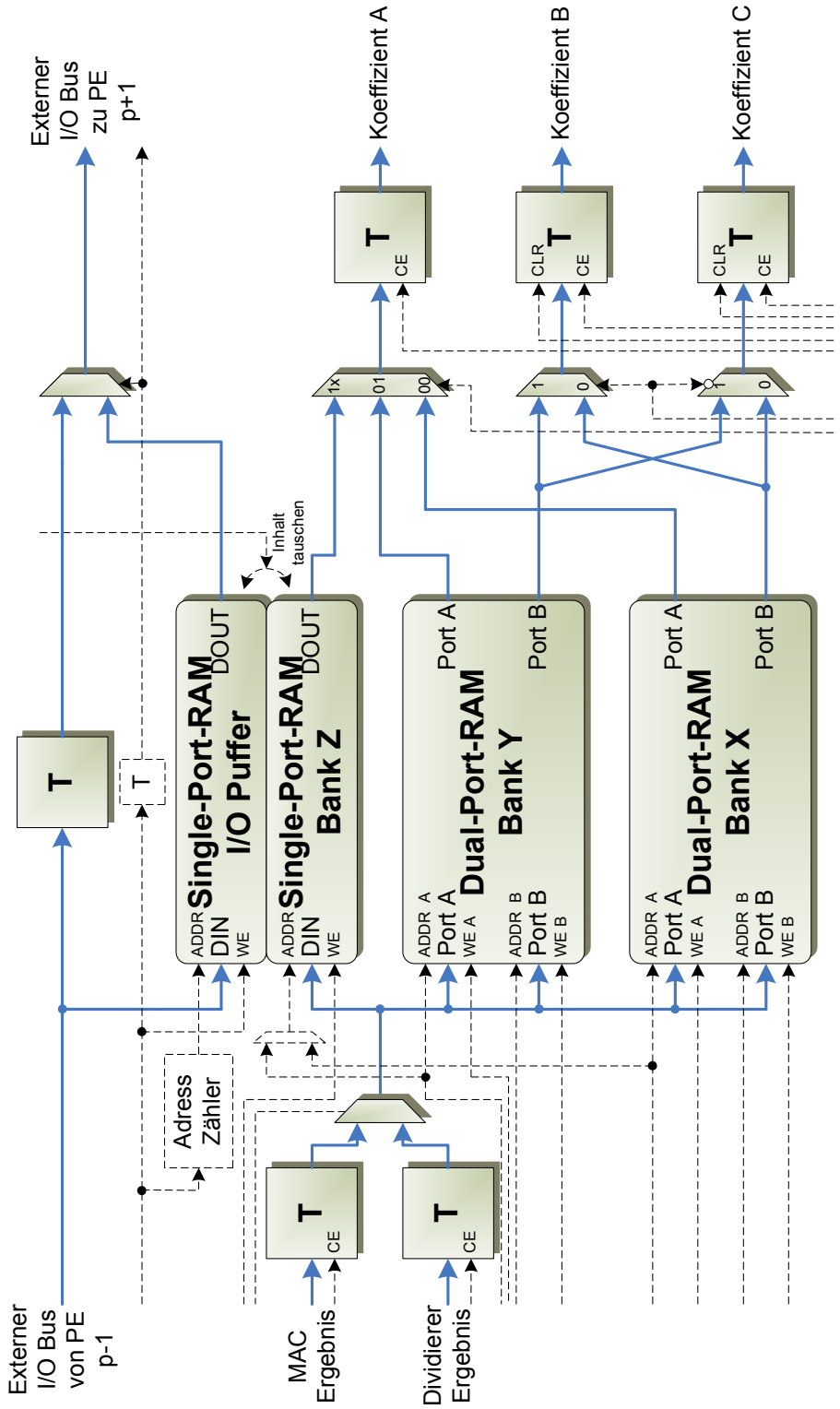


Bild 4.8: Speichereinheit mit zwei Bänken zum Speichern lokaler Daten und zwei schaltbaren Puffern für den asynchronen Datenaustausch mit der externen Peripherie. In jedem Taktzyklus können drei Koeffizienten gelesen und ein Resultat geschrieben werden.

I/O-Schnittstelle zur externen Peripherie sind zwei zusätzliche Datenpuffer vorgesehen. Auf einen der beiden Puffer (Bank Z) kann das Rechenwerk zugreifen, der andere (I/O-Puffer) ist asynchron zum PE über eine Registerkette mit der externen Peripherie verbunden. Der Zugriff auf die beiden Puffer kann wechelseitig ausgetauscht werden (*switch buffer*). Effektiv können die Puffer als physikalisch getrennte Bänke angesehen werden, deren Speicherinhalt von einem Taktzyklus auf den nächsten ausgetauscht werden kann. Um Ressourcen zu sparen werden beide Puffer nur mit jeweils einem Port ausgelegt. Eine alternative Realisierung über einen gemeinsamen Speicherbereich mit zwei Ports ist möglich, verbraucht aber mehr Ressourcen.

Ein paralleler Zugriff ist damit nur auf drei getrennte Speicherbereiche mit maximal zwei Ports pro Bereich möglich. Bei der Programmierung muss die Zuordnung der Speicherbereiche entsprechend sorgfältig gewählt werden um Zugriffskonflikte zu vermeiden.

Zum Lesen der Koeffizienten A, B und C müssen die Speicherports mit den Eingängen der MAC-Pipeline verschaltet werden. Das in Bild 4.8 gezeigte Schaltnetzwerk erlaubt mit minimalem Ressourcenverbrauch alle möglichen Kombinationen für den Lesezugriff, mit einer Ausnahme: Koeffizient C kann nicht aus Bank Z gelesen werden.

Die drei gelesenen Datenworte können in steuerbaren Registern über ein *clock enable* (CE) gespeichert werden. Für Vektoroperationen bei denen einer der Koeffizienten ein Skalar ist, kann auf diese Weise der entsprechende Speicherport nach dem Lesen des Datenwortes freigegeben werden. Die Register der Koeffizienten B und C liefern zudem über ein *clear*-Signal (CLR) einen vordefinierten konstanten Wert. Mit der Konstante -MAXINT für Koeffizient B kann bei entsprechender Wahl der Vorzeichen in der MAC-Pipeline der Koeffizient A unverändert in den Akkumulator geladen werden. Mit der Konstante *Null* für Koeffizient C kann auf die Addition im Akkumulator verzichtet werden. In beiden Fällen muss kein Speicherport für den entsprechenden Koeffizienten reserviert werden. Für die Berechnung von inneren Vektorprodukten wird der Speicherport für Koeffizient C höchstens im ersten Zyklus zum Laden des Akkumulators benötigt.

Die Ergebnisse von MAC-Pipeline und Dividierer werden jeweils in einem Register zwischengespeichert, bis ein Port der gewünschten Speicherbank frei wird. Bei der Programmierung muss ausgeschlossen werden, dass ein neues Ergebnis eintrifft bevor das vorhergehende geschrieben wurde. Mit einem Schaltnetzwerk aus fünf Multiplexern könnte an jedem Port jeweils eines der beiden Ergebnisse geschrieben werden. Der im vorhergehenden Kapitel beschriebene Dividierer kann aber nur al-

le P_d Taktzyklen einen Kehrwert berechnen und wird zudem von den Algorithmen selten genutzt. Daher teilen sich MAC-Pipeline und Dividierer für die Ergebnisse einen gemeinsamen Bus, der direkt mit allen verfügbaren Schreibports verbunden ist. Das Schaltnetzwerk wird damit auf einen Multiplexer zum Auswählen des Ergebnisses reduziert. Die Auswahl eines Ports erfolgt über eines von fünf getrennten *write enable* Signalen (WE). Jeder Port der Speicherbänke X und Y verfügt über eine eigene Adressleitung. Die Speicherbank Z verwendet die Adresse für Port A der Bank X oder Y.

Die Schnittstelle zur externen Peripherie ist dezentral realisiert. Die Eingangsdaten werden sequentiell über eine Registerkette von PE zu PE geleitet (in Bild 4.8, oben). Die PE entnehmen die ihnen zugewiesenen Datenworte der Kette und schreiben sie in den I/O-Puffer. Der ursprüngliche Inhalt einer Speicherzelle wird während des Schreibvorgangs ausgelesen (*read before write*) und ersetzt den jeweiligen Eingangswert in der Registerkette. Auf diese Weise werden die Eingangsdaten mit den Ergebnissen der Raum-Zeit-Verarbeitung aus dem vorhergehenden Datenblock vertauscht. Die Ausgangsdaten können dann am Ende der Registerkette sequentiell entnommen werden.

Die Aufteilung der Eingangsdaten auf die einzelnen PE wird über zwei parallel zum Datenbus geführte Signale gesteuert. Ein Trigger-Signal markiert jeweils das erste Datenwort einer Unterträgergruppe. Das zweite Signal aktiviert in einem PE den oben beschriebenen Datenaustausch von Ein- und Ausgangsdaten. Die Aktivierung wird, getriggert durch das erste Signal, von einem PE zum nächsten weitergereicht. Auf diese Weise werden die Unterträgergruppen sequentiell auf die PE verteilt. Die Adressen für den I/O-Puffer werden lokal generiert.

4.3.4 Steuereinheit

Die Funktionalität der Algorithmen wird mit Hilfe eines Compilers oder Assemblers auf Maschinenprogramme abgebildet. Zur Laufzeit liest die Steuereinheit sequentiell den Programmcode ein und konfiguriert den Prozessor entsprechend den ausgewerteten Befehlen. In der Regel unterstützen Signalprozessoren einen dynamischen Programmablauf mit Hilfe von Schleifen, bedingten Sprüngen oder Interrupts. Für die Inkrementierung der Schleifenindizes und Adressen oder die Auswertung der Bedingungen sind zusätzliche Operationen nötig, die neben den eigentlichen Daten im Rechenwerk verarbeitet werden.

Speziell für den SIMD-Betrieb ist der Programmablauf immer unabhängig von den verarbeiteten Daten (eine Abhängigkeit wäre mit der parallelen Verarbeitung

unterschiedlicher Datensätze nicht vereinbar). Die zur Steuerung des Programmablaufes benötigten Operationen werden daher direkt innerhalb der zentralen Steuereinheit auf einem zusätzlichen Rechenwerk ausgeführt.

Der dynamische Programmablauf der betrachteten Algorithmen beschränkt sich auf mehrfach geschachtelte Schleifen für die sequentielle Berechnung der Matrix- bzw. Vektoroperationen. Die Grenzen für die Laufvariablen der Schleifen sind schon im voraus bekannt. Die effektiven Befehle aller Schleifendurchläufe können dann offline generiert und sequentiell im Programmspeicher abgelegt werden (*loop unrolling*). Damit ergibt sich ein streng linearer Programmdurchlauf und die Steuereinheit, besonders der Zugriff auf den Programmspeicher, wird wesentlich vereinfacht. Die Bereitstellung von vier Adressen, zwei Skalierungsfaktoren und der Konfiguration des Rechenwerkes erfordert jedoch eine große Wortbreite der Befehle (VLIW, *very large instruction words*). Der Speicherbedarf für den ausgerollten Programmcode sowie der Datendurchsatz zum Lesen der Programme stellen dann hohe Anforderungen.

Die Einführung von Vektorbefehlen reduziert die Anforderungen an den Programmspeicher unter Beibehaltung der Vorteile eines linearen Programmablaufes. Bei der Berechnung von inneren Vektorprodukten und Hadamard-Produkten wird linear auf den Speicher zugegriffen. Nach Angabe einer Basisadresse sowie der Vektorlänge können alle folgenden Adressen in der Steuereinheit berechnet werden. Für den Zugriff auf die Zeilenvektoren einer spaltenweise abgelegten Matrix muss zusätzlich eine Sprungweite entsprechend der Spaltenlänge angegeben werden; im Falle einer Dreiecksmatrix muss die Sprungweite in jeder Iteration um eins reduziert bzw. erhöht werden.

Auf diese Weise können inneres Vektorprodukt und Hadamard-Produkt mit nur einem Befehl ausgeführt werden. Das äußere Produkt zweier Vektoren wird aus mehreren Hadamard-Produkten des einen Vektors mit jeweils einem Element des anderen zusammengesetzt. Bei Angabe zusätzlicher Sprungweiten kann das Prinzip auch auf die äußeren Schleifen zur Berechnung ganzer Matrixoperationen erweitert werden.

Der Zugriff auf den Speicher und die Verarbeitung innerhalb der Pipelines sind in der Regel mit Latenzzeiten von mehreren Taktzyklen verbunden. Die Stufen einer Pipeline müssen zeitlich versetzt konfiguriert werden. VLIW-Signalprozessoren wie der TMS320C6x definieren mit einem Befehlswort jeweils unabhängig voneinander die aktuelle Funktion der einzelnen Stufen. Die Pipeline-Struktur wird dann bei der Erzeugung des Maschinenprogramms unter Berücksichtigung der Latenzen aufgebaut (*software pipelining*). Dieser Ansatz ist hier nicht anwendbar, da die Ausführung der Vektorbefehle asynchron zu den Verzögerungen der Stufen erfolgt.

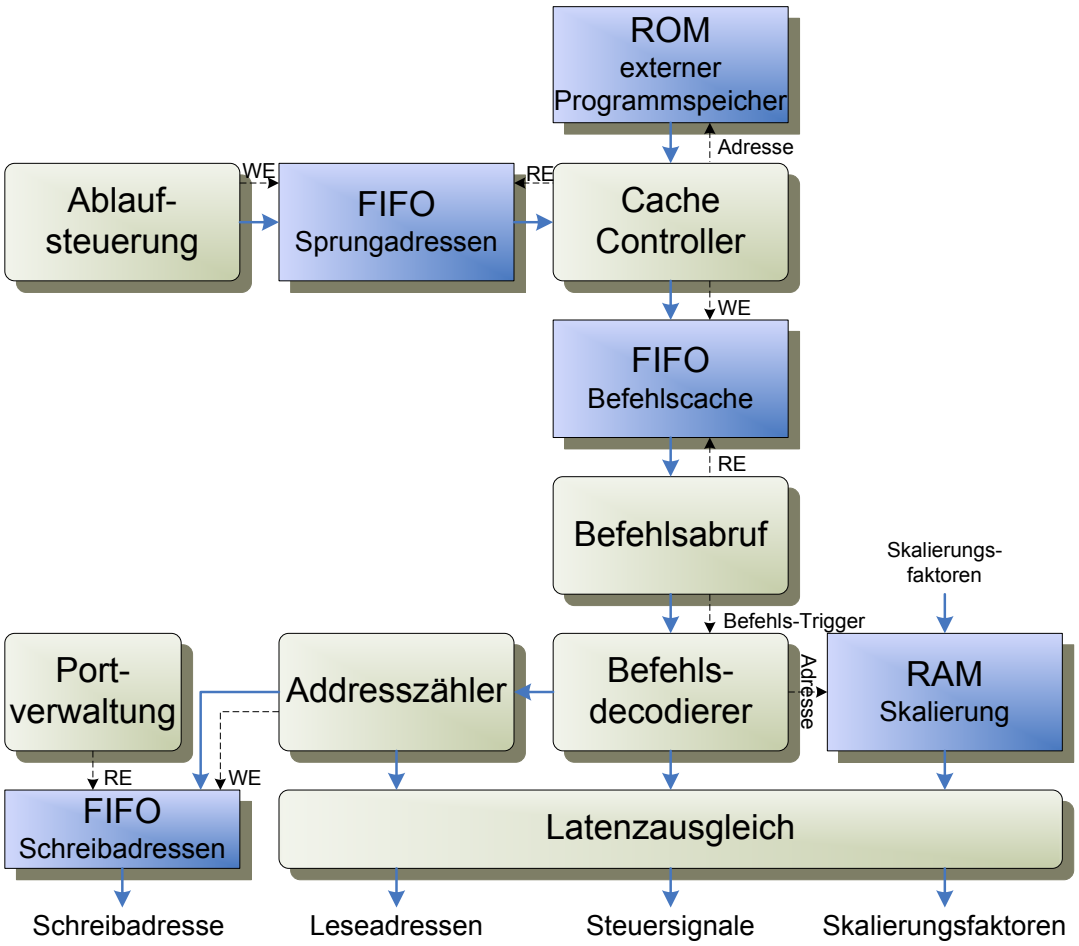


Bild 4.9: Steuereinheit zur Konfiguration der Prozessorelemente auf Basis von Vektorbefehlen in einem linearen Programmablauf mit von außen vorgegebenen Sprungadressen. Ein Cache gleicht Durchsatzspitzen im Zugriff auf den externen Programmspeicher aus. Zur Kompensation der Latenz in den Pipelines der Rechenwerke werden die Steuersignale entsprechend verzögert.

Ein Vektorbefehl enthält aus diesem Grund die Konfiguration für den gesamten Durchlauf der Pipeline, inklusive Lese- und Schreibadressen. Die Steuersignale werden dann in der Steuereinheit mit Hilfe von Registerketten verzögert, um die Latenzzeiten innerhalb der Pipeline auszugleichen. Wie weiter oben bereits erläutert, erfolgt der Schreibvorgang in Abhängigkeit der aktuellen Portbelegung, d.h. mit nicht definierter Latenz. Für die Schreibadressen werden statt der Registerketten daher FIFOs eingesetzt.

Die resultierende Architektur der Steuereinheit ist in Bild 4.9 dargestellt. Eine externe Ablaufsteuerung gibt die Sprungadressen für sequentiell abzuarbeitende Unterprogramme vor und legt sie in einem FIFO-Speicher ab. Ein Cache-Controller liest aus einem externen ROM die an den jeweiligen Adressen gespeicherten

Unterprogramme und schreibt sie in einen internen Cache. Der effektive Befehlsdurchsatz variiert mit der Vektorlänge der einzelnen Befehle. Die Schnittstelle zum externen Programmspeicher ist auf den mittleren Durchsatz ausgelegt. Der Cache ist wegen des linearen Programmablaufes als FIFO aufgebaut und dient zum Ausgleich kurzzeitiger Durchsatzspitzen. Sprungbefehle im Programmcode markieren das Ende eines Unterprogrammes. Der Cache-Controller führt die Sprünge sofort aus, indem er einen neuen Wert aus dem Adress-FIFO holt und an der entsprechenden Stelle im Programmspeicher mit dem Lesen fortfährt.

Jeder Befehl enthält eine Vektorlänge zur Spezifikation der Anzahl an Taktzyklen für die Verarbeitung. Der Befehlsabruf liest die Befehlsworte aus dem Cache-FIFO und decodiert jeweils die Vektorlänge. Die Befehlsworte werden für die entsprechende Dauer gehalten bevor das jeweils nächste gelesen wird. Jedes neue Befehlswort wird über ein Trigger-Signal signalisiert. Der Decodierer wandelt die Maschinenbefehle in Steuersignale um. Die Basisadressen und Sprungweiten übergibt er an einen Adresszähler, wo die absoluten Adressen für die einzelnen Zyklen eines Vektorbefehls generiert werden. Die Befehle enthalten Adressen für einen Speicher mit den Skalierungsfaktoren. Mit diesem indirekten Zugriff kann die Skalierung im laufenden Betrieb an Dienstgüteanforderungen oder Kanaleigenschaften angepasst werden. Die generierten Adressen, Skalierungsfaktoren und direkt vom Decodierer kommende Steuersignale werden wie zuvor erläutert verzögert um die Latenz auszugleichen. Eine Portverwaltung prüft die aktuelle Belegung der Ports und den Bedarf für einen Schreibzugriff in der Speichereinheit und lädt bei entsprechender Verfügbarkeit die zugehörige Schreibadresse aus dem FIFO.

Die Ablaufsteuerung muss die Startadressen für die Unterprogramme frühzeitig liefern, um den Cache gefüllt zu halten. Sie hat daher keinen Einfluss auf den Zeitpunkt der Ausführung. Der Prozessor tauscht über den I/O-Puffer Daten mit der externen Peripherie aus. Zur Synchronisation des Prozessors mit dem Schaltzeitpunkt für das Austauschen des Puffers wird ein sleep-Befehl eingeführt. Der sleep-Befehl versetzt den Prozessor durch Abschalten des Taktsignals in einen Ruhezustand, unmittelbar bevor die Daten für den folgenden Befehl aus dem Speicher gelesen werden. Der Prozessor kann dann von außen synchron zu einem neuen Datensatz reaktiviert werden. Wenn die volle Prozessorleistung nicht benötigt wird, z.B. bei reduzierter Nutzeranzahl oder während nicht genutzter Zeitschlitze, kann über den Ruhezustand der Stromverbrauch reduziert werden.

4.4 Fazit

Der hohe Aufwand für die Signalverarbeitung in hochratigen Mehrantennensystemen rechtfertigt eine Partitionierung auf anwendungsspezifische Hardware-Bausteine. Die Transformation in den Frequenzbereich ermöglicht eine einfache Parallelisierung der Raum-Zeit-Verarbeitung durch die Aufteilung in unabhängige Datensätze. Das hier vorgestellte Konzept für einen Parallelprozessor nutzt diese Eigenschaft mit einer SIMD-Architektur und erlaubt so eine flexible Software-Realisierung der Raum-Zeit-Verarbeitung mit einer hohen, effizient genutzten Rechenleistung. Der Prozessor kann in einem Zugangspunkt mit einheitlichen Schnittstellen abwechselnd im Sende- und Empfangsbetrieb jeweils für eine Ein- oder Mehrträgermodulation eingesetzt werden.

Die in Kapitel 3 vorgestellten Algorithmen für die lineare Frequenzbereichsentzerrung stellen typische Anwendungen für den Prozessor dar. Das Aufgabenspektrum umfasst die Kanalschätzung, eine Interpolation zwischen den Trainingssymbolen, eine Inversion oder Zerlegung der geschätzten Systemmatrix, Matrix-Vektorprodukte für die Entzerrung, Vorverzerrung oder ein *matched filter* des Datenstromes, das Lösen von Gleichungssystemen über Vorwärts/Rückwärts-Einsetzen sowie das Nachführen der Kanalschätzung oder der Gewichte über eine iterative Adaption nach dem LMS-, RLS- oder Kalman-Ansatz mit einer optionalen KLT. Die Vielseitigkeit der möglichen Anwendungen unterstreicht den Vorteil einer Software-Realisierung. Die Ausführung aller Verarbeitungsschritte in einem Prozessor auf einem gemeinsamen Speicherbereich minimiert den Datentransfer und ermöglicht einen flexiblen Einsatz der verfügbaren Rechenkapazität. Alle betrachteten Algorithmen basieren auf komplexwertigen Vektorprodukten und erlauben damit eine einheitliche Optimierung der Rechenwerke.

Die wesentlichen anwendungsspezifischen Eigenschaften der vorgeschlagenen Architektur sind:

- SIMD-Architektur mit mehreren Prozessorelementen und einer gemeinsamen Steuereinheit zur parallelen Verarbeitung mehrerer Unterträgergruppen.
- Zusammenfassung von jeweils vier parallelen Multiplizierern und Addierern zu einer komplexwertigen MAC-Pipeline für die Berechnung einer CMAC-Operation pro Taktzyklus.
- Dedizierter Dividierer für die Berechnung reeler, positiver Kehrwerte. Zur Reduktion von Ressourcenverbrauch und Latenzzeit wird ein paralleler Dividierer mit einer Division pro Taktzyklus von mehreren Prozessorelementen genutzt.
- Verzicht auf eine Registerbank zwischen Speicher und Rechenwerk. Die

MAC-Pipeline ist direkt an den lokalen Speicher angebunden, der Dividierer verarbeitet ausschließlich Ergebnisse der MAC-Pipeline.

- Zwei lokale Speicherbänke mit jeweils zwei Ports ermöglichen es, pro Taktzyklus drei Koeffizienten zu lesen und ein Ergebnis zu schreiben. Eine zusätzliche Speicherbank dient dem kontinuierlichen, asynchronen Datenaustausch mit der externen Peripherie.
- Benachbarte Prozessorelemente können Ergebnisse der MAC-Pipeline über einen Ringbus austauschen, z.B. für eine Interpolation oder Adaption über die Frequenz.
- Statischer Programmablauf mit festgelegten Sprungbefehlen. Ein externer Controller wählt die zugehörigen Sprungadressen zur Laufzeit und ermöglicht so eine dynamische Abfolge von Unterprogrammen.
- Vektorbefehle reduzieren den Speicherbedarf und erlauben eine einfach strukturierte Programmierung der Vektorprodukte.

KAPITEL 5

Zahlendarstellung im Rechenwerk

Für die Verarbeitung in Festkomma-Arithmetik muss die Skalierung der Zwischenergebnisse schon bei der Programmierung festgelegt werden. Die Dimensionierung der Wortbreiten und die Wahl der Skalierung sind entscheidend für die Leistungsfähigkeit des Systems. Dies gilt besonders für die Matrixinversion und den RLS-Algorithmus, die beide als numerisch kritisch gelten. Ein Fließkomma-Rechenwerk skaliert die Zwischenergebnisse zur Laufzeit in Abhängigkeit der Daten, reduziert damit den Programmieraufwand und erhöht die Genauigkeit. Aufgrund einer im Vergleich größeren Chipfläche, höheren Leistungsaufnahme und einer geringeren Taktrate haben sich Signalprozessoren mit Fließkomma-Arithmetik in der drahtlosen Kommunikation bisher kaum durchsetzen können.

In Kapitel 5.1 werden ausführlich die Anforderungen an eine Festkomma-Arithmetik untersucht. Anschließend wird in Kapitel 5.2 alternativ eine Fließkomma-Arithmetik vorgestellt und bewertet.

5.1 Festkomma-Arithmetik

Für den SIMD-Prozessor wird zunächst der effizientere Weg einer Festkomma-Arithmetik untersucht. In Kapitel 5.1.1 wird zunächst ein allgemeines Modell aufgestellt, das Wortbreiten und Skalierung des in Kapitel 4.3.2 eingeführten Rechenwerkes definiert. Es folgt in Kapitel 5.1.2 eine Dimensionierung der Parameter am Beispiel der Matrixinversion nach dem Kriterium eines zur Fließkomma-Referenz vergleichbaren Bitfehlerverhältnisses. Schließlich werden in Kapitel 5.1.3 die Anforderungen des numerisch ebenfalls kritischen RLS-Algorithmus an die erforderliche Speicherwortbreite untersucht.

5.1.1 Modell

Das im Folgenden verwendete Zahlenformat stellt vorzeichenbehaftete ganze Zahlen im Binärsystem mit einer Wortbreite W dar. Negative Zahlen werden im Zweierkomplement dargestellt. Das oberste Bit (*most significant bit, MSB*) übernimmt die

Funktion des Vorzeichenbits. Die Kommastelle liegt immer rechts vom untersten Bit (*least significant bit, LSB*). Damit ergibt sich ein gültiger Zahlenbereich von -2^{W-1} bis $2^{W-1}-1$.

Die Wortbreite kann innerhalb der Pipeline variieren. Zur Anpassung an den jeweiligen Wertebereich werden die Zwischenergebnisse durch Multiplikation mit einem Faktor 2^S skaliert. Dabei ist S ein vorzeichenbehafteter ganzzahliger Skalierungsfaktor. Realisiert wird die Skalierung durch bitweise Verschiebung der Binärworte, entweder über eine feste Verdrahtung der Hardware oder über so genannte *barrel shifter* mit variablen Skalierungsfaktoren. Alle außerhalb der neuen Wortbreite liegenden Bits werden abgeschnitten, d.h. es wird weder gerundet noch auf den Maximalwert begrenzt.

Die Skalierung kann als virtuelle Verschiebung der Kommastelle interpretiert werden: positive S verschieben das Komma um S Bitstellen nach rechts, negative nach links. Die relative Position des virtuellen Kommas zum Festkomma rechts vom LSB entspricht analog zur Fließkomma-Darstellung dem Exponenten E zur Basis Zwei. Der Exponent muss bei der Programmierung berücksichtigt werden, so ist z.B. die Addition zweier Zahlen nur mit gleichem Exponenten zulässig.

Bild 5.1 gibt für alle Signalwege innerhalb des Rechenwerkes jeweils links des Signals die Wortbreite und rechts den Exponenten an. Die Wortbreite des Datenspeichers sei W_{mem}, die Exponenten der Koeffizienten A,B und C seien E_a, E_b und E_c. Die Multiplikation erfordert eine Verdoppelung der Wortbreite, wobei das MSB ausschließlich für das Produkt von -MAXINT × -MAXINT benötigt wird. Der Exponent des Produktes entspricht der Summe aus E_a und E_b. Die Produkte werden mit dem festen Faktor S_{mul} vorskaliert und in der Wortbreite auf W_{add} begrenzt. Bei der Addition der Teilprodukte zur Berechnung von Real- und Imaginärteil der komplexen Multiplikation muss die Wortbreite um ein Bit erhöht werden um einen Überlauf zu verhindern. Der Exponent bleibt dabei unverändert.

Mit dem variablen Skalierungsfaktor S_{cmul} wird anschließend die hohe Dynamik der komplexen Produkte an die Wortbreite W_{acc} des Akkumulators angepasst. Am Ausgang des Akkumulators ist noch eine feste Skalierung S_{acc} vorgesehen. Negative S_{acc} erhöhen die Genauigkeit durch Berücksichtigung niederwertiger Bits in der Akkumulation, die erst bei der Skalierung des Resultates hinausgeschoben werden.

Im parallelen Zweig für den Koeffizienten C wird für die optionale Negation die Wortbreite um eins erhöht. Ähnlich wie bei der Multiplikation wird das hinzugefügte MSB jedoch nur für die Vorzeichenumkehr von -MAXINT benötigt. Die feste Skalierung am Ausgang des Akkumulators wird durch eine Vorskalierung von C mit $-S_{acc}$ kompensiert. Die Summe von C und einem Vektorprodukt hat damit immer

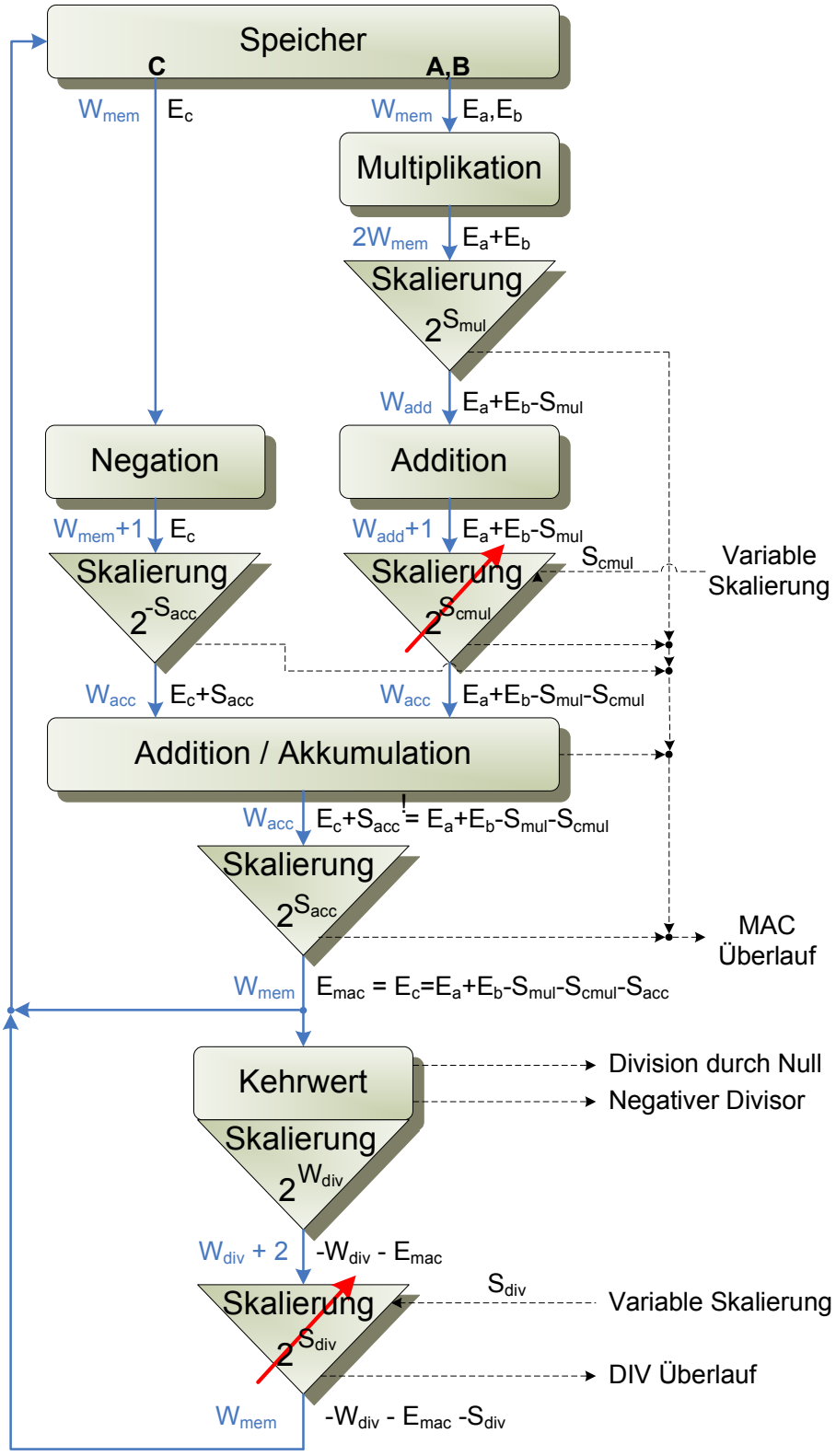


Bild 5.1: Modellierung der Festkomma-Arithmetik für das Rechenwerk aus Kapitel 4.3.2. Links der Signalwege ist jeweils die Wortbreite, rechts der Exponent angegeben. Der Exponent bestimmt die Position eines virtuellen Kommas, das in der Hardware nicht berücksichtigt wird.

den gleichen Exponenten wie C, eine Eigenschaft die den Anforderungen der betrachteten Algorithmen entspricht. Über den variablen Skalierungsfaktor S_{cmul} des Produktes müssen die Exponenten der Summanden angeglichen werden:

$$S_{cmul} \stackrel{!}{=} E_a + E_b - E_c - S_{acc} - S_{mul} \tag{5.1}$$

Entsprechend den Anforderungen der Algorithmen ist die Berechnung des Kehrwertes auf positive, reelle Zahlen beschränkt. Der Kehrwert wird mit W_{div} Nachkommastellen berechnet. Für die Division durch Eins wird eine Vorkommastelle benötigt, eine weitere Stelle für das Vorzeichenbit. Damit ist die Wortbreite nach der Division gleich $W_{div} + 2$. Der Exponent wechselt bei der Division das Vorzeichen und wird für die Verschiebung des Kommas hinter das LSB um W_{div} reduziert. Eine variable Skalierung S_{div} erlaubt eine Anpassung an den Wertebereich im Speicher.

Überläufe können ausschließlich in den Skalierungsstufen sowie im Akkumulator auftreten. Bei der Division führt außerdem ein negativer Divisor und eine Division durch Null zu einem kritischen Fehler.

5.1.2 Anforderungen für die Matrixinversion

Die Anforderungen an die Festkomma-Arithmetik sind abhängig von dem Algorithmus, den Kanaleigenschaften, den Übertragungsparametern sowie der geforderten Übertragungsqualität. Die Matrixinversion, eine typische Anwendung mit hohen Anforderungen, soll im Folgenden als *worst case* für eine Dimensionierung der Wortbreiten und festen Skalierungsfaktoren dienen. Dazu müssen zunächst sinnvolle Grenzen der Einflussgrößen spezifiziert werden. Auf Grundlage von Erfahrungswerten, hier simulierten Verteilungsdichtefunktionen, werden dann die Daten so skaliert, dass Überläufe mit ausreichender Wahrscheinlichkeit vermieden werden. Die bitgenaue Simulation der Festkomma-Arithmetik erlaubt eine Systembewertung unter dem Einfluss der Quantisierungsfehler in Abhängigkeit der Speicherwortbreite. Die Dimensionierung erfolgt schließlich über den Vergleich mit der Referenz einer Fließkomma-Arithmetik.

Moderne Kommunikationssysteme bieten Dienste mit zum Teil gegensätzlichen Anforderungen, so z.B. eine hohe Fehlersicherheit bei nahezu beliebiger Verzögerung für eine reine Datenübertragung oder aber harte Echtzeitanforderungen bei einer höheren Fehlertoleranz für die Audio- und Videoübertragung. Die jeweils erforderliche Dienstgüte wird über eine geeignete Kombination aus Kanalcodierung und Paketwiederholung (ARQ) erreicht. Im Sinne der Vereinfachung und Allgemeingültigkeit wird hier im Folgenden nur das uncodierte Bitfehlerverhältnis (BER)

betrachtet. Mit einer einheitlichen Forderung nach einem BER $\approx 10^{-2}$ kann über eine entsprechende Kanalkodierung die jeweils gewünschte Restfehlerrate erreicht werden.

Ein Empfänger mit $M = 8$ Antennen und perfekter Kanalkenntnis soll eine MMSE-Entzerrung über die Matrixinversion nach Gleichung (3.11) durchführen. Die für die Inversion entscheidende Matrixkondition ist eine durch den Kanal bestimmte Zufallsgröße. Der Erwartungswert ist abhängig von der Teilnehmeranzahl K, der räumlichen Korrelation γ sowie dem Signal-zu-Rauschverhältnis (SNR). Durch die Verringerung der Teilnehmeranzahl kann auf Kosten der spektralen Effizienz die Anzahl der Freiheitsgrade erhöht und somit die Kondition verbessert werden. Eine starke räumliche Korrelation erhöht die lineare Abhängigkeit der Spalten in der Systemmatrix, verschlechtert also ihre Kondition. Ein endliches SNR sichert eine obere Beschränkung der Kondition: bei einem niedrigen SNR geht die Δ-Matrix zunehmend in eine Diagonalmatrix über und weist unabhängig von Freiheitsgraden und Korrelation eine gute Kondition auf.

In Bild 5.2 ist das uncodierte BER über dem SNR für eine 64-QAM-Modulation mit $K = 8$ Teilnehmern und verschiedene Korrelationsfaktoren $\gamma = 0 \ldots 0.8$ aufgetragen (gestrichelte Linien). Das geforderte BER von 10^{-2} kann mit zunehmender Korrelation nur mit einem sehr großen Störabstand erreicht werden. Während theoretisch eine konstant hohe spektrale Effizienz von $48\,{}^{\text{bit}}/_{\text{s Hz}}$ erzwungen werden kann, ist dieser Betrieb in Bezug auf die Leistungseffizienz nicht mehr sinnvoll. In der Praxis sind solche idealen Bedingungen zudem auf Grund anderer Einflüsse, wie z.B. Kanalschätzfehler oder Phasenrauschen, kaum zu erreichen. Die Reduktion der Teilnehmeranzahl oder der Modulationsstufe verbessert die Kondition bzw. erhöht die Robustheit und verringert somit das erforderliche SNR. Eine Linkadaption passt so üblicherweise die spektrale Effizienz an die vorliegenden Bedingungen an. Für die Dimensionierung der Festkomma-Arithmetik wird eine Linkadaption angenommen, die für ein SNR $\leq 20\,\text{dB}$ das erforderliche BER von 10^{-2} garantiert. Die Kombinationen aus Teilnehmeranzahl und Modulationsstufe der durchgezogenen Kurven in Bild 5.2 maximieren dann die spektrale Effizienz bei den drei betrachteten Korrelationswerten. Sie stellen den *worst case* für die Dimensionierung dar.

Die Wortbreite W_{mem} des Arbeitsspeichers begrenzt die Dynamik aller Zwischenergebnisse und hat damit einen wesentlichen Einfluss auf die Festkomma-Arithmetik. Die internen Wortbreiten des Rechenwerkes sowie die festen Skalierungsfaktoren werden zunächst so gewählt, dass jeweils bis zur letzten Skalierungsstufe mit maximaler Auflösung gerechnet und Überläufe ausgeschlossen werden. Bei der Dimensionierung des Akkumulators muss die maximale Vektorlänge, hier gleich MQ, berücksichtigt

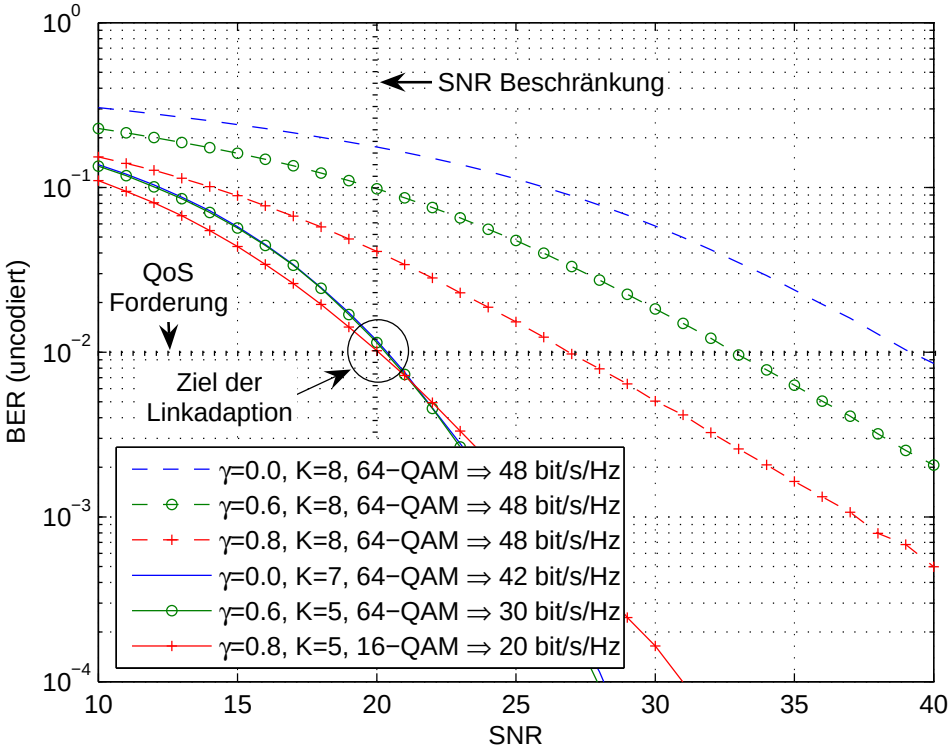


Bild 5.2: Uncodiertes Bitfehlerverhältnis über dem SNR für einen MMSE Empfänger mit $M = 8$ Antennen in Abhängigkeit der räumlichen Korrelation γ, gestrichelt für ein voll ausgelastetes System mit 64 QAM Modulation, durchgezogen für die jeweils maximale spektrale Effizienz unter der Forderung BER $\approx 10^{-2}$ für ein SNR $\leq 20\,\mathrm{dB}$ (Linkadaption).

werden. Die maximale Anzahl an zusätzlich benötigten Bitstellen ergibt sich zu $W_{mq} = \lceil \log_2(MQ) \rceil$. In Tabelle 5.1 sind die entsprechenden Wortbreiten und festen Skalierungsfaktoren unter der Spalte *Ideal* aufgelistet. Im Folgenden wird zunächst die erforderliche Speicherwortbreite spezifiziert. Anschließend werden Schritt für Schritt über die Spalten a) bis d) die internen Parameter des Rechenwerkes angepasst.

Die Kenntnis der Verteilungsdichtefunktion erlaubt eine Skalierung der Festkomma-Darstellung aller Zwischenergebnisse mit einer definierten Überlaufwahrscheinlichkeit. Die Notation der Zwischenergebnisse entspricht der aus Kapitel 3.2.2. Im Folgenden wird angenommen, dass die Matrizen $\mathbf{\Delta}$, $\mathbf{V}$ und $\mathbf{D}$ gleich skaliert sind, da sie eine vergleichbare Dynamik zeigen und die einheitliche Skalierung eine Einsparung an Zyklen für eine *in-place* Berechnung erlaubt. Die Beträge der Elemente von $\mathbf{V}$ und $\mathbf{D}$ sind außerdem durch das maximale Element der Matrix $\mathbf{\Delta}$ nach oben beschränkt und werden daher für

	Ideal	**a)**	**b)**	**c)**	**d)**	**Spec**
W_{mem}	$14\ldots32$	18				18
W_{add}	$2W_{mem}$	$20\ldots26$	24			24
S_{mul}	0	$-(2W_{mem}-W_{add}-1)$	-11			-11
S_{acc}	$-(W_{add}+1+W_{mq})$		$-6\ldots0$	-1		-1
W_{acc}	$W_{mem}-S_{acc}+W_{add}+1+W_{mq}$			$16\ldots22$	20	20
W_{div}	$2W_{mem}-2$				$18\ldots24$	23

Tabelle 5.1: Dimensionierung der Wortbreiten und festen Skalierungsfaktoren. Konfiguration *Ideal* garantiert maximale Auflösung ohne interne Überläufe. Grau hervorgehoben sind die jeweils in einer Konfiguration variierten Parameter (siehe dazu auch Bild 5.5). Unter *Spec* ist die endgültige Dimensionierung zu finden.

die Überlaufwahrscheinlichkeit nicht weiter berücksichtigt. Zur Vermeidung von trivialen Multiplikationen mit der zu eins normierten Hauptdiagonalen von $\mathbf{L}$ werden weiterhin $\mathbf{L}$ und $\mathbf{L}^{-1}$ sowie $\mathbf{D}^{-1}$ und $\mathbf{\Delta}^{-1}$ jeweils gleich skaliert.

In Bild 5.3 sind die kumulierten Verteilungsdichtefunktionen für das Kanalmodell aus Kapitel 2.3 exemplarisch für eine der *worst case* Konfigurationen dargestellt. Die Verteilungsdichte bezieht sich auf den maximalen Betrag über alle Matrix- bzw. Vektorelemente einer Realisierung. Die Darstellung der Abszisse im 2er-Logarithmus gibt für eine Festkomma-Arithmetik unmittelbar die Anzahl zusätzlich benötigter Bitstellen an. Diese im Folgenden als Dynamikreserve R bezeichnete Größe bezieht sich auf die Festkomma-Darstellung der Zahl '1'. Die Wahl des Exponenten zu

$$E = -\left((W_{mem}-1)-R\right) \tag{5.2}$$

führt dann zu einer Überlaufwahrscheinlichkeit der jeweiligen Matrix entsprechend der bei R angegebenen kumulierten Verteilungsdichte. In Tabelle 5.2 sind für alle drei Konfigurationen die Dynamikreserven mit einer Überlaufwahrscheinlichkeit kleiner 10^{-4} aufgelistet. Die relativ geringen Unterschiede zwischen den Konfigurationen rechtfertigen eine einheitliche Dimensionierung nach dem *worst case* für $\gamma = 0.8$. Durch Einsetzen der resultierenden Exponenten in Gleichung (5.1) folgen eindeutig die zur Berechnung benötigten variablen Skalierungsfaktoren S_{cmul} und S_{div}.

Mit der gewählten Wahrscheinlichkeit von 10^{-4} ist der Einfluss der Überläufe auf ein Bitfehlerverhältnis von 10^{-2} vernachlässigbar. Der Einfluss der Quantisierungsfehler ist in Bild 5.4 zu erkennen. Hier ist das uncodierte BER für eine Festkomma-Arithmetik in Abhängigkeit der Speicherwortbreite aufgetragen und wird mit einer 64 bit-Fließkomma-Arithmetik verglichen. Mit $W_{mem} \geq 18$ bit zeigt

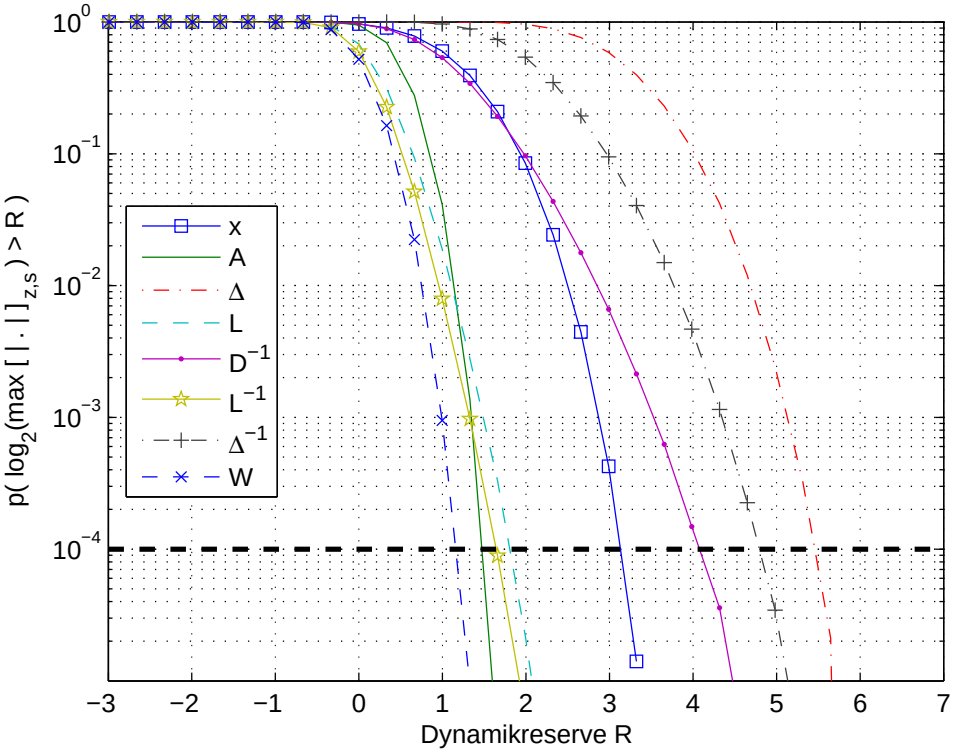


Bild 5.3: Verteilungsdichtefunktionen der Zwischenergebnisse der Matrixinversion für das jeweils größte Element der Matrizen am Beispiel der für $\gamma = 0.8$ gewählten Konfiguration der Linkadaption aus Bild 5.2 bei einem SNR $= 20\,\mathrm{dB}$. Numerisch ermittelt mit dem korrelierten Rayleigh-Kanalmodell aus Kapitel 2.3.

	Dynamikreserve R		
	$\gamma = 0.0$	$\gamma = 0.6$	$\gamma = 0.8$
$\mathbf{x}$	3	3	4
$\mathbf{A}$	2	2	2
$\mathbf{\Delta}, \mathbf{V}, \mathbf{D}$	5	5	6
$\mathbf{L}, \mathbf{L}^{-1}$	2	2	2
$\mathbf{D}^{-1}, \mathbf{\Delta}^{-1}$	4	4	5
$\mathbf{W}$	1	1	2

Tabelle 5.2: Dynamikreserve R der einzelnen Berechnungsschritte für eine Überlaufwahrscheinlichkeit von 10^{-4}. Der Korrelationsfaktor γ bezeichnet eine der drei Konfigurationen aus der Linkadaption in Bild 5.2. Die Dynamikreserve führt mit Gleichung 5.2 auf den jeweiligen Exponenten der Festkomma-Arithmetik.

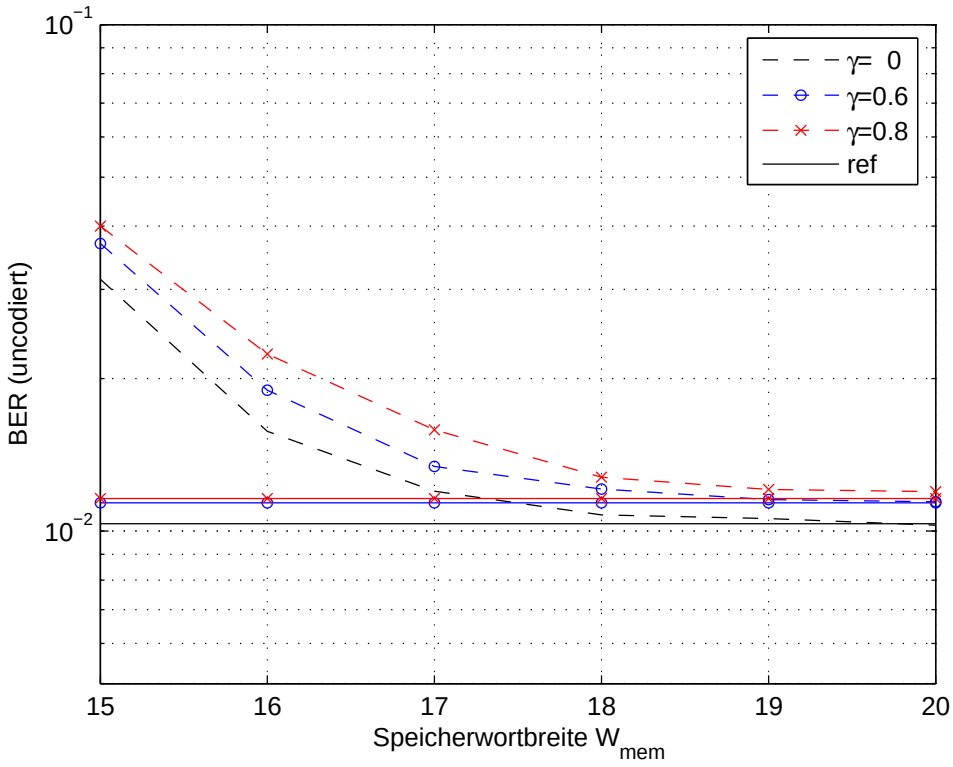
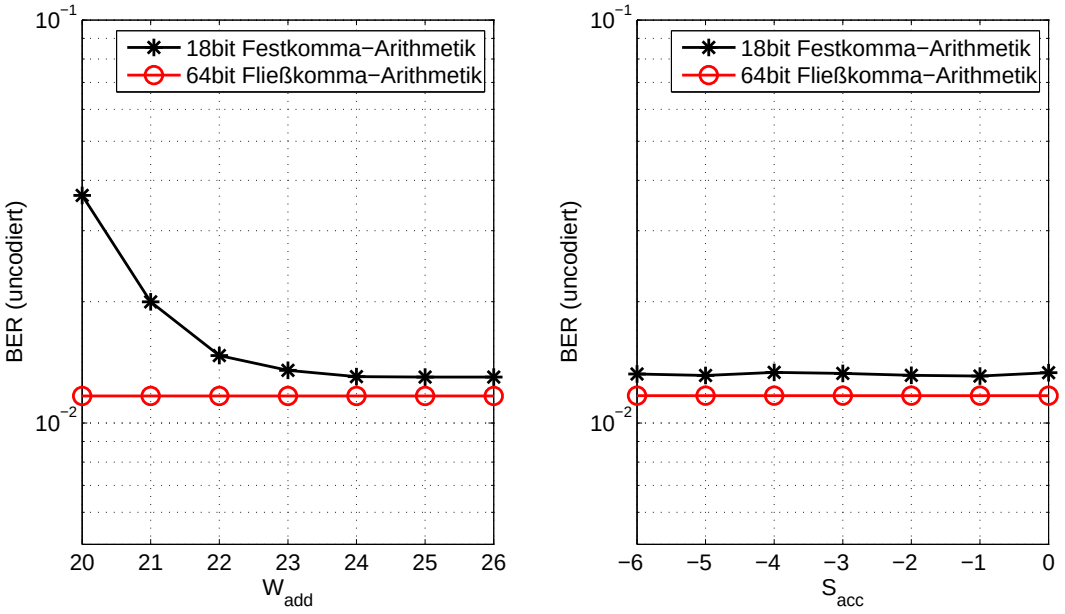


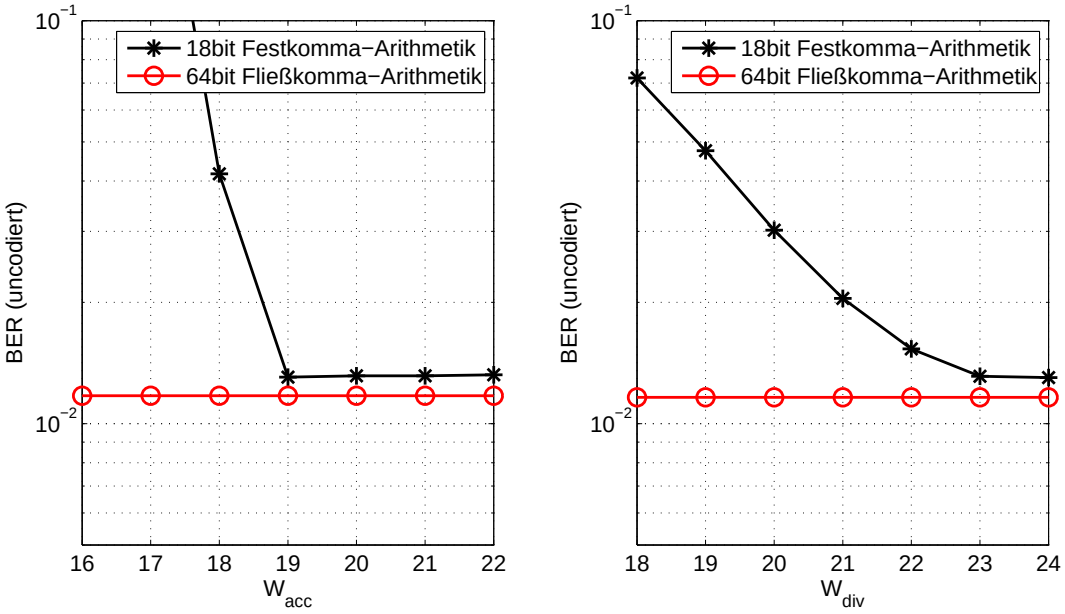
Bild 5.4: Uncodiertes Bitfehlerverhältnis für eine MMSE-Entzerrung mit Festkomma-Arithmetik in Abhängigkeit der Speicherwortbreite W_{mem} für die in Bild 5.2 definierten Konfigurationen der Linkadaption mit $\gamma = 0 \ldots 0.8$ mit 20 dB SNR. Die Skalierung ist entsprechend Tabelle 5.2 einheitlich für den *worst case* $\gamma = 0.8$ ausgelegt. Die durchgezogenen Kurven (ref) zeigen die Referenz einer 64 bit-Fließkomma-Arithmetik.

die Festkomma-Arithmetik in allen drei Konfigurationen eine ausreichend geringe Abweichung zur Fließkomma-Referenz.

Auf dieser Grundlage werden in Bild 5.5 nun die internen Parameter des Rechenwerkes bestimmt. Für eine Speicherwortbreite von $W_{mem} = 18$ werden schrittweise die Parameter a) W_{add}, b) S_{acc}, c) W_{acc} und d) W_{div} so gewählt, dass ohne merkliche Zunahme des BER der Ressourcenverbrauch minimiert wird. Die jeweilige Konfiguration der Parameter ist der entsprechenden Spalte aus Tabelle 5.1 zu entnehmen. Der feste Skalierungsfaktor S_{mul} wird abhängig von W_{add} so gewählt, dass nach der Multiplikation jeweils das obere Bit der Produkte abgeschnitten wird. Die Anzahl der relevanten Bitstellen für die Addition ist dann abhängig von dem jeweiligen Skalierungsfaktor S_{cmul}. Bild 5.5(a) zeigt eine Sättigung des BER ab einer Wortbreite von $W_{add} = 24$ bit. Der Parameter S_{acc} vergrößert die Auflösung des Akkumulators. Nach



(a) $\Rightarrow W_{add} = 24$ (b) $\Rightarrow S_{sacc} = -1$



(c) $\Rightarrow W_{acc} = 20$ (d) $\Rightarrow W_{div} = 23$

Bild 5.5: Dimensionierung der internen Wortbreiten des Rechenwerkes mit dem Ziel eines vernachlässigbaren Anstiegs des BER im Vergleich zu einer 64 bit-Fließkomma-Arithmetik für den *worst case* der Linkadaption aus Bild 5.2 mit $\gamma = 0.8$. Die Skalierung der Zwischenergebnisse erfolgt nach Tabelle 5.2. Die Konfiguration des Rechenwerkes ist jeweils der entsprechenden Spalte in Tabelle 5.1 zu entnehmen.

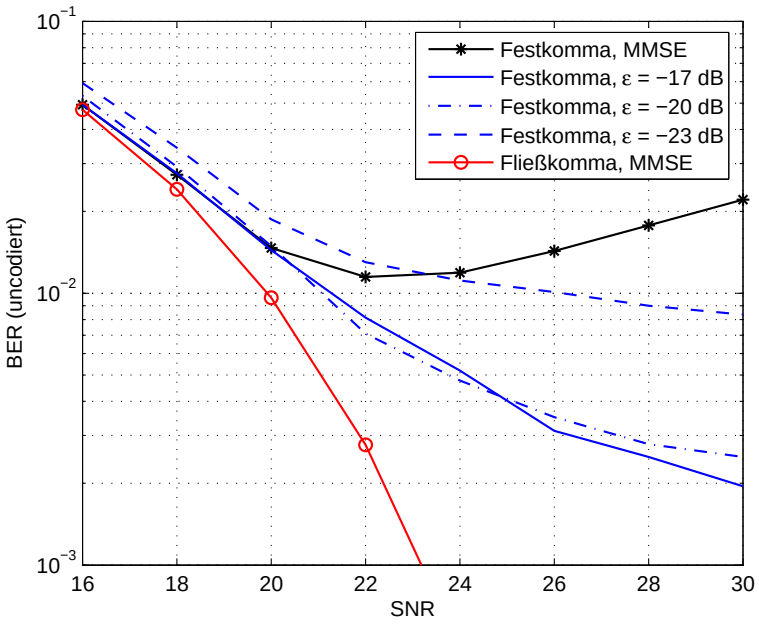
Bild 5.5(b) ist hier jedoch kein signifikanter Unterschied erkennbar. Mit $S_{acc} = -1$ wird dennoch ein Bit zum Runden des Akkumulators vorgehalten. Der Akkumulator erfordert dann gemäß Bild 5.5(c) eine Wortbreite von 19 bit. Um auch nach oben ein Bit Reserve für größere Zwischenergebissse in der Akkumulation vorzuhalten, wird im Folgenden eine Wortbreite von $W_{acc} = 20$ bit gewählt. In Bild 5.5(d) ergibt sich schließlich für den Dividierer eine Wortbreite $W_{div} = 23$. Der Skalierungsfaktor S_{div} für die Berechnung von D^{-1} ist dann gerade gleich Null, d.h. die unteren 18 bit des 25 bit breiten Dividiererausgangs werden direkt in den Speicher übernommen. Die gewählten Wortbreiten und Skalierungsfaktoren sind in Tabelle 5.1 unter der Spalte *Spec* zusammengefasst.

Unabhängig von der Linkadaption kann ein SNR > 20 dB nicht ausgeschlossen werden. Selbst bei unveränderter Parametrierung erhöht sich mit zunehmendem SNR die Überlaufwahrscheinlichkeit. Dies führt besonders bei der Einträgermodulation zu einem unerwünschten Einbruch des BER, da sich Überläufe auf einzelnen Unterträgergruppen hier auf alle übertragenen Datensymbole auswirken. Zur Vermeidung dieses Effektes kann das ϵ in Gleichung (3.11) ab einer gewissen Schwelle konstant gehalten werden. Bild 5.6 zeigt die Auswirkung auf das BER bei Ein- und Mehrträgermodulation an einem Beispiel. Mit einem konstanten $\epsilon = 20$ dB kann für beide Modulationsverfahren ein stabiler Betrieb unterhalb der geforderten BER-Schwelle garantiert werden, jedoch mit einer starken Abweichung zur Fließkomma-Referenz.

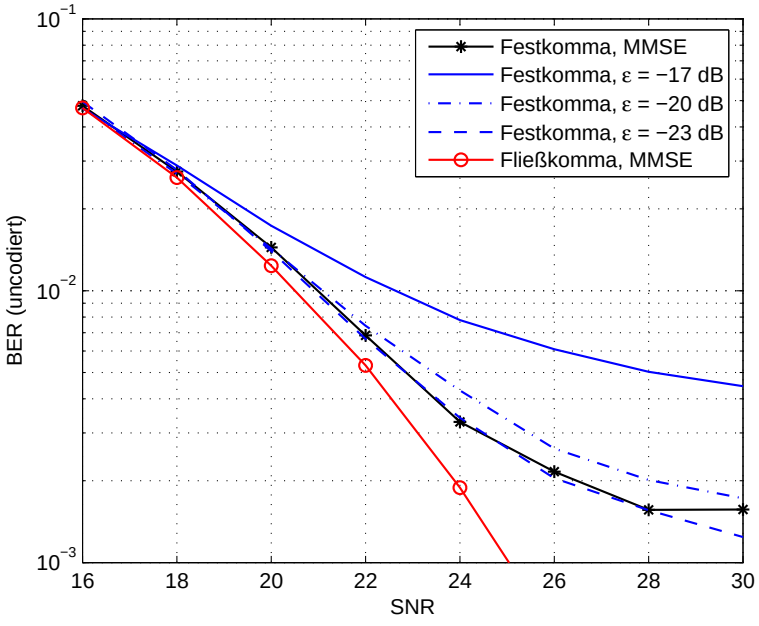
Die getroffenen Voraussetzungen zur Dimensionierung der Wortbreiten sind zusammengefasst:

- MMSE-Entzerrung mit acht Empfangsantennen über die explizite Berechnung der speziellen Wiener-Lösung nach (3.11)
- Rayleigh-verteilte Kanalkoeffizienten mit einer räumlichen Korrelation $\gamma \leq 0.8$ zwischen benachbarten Antennen an Zugangspunkt und Teilnehmerstationen
- Normierung der mittleren Leistung der Systemmatrixelemente auf den Wert 1 unter Berücksichtigung der in Tabelle 5.2 spezifizierten Dynamikreserve
- Linkadaption zur Maximierung der spektralen Effizienz für ein uncodiertes BER $\approx 10^{-2}$ mit einem SNR ≤ 20 dB

Die Dimensionierung wurde anhand von *worst case* Konfigurationen durchgeführt. Bei weniger kritischen Konfigurationen wird die Fließkomma-Referenz ebenfalls erreicht. Erst für sehr kleine BER kann sich die Restfehlerwahrscheinlichkeit der Überläufe bemerkbar machen. Innerhalb des spezifizierten SNR-Bereiches ist daher in der Regel keine Anpassung der variablen Skalierungsfaktoren erforderlich. Für den praktischen Einsatz ist eine Beschränkung auf weniger als acht Empfangsantennen



(a) Einträgermodulation



(b) Mehrträgermodulation

Bild 5.6: Frequenzbereichsentzerrung für Ein- und Mehrträgermodulation mit $N = 256$ Unterträgern, einem frequenzselektiven Kanal mit $\alpha = 3\,\mathrm{dB}$ und den Einstellungen der Linkadaption für $\gamma = 0.8$. Die Festkomma-Arithmetik ist nach Spalte *Spec* in Tabelle 5.1 dimensioniert. Festhalten des Parameters ϵ in der Matrixinversion verbessert für die Festkomma-Arithmetik das BER im hohen SNR-Bereich.

denkbar. Die Reduktion der Matrixdimension ändert jedoch an der Dimensionierung der Festkomma-Arithmetik wenig. Die Verteilungsdichten für ein System mit vier Empfangsantennen zeigen lediglich bei der Δ-Matrix eine leicht reduzierte Dynamik. Die Anforderungen an die Speicherwortbreite bleiben unverändert. Die Erweiterung des zulässigen SNR Bereiches auf 32 dB erfordert dagegen bereits ein $W_{mem} = 25$ bit für eine *worst case* Konfiguration mit $\gamma = 0.8$, $M = 8$, $K = 7$ und einer 64-QAM-Modulation. Bei einer hohen räumlichen Korrelation kann also mit einer um 7 bit breiteren Wortbreite und 12 dB mehr Sendeleistung die spektrale Effizienz von 20 auf 42 $^{\text{bit}}/_{\text{s Hz}}$ erhöht werden.

5.1.3 Anforderungen für den RLS-Algorithmus

Neben der Matrixinversion sind auch die iterativen Verfahren zur Adaption an einen zeitlich veränderlichen Kanal eine denkbare Anwendung für den Parallelprozessor. Der LMS-Algorithmus eignet sich aufgrund seiner Anfälligkeit gegen eine Korrelation der Eingangssignale hauptsächlich zur Kanalschätzung. Eine Implementierung in Festkomma-Arithmetik ist relativ unkritisch. Nach einzelnen Überläufen beim Aktualisieren der Koeffizienten konvergiert der Algorithmus wieder zum Optimum.

Zum direkten Nachführen der Gewichtsmatrix verspricht der RLS-Algorithmus durch die Dekorrelation der Eingangssignale eine schnellere Konvergenz als der LMS. Die Berechnung der Dekorrelationsmatrix $\mathbf{P}$ erfolgt jedoch in einer offenen Regelschleife ohne Rückführung des Fehlers. Quantisierungsfehler und Überläufe können die Struktur dieser Matrix soweit stören, dass der Algorithmus nicht mehr konvergiert. Die Implementierung in Festkomma-Arithmetik ist daher äußerst kritisch.

Zur Abschätzung der Anforderungen an die Wortbreite werden wie im vorhergehenden Kapitel zunächst geeignete Übertragungsparameter definiert, nun für einen zeitlich veränderlichen Kanal. In Bild 5.7 ist für eine RLS-Entzerrung das uncodierte Bitfehlerverhältnis nach $T = 100$ Iterationen mit verschiedenen Vergessensfaktoren über der maximalen Dopplerfrequenz aufgetragen. Um mit zunehmender Dopplerspreizung ein gleichbleibendes BER $\approx 10^{-2}$ zu gewährleisten, muss über eine Linkadaption die Robustheit der Übertragung erhöht werden. In der betrachteten Konfiguration ($M = 8$ Antennen, flacher Kanal mit Korrelationsfaktor $\gamma = 0.6$ und SNR $= 20$ dB) kann durch eine Adaption der Teilnehmeranzahl mit einer gleichbleibenden Modulationsstufe von 4 bit pro Symbol die höchste spektrale Effizienz erreicht werden.

Mit einem langen Gedächtnis ($\beta \approx 1$) kann der RLS einer großen Dopplerspreizung nicht mehr folgen. Ein kurzes Gedächtnis liefert dagegen in einer relativ statischen

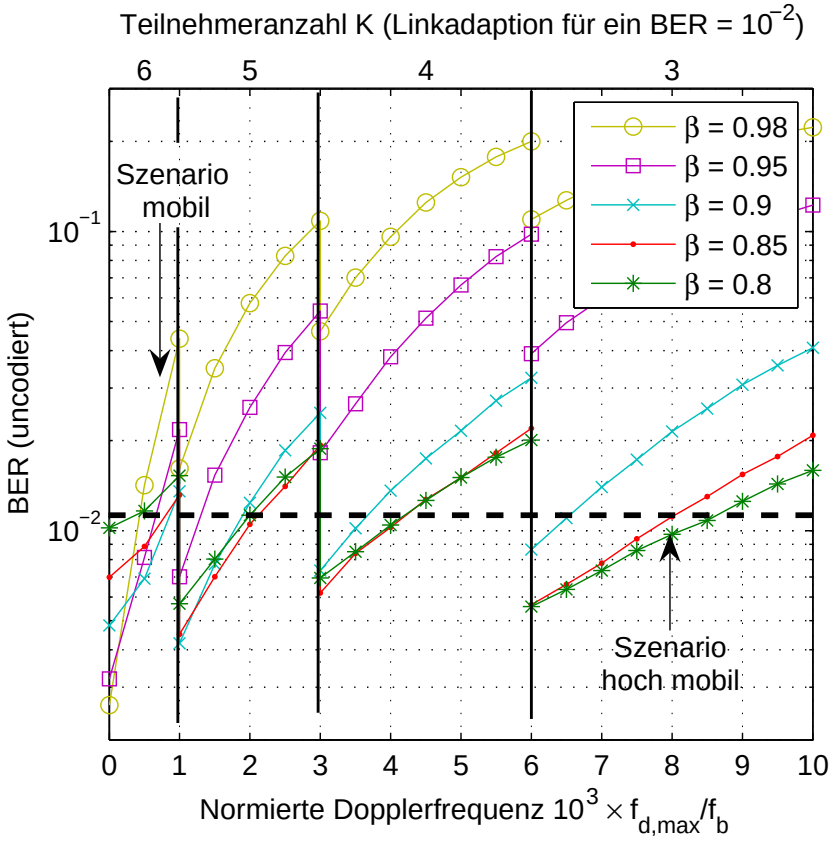
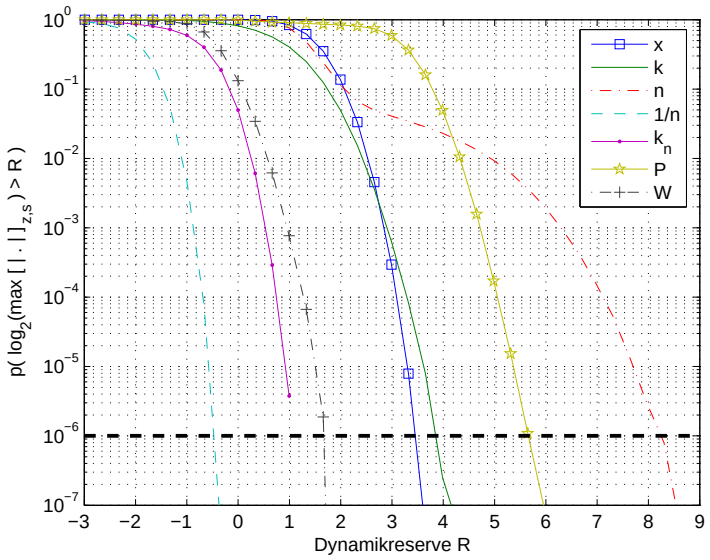


Bild 5.7: Uncodiertes Bitfehlerverhältnis in Abhängigkeit der auf die Blockrate normierten Dopplerfrequenz und des Vergessensfaktors β für eine RLS-Entzerrung mit 16 QAM Modulation, $M = 8$ Empfangsantennen über einen flachen Kanal mit Korrelationsfaktor $\gamma = 0.6$ und SNR $= 20\,\mathrm{dB}$. Eine Linkadaption wählt die Anzahl der Teilnehmer K für ein BER $\approx 10^{-2}$.

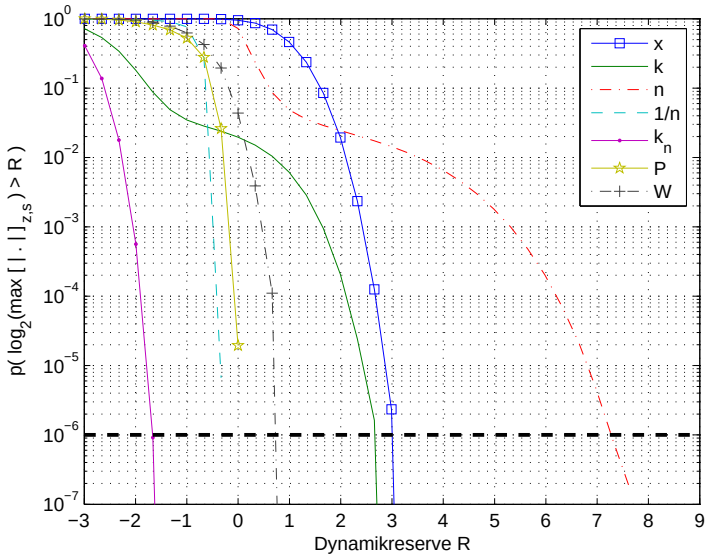
Umgebung nicht die nötige Präzision für eine hohe spektrale Effizienz. In der Literatur sind verschiedene Ansätze für eine Adaption des Vergessensfaktors zu finden (z.B. in [62]). Ein einheitlicher Vergessensfaktor von $\beta = 0.85$ bietet hier bereits einen guten Kompromiss über alle Dopplerfrequenzen, nur für eine geringe Mobilität mit $f_{d,max} < 0.001 f_b$ kann ein größerer Vergessensfaktor eine wesentliche Verbesserung erbringen.

Der Einfluss der Festkomma-Arithmetik wird nun für zwei beispielhafte Szenarien aus der Linkadaption untersucht:

- *mobil* mit $K = 6$ bei $f_{d,max} = 0.0008 f_b$ (entspricht einer Geschwindigkeit $v = 17.3\,\mathrm{km/h}$ bei einer Trägerfrequenz $f_t = 5\,\mathrm{GHz}$ und einer Blockrate $f_b = 100\,\mathrm{kHz}$)
- *hoch mobil* mit $K = 3$ bei $f_{d,max} = 0.008 f_b$ ($\widehat{=}$ $v = 173\,\mathrm{km/h}$ bei $f_t = 5\,\mathrm{GHz}$ und $f_b = 100\,\mathrm{kHz}$)



(a) $\beta = 0.85$



(b) $\beta = 0.98$

Bild 5.8: Kumulierte Verteilungsdichtefunktion der Beträge aller Zwischenergebnisse der RLS Entzerrung für Szenario *mobil* aus Bild 5.7. Numerisch ermittelt mit dem korrelierten Rayleigh-Kanalmodell aus Kapitel 2.3. Der größere Vergessensfaktor weist eine wesentlich geringere Dynamik der Größen $\mathbf{k}_n$ und $\mathbf{P}$ auf.

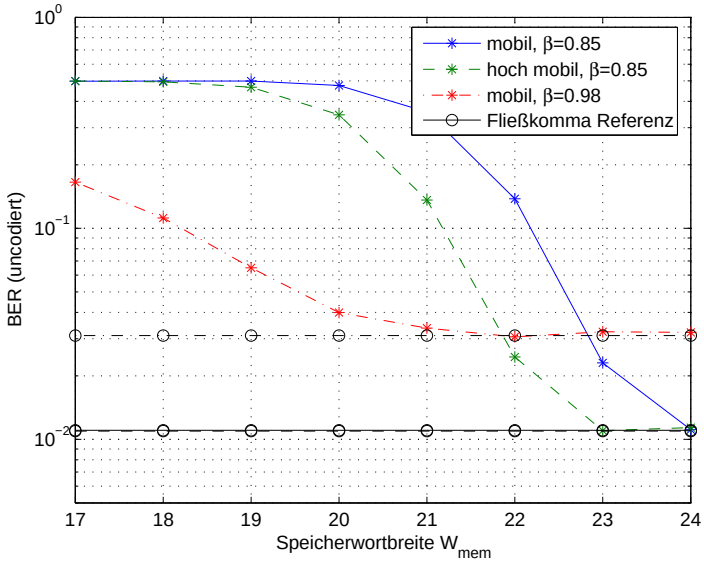
In Bild 5.8 sind für das Szenario *mobil* die kumulierten Verteilunsdichtefunktionen der maximalen Beträge aller Zwischenergebnisse jeweils für $\beta = 0.85$ und 0.98 aufgetragen. Im direkten Vergleich zeigt der größere Vergessensfaktor eine wesentlich geringere Dynamik der Größen $\mathbf{k}_n$ und $\mathbf{P}$. Das Szenario *hoch mobil* verfügt auf

	Dynamikreserve R	
	$\beta = 0.85$	$\beta = 0.98$
$\mathbf{x}$	+4	+4
$\mathbf{k}$	+4	+3
$\mathbf{n}$,β	+9	+8
$\mathbf{1/n}$	+0	+0
$\mathbf{k_n}$	+2	−1
$\mathbf{P}$	+6	+1
$\mathbf{W}$	+2	+1

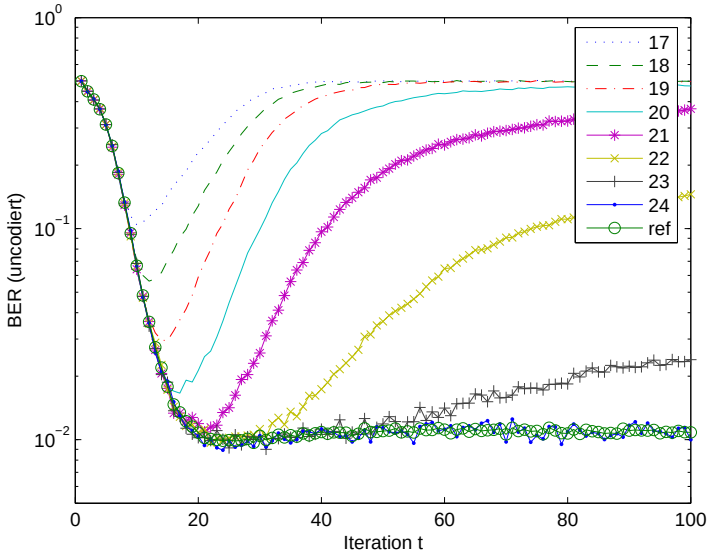
Tabelle 5.3: Dynamikreserve der einzelnen Berechnungsschritte für eine Überlaufwahrscheinlichkeit von 10^{-6} im Szenario *mobil* mit den Verteilungen für $\beta = 0.85$ und 0.98 aus Bild 5.8. Die Dynamikreserve führt mit Gleichung 5.2 auf den jeweiligen Exponenten der Festkomma-Arithmetik.

Grund der geringeren Teilnehmeranzahl über eine besser konditionierte Systemmatrix und zeigt damit eine leicht geringere Dynamik. Die jeweilige Dynamikreserve mit einer Überlaufwahrscheinlichkeit kleiner 10^{-6} ist für Szenario *mobil* in Tabelle 5.3 aufgelistet. Voraussetzung für die Anpassung an die simulierten Wahrscheinlichkeiten ist eine Normierung der Elemente des Empfangsvektors $\mathbf{x}$ auf eine mittlere Leistung von eins unter Berücksichtigung des über die Dynamikreserve spezifizierten Exponenten. Aus den geforderten Exponenten folgen dann direkt die zugehörigen Skalierungsfaktoren.

Die gewählte Skalierung schließt für die beiden gegebenen Konfigurationen Überläufe mit hoher Wahrscheinlichkeit aus. Der Einfluss der Quantisierungsfehler ist in Bild 5.9(a) in Abhängigkeit der Speicherwortbreite zu erkennen. Die internen Wortbreiten des Rechenwerkes sind entsprechend Spalte *Ideal* in Tabelle 5.1 gewählt. Für den einheitlichen Vergessensfaktor $\beta = 0.85$ wird die Referenz der Fließkomma-Arithmetik erst ab einer Wortbreite von 24 bit erreicht. Das Szenario *hoch mobil* zeigt aufgrund der besseren Kondition etwas geringere Anforderungen. Bei eingeschränkter Mobilität kann mit dem größeren Vergessensfaktor $\beta = 0.98$ die erforderliche Wortbreite um 3-4 bit gesenkt werden. Während dies im Szenario *mobil* bereits eine merkliche Verschlechterung der Bitfehlerrate zur Folge hat, ist der Verlust bei deutlich höheren Geschwindigkeiten nicht mehr tragbar. Die Betrachtung des BER über die Zeit in Bild 5.9(b) zeigt, dass bei zu geringer Wortbreite schon während der Initialisierung die Konvergenz verloren geht und die Bitfehlerrate gegen 0.5 strebt. Erst mit relativ großen Wortbreiten kann dies für einen ausreichend langen Zeitraum



(a) BER nach T=100 Iterationen in Abhängigkeit der Speicherwortbreite. Mit einem größeren Vergessensfaktor nehmen die Anforderungen ab, jedoch auf Kosten der Adaptionsfähigkeit.



(b) Entwicklung des BER über die Zeit in Abhängigkeit der Speicherwortbreite (*ref* steht für die Fließkomma Referenz), hier für das mobile Szenario mit $\beta = 0.85$.

Bild 5.9: Einfluss der Speicherwortbreite auf eine RLS Entzerrung mit Festkomma-Arithmetik für die beiden Szenarien aus Bild 5.7. Die Skalierung der Zwischenergebnisse erfolgt mit der entsprechenden Dynamikreserve aus Tabelle 5.3.

ausgeschlossen werden. Für die Matrixinversion in einer ähnlichen Konfiguration ist nach Bild 5.4 eine Wortbreite von 18 bit bereits ausreichend.

Wie bei der Matrixinversion hat das SNR auch hier einen entscheidenden Einfluss auf die Anforderungen. Für ein geringeres SNR von 10 dB kann für $\beta = 0.85$ z.B. die Dynamikreserve für $\mathbf{P}$ von 6 auf 3 und für $\mathbf{k_n}$ von 2 auf 0 verkleinert werden. Mit einer QPSK-Modulation kann ein BER $\approx 10^{-2}$ dann schon mit einer Speicherwortbreite von 20 bit erreicht werden. Ein größeres SNR von 30 bis 40 dB erfordert dagegen eine Erhöhung der Dynamikreserve auf 9 bzw. 3. Eine stabiler Betrieb ist dann erst ab einer Wortbreite von 25 bit möglich.

Ähnlich wie bei der Matrixinversion kann die Stabilität im hohen SNR-Bereich für kleinere Wortbreiten nur über ein künstlich erzeugtes Rauschen garantiert werden. Bei einem SNR von 32 dB im mobilen Szenario mit $\beta = 0.85$ bricht das BER für eine Wortbreite von 24 bit auf 0.5 ein. Durch eine modifizierte Skalierung des Eingangsvektors mit einem Reservefaktor von 21 kann das Quantisierungsrauschen zum Aktualisieren der inversen Korrelationsmatrix stark angehoben werden. Das BER sinkt dann wieder unter 10^{-3}. Zur Schätzung des Symbolvektors muss jedoch die ursprügliche Skalierung verwendet werden.

Gelockert werden die Anforderungen durch eine Verkleinerung der Matixgröße. Mit $M = 4$ Empfangsantennen und $K = 3$ bzw. 2 Teilnehmern für das mobile bzw. hoch mobile Szenario mit ansonsten unveränderten Parametern verringert sich die für ein $\beta = 0.85$ erforderliche Speicherwortbreite auf 20 bit.

5.2 Vergleich zu einer Fließkomma-Arithmetik

Im Folgenden wird eine Fließkomma-Arithmetik definiert und im direkten Vergleich zur Festkomma-Arithmetik bewertet. Die Auslegung von Rechenwerk und Speichereinheit auf komplexe Zahlen ermöglicht ein effizientes Zahlenformat mit einem gemeinsamen Exponenten und jeweils einer Mantisse für Real- und Imaginärteil. Sowohl der Exponent als auch die Mantissen seien im Zweierkomplemet dargestellt mit dem jeweiligen MSB als Vorzeichenbit. Eine Zahlendarstellung in Vorzeichen und Betrag, entsprechend dem IEEE-Standard, führt zu geringen Unterschieden in der Implementierung. Auf die im Folgenden abgeschätzten Wortbreiten sowie den Vergleich zur Festkomma-Arithmetik hat die Zahlendarstellung keine Auswirkung.

Die Wortbreite des Exponenten sei W_{exp}, die einer Mantisse W_{man}. Das Komma der Mantissen liegt um eine Bitstelle rechts vom MSB. Die Mantissen seien immer so normiert, dass der zugehörige Exponent minimiert wird. Positive Zahlen führen vor dem Komma dann immer eine '1', negative Zahlen eine '0'. Die komplexe Null

bildet eine Ausnahme und führe den kleinst möglichen Exponenten von $-2^{W_{exp}-1}$. Da die Vorkommastelle bei gegebenem Vorzeichen bekannt ist, braucht sie nicht mit im Speicher abgelegt zu werden. Die für eine komplexe Zahl benötigte Speicherwortbreite beträgt damit $W_{exp}+2(W_{man}-1)$ und muss mit der doppelten Speicherwortbreite $2W_{mem}$ für die Quadraturkomponenten der Festkomma-Arithmetik verglichen werden.

In Bild 5.10 ist das Fließkomma-Modell für das Rechenwerk aus Kapitel 4.3.2 dargestellt. Im Gegensatz zur Festkomma-Arithmetik werden hier Überläufe der Mantissen vollständig ausgeschlossen. Eine Reduktion der Wortbreite erfolgt immer durch Abschneiden der hinteren Bitstellen. Bei jeder Addition oder Negation wird dagegen die Wortbreite vorne um ein Bit erhöht. Die Rückkehr zum vorgeschriebenen Zahlenformat erfordert dann eine entsprechende Skalierung von Mantisse und Exponent.

Im komplexen Multiplizierer werden die Teilprodukte auf $W_{acc}-1$ MSBs reduziert und dann mit der Akkumulatorwortbreite W_{acc} aufaddiert. Nach der komplexen Multiplikation muss das Komma um drei Bitstellen nach vorne geschoben werden, die Summe der Exponenten der Koeffizienten A und B erhöht sich um drei. Die Mantisse des Koeffizienten C wird hinten mit Nullen aufgefüllt, der Exponent aufgrund der Wortbreitenerweiterung bei der Negation um eins erhöht.

Die komplexe Addition im Akkumulator erfordert eine Angleichung der Exponenten der beiden Summanden. Dazu wird die Mantisse des Summanden mit dem kleineren Exponenten um die Differenz der Exponenten nach rechts verschoben. Der andere Summand bleibt unverändert. Bei der Addition wird die Wortbreite dann erneut um ein Bit erhöht. Am Ausgang des Addierers erfolgt schließlich eine Normierung der Mantisse um den Exponenten wieder zu minimieren. Die Normierung prüft die Belegung der oberen Bitstellen in den Mantissen beider Quadraturkomponenten und gibt die kleinere Anzahl der ungenutzten Bitstellen für die Verschiebung der Mantissen im *barrel shifter* sowie die Korrektur des Exponenten weiter. Für die komplexe Null wird hier der kleinstmögliche Exponent gewählt.

Für die Akkumulation wird das Ergebnis auf den Eingang des Addierers zurückgeführt. Diese Schleife stellt den kritischen Pfad für die Gatterlaufzeit dar. Wird die Summe erst hinter der Normierung abgegriffen, durchläuft sie zwei *barrel shifter*, den Kreuzschalter sowie den Addierer. Da die Summe bereits im folgenden Taktzyklus wieder am Eingang vorliegen muss, können hier keine Register eingefügt werden. Um die erzielbare Taktrate nicht unnötig einzuschränken, wird die Normierung innerhalb der Schleife auf eine Bitstelle beschränkt und ist dann mit einem einfachen Multiplexer zu realisieren. Der kritische Pfad geht statt durch den zweiten

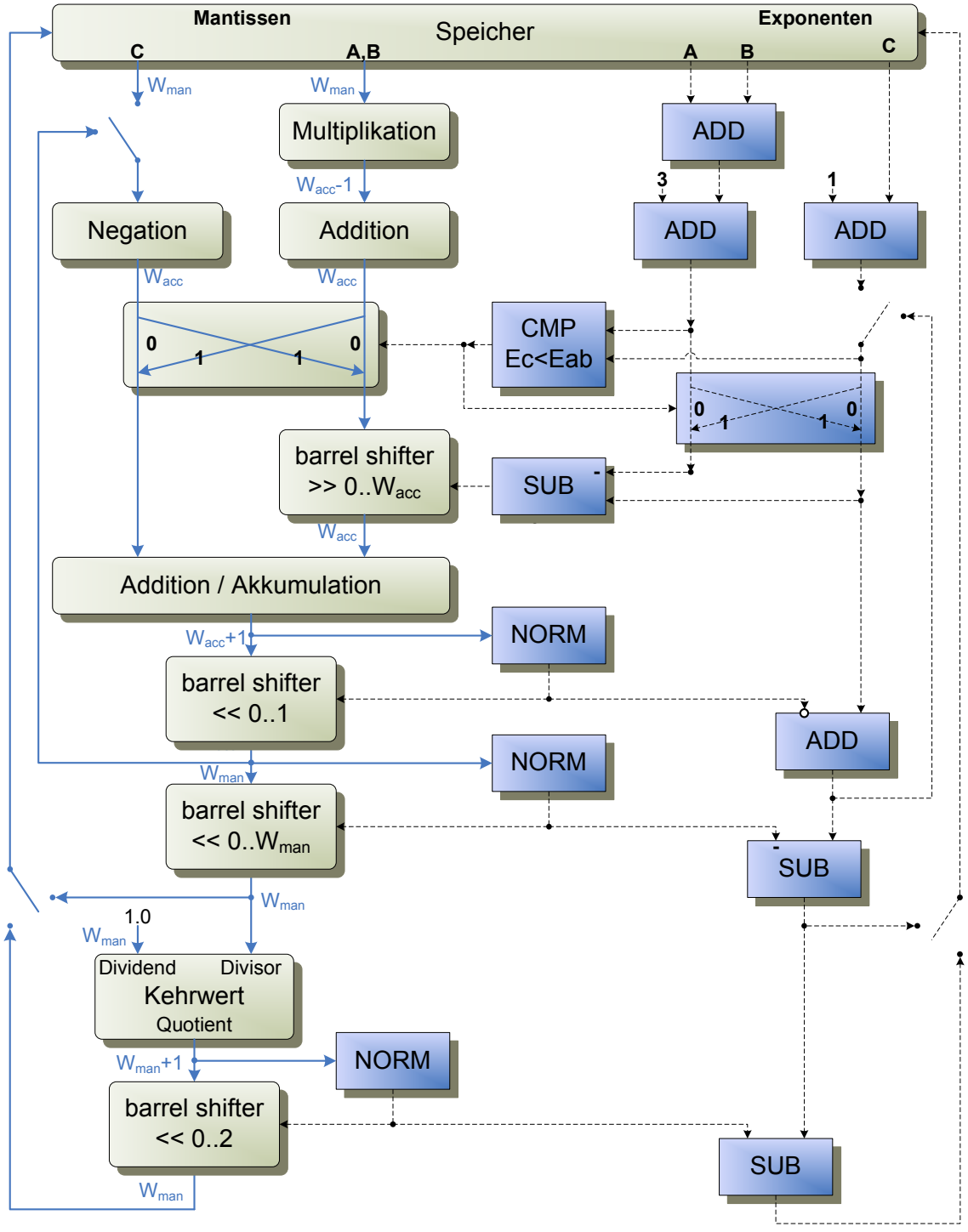


Bild 5.10: Modellierung der Fließkomma-Arithmetik für das Rechenwerk aus Kapitel 4.3.2. An den Signalwegen ist jeweils die Wortbreite der beiden Mantissen angegeben. Die Berechnung der Exponenten erfolgt mit der einheitlichen Wortbreite W_{exp}. Überläufe können nur im Exponenten, nicht in den Mantissen auftreten.

barrel shifter nur noch durch den Multiplexer und wird damit wesentlich verkürzt. Der Exponent kann während der Akkumulation dann nur noch vergrößert, nicht mehr verkleinert werden. Die Skalierung des Akkumulators richtet sich nach dem größten Exponenten der Eingangswerte bzw. der größten Zwischensumme. Kleine Zwischensummen werden entsprechend grob aufgelöst. Für die vorliegende Anwendung haben Simulationen gezeigt, dass diese Einschränkung keinen negativen Einfluss auf das BER zeigt.

Der Dividierer wird hinter die abschließende Normierung des Akkumulators geschaltet. Der Kehrwert wird nur für den Realteil berechnet, anders als bei der Festkomma-Arithmetik nun auch für negative Zahlen. Die unterschiedlichen Wertebereiche von positiven und negativen Zahlen im Zweierkomplement erfordern am Ausgang des Dividierers eine Normierung der Mantisse um bis zu zwei Bitstellen.

Um die Auswirkungen von Überläufen bei den Exponenten zu begrenzen, werden hier alle Additionen und Subtraktionen mit einer Sättigung durchgeführt. Eine gesonderte Behandlung bei Unterläufen, wie im IEEE-Standard vorgeschrieben, erfolgt nicht. Die interne Wortbreite für die Exponenten sei überall gleich W_{exp}.

Der zusätzliche Ressourcenverbrauch im Vergleich zur Festkomma-Arithmetik wird durch den zweiten *barrel shifter* zur Normierung am Ausgang des Akkumulators dominiert. Die Berechnung der Exponenten ist aufgrund der wesentlich kleineren Wortbreite unkritisch. Der *barrel shifter* innerhalb des Akkumulators ist bei der Festkomma-Artihmetik zwar ebenfalls erforderlich, dort jedoch außerhalb der Rückkoppelschleife. Gerade dieser Unterschied wird die erzielbare Taktrate für die Fließkomma-Arithmetik wesentlich einschränken. Der kritische Pfad ist mit dem *barrel shifter* und zwei zusätzlichen Multiplexern deutlich länger. Eine Latenzzeit von zwei oder mehr Zyklen bei der Addition, wie in Fließkomma-Signalprozessoren durchaus üblich, ist für die vorliegende Architektur im Zusammenhang mit den kleinen Vektorlängen der Anwendung nicht akzeptabel.

Ähnlich wie in Kapitel 5.1 wird zunächst die interne Akkumulatorwortbreite mit

$$W_{acc} = 2W_{man} + 1 \tag{5.3}$$

so gewählt, dass intern mit maximaler Auflösung gerechnet wird. Die Dynamik der in Kapitel 5.1 betrachteten Verteilungsdichtefunktionen kann mit einer Wortbreite von 4 bit für den Exponenten nicht vollständig abgedeckt werden. Zwei zusätzliche Bit erhöhen den Dynamikbereich bereits um über 280 dB und versprechen damit eine ausreichend hohe Reserve. Im Folgenden wird die Wortbreite daher zu $W_{exp} = 6$ bit gewählt.

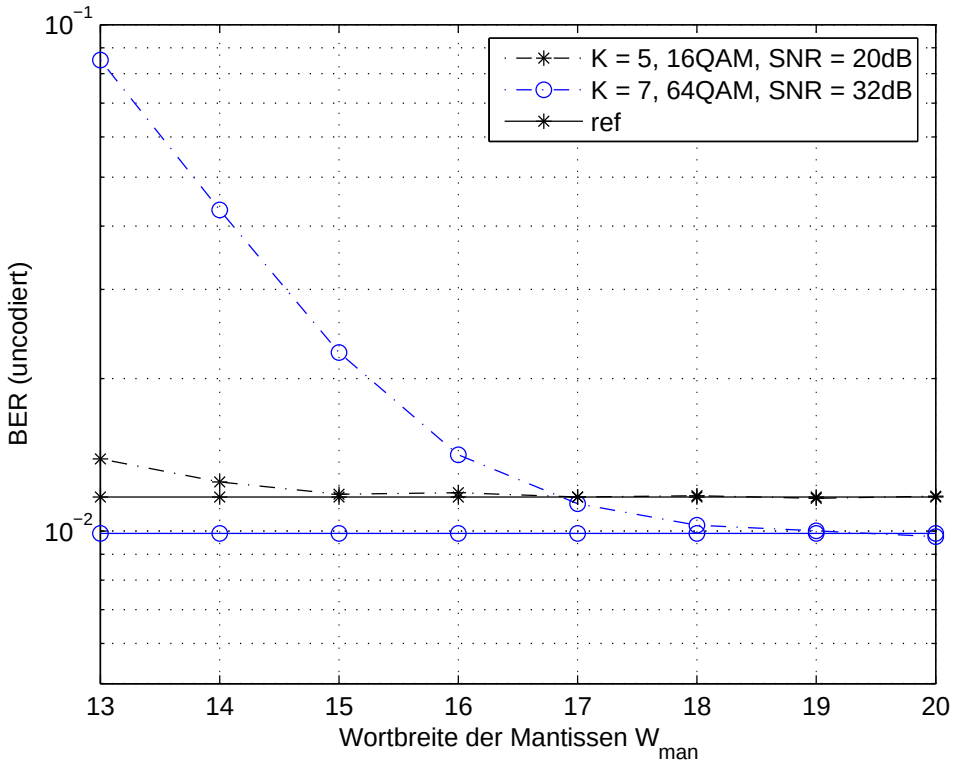


Bild 5.11: Uncodiertes Bitfehlerverhältnis für eine MMSE-Entzerrung mit Fließkomma-Arithmetik in Abhängigkeit der Wortbreite W_{man} der Mantissen für zwei *worst case* Konfigurationen aus Kapitel 5.1.2 mit $M = 8$ und $\gamma = 0.8$. Die durchgezogenen Kurven (ref) zeigen die Referenz einer 64 bit-Fließkomma-Arithmetik (IEEE *double precision*).

Bild 5.11 zeigt den Einfluss der Mantissen-Wortbreite für einen der *worst case* Konfigurationen aus Kapitel 5.1.2 sowie für die dort am Ende untersuchte Konfiguration im hohen SNR-Bereich. Als Referenz dient eine 64 bit-Fließkomma-Arithmetik nach dem IEEE-Standard (*double precision*). Für den *worst case* bei 20 dB SNR verschwindet der Fehler ab einer Wortbreite von 15 bit. Für den Speicher liegen damit die Anforderungen mit insgesamt 34 bit geringfügig unter denen der Festkomma-Arithmetik mit $2 \times 18 = 36$ bit. Im hohen SNR-Bereich ist der Unterschied größer: hier verschwindet der Fehler mit drei zusätzlichen Bitstellen pro Mantisse, gegenüber sieben Bitstellen bei der Festkomma-Arithmetik. Das Ausschließen von Überläufen innerhalb des durch W_{exp} beschränkten Dynamikbereiches, garantiert für die Fließkomma-Arithmetik zudem einen stabileren Betrieb. Ein Einbruch des BER wie in Bild 5.6 ist auch für einen MMSE-Empfänger im hohen SNR Bereich nicht zu befürchten.

Für den RLS-Algorithmus zeigt sich ein ähnliches Verhalten: für das in Kapi-

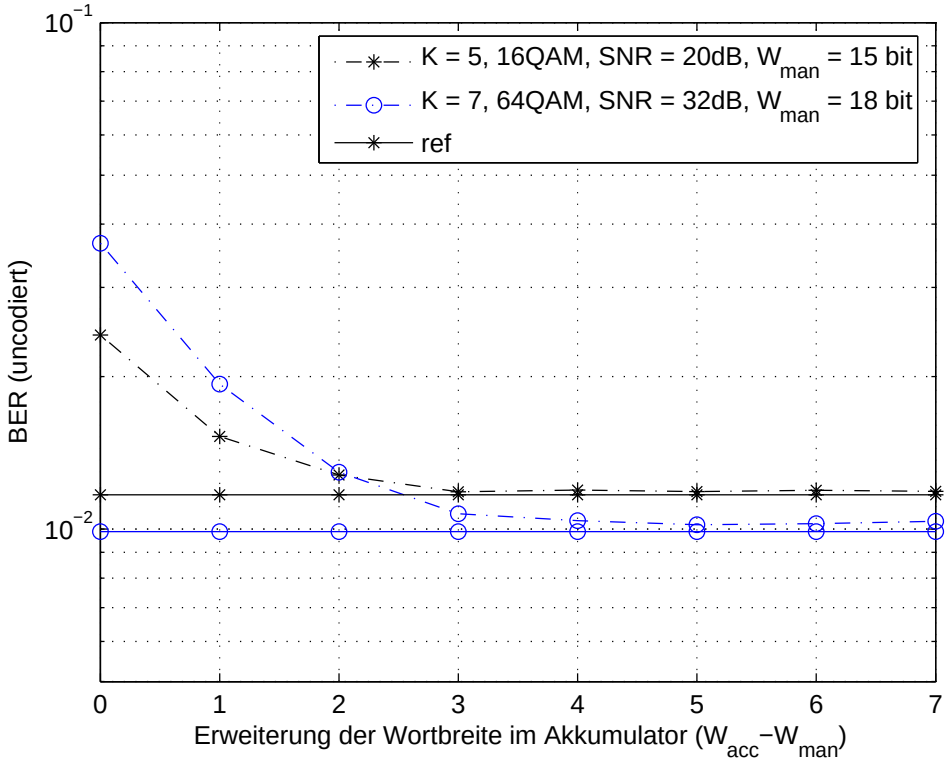


Bild 5.12: Uncodiertes Bitfehlerverhältnis für eine MMSE-Entzerrung mit Fließkomma-Arithmetik in Abhängigkeit der zusätzlichen Anzahl an Bitstellen für den Akkumulator im Vergleich zur Mantisse. Es gelten die gleichen Konfigurationen wie in Bild 5.11. Die durchgezogenen Kurven (ref) zeigen die Referenz einer 64 bit-Fließkomma-Arithmetik (IEEE *double precision*).

tel 5.1.3 definierte Szenario *mobil* mit einem Vergessensfaktor $\beta = 0.85$ erfordert die Fließkomma-Arithmetik eine Gesamtwortbreite von 44 bit gegenüber 48 bit der Festkomma-Arithmetik. Eine Erhöhung des SNR auf 32 dB bei gleicher Konfiguration führt zu einer deutlichen Abweichung von der 64 bit-Referenz. Anders als bei der Festkomma-Arithmetik bleibt das BER dagegen auch ohne zusätzliche Maßnahmen unterhalb der geforderten Schwelle von 10^{-2}.

In Bild 5.12 ist schließlich der Einfluss der Akkumulatorwortbreite auf das BER dargestellt, am Beispiel der Matrixinversion für die Konfigurationen aus Bild 5.11 mit einer Mantissenwortbreite von $W_{man} = 15$ bzw. 18 bit. In beiden Konfigurationen reichen etwa $3-4$ bit zusätzlich zur jeweiligen Mantissenwortbreite aus. Damit wird im Wesentlichen die Auflösung des komplexen Multiplizierers an das Niveau der Mantisse angepasst, denn die dort zum Ausschließen von Überläufen vorgehaltene Reserve von 3 bit wird nur in Ausnahmefällen ausgeschöpft.

5.3 Fazit

Die bitgenaue Simulation der Festkomma-Arithmetik nach dem Modell aus Kapitel 5.1.1 ermöglicht eine Dimensionierung der Wortbreiten anhand der Systemleistung, hier dem uncodierten BER. Die Anforderungen an die Wortbreiten hängen ab von den eingesetzten Algorithmen, den gewählten Übertragungsparametern, den zu erwartenden Kanalbedingungen und nicht zuletzt von der geforderten Übertragungsqualität. Hier wurde eine Dimensionierung am Beispiel der Matrixinversion durchgeführt, mit einer *worst case* Annahme von $M = 8$ Empfangsantennen, einem SNR $\leq 20\,\mathrm{dB}$ und einer räumlichen Korrelation $\gamma \leq 0.8$ zwischen benachbarten Antennen an Zugangspunkt und Teilnehmerstationen. Teilnehmeranzahl und Modulationsstufe werden durch eine Linkadaption so gewählt, dass mit maximaler spektraler Effizienz ein BER $\approx 10^{-2}$ erreicht wird. Innerhalb dieser Grenzen erreicht eine 18 bit-Festkomma-Arithmetik annähernd die Referenz einer 64 bit-Fließkomma-Arithmetik.

Die spektrale Effizienz kann für ein SNR $> 20\,\mathrm{dB}$ aufgrund der begrenzten Dynamik jedoch nicht weiter erhöht werden. Bei gleichbleibender spektraler Effizienz muss zudem im hohen SNR-Bereich eine Beschränkung der Matrixkondition erzwungen werden. Diese Abweichung vom MMSE-Optimum führt zu einer Sättigung des BER, allerdings unterhalb der geforderten Schwelle. Mit einer Speicherwortbreite von 25 bit pro Quadraturkomponente kann auch bei einer hohen räumlichen Korrelation ($\gamma = 0.8$) noch eine spektrale Effizienz von $42\,{}^{\mathrm{bit}}/_{\mathrm{s\,Hz}}$ erreicht werden (abzüglich der Verluste durch zyklische Erweiterung, Trainingssymbole und Coderedundanz). Dazu ist bereits ein SNR von 32 dB erforderlich. Noch größere Wortbreiten werden für die Inversion daher kaum von praktischer Bedeutung sein.

Die Anforderungen des RLS an die Wortbreite liegen deutlich höher. Unter ähnlichen Voraussetzungen wie bei der Matrixinversion (BER $\approx 10^{-2}$ mit SNR $\leq 20\,\mathrm{dB}$ und $\gamma = 0.6$) benötigt der RLS eine Wortbreite von 24 bit. Neben dem SNR zeigt hier der Vergessensfaktor einen großen Einfluss. Durch Einschränkung der Mobilität auf eine Dopplerfrequenz deutlich unter einem Tausendstel der Blockrate kann mit einem Vergessensfaktor von 0.98 die erforderliche Wortbreite auf 20 bit gesenkt werden. Mit dieser Dimensionierung ist auch hier die spektrale Effizienz auf die bei einem SNR $= 20\,\mathrm{dB}$ erreichbare Grenze beschränkt. Ähnlich wie bei der Matrixinversion kann im hohen SNR-Bereich durch künstliches erzeugtes Rauschen die Übertragungsqualität verbessert werden.

Entscheidend für beide Ansätze ist eine geeignete Normierung von Empfangsvektor und geschätzter Systemmatrix sowie die Skalierung der Zwischenergeb-

nisse. Die hier vorgeschlagenen Zahlenwerte basieren auf numerisch ermittelten Überlaufwahrscheinlichkeiten für das korrelierte Rayleigh-Kanalmodell. Dies kann im Einzelfall wesentlich von der Realität abweichen, so dass eine Anpassung der Skalierungsfaktoren an die jeweiligen Kanaleigenschaften erforderlich wäre. Unter der Voraussetzung einer vergleichbaren Verteilung der Matrixkondition sind große Abweichungen jedoch nicht zu erwarten. Ein Wechsel der Skalierungsfaktoren in Abhängigkeit der Übertragungsparameter (z.B. Teilnehmeranzahl oder Modulationsstufe) hat sich als unnötig erwiesen.

Zum Vergleich wurde anschließend eine Fließkomma-Arithmetik für das Rechenwerk spezifiziert und bewertet. Durch Ausnutzung der komplexwertigen MAC-Struktur des Rechenwerkes bleibt der Ressourcenverbrauch durchaus vergleichbar mit dem der Festkomma-Arithmetik: im Wesentlichen werden zusätzlich ein *barrel shifter* sowie einige Multiplexer und kleine Addierer benötigt. Die erforderliche Gesamtwortbreite für ein SNR $\leq 20\,\mathrm{dB}$ ist geringfügig kleiner als bei der Festkomma-Arithmetik. Für den hohen SNR-Bereich nimmt die Einsparung durch die Fließkomma-Arithmetik zu. Da Überläufe der Mantissen ausgeschlossen werden, ist grundsätzlich ein stabilerer Betrieb möglich. Das Hinzufügen von künstlichem Rauschen im hohen SNR-Bereich ist unnötig.

Größtes Manko der Fließkomma-Arithmetik ist eine deutliche Einschränkung der Taktrate im Vergleich zur Festkomma-Arithmetik. Der kritische Pfad läuft durch die Rückkoppelschleife des Akkumulators und verlängert sich bei der Fließkomma-Arithmetik um einen *barrel shifter* und zwei Multiplexer. Die Taktrate kann nur durch Einfügen von zusätzlichen Registern innerhalb der Schleife erhöht werden. Die Akkumulation erfolgt dann jedoch in getrennten Teilsummen. Die anschließende Addition dieser Teilsummen im Akkumulator erfordert zusätzliche Zyklen und verringert bei kurzen Vektorlängen deutlich die Effizienz.

Zusammengefasst lässt sich das Rechenwerk mit ähnlichem Ressourcenverbrauch in Fest- oder Fließkomma-Arithmetik realisieren. Die erforderliche Wortbreite hängt im Wesentlichen davon ab, in wie weit auch bei einer schlechten räumlichen Trennbarkeit noch ein stabiler Betrieb gewährleistet werden muss. In einem praktischen System sind diese Anforderungen immer durch das verfügbare SNR begrenzt. Die Fließkomma-Arithmetik vereinfacht die Programmierung und garantiert ein zuverlässigeres Verhalten im hohen SNR-Bereich, jedoch nur mit Einbußen in der Taktrate. Die in Kapitel 4 vorgestellte Prozessorarchitektur gilt unabhängig von der Zahlendarstellung in den Rechenwerken.

KAPITEL 6

Umsetzung in einem Demonstrator

Im Rahmen des vom BMBF geförderten Verbundprojektes HyEff wurde am Institut für Hochfrequenztechnik der RWTH Aachen ein echtzeitfähiger MIMO-Demonstrator entwickelt. Der Demonstrator war weltweit eines der ersten Systeme, das eine breitbandige MIMO-Übertragung über einen frequenzselektiven Kanal ermöglichte. Für die Raum-Zeit-Signalverarbeitung des Demonstrators wurde eine VLSI-Beschreibung der hier vorgeschlagenen Prozessorarchitektur erstellt und auf einer FPGA-Plattform umgesetzt.

In Kapitel 6.1 wird zunächst ein Überblick des Systems und der Spezifikationen gegeben. Es folgt eine detaillierte Beschreibung der eingesetzten digitalen Hardware in Kapitel 6.2. Anschließend wird in Kapitel 6.3 das VLSI-Design der digitalen Transceiver, der FFT-Prozessoren, des Parallelprozessors und der Ablaufsteuerung erläutert. Nach einer kurzen Beschreibung der Methodik zur Programmierung des Prozessors in Kapitel 6.4 wird in Kapitel 6.5 schließlich der Ressourcenverbrauch dargestellt und die Leistungsaufnahme abgeschätzt.

6.1 Systemüberblick

In den Jahren 2001-2003 förderte das Bundesministerium für Bildung und Forschung im Schwerpunktprogramm *Mobile Kommunikation* das Verbundprojekt *HyEff*. Losgelöst von den bestehenden Standards sollten neue Technologien zur Steigerung der Übertragungseffizienz in der drahtlosen Kommunikation erforscht werden. In einem Teilvorhaben (Förderkennzeichen 01BU154) wurde am Institut für Hochfrequenztechnik der RWTH-Aachen ein MIMO-Demonstrator mit der folgenden Zielsetzung aufgebaut:

- Praxisnahe Bewertung der Anforderungen und des Implementierungsaufwandes
- Entwicklung konkreter Realisierungskonzepte für die physikalische Schicht
- Demonstration der Machbarkeit unter realen Bedingungen

Der Demonstrator ist ausgelegt für die Anwendung in WLAN-Szenarien mit einer paketbasierten Datenübertragung zwischen einem Zugangspunkt und mehreren

mobilen Teilnehmern. Bei eher geringer Reichweite ($< 30\,\mathrm{m}$) und Mobilität ($< 50\,{}^{\mathrm{km}}/_{\mathrm{h}}$) wird eine hohe Übertragungsrate von mehreren hundert Mbit pro Sekunde mit einer hohen spektralen Effizienz gefordert.

Das umgesetzte Übertragungsverfahren ist wie folgt spezifiziert:

- Mehrnutzersystem mit Mehrfachzugriff über verschiedene Zeitschlitze, Spreizcodes sowie die räumliche Selektivität (TD/CD/SDMA)
- Bidirektionale Übertragung im Zeit-Duplex (TDD)
- Analogbandbreite von etwa 30 MHz für eine Chiprate von $f_{\mathrm{chip}} = 25\,{}^{\mathrm{Mchip}}/_{\mathrm{s}}$
- Universeller Einsatz von Einträger- und Mehrträgermodulation mit einer einheitlichen Blockstruktur:
  - Zyklisch erweiterte Datenblöcke der Länge $T_b = 11.84\,\mu\mathrm{s}$, bestehend aus jeweils $N = 256$ Chips und einer zyklischen Erweiterung von $L = 40$ Chips
  - Kontinuierliche Pulsformung durch eine vierfach überabgetastete Filterung mit einem Root-Raised-Cosine-Filter mit 32 Koeffizienten (d.h. $W_f = 8$ Chips)
  - Es folgt eine zulässige Kanallänge von $L - W_f = 32$ Chips oder $1.28\,\mu\mathrm{s}$
- Spreizung über Hadamard-Sequenzen der Länge $Q = 1, 2, 4$ oder 8 (jeweils einheitlich für alle Teilnehmer in einem Zeitschlitz)
- Unterstützte Modulationsarten: BPSK, QPSK, 8-PSK und 16-QAM
- Rahmenstruktur zunächst beliebig konfigurierbar

Die Realisierung beruht auf einer zentralen digitalen Signalverarbeitung für den Zugangspunkt und alle Teilnehmerstationen. Die Signalverarbeitung versorgt über Kabelverbindungen im Raum verteilte, aktive Sende/Empfangsmodule (*transmit/receive- oder TR-Modul*) mit angeschlossener Antenne. Der resultierende Aufbau für den geplanten Endausbau mit $M = 8$ Antennen am Zugangspunkt und $K = 8$ Antennen für die Teilnehmerstationen ist in Bild 6.1 dargestellt.

Zugangspunkt und Teilnehmerstationen nutzen identische TR-Module, die im Rahmen des Projektes am Institut entwickelt wurden und wie folgt spezifiziert sind:

- Trägerfrequenz im X-Band bei $f_t = 10.525\,\mathrm{GHz}$
- Bandbreite von etwa 30 MHz
- einstufige analoge Frequenzumsetzung in Sender und Empfänger mit einer Zwischenfrequenz von 175 MHz
- Sender mit 1 dB-Kompressionspunkt bei 25 dBm, Sendeverstärkung mit einer Dynamik von 50 dB einstellbar
- Empfänger mit einer Rauschzahl von etwa 1 dB, Empfangsverstärkung mit einer Dynamik von 30 dB einstellbar.

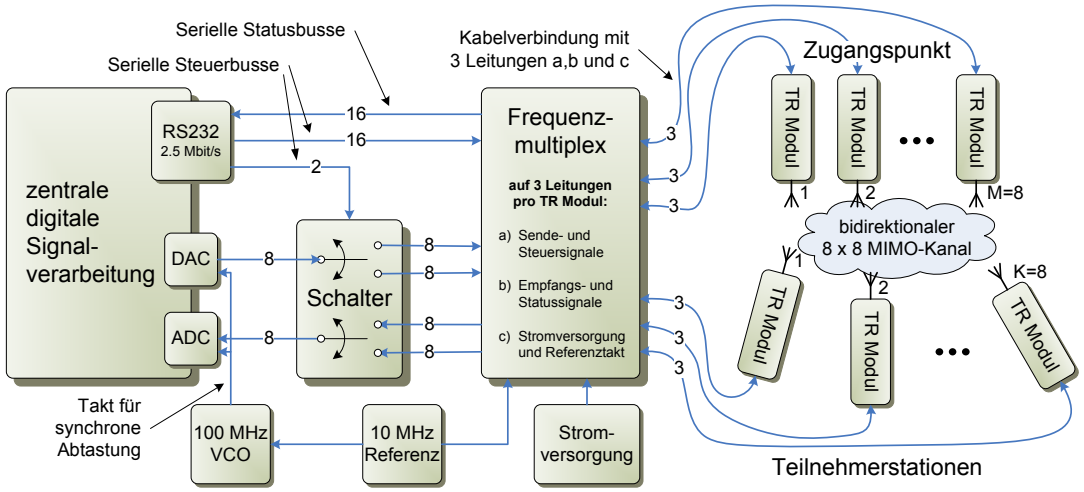
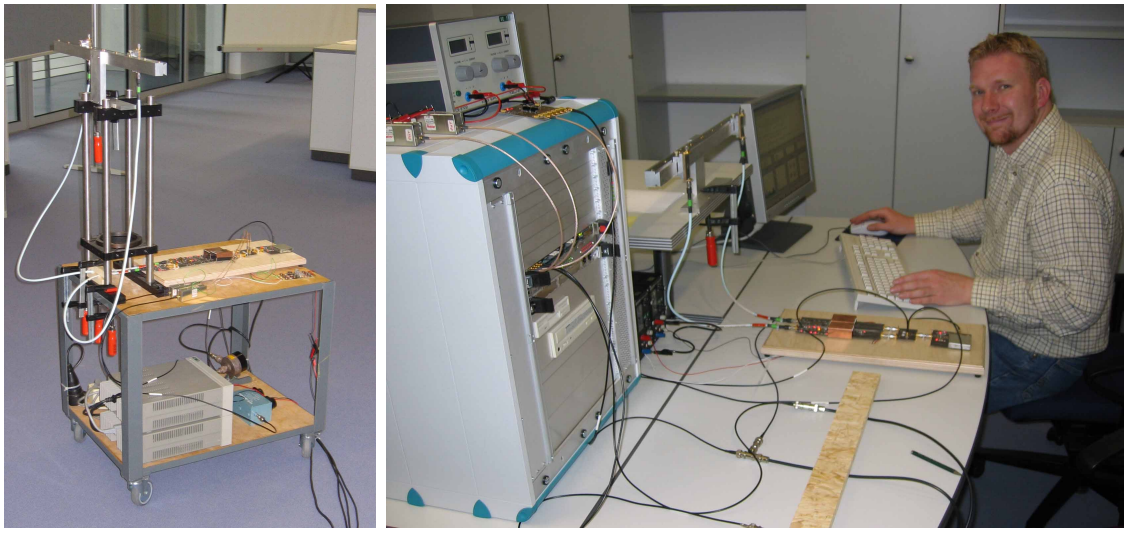


Bild 6.1: Systemaufbau des MIMO-Demonstrators mit einer zentralen Signalverarbeitung für Zugangspunkt und Teilnehmer. Alle TR-Module sind über Kabelverbindungen angebunden. Wechselschalter erlauben eine bidirektionale Übertragung im TDD-Betrieb.

- Integrierter Oszillator für die Erzeugung der Mischerfrequenzen von Sender und Empfänger, über eine PLL an eine externe 10 MHz-Referenz phasenstarr angebunden
- Sende/Empfangsumschalter für die Antenne mit einem zusätzlichen Tor zum Einspeisen und Auskoppeln von Kalibriersignalen
- Konfiguration des Betriebszustandes (Verstärkung in Sender und Empfänger, Wahl zwischen Sende-, Empfangs- oder Kalibrierbetrieb) sowie die Rückmeldung von Statusinformationen (aktuelle Sendeleistung) alle 10 μs über zwei unidirektionale, serielle Busse

Die zentrale Signalverarbeitung verfügt über eine analoge Schnittstelle mit acht DA-Wandlern und acht AD-Wandlern. Über einen Schalter kann jeder Wandler wahlweise mit einem Sender bzw. Empfänger des Zugangspunktes oder aber der Teilnehmerstationen verbunden werden. Durch eine wechselseitige Schalterstellung können beide Übertragungsrichtungen im Zeit-Duplex betrieben werden. Das Konzept der zentralen Signalverarbeitung reduziert so die Anzahl der benötigten Wandler um die Hälfte. Durch die abwechselnde Nutzung gemeinsamer Komponenten können auch Logikressourcen innerhalb der Signalverarbeitung eingespart werden. Die vorhandene Rechenleistung kann zudem flexibel zwischen Basisstation und Teilnehmerstationen aufgeteilt werden. Auf die Rahmen- und Abtastratensynchronisation kann verzichtet werden. Das Konzept erlaubt zudem ein ideales Feedback zwischen den Kommunikationsteilnehmern, z.B. für die Leistungsregelung oder den Austausch von



(a) Antennen und Sendezüge für die *mobile* Teilnehmerstation

(b) Antennen und Empfangszüge des Zugangspunktes mit der zentralen Signalverarbeitung im Industrie-PC links.

Bild 6.2: Eine erste Ausbaustufe des Demonstrators für eine Präsentation auf dem Statusseminar *Mobile Kommunikation* des BMBF in Frankfurt/Oder im Juli 2004. Gezeigt wurde eine unidirektionale 2×2-Übertragung im X-Band wahlweise mit einer Raum-Zeit-Entzerrung im Empfänger oder einer Raum-Zeit-Vorverzerrung mit idealem Feedback im Sender. Mit einer 16-QAM-Modulation wird eine Rohdatenrate von 200 $^{\text{Mbit}}/_{\text{s}}$ erreicht.

Kanalkoeffizienten. Schließlich wird die Auswertung von Messreihen vereinfacht, da Sende- und Empfangsdaten unmittelbar verglichen werden können.

Alle TR-Module werden an eine zentrale 10 MHz-Referenz angebunden. Damit ist das gesamte System phasenstarr gekoppelt, eine Frequenzsynchronisation wird überflüssig. Die Stromversorgung der TR-Module erfolgt ebenfalls zentral. Neben den Sende- und Empfangsdaten (jeweils bei der Zwischenfrequenz von 175 MHz) müssen also auch das 10 MHz-Referenzsignal, die Stromversorgung sowie die zwei seriellen Busse zur Konfiguration an alle TR-Module verteilt werden. Um die Anzahl der Kabel gering zu halten, werden jeweils zwei Signale unterschiedlicher Frequenz auf eine Leitung gemultiplext: Sende- und Steuersignale, Empfangs- und Statussignale sowie Stromversorgung und Referenzsignal. Die RS232-Schnittstellen für die Steuer- und Statusbusse sind in einem der FPGAs integriert. Mit einer Bitrate von 2.5 $^{\text{Mbit}}/_{\text{s}}$ kann während eines Datenblockes die Konfiguration für den jeweils nächsten übermittelt werden. Die Umschaltung zwischen Uplink- und Downlink-Betrieb erfolgt ebenfalls über zwei serielle Busse.

Zur Fertigstellung dieser Arbeit befindet sich der Demonstrator noch im Aufbau. Bild 6.2 zeigt Photos einer ersten Ausbaustufe für die unidirektionale Übertragung mit zwei Sende- und zwei Empfangsantennen. Für die Signalverarbeitung kommt eine Vorstufe der in dieser Arbeit entwickelten Prozessorarchitektur zum Einsatz. Mit einer 16-QAM-Modulation wird bereits eine Rohdatenrate von 200 $^{\text{Mbit}}/_{\text{s}}$ bei einer spektralen Effizienz von 8 $^{\text{bit}}/_{\text{s Hz}}$ erreicht. Mit einem idealen Feedback der Kanalschätzung innerhalb der zentralen Signalverarbeitung konnte erstmals auch die Raum-Zeit-Vorverzerrung demonstriert werden.

6.2 Digitale Hardware

Die zentrale Signalverarbeitung wurde mit einem kommerziellen FPGA-System der Firma Nallatech realisiert. Zwei cPCI-Trägerkarten (BenERA) befinden sich zusammen mit einer Slot-CPU-Karte (Host) in einem 19”-Rack mit 6 Höheneinheiten. Jeder Träger verfügt über vier Steckplätze für den firmenspezifischen DIME-II Formfaktor. Das System umfasst insgesamt 6 DIME-II Module: vier identische Wandlermodule (BenADDA) und zwei unterschiedlich konfigurierte Prozessormodule (BenBLUE-II). Bild 6.3 zeigt Photos der einzelnen Komponenten.

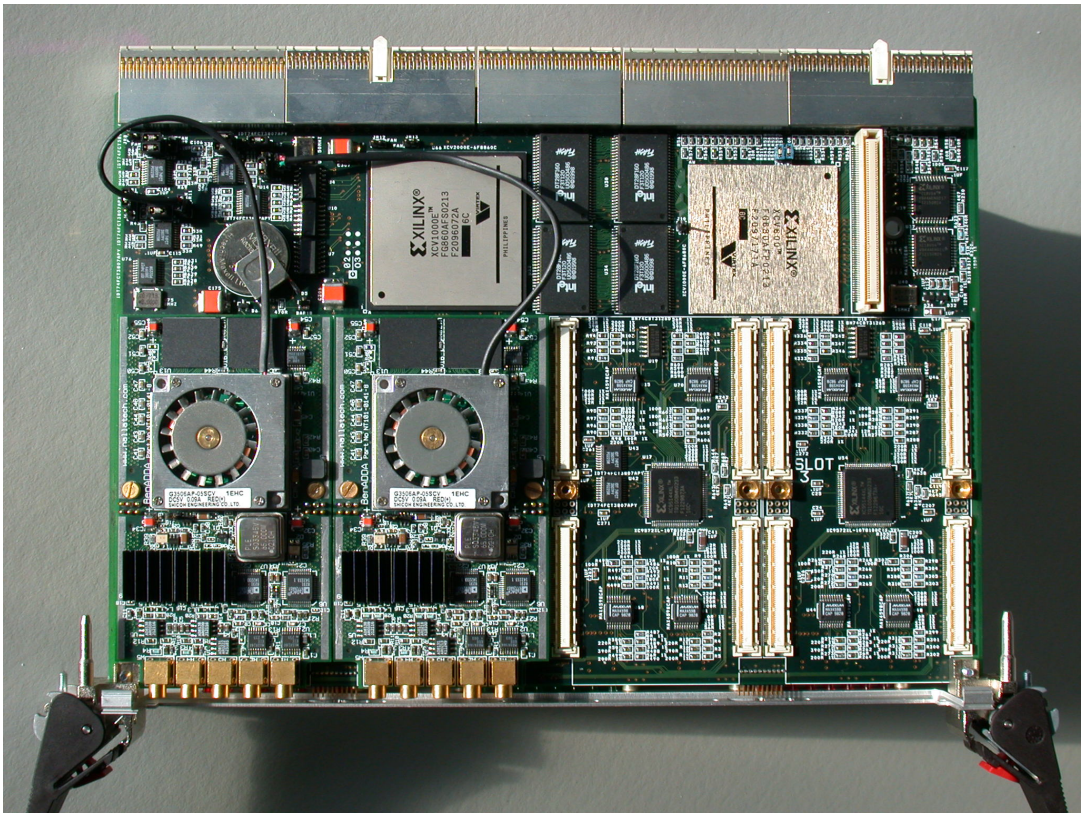
Trägerkarten und Module sind wie folgt ausgestattet:

BenERA cPCI-Träger:

- Ein frei konfigurierbarer Virtex-E FPGA (XCV1000E-5)
- Vier DIME-II Steckplätze
- Ring-Bus zur Verbindung benachbarter Module mit jeweils 122 bit
- Direkte Verbindungen zwischen allen Modulen mit jeweils 24 bit
- Gemeinsamer Bus für alle Module mit 64 bit
- Verbindung mehrerer Trägerkarten über Backplane mit 247 bit
- Bustakt je nach Bus und FPGA Konfiguration zwischen 66 MHz und 200 MHz
- Stromversorgung bis zu 32 A pro Modul

BenADDA Wandlermodul:

- Ein Virtex-II FPGA (XC2V3000-5)
- 4 MByte ZBT-SRAM in zwei Bänken mit jeweils 32 bit Breite und einer Taktrate bis 133 MHz
- Zwei AD-Wandler (Analog Devices AD6645) mit 105 MSPS Abtastrate und 14 bit Auflösung



(a) cPCI-Tragerkarte BenERA mit vier DIME-II Steckplätzen, hier bestückt mit zwei BenADDA Wandlermodulen. Jedes Wandlermodul verfügt über jeweils zwei analoge Ein- und Ausgänge sowie einen externen Clock-Eingang.

(b) Das Prozessormodul Benblue-II mit zwei Virtex-II FPGAs und 4 MByte ZBT-SRAM.

(c) 19"-Rack in einem geschlossenen Gehäuse mit zwei BenERA-Karten und einer Slot-CPU-Karte als Host-Rechner.

Bild 6.3: Das für den Demonstrator angeschaffte modulare FPGA System, bestehend aus einem Hostrechner, zwei cPCI-Tragerkarten sowie einigen DIME-II Modulen. Der Host läuft unter *Windows XP* und ermöglicht über eine MATLAB-Umgebung den direkten Zugriff auf die FPGAs.

- Zwei DA-Wandler (Analog Devices AD9772) mit 150 MSPS Abtastrate und 14 bit Auflösung
- Externer Clock-Eingang für eine synchrone Abtastung über mehrere Module
- Nutzbare Busbreite auf dem Ring-Bus des BenERA ist auf 64 bit beschränkt

BenBLUE-II Prozessormodul:

- Zwei Virtex-II FPGAs (Ein Modul mit XC2V6000-6, das andere mit XC2V4000-4 konfiguriert)
- 4 MByte ZBT-SRAM in zwei Bänken mit jeweils 64 bit Breite und einer Taktrate bis 133 MHz
- Stecker mit 80 digitalen I/O-Leitungen an der Frontseite mit direkter Verbindung zu einem der FPGAs
- Busbreite von 159 bit zwischen den beiden FPGAs

In Bild 6.4 ist die gewählte Konfiguration für ein 8x8-MIMO-System dargestellt. Jede Trägerkarte ist mit zwei Wandlermodulen bestückt. Die Abtastung der Zwischenfrequenzsignale mit 100 MHz führt zu einer Umsetzung der analogen Zwischenfrequenz (175 MHz) auf eine digitale Zwischenfrequenz von 25 MHz. Auf dem lokalen FPGA erfolgt dann die Umsetzung von der digitalen Zwischenfrequenz ins Basisband sowie die Reduktion auf Symbolrate mit einem dezimierenden Empfangsfilter. Die Speisung der Sender erfolgt entsprechend umgekehrt: Pulsformung und digitale Frequenzumsetzung vom Basisband auf 25 MHz, digital/analog-Wandlung mit 100 MHz und Selektion des Frequenzbandes bei 175 MHz mit einem Bandpassfilter. Um eine synchrone Abtastung zu gewährleisten, wird das Taktsignal für die Wandler extern erzeugt und mit gleichen Laufzeiten an die Wandlermodule verteilt. Das VLSI-Design für die FPGAs ist identisch für alle Wandlermodule. Der externe Speicher kann als Sende- und Empfangsspeicher für den Betrieb mit einer Offline-Signalverarbeitung eingesetzt werden.

Für die Echtzeit-Signalverarbeitung müssen die Basisbandsignale aller Sender und Empfänger auf einem Prozessormodul zusammengeführt werden. Bei einem Symboltakt von 25 MHz und einer Wortbreite von 32 bit pro Symbol führt das zu einem Datentransfervolumen von 800 $^{\text{Mbit}}/_{\text{s}}$ pro Kanal, d.h. für acht Sende- und Empfangskanäle insgesamt 12.8 $^{\text{Gbit}}/_{\text{s}}$. Sende- und Empfangsdaten werden mit einer Busbreite von jeweils 64 bit und 100 MHz Bustakt über den Ringbus auf das Prozessormodul geleitet. Die vier Kanäle der zwei Wandlermodule auf jedem Träger teilen sich jeweils einen 32 bit-Bus im zeitlichen Multiplex. Mit einigen zusätzlich benötigten Steuersignalen sind damit die I/O-Ressourcen des BenERA (Ringbus und Direktverbindungen) bis an die Grenzen erschöpft.

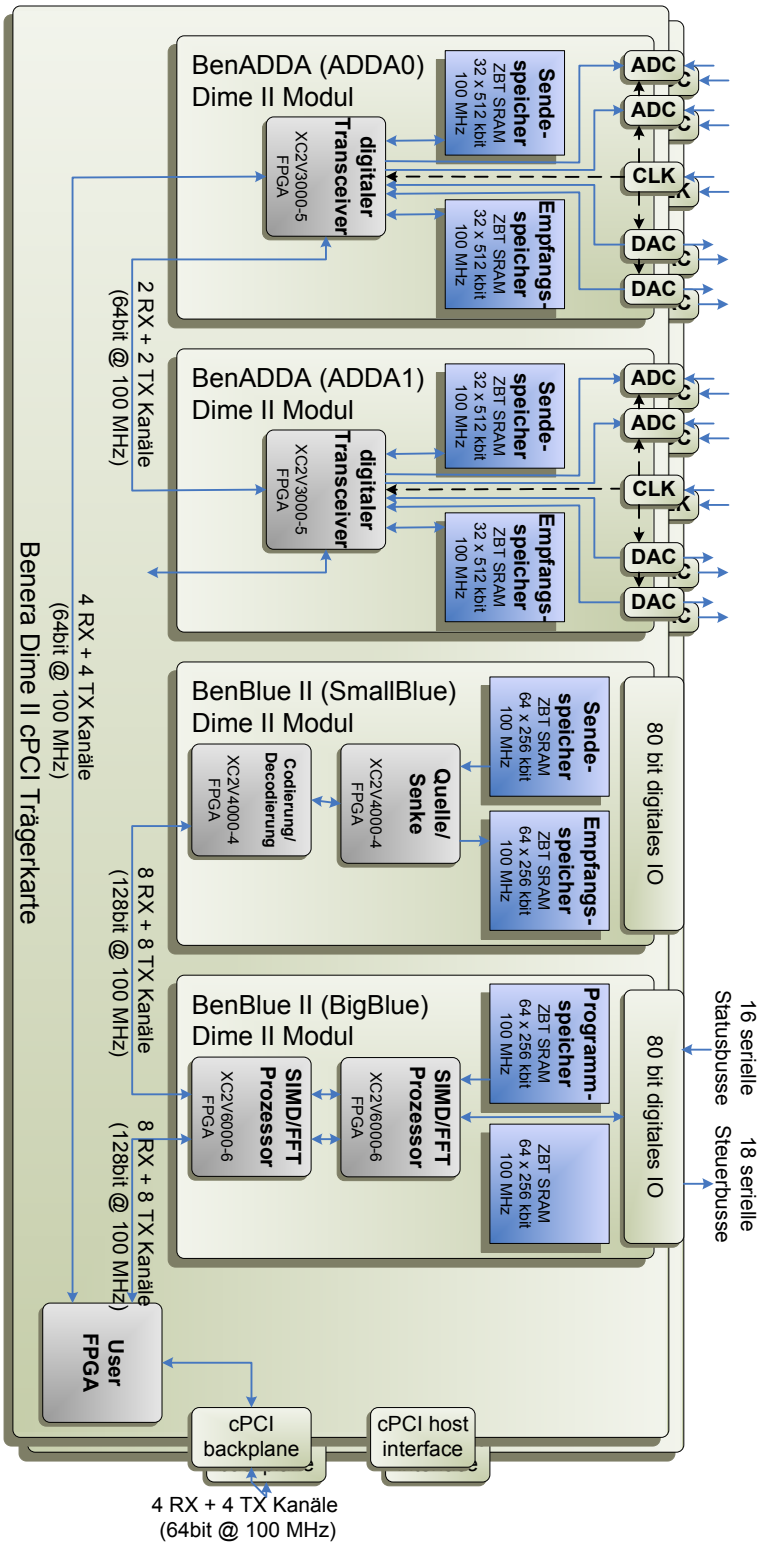


Bild 6.4: Konfiguration der digitalen Hardware für jeweils acht Sende- und Empfangskanäle mit sechs Modulen auf zwei Trägerkarten. Der Datentransfer zwischen den Modulen erfordert pro Kanal 800 $^{\text{Mbit}}/_{\text{s}}$. Jeweils vier Kanäle werden in einer Richtung im Zeitmultiplex übertragen.

FPGA Typ	**Anzahl**	**Slices** 2 Flip-Flops + 2 LUTs	**BlockRAMs** 18 kbit, zwei Ports	**Multiplizierer** 18 x 18 bit	**Speed-grade** (-4, -5, -6)
XC2V3000	4	14336	96	96	-5
XC2V4000	2	23040	120	120	-4
XC2V6000	2	33792	144	144	-6

Tabelle 6.1: Für den Demonstrator verfügbare FPGA-Ressourcen. Alle Bausteine stammen aus der Virtex-II Famile von Xilinx. Ein Speedgrade -6 erlaubt die höchsten Taktraten.

Alle acht Sende- und Empfangskanäle laufen auf einem der Prozessormodule zusammen. Hier sind die FFT-Prozessoren, der schaltbare Datenfluss für die unterschiedlichen Betriebszustände sowie der SIMD-Prozessror für die Raum-Zeit-Signalverarbeitung realisiert. Der externe Speicher wird als Programmspeicher für den SIMD-Prozessor verwendet. Die Steuer- und Statusbusse für die TR-Module und den Uplink/Downlink-Schalter werden über die digitale I/O-Schnittstelle auf der Frontseite mit einem FPGA verbunden. Über Registerzugriffe können von dort aus zentral alle Konfigurationen gesetzt und Statusinformationen ausgelesen werden.

Die Eingangsdaten von der Quelle sowie die Ausgangsdaten für die Senke werden über den Ringbus mit dem benachbarten Prozessormodul ausgetauscht. Auf dem zweiten Prozessormodul ist die Realisierung von Codierung und Decodierung, Modulation und Demodulation sowie eine Auswertung der Bitfehlerrate vorgesehen. In einem ersten Schritt werden diese Operationen noch offline auf dem Host-Rechner durchgeführt. Die beiden Bänke des externen Speichers werden unabhängig voneinander als Sende- und Empfangsspeicher eingesetzt. Für ein Folgeprojekt wird das kleinere BenBlue-II Modul durch ein BenDATA-II Modul mit einem Virtex-II Pro FPGA ersetzt. Mit dem integrierten PowerPC soll der Demonstrator um eine echtzeitfähige MAC-Schicht erweitert werden.

Über den cPCI-Bus und den gemeinsamen 64 bit-Bus auf dem BenERA kann vom Host-Rechner auf jeden FPGA zugegriffen werden. Eine MATLAB-API ermöglicht die Konfiguration der FPGAs, den Austausch von größeren Datenblöcken über einen DMA Zugriff sowie das Setzen von Steuerregistern und das Auslesen von Statusregistern innerhalb der FPGAs.

In Tabelle 6.1 sind die Ressourcen der im System verwendeten FPGAs aufgelistet. Die Logik-Ressourcen werden in *Slices* gezählt, mit jeweils zwei Flip-Flops und zwei look-up-tables (LUTs) mit je vier Eingängen und einem Ausgang. Neben der

konfigurierbaren Logik stehen fest verdrahtete Multiplizierer und Speicherblöcke (*BlockRAMs*) zur Verfügung. Die BlockRAMs bieten zwei Ports, eine Speicherkapazität von 18 kbit und eine Wortbreite bis 36 bit. Die Multiplizierer haben eine Eingangswortbreite von 18×18 bit. Die dedizierten Komponenten sind aufgrund der festen und optimierten Verdrahtung in der Leistungsaufnahme und der Chipgröße wesentlich effizienter als eine vergleichbare Realisierung über die Logikbausteine, so erfordert ein 18×18 bit Multiplizierer etwa 200 Slices, ein 18 kbit Dual-Port-Speicher schon 1300 Slices.

Das *speed grade* ist schließlich eine Einstufung der maximalen Gatterlaufzeiten und Leitungsverzögerungen innerhalb des FPGAs. Mit einem speed grade -6 ist nach Erfahrungswerten in der praktischen Anwendung eine Taktrate bis 200 MHz möglich, mit -4 nur bis etwa 100 MHz.

6.3 VLSI-Design

Die digitale Signalverarbeitung wurde in der Beschreibungssprache VHDL umgesetzt. Die Funktionalität wurde mit dem Tool *HDL-Designer* von Mentor Graphics entweder direkt in VHDL oder auch in Blockschaltbildern und Zustandsgraphen beschrieben. Die graphische Darstellung erlaubt mit kurzer Entwicklungszeit die übersichtliche Verschaltung von Einzelkomponenten, die meist aus frei verfügbaren Bibliotheken (*intellectual property*, IP) entnommen werden konnten (Xilinx *LogiCORE* sowie Mentor Graphics *ModuleWare*). Der *HDL-Designer* setzt die graphische Beschreibung automatisiert in VHDL um.

Die funktionale Verifikation des Codes erfolgt mit *Modelsim*, die Synthese zur Erzeugung generischer Netzlisten mit *PrecisionRTL*, beide Tools ebenfalls von Mentor Graphics. Die generischen Netzlisten wurden schließlich mit der integrierten Entwicklungsumgebung von Xilinx (ISE) auf die Logikressourcen der FPGAs abgebildet (MAP) und dann auf den verfügbaren Bausteinen platziert und geroutet (PAR).

Es folgt eine kurze Beschreibung der drei wesentlichen Komponenten im VLSI-Design: die digitalen Transceiver, die Fourier-Transformation und der Parallelprozessor. Anschließend wird noch auf die programmierbare Ablaufsteuerung eingegangen. Eine Auflistung der beanspruchten Logikressourcen ist dann in Kapitel 6.5 zu finden.

6.3.1 Digitale Transceiver

Auf jedem der vier Wandlermodule befinden sich zwei digitale Transceiver. Sender und Empfänger bestehen im wesentlichen aus einem interpolierenden bzw. dezi-

mierenden FIR-Filter mit integriertem IQ-Mischer. Da die Zwischenfrequenz genau einem Viertel der Abtastrate entspricht, kann der Mischeroszillator über die wiederholte Folge $\{1, \mathrm{j}, -1, -\mathrm{j}\}$ dargestellt werden. Die trivialen Multiplikationen mit ± 1 sind über eine entsprechende Vorzeichenumkehr innerhalb der Filterstruktur realisiert. Real- und Imaginärteil des Mischeroszillators sind jeweils abwechselnd gleich Null. Diese Kenntnis wird in der Filterstruktur ausgenutzt und halbiert so den Aufwand.

Entsprechend der Modellierung in Kapitel 2.2 erweitert der Sender die Datenblöcke zyklisch und führt dann eine lineare Filterung der aneinander gereihten Blöcke durch. Der Empfänger verwirft die zyklische Erweiterung des Senders und führt eine zyklische Filterung der verbleibenden Datenblöcke aus.

Die Filter sollen neben der Pulsformung und Bandselektion auch frequenzselektive Verzerrungen der analogen Frontends im Sinne eines reziproken Gesamtsystems ausgleichen. Die Filterkoeffizienten können für die Kalibrierung während des Betriebes neu geladen werden. Für die abwechselnde Nutzung der Transceiver in Zuganspunkt und Teilnehmerstationen stehen zwei parallele Koeffizientensätze zur Auswahl. Es können beliebige komplexwertige Impulsantworten bis zu einer Länge von 32 Koeffizienten verwendet werden, eine Symmetrie wird nicht vorausgesetzt. Realisiert sind die Filter mit jeweils 16 Multiplizierern und 16 Addierern bei einer Taktrate von 100 MHz. Die Wortbreite beträgt 16 bit.

6.3.2 Datenfluss und Fourier-Transformation

Das in Kapitel 4.1 vorgestellte Systemkonzept für die vier Betriebszustände des Zugangspunktes (Entzerrung und Vorverzerrung für Ein- und Mehrträgermodulation) wurde hier im Sinne der zentralen Signalverarbeitung erweitert. Der TDD-Betrieb erlaubt für Zugangspunkt und Teilnehmerstationen eine wechselseitige Nutzung gemeinsamer Ressourcen für Quelle, Sender, Empfänger, Senke und FFT-Prozessoren. Der resultierende Datenfluss ist in Bild 6.5 dargestellt.

Quelle und Senke sind in Abhängigkeit der Übertragungsrichtung jeweils entweder dem Zugangspunkt oder den Teilnehmerstationen zugeordnet. Sende- und Empfangsdaten laufen kontinuierlich von links nach rechts durch zwei parallele Zweige. Im oberen Zweig wird ein FFT-Prozessor, der Parallelprozessor für die Raum-Zeit-Verarbeitung sowie ein IFFT-Prozessor durchlaufen. Im unteren Zweig dienen FIFO-Speicher zum Ausgleich der Verarbeitungszeit der Prozessoren; die einzelnen Stufen werden jeweils synchron auf beiden Zweigen ausgelesen.

Über die vier Kreuzschalter kann der Datenfluss umgeleitet und damit der

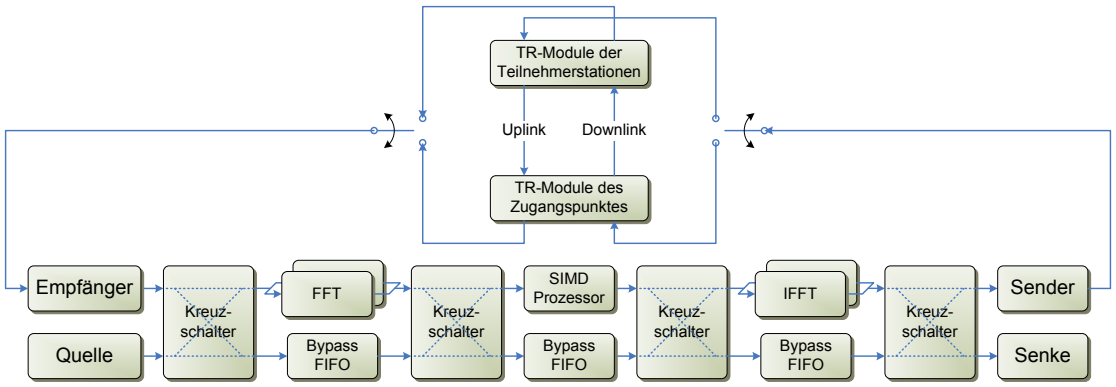


Bild 6.5: Zentrale Signalverarbeitung von Sende- und Empfangsdaten in einer gemeinsamen Verarbeitungskette. Quelle und Senke sind in Abhängigkeit der Übertragungsrichtung jeweils entweder dem Zugangspunkt oder den Teilnehmerstationen zugeordnet. Die vier Kreuzschalter erlauben einen Wechsel zwischen Entzerrung und Vorverzerrung mit Ein- oder Mehrträgermodulation.

gewünschte Betriebszustand gewählt werden: Einträger-Vorverzerrung (×===), Mehrträger-Vorverzerrung (=×==), Mehrträger-Entzerrung (==×=) oder Einträger-Entzerrung (===×). Während des laufenden Betriebes sind alle Komponenten zu jeder Zeit voll ausgelastet. Aufgrund der Latenzzeiten kann es jedoch beim Umschalten zwischen zwei Betriebszuständen zu Ressourcenkonflikten kommen.

Für die Fourier-Transformation wurde der IP-Kern *Fast Fourier Transform v3.0* aus dem LogiCORE Programm von Xilinx verwendet. Für die Echtzeitverarbeitung eines Datenblockes stehen 11.84 μs zur Verfügung, das entspricht 296 Chips für die Daten sowie die zyklische Erweiterung. Mit einem Takt von 100 MHz führt der Kern innerhalb von 10.24 μs nacheinander vier Transformationen der Länge 256 aus. Für acht Sende- und Empfangskanäle werden also jeweils zwei parallele Prozessoren für die FFT und die IFFT benötigt. Der Kern verwendet eine Festkomma-Arithmetik mit jeweils 16 bit für Real- und Imaginärteil. Die Skalierung der Zwischenergebnisse kann im Betrieb eingestellt werden. Die Ein- und Ausgabe der Daten erfolgt kontinuierlich mit einer Latenzzeit von etwa 5.5 μs.

Die FFT-Prozessoren bestimmen den Datentransfer zwischen den einzelnen Stufen der Verarbeitungskette. Innerhalb der Blockdauer werden auf zwei parallelen 64 bit-Bussen nacheinander jeweils vier Datenblöcke à 256 Chips übertragen. Der Bustakt entspricht der Taktrate von 100 MHz. Die Verarbeitung wäre damit theoretisch innerhalb von 10.24 μs möglich, die noch verfügbaren 1.6 μs der zyklischen Erweiterung geben Reserve für Latenzzeiten bei den Speicherzugriffen und dem Datentransfer. Der effektive Gesamtdurchsatz der FFT-Prozessoren beträgt 356 MSPS.

6.3.3 Parallelprozessor

Der Parallelprozessor für die Raum-Zeit-Verarbeitung wurde nach dem Konzept aus Kapitel 4.3 in VHDL umgesetzt und in zwei Konfigurationen für das BenBLUE-II Modul realisiert. Im Folgenden werden zunächst die Komponenten spezifiziert, die in beiden Konfigurationen identisch sind. Erst im Anschluss werden dann die spezifischen Eigenschaften der Konfigurationen erläutert.

Die Dimensionierung des Prozessors erfolgte mit dem Ziel einer möglichst hohen Rechenleistung unter den Randbedingungen der verfügbaren Ressourcen der zwei Virtex-II XC2V6000-6 FPGAs. Im Sinne einer möglichst hohen Taktrate wird nach den Vorüberlegungen in Kapitel 5 für die Zahlendarstellung die Festkomma-Arithmetik gewählt.

Die dedizierten BlockRAMs und Multiplizierer begrenzen die Speicherwortbreite auf maximal 18 bit. Nach den Simulationsergebnissen aus Kapitel 5.1 ist dies für eine Matrixinversion unter den dort definierten Voraussetzungen ausreichend, der RLS-Algorithmus kann dagegen nur eingeschränkt betrieben werden. Die übrigen Wortbreiten und festen Skalierungsfaktoren wurden entsprechend der Spalte (*Spec*) in Tabelle 5.1 ausgelegt. Die variablen Skalierungsfaktoren verfügen über eine Wortbreite von 4 bit und können mit einem festen Offset in der MAC-Pipeline von -10 bis 5, im Dividierer zwischen -8 und 7 gewählt werden.

Für die komplexe Multiplikation werden pro PE vier der dedizierten Multiplizierer verwendet. Die beiden Speicherbänke X und Y sind jeweils mit zwei parallelen BlockRAMs realisiert, einer für den Realteil und einer für den Imaginärteil. Damit verfügt jedes PE über 72 kbit bzw. 2048 Worte lokalen Speicher. Jeweils acht PE teilen sich einen Dividierer entsprechend Bild 4.7 mit $P_d = 8$. Für den Dividierer wurde der LogiCORE IP-Kern *Pipelined Divider v3.0* von Xilinx verwendet, konfiguriert für eine Division pro Taktzyklus mit einer Latenzzeit von 25 Taktzyklen.

Für den Datenaustausch zwischen den PE wird das Ergebnis einer MAC-Pipeline sowohl zum lokalen Speicher als auch zum Speicher des jeweils nächsten PE verteilt. Beim Schreiben in den Speicher kann dann zwischen dem lokalen Ergebnis und dem des vorhergehenden PE gewählt werden. Der Datenaustausch erfolgt dabei zyklisch, d.h. das letzte PE gibt die Daten an das erste weiter.

Die Steuereinheit verarbeitet eine lineare Folge von Vektorbefehlen wie in Kapitel 4.3.4 erläutert. Ein Vektorbefehl besteht aus einem Basisbefehlswort sowie einer optionalen Befehlserweiterung. Basisbefehl und Erweiterung sind jeweils 64 bit breit und werden über einen Befehlscode (opcode) unterschieden. Der Basisbefehl enthält die Vektorlänge, die Basisadressen zum Lesen der drei Koeffizienten und Schreiben

der Vektorprodukte, Vorzeichenwahl und Adressierung der variablen Skalierung für die MAC-Pipeline sowie die Konfiguration der Ausgangsregister in der Speichereinheit. Die Befehlserweiterung enthält vier Sprungweiten für die Addressberechnung, Ziel- und Skalierungsadresse für den Dividierer, Anweisungen für Programmsprünge, den Ruhezustand sowie die Rotation der Ergebnisse zwischen den PE und schließlich eine Erweiterung für größere Vektorlängen bis zu 128 Elementen. Für innere Vektorprodukte in zusammenhängenden Speicherbereichen (Sprungweite = 1) kann auf die Befehlserweiterung verzichtet werden. Die Steuereinheit verwendet in diesem Fall fest definierte Default-Werte. Die konkrete Syntax der Befehle und die bei fehlender Erweiterung angenommenen Default-Werte sind in Anhang B aufgelistet.

Die Schnittstelle zum externen Programmspeicher liest 64 bit Worte mit einem Takt von 100 MHz in den internen Cache. Der Cache ist als FIFO-Speicher mit einer Wortbreite von 128 bit und einer Tiefe von 511 Worten ausgelegt. Die eingelesenen Befehlsworte werden zu einem 70 bit breiten Steuersignal decodiert, das dann sternförmig an die PE verteilt wird. Der Latenzausgleich für die Pipelines der Rechenwerke erfolgt zur Laufzeit über Registerketten innerhalb der Steuereinheit.

Für eine der beiden Konfigurationen, *Space-Time-Processor-200* (STP-200), wurden $P = 32$ PE zusammen mit der Steuereinheit auf einem XC2V6000-6 integriert (siehe Primary FPGA in Bild 6.6). Das Design erfüllt die Timingspezifikationen für den Betrieb mit einer Taktrate von 200 MHz. Die im Vergleich zu den FTT-Prozessoren doppelte Taktrate erlaubt einen Multiplex der beiden parallel transformierten Datenströme auf eine 32 bit breite Registerkette. In der Speichereinheit müssen I/O-Puffer und Speicherbank Z als verteilte Speicher über die Logikelemente realisiert werden, da die Speicherbänke X und Y bereits 128 der 144 BlockRAMs belegen. Wie in Kapitel 4.3.4 vorgeschlagen, werden hierzu zwei Speicher mit jeweils einem 32 bit Port und einer Speichertiefe von 64 Worten eingesetzt.

Die vier FFT-Prozessoren mit dem im vorhergehenden Kapitel beschriebenen Datenfluss befinden sich auf dem zweiten FPGA (secondary). Die Schnittstelle zwischen den FPGAs bilden im Wesentlichen nur Ein- und Ausgang der Registerkette mit jeweils 32 bit bei einem Takt von 200 MHz. Die interne Vernetzung der PE setzt sich zusammen aus der Registerkette (I/O-Bus), dem Ring-Bus für den zyklischen Austausch der MAC-Ergebnisse, jeweils zwei Registerketten für Eingang und Ausgang jedes Dividierers sowie der sternförmigen Verteilung der Steuersignale (Steuer-Bus).

Für die hohe Taktrate müssen eine Reihe von Registern innerhalb der Verarbeitungsketten vorgesehen werden. Die resultierende Latenzzeit von Speicher zu Speicher beträgt 11 Taktzyklen durch die MAC-Pipeline und 45 Taktzyklen durch MAC-Pipeline und Dividierer.

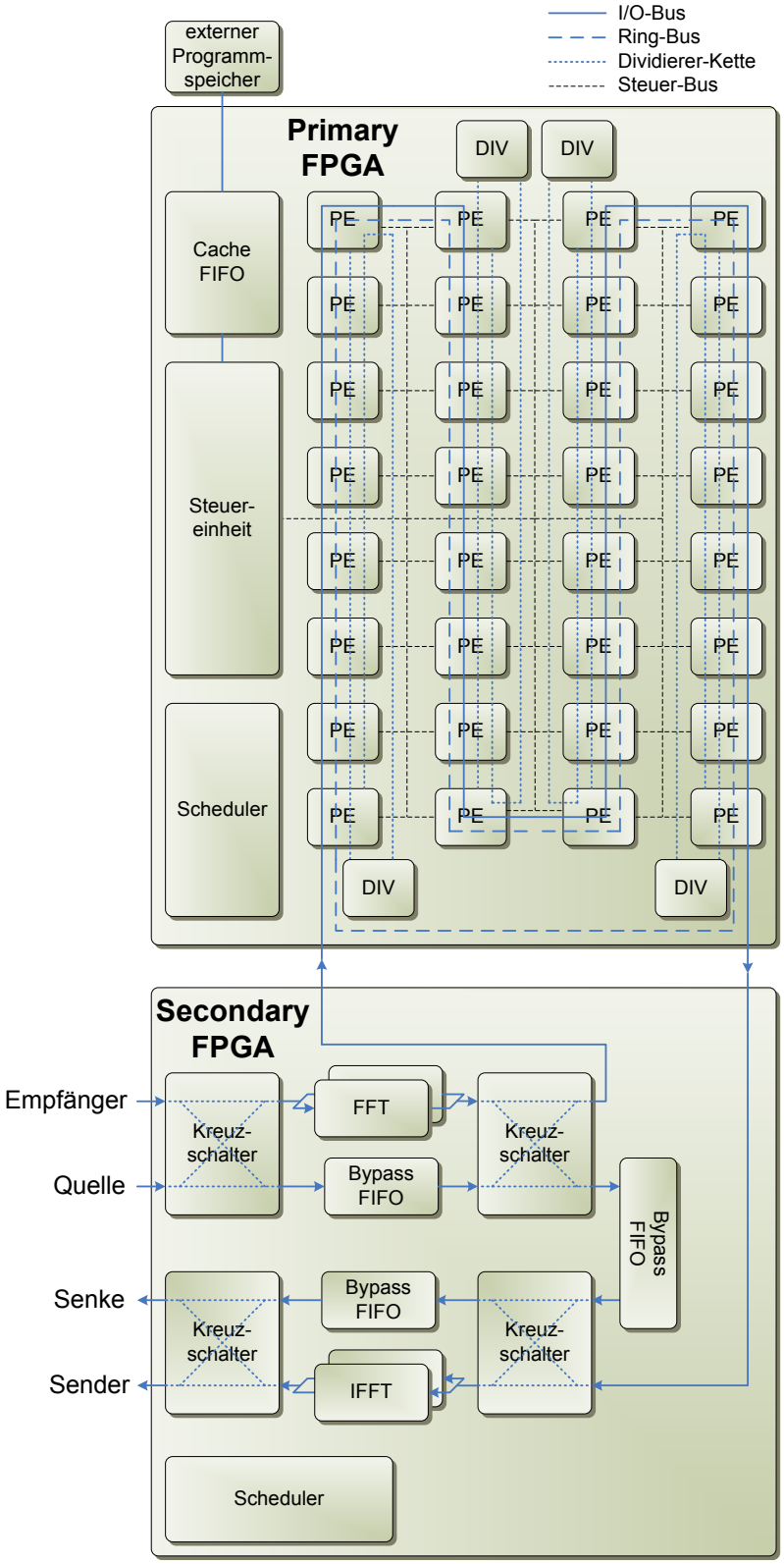


Bild 6.6: Prozessorkonfiguration STP200. Der SIMD-Prozessor mit 32 Prozessorelementen ist auf einem FPGA integriert, die vier FFT-Prozessoren auf dem anderen. Die Schnittstelle zwischen den FPGAs bilden zwei 32 bit-Busse, getaktet mit 200 MHz.

Mit 128 Multiplizierern ergibt sich für den mit 200 MHz getakteten Parallelprozessor eine theoretische Rechenleistung von 25,6 $^{\mathrm{GMAC}}/_{\mathrm{s}}$. Hinzu kommen die dedizierten Dividierer sowie rund 4,8 $^{\mathrm{GMAC}}/_{\mathrm{s}}$ der FFT-Prozessoren. Die Analyse des endgültigen Designs hat für den Parallelprozessor jedoch eine deutliche Überschreitung der zulässigen Leistungsaufnahme ergeben. Zudem ist eine ausreichende Wärmeabfuhr nicht mit vertretbarem Aufwand zu erreichen (mehr dazu in Kapitel 6.5).

Als Alternative wurde eine zweite Konfiguration (STP100, Bild 6.7) erstellt: mit einer reduzierten Taktrate von 100 MHz, ebenfalls $P = 32$ PE, jedoch zu gleichen Teilen verteilt auf beide FPGAs. Die Reduktion der Taktrate erfordert eine Erweiterung der Registerkette für die Ein- und Ausgabe der Daten auf 64 bit. Aufgrund einer gleichmäßigeren Belegung der Ressourcen können I/O-Puffer und Speicherbank Z nun über zwei BlockRAMs für Real- und Imaginärteil realisiert werden. Ein Achtel des gemeinsamen Speicherbereiches wird als *switch buffer* zum Austausch der Daten genutzt, der Rest ist frei verfügbar z.B. zum Ablegen der Spreizcodes, Trainingssequenzen oder der Filterkoeffizienten für die Interpolation. Die Latenzzeit durch die MAC-Pipeline hat sich aufgrund der niedrigeren Taktrate auf 9 Zyklen reduziert. Der Dividierer wurde unverändert übernommenen.

Neben den PE sind auch die 4 FFT-Prozessoren gleichmäßig auf beide FPGAs verteilt. Die Steuereinheit befindet sich wegen des Zugriffs auf den externen Programmspeicher auf dem oberen (Primary) FPGA. Die Schnittstelle zwischen den FPGAs ist deutlich aufwändiger als bei der ersten Konfiguration: die Registerkette, nur in einer Richtung, jedoch mit 64 bit, das Bypass-Signal mit 64 bit, die Konfigurationssignale für die PE und Dividierer mit 70 bit sowie zwei mal 36 bit für die zyklische Rotation der MAC-Ergebnisse zwischen den PE. Die Anzahl der benötigten I/O Leitungen (über 270) muss mit einer Datenübertragung auf beiden Taktflanken (DDR, *double data rate*) halbiert werden.

Die interne Vernetzung der PE ist identisch zu der ersten Konfiguration. Die Rechenleistung des Parallelprozessors halbiert sich auf 12,8 $^{\mathrm{GMAC}}/_{\mathrm{s}}$, die der FFT-Prozessoren bleibt bei 4,8 $^{\mathrm{GMAC}}/_{\mathrm{s}}$.

6.3.4 Ablaufsteuerung

Die Verarbeitung erfolgt blockweise im festen zeitlichen Raster der Blockdauer von 11.84 μs. Die verschiedenen Komponenten der Verarbeitungskette in Bild 6.5 werden über so genannte Scheduler konfiguriert. Ein Scheduler arbeitet ein vorgefertigtes Programm ab und ersetzt damit einen aktiven Echtzeitkontroller. Für jeden Block wird ein Konfigurationswort aus einem Programmspeicher ausgelesen und

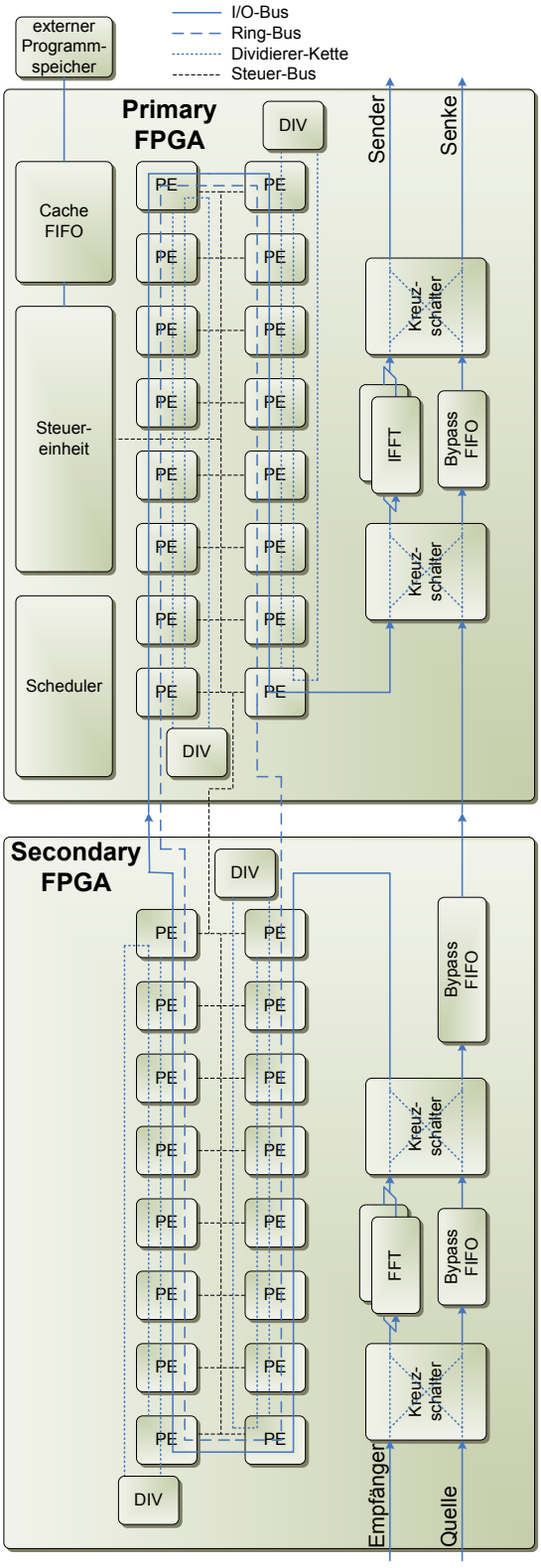


Bild 6.7: Prozessorkonfiguration STP100. Die 32 Prozessorelemente des SIMD-Prozessors und die vier FFT-Prozessoren sind hier jeweils gleichmäßig auf beide FPGAs verteilt. Die Schnittstelle zwischen den FPGAs ist mit insgesamt 135 bit aufwändiger als bei STP200.

zur Ansteuerung an die Komponenten verteilt. Die Latenzzeiten der Verarbeitungskette werden durch entsprechende Verzögerungsglieder ausgeglichen. Konfigurierbar sind unter anderem die Koeffizientensätze in den Transceivern, die Skalierung der FFTs, Stellung der Kreuzschalter für den Datenfluss, Sprungadressen für den Parallelprozessor sowie Lese- und Schreibadressen für die externen Speicher. Auch die TR-Module sowie der Schalter für Uplink- und Downlink-Betrieb werden über die seriellen Steuerbusse synchron zur internen Verarbeitung mit einem Scheduler konfiguriert.

6.4 Programmierung des Parallelprozessors

Der Binärcode für den Parallelprozessor wird mit Hilfe von MATLAB erzeugt. Die MATLAB-Umgebung erlaubt eine komfortable und strukturierte Beschreibung der Algorithmen mit Hilfe von Schleifen, Programmverzweigungen sowie eine übersichtliche Indizierung der Matrizen. Erzeugt wird eine MATLAB-Quelldatei (M-File), in der dann mit einer spezifizierten Syntax Vektorbefehle verwendet werden. Bild 6.8 zeigt exemplarisch den Quellcode für die LDL-Zerlegung der Δ-Matrix.

Die bitgenaue Modellierung des Rechenwerkes aus Kapitel 5.1.1 ermöglicht die Verifikation der Funktionalität anhand von Testdaten. Für die Erzeugung des Maschinencodes wird beim Aufruf des Quellcodes zunächst jeder verwendete Vektorbefehl sequentiell in einer Liste abgespeichert. Unter Berücksichtigung der tatsächlichen Latenzzeiten wird im Anschluss der resultierende Programmablauf auf Ressourcenkonflikte untersucht und für den Lesezugriff die Verfügbarkeit zuvor berechneter Ergebnisse überprüft. Bei einem Konflikt wird zunächst versucht, unabhängig durchführbare Befehle aus anderen Programmen einzufügen (Befehls-Interleaving). So können z.B. die Matrixinversionen mehrerer Unterträgergruppen sowie eine parallel durchgeführte Gewichtung in unabhängigen Programmlisten definiert und dann verschachtelt abgearbeitet werden. Erst wenn dies nicht mehr möglich ist, werden Wartezyklen eingefügt.

Zur Vermeidung von Wartezyklen muss auf die Partitionierung der lokalen Daten auf die drei Speicherbänke X, Y und Z geachtet werden. Auf jede Bank sind gleichzeitig maximal zwei Zugriffe möglich. Die Zuordnung erfolgt manuell durch den Programmierer bei der Initialisierung der Variablen.

```
function [Delta, L, inst_tab] = LDL_Zerlegung(Delta, L, inst_tab)
% Bittrue calculation of LDL Decomposition of rectangular, symmetrical matrix
% Delta. Only the lower triangular part of Delta is used. It will be overwritten by
% temporal matrix V in the lower triangular part and by the inverse of diagonal
% matrix D on the main diagonal. Matrix L is also lower triangular, the unity
% main diagonal is not written.
%
% Globally defined scaling table entries 'copy', 'Di', 'V' and 'L' are used.
%
% The required vector commands are sequentially appended to instruction table inst_tab
%
% syntax of vector instruction:
% result = vec_inst(vector_length, A, B, C, mac_result, mac_scaling, div_result,
%                   div_scaling, conjugate_abc, sign_abc, instruction_table)
K = size(Delta,1); % get matrix dimension
for s = 1:K % loop over all columns
   % main diagonal
   if s == 1 % on first column coefficient is loaded through MAC pipeline and inverted
      [Delta(s, s), inst_tab] = vec_inst(1, Delta(s, s), [], [], [], 'copy', Delta(s, s), 'Di', ...
                                         [0,0,0], [0,0], inst_tab);
   else % calculation of D matrix entry and inversion
      [Delta(s, s), inst_tab] = vec_inst(s-1, L(s, 1:s-1), Delta(s, 1:s-1), Delta(s, s), ...
                                         [], 'V', Delta(s, s), 'Di', [0,1,0], [1,0], inst_tab);
   end
   % rest of column (no operations generated for first and last column)
   for z = s+1:K % calculate V matrix entries for all rows in column
      [Delta(z, s), inst_tab] = vec_inst(s-1, L(z, 1:s-1), Delta(s, 1:s-1), Delta(z, s), ...
                                         Delta(z, s), 'V', [], [], [0,1,0], [1,0], inst_tab);
   end
   if s < K % normalizing V by multiplying with inverted D to get L
      [L(s+1:K, s), inst_tab] = vec_inst(K-s, Delta(s+1:K, s), Delta(s, s), [], ...
                                         L(s+1:K, s), 'L', [], [], [0,0,0], [0,0], inst_tab);
   end
end
```

Bild 6.8: Quellcode für die LDL-Zerlegung in MATLAB Notation. Ein Aufruf der Funktion in MATLAB berechnet die LDL-Zerlegung der übergebenen Matrix $\mathbf{\Delta}$ in einer bitgenauen Emulation des Rechenwerkes und speichert die verwendeten Vektorbefehle sequentiell in einer Liste. Der Maschinencode für den Parallelprozessor wird dann auf Basis der Befehlsliste unter Berücksichtigung möglicher Ressourcenkonflikte erzeugt.

Komponente	MUL	BRAM	DFF	LUT	Slices	Anzahl	Anteil Slices	
							Pri	Sec
Prozessorelement	4	6	766	756	595	×16	57%	61%
Dividierer	-	-	1.065	413	541	×2	7%	7%
Steuereinheit	-	4	795	679	638	×1	4%	-
FFT Prozessor	12	4	2.603	2.812	2155	×2	26%	28%
Gesamt (Pri)	88	116	21.993	20.642	16.839	-	100%	-
Gesamt (Sec)	88	124	20.616	17.787	15.636	-	-	100%

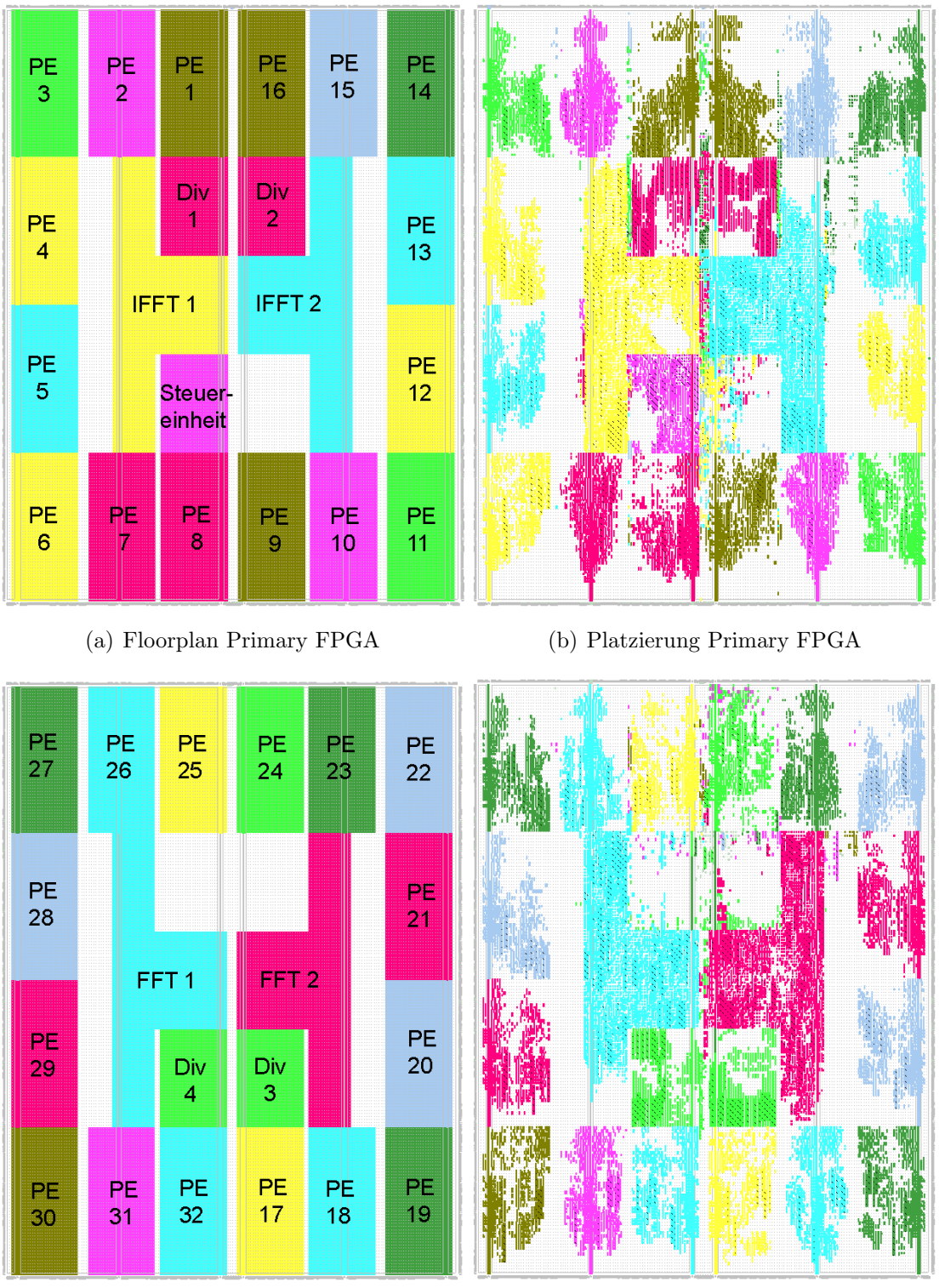
Tabelle 6.2: Ressourcenverbrauch der verschiedenen Komponenten am Beispiel der Konfiguration STP100

6.5 Ressourcenverbrauch

Anhand der FPGA-Realisierungen soll nun der Ressourcenverbrauch des Parallelprozessors abgeschätzt werden. Die Anzahl der benötigten Logikelemente erlaubt für eine FPGA-Realisierung eine recht genaue Aufwandsbewertung unabhängig von Hersteller und Produktfamilie. Auf dem Virtex-II sind jeweils zwei D-Flip-Flops (DFF) und zwei konfigurierbare Look-up-Tables (LUT) zu einer *Slice* zusammengefasst. Dedizierte Komponenten, wie Multiplizierer (MUL) und BlockRAMs (BRAM), müssen gesondert gezählt werden.

In Tabelle 6.2 ist der Ressourcenverbrauch der einzelnen Komponenten am Beispiel der Konfiguration STP100 aufgetragen. Knapp 2/3 der belegten Slices entfallen auf die Prozessorelemente, etwa 1/4 auf die FFT-Prozessoren. Die Dividierer sowie die Steuereinheit fallen mit 7% bzw. 4% kaum ins Gewicht. Der Gesamtverbrauch zeigt eine gleichmäßige Verteilung auf die beiden FPGAs *Primary* und *Secondary*. Auf beiden FPGAs wird nur etwa die Hälfte der jeweils 33792 verfügbaren Slices belegt. Die alternative Konfiguration (STP200) belegt dagegen 83% und 38% der beiden FPGAs. Die Zahlenwerte werden bei der Abbildung der generischen Netzliste auf die Ressourcen der Zielplattform ermittelt. Dabei wurden jeweils nur unmittelbar zusammenhängende Gatter auf eine Slice abgebildet. Das Zusammenfassen unabhängiger Gatter reduziert zwar den Ressourcenverbrauch, erschwert jedoch das Einhalten der Timing-Spezifikationen.

Die Bilder 6.9 und 6.10 zeigen für beide Konfigurationen die Platzierung der Logikelemente auf der Chipfläche. Über den Floorplan (jeweils links) beschränkt der



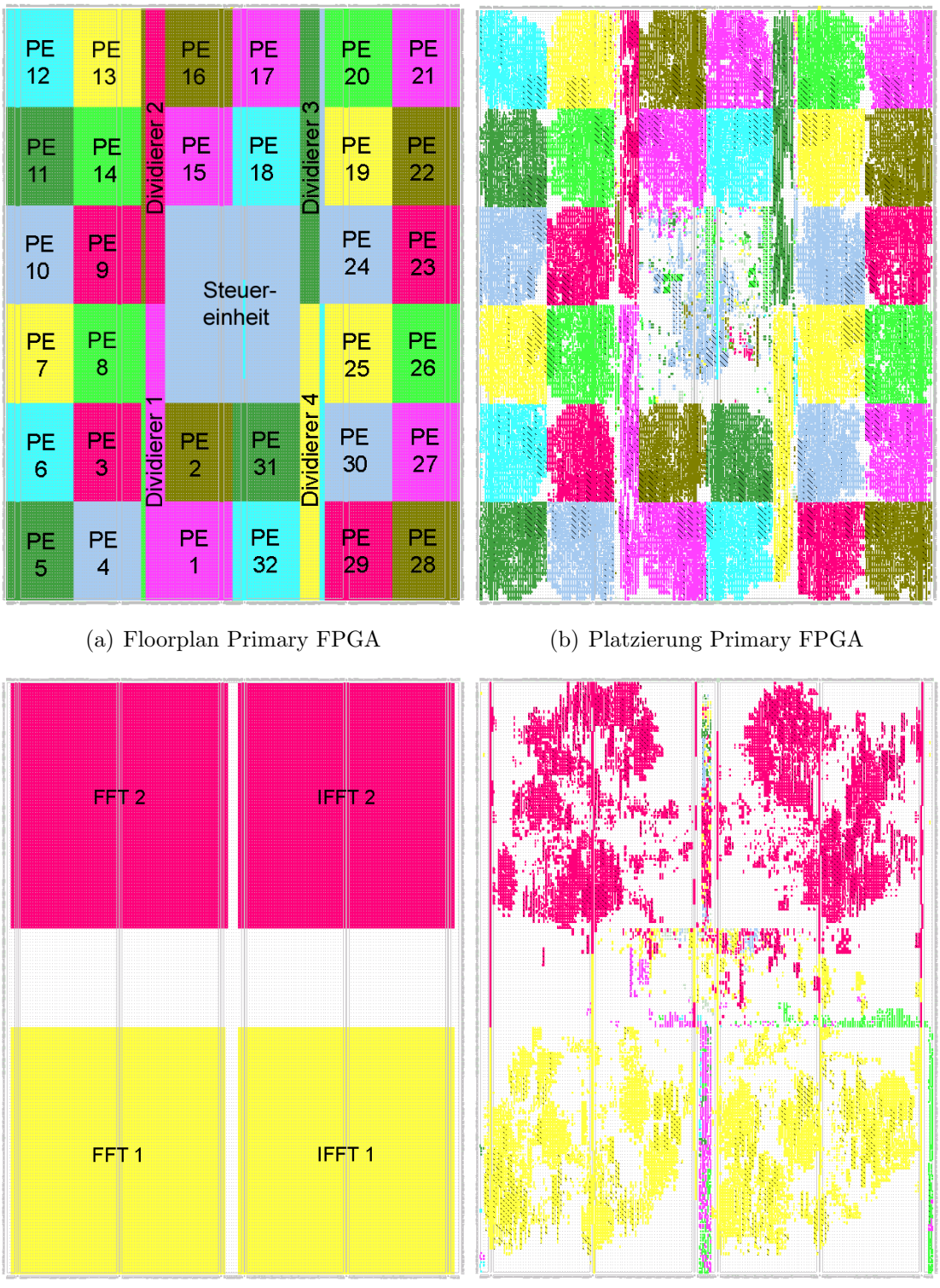
(a) Floorplan Primary FPGA

(b) Platzierung Primary FPGA

(c) Floorplan Secondary FPGA

(d) Platzierung Secondary FPGA

Bild 6.9: Floorplan und Platzierung für die STP100 Konfiguration. Beide FPGAs arbeiten mit einer Taktrate von 100 MHz.



(a) Floorplan Primary FPGA

(b) Platzierung Primary FPGA

(c) Floorplan Secondary FPGA

(d) Platzierung Secondary FPGA

Bild 6.10: Floorplan und Platzierung für die STP200 Konfiguration. Der SIMD-Prozessor auf dem Primary FPGA erreicht eine Taktrate von 200 MHz, die FFT-Prozessoren auf dem Secondary FPGA arbeiten mit 100 MHz.

Entwickler die Platzierung einzelner Komponenten der Design-Hierarchie auf die gekennzeichneten Bereiche. Das komplexe Optimierungsproblem der Platzierung wird somit in kleinere lokale Probleme zerlegt. Zur Reduktion der Laufzeiten müssen die Verbindungen zwischen den Komponenten möglichst kurz gehalten werden, daher die serielle Anordnung der PE. Der Floorplan richtet sich außerdem nach der Lage der BlockRAMs und dedizierten Multiplizierer, die in sechs vertikalen Spalten über die Chipfläche verteilt sind. Mit Ausnahme der Steuereinheit von STP200 sind alle Bereiche exklusiv für die betreffenden Komponenten reserviert. Die übrige Logik, nach Tabelle 6.2 etwa 5% der Slices, verteilt sich auf die frei gebliebenen Flächen. Die Platzierung (jeweils rechts) zeigt farbig markiert die aus dem Optimierungsprozess resultierende Belegung der einzelnen Slices. In STP200 zeigt sich deutlich die hohe Dichte der Belegung für den Parallelprozessor, in STP100 dagegen eine ausgewogene Belegung der beiden FPGAs mit einer ähnlichen Verteilung der Slices.

Auf Grundlage der platzierten und gerouteten Netzlisten kann die Leistungsaufnahme sowie die Kerntemperatur für den laufenden Betrieb abgeschätzt werden. Das Programm *XPower* von Xilinx erlaubt eine recht präzise Vorhersage durch Berücksichtigung sämtlicher interner Schaltvorgänge, die für eine definierte äußere Anregung mit einem Simulator wie Modelsim aufgezeichnet werden können. Dieser Prozess war aufgrund der hohen Komplexität des Designs zu aufwändig. Stattdessen wurde für alle internen Signalwege eine einheitliche, auf die jeweilige Taktfrequenz von 100 MHz bzw. 200 MHz bezogene Schaltfrequenz definiert. In den vorliegenden Schaltungen dominieren Datenbusse, die Binärzahlen im Zweierkomplement führen. Unter der Annahme, dass die einzelnen Bitstellen unkorreliert und gleichverteilt zwischen '0' und '1' wechseln, ergibt sich dann eine mittlere Schaltfrequenz von 50% der Taktrate.

Zur Berechnung der Kerntemperatur wird eine Umgebungstemperatur von $T_u = 35\,^{\circ}\mathrm{C}$ sowie eine passive Kühlung über einen hochwertigen Kühlkörper angenommen. Der Kühlkörper verfügt über einen effektiven thermischen Widerstand von $\Theta_k = 2{,}0\,^{\circ}\mathrm{C/w}$ inkl. Klebemittel. Ein wesentlich geringerer Widerstand ist auch mit einer aktiven Kühlung kaum erreichbar. Der thermische Widerstand des FPGA Gehäuses (FF1152) ist zu $\Theta_g = 0{,}5\,^{\circ}\mathrm{C/w}$ spezifiziert. Bei einer Leistungsaufnahme P_v ergibt sich die Kerntemperatur (*junction temperature*) T_j zu

$$T_j = T_u + (\Theta_g + \Theta_k) P_v \tag{6.1}$$

Tabelle 6.3 zeigt die resultierenden Schätzwerte für die Leistungsaufnahme und die Kerntemperatur, jeweils für beide FPGAs der zwei Konfigurationen. Die bei Platzierung und Routen vorausgesetzten Timing-Spezifikationen der Gatterlaufzeiten

Konfiguration	**FPGA**	**äquivalente Gatteranzahl** [Millionen Gatter]	**Leistungs-verbrauch** [W]	**Kern-temperatur** [°C]
STP100	Primary	8,5	15,1	73
	Secondary	9	13,2	68
STP200	Primary	11	38,5	131
	Secondary	7	11,9	65

Tabelle 6.3: Gatteranzahl, Verlustleistung und Kerntemperatur für die jeweiligen FPGAs der beiden Konfigurationen. Der Leistungsverbrauch und die daraus resultierende Kerntemperatur folgen aus einer Analyse der platzierten und gerouteten Netzlisten durch das Programm XPower. Die Gatteranzahl ist nur bedingt aussagekräftig.

werden nur bis zu einer Kerntemperatur von 80 °C garantiert. Bei höheren Temperaturen muss mit Timing-Fehlern gerechnet werden. Ab 125 °C kann der Kern dauerhaft beschädigt werden. Die maximal zulässige Leistungsaufnahme liegt bei etwa 20 W.

Das Design für den Primary FPGA bei STP200 überschreitet sowohl mit der Leistungsaufnahme als auch mit der Kerntemperatur deutlich die zulässigen Grenzwerte und ist somit nicht realisierbar. Eine Halbierung der Taktrate von 200 MHz auf 100 MHz wirkt sich in etwa linear auf die Verlustleistung aus. Im Redesign (STP100) wurde zudem auf eine gleichmäßigere Verteilung der beanspruchten Ressourcen geachtet. Mit etwa 15 W pro FPGA und einer Kerntemperatur um die 70 °C liegt diese Konfiguration innerhalb der zulässigen Grenzwerte.

Für die Kombination aus FFT-Prozessoren und Parallelprozessor wird unabhängig von der Konfiguration eine äquivalente Gatteranzahl von etwa 18 Millionen Gattern genannt. Die äquivalente Gatteranzahl gibt unabhängig von der physikalischen Umsetzung an, wie viele NAND-Gatter zur Abbildung der Funktionalität benötigt werden. Sie erlaubt den Schluss von der hier durchgeführten FPGA-Realisierung auf eine mögliche ASIC-Realisierung. Die Methodik der Berechnung einer äquivalenten Gatteranzahl ist jedoch nicht eindeutig definiert, besonders im Bereich der Speicherlogik. Die Herstellerangaben sind aus Marketinggründen oft nach oben geschönt und daher nur bedingt aussagekräftig.

Für eine grobe Abschätzung der Größenordnung kann dennoch ein Vergleich zu anderen Prozessoren gezogen werden. Ein Pentium 4 mit Prescott Kern besteht aus 125 Millionen Transistoren, das entspricht etwa 32 Millionen Gattern bei vier

Transistoren pro NAND-Gatter. Aktuelle Hochleistungs-DSPs von TI (C64x) sind mit etwa 65 Millionen Transistoren, d.h. 16 Millionen Gattern spezifiziert. Der mit 256 GFLOPS angekündigte Cell erfordert bereits 234 Millionen Transistoren, bzw. knapp 60 Millionen Gatter. Der Vergleich der Gatteranzahl mit dem C64x zeigt, dass eine ASIC-Realisierung auf einem Chip mit wenigen Watt Verlustleistung möglich sein sollte.

Mit einer Belegung von 50% der verfügbaren Logikressourcen und der halben von den Timing-Spezifikationen her möglichen Taktrate erreicht die Konfiguration STP100 auf der Virtex-II Architektur bereits die Grenzen der Realisierbarkeit. Die nächste Generation (Virtex-4) verspricht mit einem verbesserten Prozess in der Herstellung eine deutlich höhere Leistungseffizienz. Eine Portierung auf die Virtex-4 Architektur sollte damit auch eine FPGA-Realisierung des Parallelprozessors in der STP200 Konfiguration mit den angestrebten 25,6 $^{\mathrm{GMAC}}/_{\mathrm{s}}$ ermöglichen.

6.6 Fazit

Das Institut für Hochfrequenztechnik verfolgt mit dem hier vorgestellten MIMO-Demonstrator ein ehrgeiziges Konzept mit dem Ziel, Realisierungskonzepte zu erarbeiten, den erforderlichen Aufwand abzuschätzen und schließlich die Machbarkeit unter realistischen Bedingungen zu zeigen. Bis zur Fertigstellung der vorliegenden Arbeit konnte in einer Zwischenlösung eine unidirektionale Übertragung im X-Band mit zwei Sende- und zwei Empfangsantennen umgesetzt werden. Mit einer Rohdatenrate von 200 $^{\mathrm{Mbit}}/_{\mathrm{s}}$ war dies eine der weltweit ersten breitbandigen MIMO-Übertragungen. Das gewählte Konzept zeigt ein großes Potential für eine Umsetzung der räumlich-zeitlichen Vorverzerrung, die zwar in der Theorie ausführlich behandelt, in der Praxis bisher aber nie demonstriert wurde.

Voraussetzung dazu ist eine leistungsfähige digitale Signalverarbeitung. Die für den Demonstrator angeschaffte digitale Hardware verspricht mit insgesamt acht FPGAs allein über dedizierte Multiplizier eine Rechenleistung über 100 $^{\mathrm{GMAC}}/_{\mathrm{s}}$. Die verfügbare Hardware sowie die Anforderungen des Demonstrators stellen die Randbedingungen für eine konkrete Umsetzung des Parallelprozessors: der Echtzeitbetrieb mit jeweils acht Sende- und Empfangsantennen für die Ein- und Mehrträgermodulation mit 256 Unterträgern und einer Chiprate von 25 MHz für eine Rohdatenrate bis zu 800 $^{\mathrm{Mbit}}/_{\mathrm{s}}$.

Nach dem Konzept einer zentralen Signalverarbeitung für Zugangspunkt und Teilnehmerstationen wurde mit acht digitalen Transceivern und vier FFT-Prozessoren eine universelle Umgebung geschaffen, die über einheitliche Schnittstel-

len zum Parallelprozessor die Entzerrung und Vorverzerrung mit Ein- und Mehrträgermodulation erlaubt. Der Parallelprozessor wurde mit 32 Prozessorelementen und vier Dividierern in einer Festkomma-Arithmetik umgesetzt. Mit einer Speicherwortbreite von 18 bit pro Quadraturkomponente werden die dedizierten Komponenten der Virtex-II Architektur voll ausgeschöpft. Die Programmierung erfolgt über Vektorbefehle mit einem oder zwei 64 bit-Befehlsworten pro Vektorprodukt.

Die Analyse der beanspruchten Ressourcen zeigt, wie für den SIMD-Ansatz erhofft, einen vernachlässigbaren Anteil der Steuereinheit. Die Dividierer belegen mit 10% ebenfalls nur einen kleinen Anteil des Parallelprozessors. Die insgesamt für die Raum-Zeit-Verarbeitung erforderlichen Logikressourcen und Multiplizierer verteilen sich zu etwa 70% auf den Parallelprozessor, zu 30% auf die FFT-Prozessoren.

Die Integration des gesamten Parallelprozessors auf einem FPGA mit einer Taktrate von 200 MHz hat sich aufgrund einer deutlichen Überschreitung der zulässigen Leistungsaufnahme und Kerntemperatur auf der Virtex-II Architektur als nicht realisierbar herausgestellt. Erst die Halbierung der Taktrate auf 100 MHz sowie eine ausgewogenere Aufteilung der Prozessorelemente und FFT-Prozessoren auf zwei FPGAs hat in einem Redesign zu einer realisierbaren Lösung geführt. Da so nur etwa die Hälfte der Logikressourcen belegt und die Gatterlaufzeiten nicht ausgeschöpft sind, sollte für diesen Ansatz eine Erweiterung auf die Fließkomma-Arithmetik nach Kapitel 5.2 durchaus möglich sein. Eine mit bestehenden Signalprozessoren vergleichbare Gatteranzahl verspricht eine mögliche ASIC-Umsetzung mit wenigen Watt Verlustleistung.

Aufgrund des notwendigen Redesigns und einer noch nicht ausreichenden Kühlung der FPGAs konnte das endgültige Design vor Fertigstellung dieser Arbeit nicht mehr in der Hardware getestet werden. Lauffähig ist eine Vorstufe mit acht Prozessorelementen, die für das bestehende 2×2-System auch völlig ausreicht. Die Inbetriebnahme der hier vorgestellten neuen Generation sowie eine Erweiterung durch Anbindung eines MAC-Prozessors sind im Rahmen eines Folgeprojektes in Arbeit.

Für die Verarbeitung der Algorithmen ist eine hohe Effizienz nahe der theoretischen Grenze zu erwarten. Die Vektorprodukte werden ohne zusätzliche Zyklen mit genau einer komplexen MAC-Operation pro Taktyklus berechnet. Bei einer entsprechenden Partitionierung der Daten auf die Speicherbänke sind Wartezyklen aufgrund von Zugriffskonflikten im Speicher kaum zu erwarten. Auch die recht hohe Latenzzeit von 9 Zyklen in der MAC-Pipeline ist mit einem Befehlsinterleaving von $256/32 = 8$ unabhängigen Datensätzen pro Prozessorelement unkritisch.

Innerhalb der Blockdauer von 11,84 μs ermöglicht der Prozessor nach der STP100-Konfiguration 148 CMACs pro Unterträger. Nach den Abschätzung aus Tabelle 3.2

ist dies für ein 4×3-System völlig ausreichend, für das 8×7-System eher knapp bemessen. Die Portierung auf einen Virtex-4 oder einen ASIC würde mit einer Verdopplung der Taktrate auch hier ausreichend Reserve schaffen.

KAPITEL 7

Zusammenfassung und Ausblick

Die MIMO-Technologie ermöglicht eine drahtlose Datenübertragung mit einer hohen spektralen Effizienz. Für die wirtschaftliche Umsetzung der aufwändigen Signalverarbeitung in zukünftigen breitbandigen Funksystemen besteht ein Bedarf an effizienten Parallelprozessoren. Die Transformation in den Frequenzbereich schafft die Voraussetzung für eine effiziente Parallelisierung der Raum-Zeit-Signalverarbeitung in Verbindung mit einer flexiblen Programmierbarkeit. Während sich dieser Ansatz mit OFDM bereits in allen breitbandigen Systemen durchgesetzt hat, ist in Zukunft, z.B. zur Reduktion der Sendedynamik, auch eine Kombination mit einer Einträgermodulation denkbar. Unter der Voraussetzung einer zyklisch erweiterten Blockübertragung ist die Signalverarbeitung beider Modulationsverfahren selbst unter Einbeziehung von Spreizverfahren nahezu identisch. Über einen Overlap-Save-Ansatz kann auch eine kontinuierliche Einträgermodulation unterstützt und so ohne zyklische Erweiterung die Effizienz weiter gesteigert werden. Selbst wenn eine Zeitbereichsentzerrung mit einem vergleichbaren Rechenaufwand möglich ist, wird sie kaum so flexibel und effizient zu parallelisieren sein.

Der hier vorgeschlagene Parallelprozessor nutzt die Orthogonalität der Unterträger im Frequenzbereich für die parallele Verarbeitung auf einer größeren Anzahl identischer Prozessorelemente mit einer gemeinsamen Steuereinheit. Der Parallelisierungsgrad dieser SIMD-Architektur ist prinzipiell nur durch die minimale Anzahl unabhängiger Unterträgergruppen beschränkt. Die Optimierung von Rechenwerk, Speichereinheit und Steuereinheit auf die sequentielle Berechnung komplexwertiger Vektorprodukte ermöglicht die effiziente Ausführung aller betrachteten Algorithmen nahe der theoretischen Rechenkapazität.

Die Speicherschnittstelle, mit bis zu drei Lese- und einem Schreibzugriff pro Taktzyklus, liefert den erforderlichen Durchsatz für die beiden parallelen Pipelines des Rechenwerkes zur Berechnung einer komplexen MAC-Operation pro Zyklus sowie einer reelwertigen Kehrwertbildung alle acht Zyklen. Entscheidend zum Erreichen der hohen Effizienz ist die Beschränkung auf einen linearen Programmablauf in der Steuereinheit. Ein Anstieg der Codegröße, z.B. durch das erforderliche Ausrollen der Schleifendurchläufe, wird mit der Einführung von Vektorbefehlen zum Teil kompen-

siert. Im Gegensatz zu anderen VLIW-Prozessoren (z.B. TIs C6x) ermöglichen die Vektorbefehle mit einem in der Steuereinheit integrierten Latenzausgleich zudem eine sehr direkte und geradlinige Programmierung.

Der Anwendungsbereich des Prozessors umfasst trotz spezifischer Optimierung alle gängigen Verfahren zur linearen Mehrteilnehmerdetektion:

- Kanalschätzung auf Basis von Trainingssymbolen
- Interpolation geschätzter Kanalkoeffizienten oder der Gewichtsfaktoren
- Inversion oder Zerlegung der Systemmatrix
- Matrix-Vektorprodukte für die Entzerrung und Vorverzerrung der Daten
- Lösen von Gleichungssystemen über Vorwärts-/Rückwärtseinsetzen
- Nachführen der Kanalkoeffizienten oder der Gewichtsfaktoren über eine iterative Adaption mit LMS, RLS oder Kalman-Filter

Einige der Verfahren (z.B. Interpolation oder Adaption über die Frequenz) nutzen entgegen dem eigentlichen Prinzip der Unabhängigkeit gerade die Ähnlichkeit benachbarter Unterträger. Ein optional durchlaufener Ring-Bus am Ausgang der MAC-Pipeline ermöglicht die zyklische Rotation von Daten zwischen den Prozessorelementen ohne zusätzliche Beanspruchung von Taktzyklen.

Mit der Erweiterung der Rechenwerke um einen Entscheider wird auch eine Reihe der nichtlinearen Verfahren mit sukzessiver oder paralleler Entscheidungsrückführung auf dem Prozessor zu implementieren sein. Nicht vereinbar mit der Architektur sind dagegen datenabhängige Addressierungen (z.B. zur Optimierung der Detektionsreihenfolge), wesentliche Abweichungen von den komplexwertigen Vektorprodukten, (z.B. Codierung bei den Turbo-Verfahren) sowie die Einbindung der Fourier-Transformation, z.B. für die Einträgermodulation mit Entscheidungsrückführung.

Für eine anwendungsspezifische Dimensionierung der Wortbreiten des Prozessors muss ein praktisch relevanter Betriebsbereich definiert werden. In einem modernen Funksystem wird eine Linkadaption die frei wählbaren Übertragungsparameter, wie Teilnehmeranzahl, Modulationsstufe oder Spreizfaktor, den momentanen Kanaleigenschaften, im Wesentlichen der Matrixkondition und dem Störabstand, anpassen. Unter der Voraussetzung einer Optimierung nach dem MMSE-Kriterium sowie einer gegebenen Anforderung an die Übertragungsqualität beschränkt das verfügbare SNR den zulässigen Bereich für die Matrixkondition.

Am Beispiel der Matrixinversion mit einem SNR $\leq$ 20 dB für ein uncodiertes BER $\leq 10^{-2}$ erreicht eine Festkomma-Artithmetik ab einer Speicherwortbreite von 18 bit pro Quadraturkomponente die Referenz einer 64 bit-Fließkomma-Arithmetik. Der RLS-Algorithmus erfordert unter entsprechenden Bedingungen bereits eine

deutlich höhere Wortbreite. Die Einträgermodulation reagiert im Verglich zur Mehrträgermodulation grundsätzlich empfindlicher auf Überläufe und muss im hohen SNR-Bereich jenseits von 20 dB durch Hinzufügen künstlichen Rauschens stabil gehalten werden.

Unter identischen Bedingungen sind mit einer Fließkomma-Arithmetik insgesamt 34 bit für beide Quadraturkomponenten nötig. Erst im hohen SNR Bereich über 20 dB kann mit einer Fließkomma-Arithmetik die Wortbreite im Vergleich zur Festkomma-Arithmetik merklich reduziert werden. Dabei wird eine komplexwertige Zahlendarstellung angenommen, mit jeweils einer Mantisse für Real- und Imaginärteil und einem gemeinsamen Exponenten. Durch den Ausschluss von Überläufen in den Mantissen bleibt auch die Einträgermodulation ohne künstliches Rauschen stabil. Ansonsten liegt der Vorteil gegenüber der Festkomma-Arithmetik nur in der einfacheren Programmierung ohne Festlegung von Skalierungsfaktoren. Während sich der zusätzliche Ressourcenverbrauch durchaus in Grenzen hält, ist aufgrund eines kritischen Pfades im Akkumulator mit einer deutlichen Einschränkung der Taktrate zu rechnen.

Die vorgeschlagene Prozessorarchitektur wurde auf einer FPGA-Plattform umgesetzt und in einem leistungsfähigen MIMO-Demonstrator angewendet. Die erstellte VHDL-Beschreibung des Prozessors ist in der Anzahl der Prozessorelemente frei skalierbar. Endgültig auf die Hardware abgebildet wurden insgesamt drei Konfigurationen. Eine Vorstufe mit acht Prozessorelementen und eingeschränkter Funktionalität wird im Komplettsystem zur Übertragung mit zwei Sende- und zwei Empfangsantennen bereits betrieben. Die Integration des Parallelprozessors auf einem Virtex-II FPGA ist von den Ressourcen her mit 32 Prozessorelementen bei einer Taktrate von 200 MHz zwar möglich, aufgrund einer zu hohen Leistungsaufnahme und Verlustwärme aber nicht realisierbar. Die gleichmäßige Verteilung der 32 Prozessorelemente und vier FFT-Prozessoren auf zwei FPGAs hat mit einem 100 MHz-Takt schließlich zu einer realisierbaren Lösung geführt.

Mit 12,8 $^{\mathrm{GMAC}}/_{\mathrm{s}}$ für den Parallelprozessor und zusätzlichen 4,8 $^{\mathrm{GMAC}}/_{\mathrm{s}}$ der FFT-Prozessoren kann diese Konfiguration als derzeit leistungsfähigster MIMO-Prozessor bezeichnet werden. Nach der Aufwandsabschätzung der Algorithmen ermöglicht er eine hochadaptive lineare Raum-Zeit-Verarbeitung mit bis zu fünf parallelen Datenströmen. Aufgrund hoher Kosten und einer geschätzten Leistungsaufnahme von etwa 30 W ist die FPGA-Umsetzung zu einer Multi-Prozessor-Lösung mit 6 TigerSHARCs nur bedingt konkurenzfähig. Erst in einer ASIC-Umsetzung wird sich die hohe Effizienz der Architektur rechnen.

Für den Ausblick ist zunächst die tatsächliche Inbetriebnahme der erstellten Konfi-

guration und die Anwendung im Demonstrator zu nennen. Die prognostizierte hohe Effizienz in der Ausführung sollte in der Praxis mit einem Benchmarking typischer Verfahren nachgewiesen werden. Dabei sind auch Größe des Programmcodes sowie die Durchsatzanforderungen für den Zugriff auf den Programmspeicher von Interesse. In der Anwendung bietet die Umsetzung der Vorverzerrung mit einer leistungsfähigen Echtzeitverarbeitung sicherlich das größte Potential.

Die Erweiterung des Rechenwerkes um einen konfigurierbaren Entscheider würde mit einigen nichtlinearen Verfahren den Anwendungsbereich des Prozessors erweitern. Der Ausbau auf die hier vorgeschlagene Fließkomma-Arithmetik schließt mit den Überläufen potentielle Fehlerquellen aus und vereinfacht die Umsetzung neuer Anwendungen für den Demonstrator. Eine Portierung auf die Virtex-4 Architektur ermöglicht mit relativ geringem Entwicklungsaufwand eine Erhöhung der Rechenleistung über weitere Prozessorelemente und höhere Taktraten. Die Portierung der Architektur auf einen ASIC ist zwar sicherlich mit höherem Aufwand und Kosten verbunden, bietet aber zur Zeit die einzige Möglichkeit die hohe Rechenleistung wirtschaftlich mit einer vertretbaren Verlustleistung umzusetzen.

Abschließend ist festzustellen, dass die hier vorgeschlagene Prozessorarchitektur eine effiziente und flexible Umsetzung der Raum-Zeit-Signalverarbeitung für die breitbandige Funkkommunikation mit adaptiven Gruppenantennen ermöglicht. Die Anwendung in einem Demonstrator hat gezeigt, dass bereits heute eine Umsetzung auf einem oder zwei FPGAs möglich ist und die Übertragung hoher Datenraten von mehreren hundert Mbit pro Sekunde zulässt.

ANHANG A

Erweiterungen des RLS Algorithmus

A.1 Iterative Kahunen-Loéve-Transformation

Der PAST Algorithmus [63] (*projection approximation subspace tracking*) berechnet iterativ eine Näherung $\mathbf{T}_\mathrm{s}$ für eine Basis des Signalunterraumes $\mathbf{U}_\mathrm{s}$. Yang formuliert ein Optimierungsproblem, das sehr effizient mit den bekannten Iterationsalgorithmen (z.B. LMS oder RLS) gelöst werden kann. Der Fehler durch die Näherung ist sehr gering; die Spalten von $\mathbf{T}_\mathrm{s}$ sind annähernd orthonormal. Die Basis ist jedoch nicht eindeutig, $\mathbf{T}_\mathrm{s}$ und $\mathbf{U}_\mathrm{s}$ unterscheiden sich durch eine unbekannte orthononormale Matrix $\mathbf{Q}$:

$$\mathbf{T}_\mathrm{s} = \mathbf{U}_\mathrm{s}\mathbf{Q} \tag{A.1}$$

Die Matrix $\mathbf{T}_\mathrm{s}$ wird zur Transformation des Empfangsvektors genutzt um die Filterordnung von MQ auf K zu reduzieren. Das allgemeine Prinzip ist in der Literatur unter dem Begriff *transform domain* bekannt [64]. Die Verwendung des Signalunterraumes entspricht der Kahunen-Loéve-Transformation (KLT). Mit dem transformierten Eingangsvektor kann die eigentliche Entzerrung dann mit reduziertem Aufwand durchgeführt werden. Die Kombination der KLT mit einem RLS zur Entzerrung der transformierten Empfangsdaten und einem PAST-RLS zum Nachführen der Transformationsmatrix wird im Folgenden als KLT-RLS bezeichnet. Die Berechnungsvorschrift ist in Bild A.1 angegeben.

Durch die Transformation in (A.2) wird der RLS auf die Dimension K reduziert. Der PAST-RLS beruht auf dem selben transformierten Eingangsvektor $\mathbf{x}_\mathrm{t}$ wie der RLS. Der Kalman-Gain $\mathbf{k}_\mathrm{n}$ für (A.11) muss nicht neu berechnet werden. Der zusätzliche Aufwand für das Nachführen der Transformationsmatrix $\mathbf{T}_\mathrm{s}$ entspricht daher nur dem eines LMS der Dimension K mit MQ Kanälen.

Anders als bei anderen Anwendungen ist hier der Rang des Signalunterraumes mit K fest vorgegeben. Die Projektion ist dann verlustlos, d.h. das Produkt $\mathbf{W}_\mathrm{t}\mathbf{T}_\mathrm{s}^H$ konvergiert im statischen Fall exakt zur Wienerlösung. Im Optimum gilt:

KLT-RLS Algorithmus

- *Transformation des Empfangsvektors mit der KLT*

$$\mathbf{x}_{\mathrm{t}}^{(l)} = \mathbf{T}_{\mathrm{s}}^{(l-1)\,H}\mathbf{x}^{(l)} \tag{A.2}$$

- *RLS zur Entzerrung und Interferenzbefreiung*

$$\begin{aligned}
\hat{\mathbf{d}}^{(l)} &= \mathbf{W}_{\mathrm{t}}^{(l-1)}\mathbf{x}_{\mathrm{t}}^{(l)} && \text{(A.3)}\\
\mathbf{e}_{\mathrm{d}}^{(l)} &= \mathbf{d}^{(l)} - \hat{\mathbf{d}}^{(l)} && \text{(A.4)}\\
\mathbf{k}^{(l)} &= \mathbf{P}^{(l-1)}\mathbf{x}_{\mathrm{t}}^{(l)} && \text{(A.5)}\\
\mathbf{k}_{\mathrm{n}}^{(l)} &= \frac{\mathbf{k}^{(l)}}{\beta + \mathbf{x}_{\mathrm{t}}^{(l)\,H}\mathbf{k}^{(l)}} && \text{(A.6)}\\
\mathbf{P}^{(l)} &= \frac{1}{\beta}\left(\mathbf{P}^{(l-1)} - \mathbf{k}^{(l)}\mathbf{k}_{\mathrm{n}}^{(l)\,H}\right) && \text{(A.7)}\\
\mathbf{W}_{\mathrm{t}}^{(l)} &= \mathbf{W}_{\mathrm{t}}^{(l-1)} + \mathbf{e}_{\mathrm{d}}^{(l)}\mathbf{k}_{\mathrm{n}}^{(l)\,H} && \text{(A.8)}
\end{aligned}$$

- *PAST-RLS zum Nachführen der Transformationsmatrix*

$$\begin{aligned}
\hat{\mathbf{x}}^{(l)} &= \mathbf{T}_{\mathrm{s}}^{(l-1)}\mathbf{x}_{\mathrm{t}}^{(l)} && \text{(A.9)}\\
\mathbf{e}_{\mathrm{x}}^{(l)} &= \mathbf{x}^{(l)} - \hat{\mathbf{x}}^{(l)} && \text{(A.10)}\\
\mathbf{T}_{\mathrm{s}}^{(l)} &= \mathbf{T}_{\mathrm{s}}^{(l-1)} + \mathbf{e}_{\mathrm{x}}^{(l)}\mathbf{k}_{\mathrm{n}}^{(l)\,H} && \text{(A.11)}
\end{aligned}$$

Bild A.1: Berechnungsvorschrift für den KLT-RLS-Algorithmus

$$\mathbf{W}_{\mathrm{t}} = \mathbf{R}_{\mathrm{dd}}\mathbf{V}_{\mathrm{A}}\mathbf{\Sigma}_{\mathrm{As}}\mathbf{\Lambda}_{\mathrm{s}}^{-1}\mathbf{Q} \tag{A.12}$$

mit der Diagonalmatrix $\mathbf{\Sigma}_{\mathrm{As}}$ und der orthonormalen Matrix $\mathbf{V}_{\mathrm{A}}$, die aus den Singulärwerten bzw. den rechten Singulärvektoren der Systemmatrix $\mathbf{A}$ bestehen. Für unkorrelierte Datenströme hat $\mathbf{W}_{\mathrm{t}}$ orthogonale Spaltenvektoren. Eine Transformation mit den Eigenvektoren führt zu einer Dekorrelation der Empfangssignale. Die inverse Korrelationsmatrix in (A.7) wäre dann eine Diagonalmatrix. Diese Struktur wird aber durch die unbekannte Matrix $\mathbf{Q}$ zerstört.

A.2 Kalman-Filter

Für die lineare Entzerrung können die optimalen Gewichtskoeffizienten als Zustand in einem Zustandsraum angesehen werden. Die iterative Veränderung des Zustandes sowie dessen Beobachtung wird durch das folgende Zustandsraummodell beschrieben:

$$\begin{aligned} \mathbf{W}^{(l+1)^H} &= \mathbf{F}\,\mathbf{W}^{(l)^H} + \mathbf{N}_{\mathrm{W}}^{(l)} \\ \mathbf{d}^{(l)^H} &= \mathbf{x}_{\mathrm{e}}^{(l)^H}\mathbf{W}^{(l)^H} + \mathbf{e}_{\mathrm{o}}^{(l)^H} \end{aligned} \quad \text{mit} \quad \begin{aligned} E\left\{\mathbf{N}_{\mathrm{W}}\mathbf{N}_{\mathrm{W}}{}^H\right\} &= \mathbf{R}_{\mathrm{N_W}} \\ E\left\{\mathbf{e}_{\mathrm{o}}^H\mathbf{e}_{\mathrm{o}}\right\} &= \sigma_e \end{aligned} \tag{A.13}$$

Die Übergangsmatrix $\mathbf{F}$ definiert die Abhängigkeit eines Zustandes vom vorangegangenen Zustand. Nicht deterministische Zustandsänderungen werden über die additive Rauschmatrix $\mathbf{N}_{\mathrm{W}}^{(l)}$ modelliert. Die Gewichtung des erweiterten Empfangsvektors $\mathbf{x}_{\mathrm{e}}$ mit der Zustandsmatrix führt mit dem Schätzfehler $\mathbf{e}_{\mathrm{o}}$ zu dem gesendeten Symbolvektor $\mathbf{d}$. Der Symbolvektor ist damit eine durch den zufälligen Schätzfehler gestörte Beobachtung des gewünschten Zustandes. Das Modell soll den optimalen MMSE-Empfänger nachbilden. Der Erwartungswert σ_e des Schätzfehlers ist dann durch (3.20) bzw. (3.21) gegeben und damit abhängig vom momentanen Kanal sowie dem SNR. Die Darstellung über die Hermitesche dient der Beschreibung des Mehrnutzerfalls mit der in der Literatur üblichen Notation.

Das Kalman-Filter erlaubt nun die iterative Schätzung des Zustandes mit minimalem quadratischem Fehler. Der prinzipielle Aufbau des Kalman-Entzerrers ist in Bild A.2 dargestellt und wird dem üblichen Aufbau für die Systemidentifikation gegenübergestellt. Die Berechnungsvorschrift ist in Bild A.3 zu finden. Für den Parametersatz $\mathbf{F} = \mathbf{I}_{\mathrm{MQ}}$, $\mathbf{R}_{\mathrm{N_W}} = \mathbf{0}$, $\sigma_e = 1$ und $\mathbf{x}_{\mathrm{e}} = \mathbf{x}$ ist das Kalman-Filter identisch mit dem klassischen RLS. Mit $\mathbf{F} = \frac{1}{\sqrt{\beta}}\mathbf{I}_{\mathrm{MQ}}$ und einer zeitabhängigen Normierung von Zustand und Referenzsignal kann auch der RLS mit exponentiellem Vergessensfaktor nachgebildet werden [65]. Es ist klar ersichtlich, dass beide Ansätze von einem statischen Zustand ausgehen. Durch die Anpassung der Modellparameter an den zeitvarianten Kanal soll nun die Adaptionsgeschwindigkeit verbessert werden.

Die Veränderungen des Kanals sind stetig, d.h. es besteht eine hohe Abhängigkeit zwischen aufeinander folgenden Zuständen. Ein gängiger Ansatz ist die Verwendung eines linearen AR-Modells (AR, *auto regressive*). In [66] wird der Zustandsraum um die vergangenen Zustände erweitert, das AR-Modell mit der Übergangsmatrix realisiert und die AR-Koeffizienten dann adaptiv bestimmt. Um den Rechenaufwand gering zu halten wird im Folgenden ein AR-Modell zweiter Ordnung mit festen Koeffizienten für eine lineare Prädiktion angesetzt. Für die Parameter des Zustandsraummodells (A.13) gilt dann

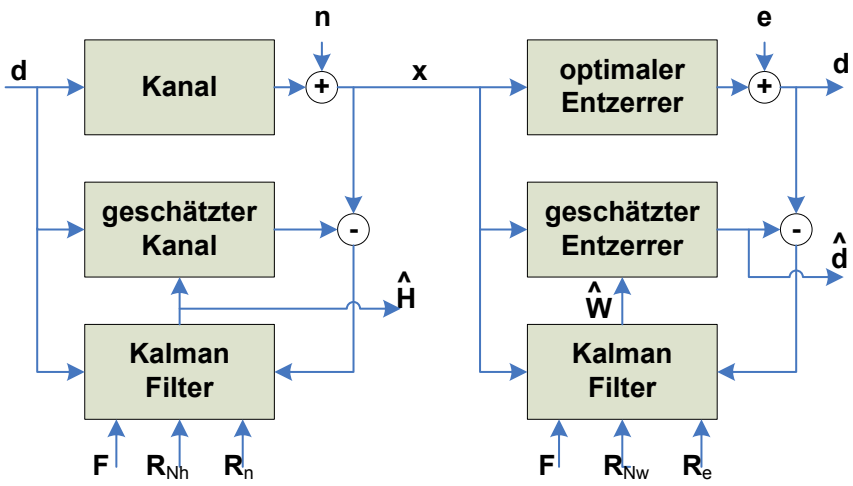


Bild A.2: Das Kalman-Filter für die Systemidentifikation (links) und für die Entzerrung (rechts)

Kalman Filter:

$$\hat{\mathbf{d}}^{(l)} = \mathbf{W}_{\text{Kalman}}^{(l)} \mathbf{x}_{\text{e}}^{(l)} \tag{A.14}$$

$$\mathbf{e}^{(l)} = \mathbf{d}^{(l)} - \hat{\mathbf{d}}^{(l)} \tag{A.15}$$

$$\mathbf{k}^{(l)} = \mathbf{P}^{(l)} \mathbf{x}_{\text{e}}^{(l)} \tag{A.16}$$

$$\mathbf{k}_{\text{n}}^{(l)} = \frac{\mathbf{k}^{(l)}}{\sigma_e + \mathbf{x}_{\text{e}}^{(l)H} \mathbf{k}^{(l)}} \tag{A.17}$$

$$\mathbf{P}^{(l+1)} = \mathbf{F} \left(\mathbf{P}^{(l)} - \mathbf{k}^{(l)} \mathbf{k}_{\text{n}}^{(l)H} \right) \mathbf{F}^H + \mathbf{R}_{\text{N}_\text{W}} \tag{A.18}$$

$$\mathbf{W}_{\text{Kalman}}^{(l+1)} = \left(\mathbf{W}_{\text{Kalman}}^{(l)} + \mathbf{e}^{(l)} \mathbf{k}_{\text{n}}^{(l)H} \right) \mathbf{F}^H \tag{A.19}$$

Bild A.3: Iterative Berechnungsvorschrift für das Kalman-Filter

$$\mathbf{F} = \begin{bmatrix} 2\mathbf{I}_{\text{MQ}} & -\mathbf{I}_{\text{MQ}} \\ \mathbf{I}_{\text{MQ}} & \mathbf{0} \end{bmatrix}, \quad \mathbf{R}_{\text{N}_\text{W}} = \begin{bmatrix} \sigma_w \mathbf{I}_{\text{MQ}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \quad \mathbf{x}_{\text{e}}^{(l)} = \begin{bmatrix} \mathbf{x}^{(l)} \\ \mathbf{0} \end{bmatrix} \tag{A.20}$$

Anders als beim RLS wird hier die Schätzung nicht a priori sondern mit dem prädiktierten Zustand für den aktuellen Zeitpunkt durchgeführt. Das Kalman-Filter kann auch für den PAST zum Nachführen der KLT eingesetzt werden.

ANHANG B

Befehlswortdefinition für den Parallelprozessor

Bit	Basic Vector Operation (inner product)	Extension Word (other products or reciprocal)	Extension (default)
63	opcode = 0	opcode = 1	
62	*free for further opcodes*	*free for further opcodes*	
61	vector length lsbs	vector length msbs (for length extension)	0000
60			
59			
58			
57	memory bank selection coefficient A (X:00, Y:01, Z:10, no read:11)	goto next program address	0
56		sleep	0
55	memory bank selection coefficient B (X:0, Y:1)	MAC result rotation (no rotation:0, forward rotation: 1)	0
54	memory bank selection MAC result (X:00, Y:01, Z:10, no write:11)	memory bank selection reciprocal (X:00, Y:01, Z:10, no reciprocal:11)	11
53			
52	clear/keep coefficient B		
51	clear/keep coefficient C		
50	clear(0) or keep(1) selection		
49	sign product A*B (+:0, -:1)		
48	sign coefficient C (+0, -:1)		
47	conjugate coefficient A		
46	conjugate coefficient B		
45	conjugate coefficient C		
44	scaling LUT addr for MAC unit	scaling LUT addr for reciprocal	
43			
42			
41			
40			

Bit	Basic Vector Operation (inner product)	Extension Word (other products or reciprocal)	Extension (default)
39	source address coefficient A	jump decrement coefficient A	0
38		address jump width coefficient A	1
37			
36			
35			
34			
33			
32			
31			
30			
29	source address coefficient B	jump decrement coefficient B	0
28		address jump width coefficient B	1
27			
26			
25			
24			
23			
22			
21			
20			
19	source address coefficient C	jump decrement coefficient C	0
18		address jump width coefficient C	0
17			
16			
15			
14			
13			
12			
11			
10			
9	destination address MAC result	jump decrement result address	0
8		address jump width MAC result or destination address reciprocal result (including jump bit)	0
7			
6			
5			
4			
3			
2			
1			
0			

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

SYMBOLVERZEICHNIS

Symbol	Bedeutung
α	Dämpfung pro Chipdauer in Dezibel für exponentiell abfallende Leistungsverteilung der Impulsantwort
α_V	Normierungsfaktor für die Vorverzerrung nach dem PCMMSE-Kriterium
$\mathbf{a}$	Übertragungsvektor mit Kanal und Spreizung für eine Unterträgergruppe
A	Koeffizient für das Produkt in der MAC-Pipeline
$\mathbf{A}$	Übertragungsmatrix mit Kanal und Spreizung für eine Unterträgergruppe
$\mathbf{A}_{\mathrm{MMSE}}$	Schätzung der Übertragungsmatrix nach dem MMSE-Kriterium
β	Vergessensfaktor für den RLS-Algorithmus
$\beta_{d,max}$	Phasenmass für die maximale, auf die Blockrate bezogene Dopplerverschiebung
B	Koeffizient für das Produkt in der MAC-Pipeline
$\mathfrak{c}(n)$	Spreizcode der Länge Q
c	Lichtgeschwindigkeit
$\mathbf{c}$	Codevektor
C	Summand für die Addition mit dem Produkt innerhalb der MAC-Pipeline
$\mathbf{C}_{\mathrm{corr}}$	Koeffizientenmatrix zur Korrelation des zufälligen Kanalvektors
$\mathbf{c}_{\mathrm{fd}}$	Codevektor im Frequenzbereich für Frequenzbereichsspreizung
$\mathbf{C}_{\mathrm{fd}}$	Codematrix einer Unterträgergruppe für Frequenzbereichsspreizung
$\mathbf{c}_{\mathrm{td}}$	Codevektor im Frequenzbereich für Zeitbereichsspreizung
$\mathbf{C}_{\mathrm{td}}$	Codematrix einer Unterträgergruppe für Zeitbereichsspreizung
$\mathfrak{d}_c(n)$	Folge mit $n = 1 \ldots N$ zu sendenden Chips
$\mathfrak{d}_s(u)$	Folge mit $u = 1 \ldots U$ zu sendenden Symbolen
$\mathfrak{d}_{\ddot{u}}^{(t)}(n_{\ddot{u}})$	Überabgetastete Chipfolge $\mathfrak{d}_c(n)$
D	Resultat der MAC-Pipeline
$\mathbf{\Delta}$	Symmetrische Matrix für die Interferenzreduktion nach der speziellen Wienerlösung in (3.11)
$\mathbf{d}$	Datenvektor für eine Unterträgergruppe
$\hat{\mathbf{d}}$	Schätzwert des Symbolvektors
$\mathbf{d}_{\mathrm{c}}$	Chipvektor mit den N zu sendenden Chips
$\mathbf{d}_{\mathrm{s}}$	Frequenzbereichssymbolvektor
$\mathbf{d}_{\ddot{\mathrm{u}}}$	Überabgetasteter Chipvektor $\mathbf{d}_{\mathrm{c}}$
$\mathbf{D}$	Diagonalmatrix der LDL-Zerlegung
ϵ	Quotient aus der mittleren Rauschleistung pro Unterträger zur mittleren Leistung der Sendesymbole

Symbol	Bedeutung
E	Exponent zur Basis 2 in der Fest- und Fließkommadarstellung
E_a, E_b, E_c, E_{mac}	Exponent der Koeffizenten A, B und C sowie des Ergebnisses der MAC-Pipeline
$\mathbf{e}$	Fehlervektor
f_b	Blockrate
f_{chip}	Chiprate
$f_{d,max}$	maximale Dopplerfrequenz
f_s	Abtastrate in Sender und Empfänger
f_t	Trägerfrequenz
$\mathbf{F}$	Übergangsmatrix im Zustandsraummodell
γ_k	Korrelation benachbarter Antennen an den Teilnehmerstationen
γ_m	Korrelation benachbarter Antennen am Zugangspunkt
$\mathfrak{g}(n_{\ddot{u}})$	Impulsantwort von Sende- und Empfangsfilter
$\mathbf{g}$	Übertragungsfunktion von Pulsform- und Empfangsfilter
$\mathbf{G}$	Cholesky-Matrix
$\mathfrak{h}_t(n_{\ddot{u}})$	Impulsantwort des zeitdiskreten äquivalenten Tiefpasssystems des Übertragungskanals
$\mathbf{h}_{\mathrm{corr}}$	korrelierter Zufallsvektor mit Übertragungsfaktoren des stochastischen Kanalmodells
$\mathbf{h}_{\mathrm{eff}}$	Vektor mit effektiven Übertragungsfaktoren aller Unterträger
$\mathbf{H}_{\mathrm{eff}}$	Kanalmatrix mit effektiven Übertragungsfaktoren einer Unterträgergruppe
$\mathfrak{h}_{iid}$	Zufallsvariable für Kanalimpulsantwort mit N voneinander unabhängigen, komplex-normalverteilten Taps
$\mathbf{h}_{\mathrm{iid}}$	Zufallsvariable für Übertragungsfunktion mit N voneinander unabhängigen, komplex-normalverteilten Koeffizienten
$\mathbf{h}_{\mathrm{t}}$	Übertragungsfunktion des äquivalenten Tiefpasssystems des Kanals
$\mathbf{I}_{\mathrm{K}}$	Einheitsmatrix der Dimension K
κ_2	Matrixkondition in der 2-Norm
$\mathbf{k}$	Dekorrelierter Empfangsvektor
$\mathbf{k}_{\mathrm{n}}$	Kalman-Gain
K	Anzahl der Antennen an den Teilnehmerstationen
λ_{R}, λ_{Δ}	Eigenwerte von $\mathbf{R}_{\mathrm{xx}}$ bzw. $\mathbf{\Delta}$
$\mathbf{\Lambda}_{\mathrm{R}}$, $\mathbf{\Lambda}_{\Delta}$	Diagonalmatrizen mit den Eigenwerten von $\mathbf{R}_{\mathrm{xx}}$ bzw. $\mathbf{\Delta}$
L	Länge der zyklischen Erweiterung
L_i	Länge des Interpolationsfilters
$\mathbf{L}$	Untere Dreiecksmatrix mit zu eins normierter Hauptdiagonalen (aus der LDL-Zerlegung)
μ	Schrittweite für den LMS-Algorithmus
M	Anzahl der Antennen am Zugangspunkt
n	Normierungsfaktor zur Berechnung des Kalman-Gains
$\mathbf{n}$	Rauschvektor

Symbol	Bedeutung
$\tilde{\mathbf{n}}$	Mit der Spreizmatrix gewichteter Rauschvektor am Empfänger der Teilnehmerstationen
N	Anzahl an Unterträgern (entspricht Anzahl an Chips pro Datenblock)
N_i	Anzahl an Stützstellen für eine Interpolation über die Frequenz
$\mathbf{N}_{\mathrm{W}}$	Additive Rauschmatrix für zufällige Änderungen im Zustandsraummodell
$\mathfrak{p}_e(n_{\ddot{u}})$	Folge mit den Abtastwerten von dem A/D-Wandler im Empfänger
$\mathfrak{p}_s(n_{\ddot{u}})$	Folge mit den Abtastwerten für den D/A-Wandler im Sender
P	Anzahl paralleler Prozessorelemente
P_d	Anzahl der Prozessorelemente, die einen gemeinsamen Dividierer nutzen
P_v	Verlustleistung
$\mathbf{P}$	Schätzwert der inversen deterministischen Korrelationsmatrix des Eingangsvektors
Q	Spreizfaktor
$\mathbf{Q}$	Orthonormale Matrix für die Abbildung zwischen Signalunterraum und der Näherung durch den PAST-Algorithmus
r	roll-off Faktor für raised-cosine Pulsformfilter
R	Dynamikreserve bezogen auf die Festkomma-Darstellung der Zahl '1'
r_i	Interpolationsfaktor
$\mathbf{R}_{\mathrm{dd}}$	Autokorrelationsmatrix des Sendevektors
$\mathbf{R}_{\mathrm{dx}}$	Korrelationsmatrix zwischen Sende- und Empfangsvektor
$\mathbf{R}_{\mathrm{nn}}$	Autokorrelationsmatrix des Rauschvektors
$\mathbf{R}_{\mathrm{xx}}$	Autokorrelationsmatrix des Empfangsvektors
$\mathbf{R}_{\mathrm{hh}}$	Korrelationsmatrix des Zufallsvektors $\mathbf{h}_{\mathrm{corr}}$
$\mathbf{R}_{\mathrm{k}}$	räumliche Korrelationsmatrix an den Teilnehmerstationen
$\mathbf{R}_{\mathrm{m}}$	räumliche Korrelationsmatrix am Zugangspunkt
$\mathbf{R}_{\mathrm{n}}$	Frequenzkorrelationsmatrix
$\mathbf{R}_{\mathrm{N_W}}$	Autokorrelationsmatrix der zufälligen Veränderungen im Zustandsraummodell
$\mathbf{R}_{\mathrm{t}}$	Zeitkorrelationsmatrix
σ_a	Singulärwerte der Systemmatrix
σ_d	mittlere Leistung der Sendesymbole
σ_e	mittlere teilnehmerspezifische Empfangsleistung pro Antenne und Unterträger
σ_n	mittlere Rauschleistung pro Empfangsantenne und pro Unterträger
Σ_A	Diagonalmatrix mit den Singulärwerten der Systemmatrix
$\mathfrak{s}(n)$	Chipfolge $\mathfrak{d}_c(n)$ mit zyklischer Erweiterung
$\mathfrak{s}_{\ddot{u}}(n_{\ddot{u}})$	Überabtastung und Aneinanderreihung aller Sendefolgen $\mathfrak{s}^{(t)}(n)$
S	Vorzeichenbehafteter, ganzzahliger Skalierungsfaktor

Symbol	Bedeutung
S_{acc}	Fester Skalierungsfaktor am Ausgang des Akkumulators in der MAC-Pipeline
S_{div}	Variabler Faktor zur Skalierung der Kehrwerte
S_{cmul}	Variabler Faktor zur Skalierung des komplexen Produktes in der MAC-Pipeline
S_{mul}	Fester Skalierungsfaktor im Anschluss an die Multiplikation in der MAC-Pipeline
Θ_g	thermischer Widerstand des Chipgehäuses
Θ_k	thermischer Widerstand des Kühlkörpers inkl. Klebemittel
T	Anzahl an Datenblöcken pro Zeitschlitz
T_b	Blockdauer
T_{chip}	Chipdauer
T_j	Kerntemperatur (*junction temperature*)
T_s	Taktperiode der Abtastung in Sender und Empfänger
$\mathbf{T}_\mathrm{S}$	Näherung für die Basis des Signalunterraumes durch den PAST-Algorithmus
T_u	Umgebungstemperatur
$\ddot{U}$	Ganzzahliger Faktor für Überabtastung in Sender und Empfänger
U	Anzahl an Unterträgergruppen (entspricht Anzahl übertragener Symbole pro Datenblock)
$\mathbf{U}_\mathrm{A}$, $\mathbf{U}_\mathrm{R}$, $\mathbf{U}_\Delta$	linke Singulärvektoren der Matrizen $\mathbf{A}$, $\mathbf{R}_{\mathrm{xx}}$, $\mathbf{\Delta}$
$\mathbf{U}_\mathrm{An}$	Rauschunterraum der Systemmatrix
$\mathbf{U}_\mathrm{As}$	Signalunterraum der Systemmatrix
v	Geschwindigkeit mobiler Teilnehmer nach dem Jakes-Modell
$\mathbf{V}$	Gewichtsmatrix für die lineare Vorverzerrung
$\mathbf{V}_\mathrm{A}$	rechte Singulärvektoren der Systemmatrix
W	Wortbreite der Festkommadarstellung
W_{add}, W_{acc}, W_{div}	Interne Wortbreiten der Festkomma-Arithmetik für die Addierer, den Akkumulator und den Dividierer
W_{exp}	Wortbreite des Exponenten in der Fließkomma-Arithmetik
W_{man}	Wortbreite der Mantissen in der Fließkomma-Arithmetik
W_{mem}	Wortbreite des lokalen Speichers (pro Quadraturkomponente in der Festkomma-Arithmetik und gesamt in der Fließkomma-Arithmetik)
W_{mq}	Zusätzliche Bitstellen für den Akkumulator bei maximaler Vektorlänge
W_f	Filterlänge des Pulsformfilters
W_h	Länge der Kanalimpulsantwort
$\mathbf{W}$	Gewichtsmatrix für die lineare Entzerrung einer Unterträgergruppe
$\mathbf{W}_\mathrm{LMS}$	Aktueller Gewichtsvektor bei der Adaption mit dem LMS-Algorithmus
$\mathbf{W}_\mathrm{MMSE}$	Gewichtsmatrix zur Minimierung des mittleren quadratischen Fehlers

Symbol	**Bedeutung**
$\mathbf{W}_{\mathrm{RLS}}$	Aktueller Gewichtsvektor bei der Adaption mit dem RLS-Algorithmus
$\mathbf{W}_{\mathrm{t}}$	Gewichtsmatrix zur Entzerrung des transformierten Empfangsvektors
$\mathfrak{x}(n)$	Auf Chiprate abgetastete Empfangsfolge eines Datensymbols
$\mathfrak{x}_{\ddot{u}}(n_{\ddot{u}})$	Überabgetastetes Empfangssignal nach der Filterung
$\mathbf{x}$	Empfangsvektor einer Unterträgergruppe mit QM Elementen
$\mathbf{x}_{\mathrm{t}}$	Transformierter Empfangsvektor, hier durch die Kahunen-Loéve-Transformation
$\tilde{\mathbf{x}}$	Mit dem *matched filter* gewichteter Empfangsvektor
$\mathbf{x}_{\ddot{u}}$	Überabgetasteter Empfangsvektor nach der Filterung
$\mathbf{y}$	Lösungsvektor nach dem Vorwärtseinsetzen
$\mathbf{Z}$	Wiederholungsmatrix

ABKÜRZUNGSVERZEICHNIS

Abkürzung	Bedeutung
ADC	Analog to Digital Converter
API	Application Programming Interface
AR	Auto Regressive
ARQ	Automatic Repeat Request
ASIC	Application Specific Integrated Circuit
ADI	Analog Devices, Inc.
AWGN	Additive White Gaussian Noise
BER	Bit Error Ratio
BLE	Block Linear Equalizer
BMBF	Bundesministerium für Bildung und Forschung
BPSK	Binary Phase Shift Keying
BRAM	Block Random Access Memory
CDMA	Code Division Multiple Access
CLK	Clock
CMAC	Complex Multiply Accumulate
COFDM	Coded Orthogonal Frequency Division Duplex
cPCI	compact Peripheral Component Interconnect
CPU	Central Processing Unit
DAB	Digital Audio Broadcast
DAC	Digital to Analog Converter
DDR	Double Data Rate
DFF	D-Flip-Flop
DIV	Division
DSL	Digital Subscriber Line
DSP	Digital Signal Processor
DVBT	Digital Video Broadcast Terrestrial
FDD	Frequency Division Duplex
FFT	Fast Fourier Transform
FIFO	First In, First Out
FIR	Finite Impulse Response
FLOPS	FLoating point Operations Per Second
FPGA	Field Programmable Gate Array
HDL	Hardware Description Language
HyEff	Hyper Efficiency
IEEE	Institute of Electrical and Electronics Engineers
IFFT	Inverse Fast Fourier Transform
IID	Independently Identically Distributed
IIR	Infinite Impulse Response
IP	Intellectual Property

Abkürzung	**Bedeutung**
IQ	In-phase/Quadrature-phase
IS95	Interim Standard-95
ISE	Integrated Software Environment
KLT	Kahunen Loéve Transformation
LMS	Least Mean Square
LSB	Least Significant Bit
LUT	Look-Up-Table
MAC	Multiply Accumulate (auch Medium Access Controller)
MC-CDMA	Multi-Carrier Code Division Multiple Access
MF	Matched Filter
MIMO	Multiple Input, Multiple Output
MIPS	Million Instruction Per Second
MMSE	Minimum Mean Square Error
MSB	Most Significant Bit
MSPS	Mega Samples Per Second
MUL	Multiplier
OFDM	Orthogonal Frequency Division Duplex
PAPR	Peak to Average Power Ratio
PAR	Place and Route
PAST	Projection Approximation Subspace Tracking
PCMMSE	Power Constrained Minimum Mean Square Error
PE	Prozessor-Element
PLL	Phase Locked Loop
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QoS	Quality of Service
QPSK	Quadrature Phase Shift Keying
RLS	Recursive Lest Squares
ROM	Read Only Memory
RX	Receiver
SDMA	Space Division Multiple Access
SIMD	Single Instruction, Multiple Data
SNR	Signal to Noise Ratio
SRAM	Synchronous Random Access Memory
STP	Space Time Processor
TDD	Time Division Duplex
TDMA	Time Division Multiple Access
TI	Texas Instruments
TX	Transmitter
UMTS	Universal Mobile Telecommunications System
UTRA	UMTS Terrestrial Radio Access
VCO	Voltage Controlled Oscillator
VBLAST	Vertical Bell-labs Layered Space Time
VHDL	VLSI Hardware Description Language

Abkürzung	**Bedeutung**
VLIW	Very Large Instruction Words
VLSI	Very Large Scale Integration
W-CDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Network
WIMAX	Worldwide Microwave Interoperability Forum
WSSUS	Wide Sense Stationary Uncorrelated Scattering
ZBT	Zero Bus Turnaround
ZF	Zero Forcing

LITERATURVERZEICHNIS

[1] J. Kennedy. Digital Music Report 2005. Technical report, International Federation of the Phonographic Industry (IFPI), 2005.

[2] IEEE. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications Amendment 4: Further Higher Data Rate Extension in the 2.4 GHz. Technical report, IEEE Std 802.11g-2003, 2003.

[3] IEEE. IEEE Standard for Local and Metropolitan Area Networks–Part 16: Air Interface for Fixed Broadband Wireless Access Systems. Technical report, IEEE Std 802.16-2004, 2004.

[4] G.E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), Apr 1965.

[5] G.J. Foschini and M.J. Gans. On limits of wireless communications in a fading environment when using multiple antennas. *Wireless Personal Communications*, 6(3):311–335, March 1998.

[6] L. Zheng and D.N.C. Tse. Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels. *IEEE Transactions on Information Theory*, 49(5):1073–1096, May 2003.

[7] C. M. Walke and B. Rembold. Joint Detection and Joint Pre-Distortion Techniques for SD/TD/CDMA Systems. *Frequenz - Zeitschrift für Telekommunikation*, 55:204–213, July 2001.

[8] V. Tarokh, N. Seshadri, and A. R. Calderbank. Space-time codes for high data rate wireless communication: Performance criterion and code construction. *IEEE Transactions on Information Theory*, 44:744–765, March 1998.

[9] E. Telatar. Capacity of Multi-antenna Gaussian Channels. *European Transactions on Telecommunications*, 10(6):585–595, November-December 1999.

[10] E. Jorswieck, A. Sezgin, H. Boche, and E. Costa. Optimal transmit strategies in MIMO Ricean Channels with MMSE Receiver. In *Proc. IEEE Vehicular Techn. Conf. (VTC 2004) fall*, Los Angeles, USA, September 2004.

[11] M. Schubert and H. Boche. Solution of the Multi-User Downlink Beamforming Problem with Individual SINR Constraints. *IEEE Trans. on Vehicular Techn.*, 53(1):18–28, January 2004.

[12] D. Gesbert, M. Shafi, D. Shiu, and P. Smith. From theory to practice: An overview of space-time coded MIMO wireless systems. *IEEE Journal on Selected Areas in Communications*, 21(3):281–302, April 2003.

[13] Working Group 8. White paper series: Space-time signal processing. Technical report, BMBF, Förderschwerpunkt Mobile Internet, 2004.

[14] W. Keusgen and C. M. Walke. A System Model Considering the Influence of Front-End Imperfections on the Reciprocity of Up- and Downlink System Impulse Responses. In *ASST 2001*, pages 243–248, Aachen, Germany, September 2001.

[15] L. Brühl, C. Degen, W. Keusgen, B. Rembold, and C. M. Walke. Investigation of Front-End Requirements for MIMO-Systems Using Downlink Pre-Distortion. In *5th European Personal Mobile Communications Conference*, Glasgow, Scotland, April 2003.

[16] ETSI EN 300 744. Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television. Technical report, European Standard (Telecommunications series), Valbonne, France, July 1999.

[17] ETSI ETS 300 401. Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers. Technical report, European Standard (Telecommunications series), Valbonne, France, Feb 1995.

[18] D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson. Frequency Domain Equalization for Single-Carrier Broadband Wireless Systems. *IEEE Communications Magazine*, 40(4):58–66, April 2002.

[19] 3GPP TSG-RAN WG1. Feasibility Study for OFDM for UTRAN enhancement. Technical Report TR 25.892 v. 1.1.0, 3GPP, March 2004.

[20] T. Ojanpera and R. Prasad. *Wideband CDMA For Third Generation Mobile Communications.* Artech House Publishers, 1998.

[21] S. Hara and R. Prasad. Overview of Multicarrier CDMA. *IEEE Communications Magazine*, 35(12):126–133, December 1997.

[22] C. Ibars and Y Bar-Ness. The principle of time-frequency duality of DS-CDMA and MC-CDMA. In *Conference on Information Sciences and Systems*, Princeton University, March 2002.

[23] L. Brühl and B. Rembold. Unified Spatio-Temporal Frequency Domain Equalization for Multi- and Single-Carrier CDMA Systems. In *Proceedings of the IEEE Vehicular Technology Conference 2002 Fall*, Vancouver, Canada, Sep. 2002.

[24] S. Verdu. *Multiuser Detection.* Cambridge University Press, Cambridge, UK, 1998.

[25] A. Klein. *Multi-user detection of CDMA signals - algorithms and their application to cellular mobile radio.* Number 423 in Fortschritt-Berichte VDI 10. VDI-Verlag GmbH, Düsseldorf, Germany, 1996.

[26] M. Vollmer, M. Haardt, and J. Götze. Comparative Study of Joint-Detection Techniques for TD-CDMA Based Mobile Radio Systems. *IEEE Journal on Selected Areas in Communications*, 19(8):1461–1475, August 2001.

[27] S. Haykin. *Adaptive Filter Theory.* Prentice-Hall, New Jersey, third edition, 1996.

[28] A. Bury. *Efficient multi-carrier spread spectrum transmission.* Number 685 in Fortschritt-Berichte VDI, Reihe 10. VDI Verlag, Düsseldorf, 2001.

[29] J. Egle and J. Lindner. Iterative joint equalization and decoding based on soft cholesky equalization for general complex valued modulation symbols. In *Proc. 6th International Symposium on DSP for Communication Systems*, pages 163–170, Sydney-Manley/Australia, January 2002.

[30] A. Engelhart, W. G. Teich, J. Lindner, G. Jeney, S. Imre, and L. Pap. A survey of multiuser/multisubchannel detection schemes based on recurrent neural networks. *Wireless Communications and Mobile Computing*, 2(3):269–284, May 2002.

[31] P.W. Wolniansky, G.J. Foschini, G.D. Golden, and R.A. Valenzuela. V-blast: An architecture for realizing very high data rates over the rich-scattering wireless channel. In *Proc.ISSSE-98*, Pisa, Italy, Sep 1998.

[32] D. Wübben, R. Böhnke, J. Rinas, V. Kühn, and K.D. Kammeyer. Efficient algorithm for decoding layered space-time codes. *IEE Electronic Letters*, 37(22):1348–1350, Oct 2001.

[33] D. Wübben, R. Böhnke, V. Kühn, and K.D. Kammeyer. MMSE Extension of V-BLAST based on Sorted QR Decomposition. In *IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall)*, Orlando, Florida, USA, Oct 2003.

[34] D. Pham et. al. The design and implementation of a first-generation cell processor. In *International Solid-State Circuits Conference Technical Digest*, Feb 2005.

[35] D. Bartolomé, A. Pascual-Iserte, A.I. Pérez-Neira, and P. Rosson. From a Theoretical Framework to a Feasible Hardware Implementation of Antenna Array Algorithms for WLAN. In *IST Mobile Communications Summit*, Aveiro, Portugal, June 2003.

[36] I.P. Seskar and N.B. Mandayam. A Software Radio Architecture for Linear Multiuser Detection. *IEEE Journal on Selected Areas in Communications*, 17(5):814–823, May 1999.

[37] V. Jungnickel, T. Haustein, and A. Forck. Real-Time Concepts for MIMO-OFDM. In *CIC IEEE Global Mobile Congress*, Shanghai, China, Oct 2004.

[38] S. Rajagopal, S. Bhashyam, J.R. Cavallaro, and B. Aazhang. Real-Time Algorithms and Architectures for Multiuser Channel Estimation and Detection in Wireless Base-Station Receivers. *IEEE Transactions on Wireless Communications*, 1(3):468–479, July 2002.

[39] S. Rajagopal, S. Bhashyam, J.R. Cavallaro, and B. Aazhang. Efficient VLSI Architectures for Multiuser Channel Estimation in Wireless Base-Station Receivers. *Journal of VLSI Signal Processing*, 31(2):143–156, June 2002.

[40] Z. Guo and P. Nilsson. An ASIC Implementation for V-BLAST Detection in $0.35\mu m$ CMOS. In *IEEE International Symposium on Signal Processing and Information Technology*, Rome, Italy, Dec 2004.

[41] U. Girola, A. Picciriello, and D. Vincenzoni. Smart Antenna Receiver Based on a Single Chip Solution for GSM/DCS Baseband Processing. In *Automation and Test in Europe (DATE 2000)*, Paris, France, March 2000.

[42] R. van Nee and R. Prasad. *OFDM for Wireless Multimedia Communications*. Universal Personal Communications. Artech House, Boston and London, 2000.

[43] H.D. Lüke. *Signalübertragung*. Springer-Verlag, Berlin Heidelberg, 6. edition, 1995.

[44] C. Degen and L. Brühl. Multi-user detection in OFDM systems using CDMA and multiple antennas. *European Transactions on Telecommunications*, 14(6):481–490, Nov/Dec 2003.

[45] C.M. Walke. *On the Spectral Efficiency of Interference-limited Mobile Radio Systems with Space/Time/Code Division Multiple Access*. PhD thesis, RWTH Aachen University, October 2002. Shaker Verlag.

[46] K. Yu and B. Ottersten. Models for MIMO propagation channels, a review. *Special Issue an Adaptive Antennas and MIMO Systems, Wiley Journal on Wireless Communications and Mobile Computing*, 2(7):553–666, Nov 2002.

[47] J.P. Kermoal, L. Schumacher, K.I. Pedersen, P.E. Mogensen, and F. Frederiksen. A stochastic MIMO radio channel model with experimental validation. *IEEE Journal on Selected Areas in Communications*, 20(6):1211–1226, Aug 2002.

[48] W. C. Jakes. *Microwave Mobile Communications*. John Wiley & Sons, New York, 1974.

[49] A. D. Whalen. *Detection of Signals in Noise*. Academic Press, Inc., New York, 1971.

[50] B. Steiner and P. Jung. Optimum and Suboptimum Channel Estimation for the Uplink of CDMA Mobile Radio Systems with Joint Detection. In *European Transactions on Telecommunications and Related Technologies*, volume 5, pages 39–50, January 1994.

[51] T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice Hall, 1st edition, March 2000.

[52] A. Carini and E. Mumolo. Fast square-root RLS adaptive filtering algorithms. *Signal Processing*, 57(3):233–250, March 1997. Publisher: Elsevier.

[53] R. Merched and A.H. Sayed. Extended Fast Fixed Order RLS Adaptive Filters. *IEEE Transactions on Signal Processing*, 49(12):3015–3031, Dec 2001.

[54] G. Moschytz and M. Hofbauer. *Adaptive Filter*. Springer-Verlag, 2000.

[55] M. Joham, K. Kusume, M.H. Gzara, W. Utschick, and J.A. Nossek. Transmit Wiener Filter for the Downlink of TDD DS-CDMA Systems. In *IEEE 7th Int. Symp. on Spread-Spectrum Tech. & Appli.*, Prague, Czech Republic, September 2002.

[56] B. Heyne, J. Götze, and M. Bücker. Implementation of a Cordic Based FFT on a Reconfigurable Hardware Accelerator. In *3rd Karlsruhe Workshop on Software Radios*, Karlsruhe, Germany, March 2004.

[57] J.S. Lee and M.H. Sunwoo. Design of New DSP Instructions and Their Hardware Architecture for High-Speed FFT. *Journal of VLSI Signal Processing*, 33:247–254, 2003.

[58] M.J. Flynn. Some Computer Organizations and Their Effectiveness. *IEEE Transactions on Computing*, C-21(9):948–960, 1972.

[59] H.T. Kung and C.E. Leiserson. Systolic arrays (for VLSI). In *Sparse Matrix Proc. 1978, Society for Industrial and Applied Mathematics*, pages 256–282, Philadelphia, 1979.

[60] C. T. Djamegni. Synthesis of space-time optimal systolic algorithms for the cholesky factorization. *Discrete Mathematics and Theoretical Computer Science*, 5:109–120, 2002.

[61] S.F. Oberman and M.J. Flynn. Division algorithms and implementations. *IEEE TRANSACTIONS ON COMPUTERS*, 46(8):833–854, Aug 1997.

[62] C.F. So, S.C. Ng, and S.H. Leung. Gradient based variable forgetting factor RLS algorithm. *Signal Processing*, 83:1163–1175, 2003. Publisher: Elsevier.

[63] B. Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal Processing*, 43:95–107, Jan 1995.

[64] B. Farhang-Boronjeny and S. Gazor. Selection of orthonormal transforms for improving performance of transform domain normalized LMS algorithm. In *IEEE Communications, Radar and Signal Processing*, volume 139, pages 327–335, Oct 1992.

[65] A.H. Sayed and T. Kailath. A State-Space Approach to Adaptive RLS Filtering. *IEEE Signal Processing Magazine*, 11(3):18–60, July 1994.

[66] M.K. Tsatsanis, G.B. Giannakis, and G. Zhou. Estimation and equalization of fading channels with random coefficients. *Signal Processing*, 53:211–229, 1996. Publisher: Elsevier.

LEBENSLAUF

Name	Lars Brühl
Geboren	am 31.5.1973 in Euskirchen
Staatsangehörigkeit	deutsch

Schulausbildung

08/1979 - 07/1983	Gertrudisschule in Euskirchen
08/1983 - 06/1992	Emil-Fischer-Gymnasium in Euskirchen

Studium

10/1992 - 09/1994	Vordiplom Elektrotechnik an der RWTH Aachen
10/1994 - 12/1997	Hauptdiplom, Fachrichtung Nachrichtentechnik, RWTH Aachen
19.12.1997	Abschluss des Diplomstudienganges Elektrotechnik mit Auszeichnung

Tätigkeiten während des Studiums

11/1994 - 12/1995	Studentische Hilfskraft am Institut für Elektrische Anlagen und Energiewirtschaft
04/1996 - 03/1997	Studentische Hilfskraft am Institut für Integrierte Systeme der Signalverarbeitung
01/1997 – 02/1997	Fachpraktikum bei Synopsys, DSP-Solution Group, Herzogenrath

Berufstätigkeit

02/1998 – 10/2004	Wissenschaftlicher Angestellter am Institut für Hochfrequenztechnik der RWTH Aachen mit dem Arbeitsgebiet: Digitale Signalverarbeitung für adaptive Mehrantennensysteme
ab 11/2004	Systemingenieur, Telefunken Radio Communication Systems GmbH & Co. KG, Ulm-Böfingen