

# Diplomarbeit

## Kanalmessungen mit DVB-T und MATLAB

Holger Meuel

Matrikel-Nr. 2723426

29. März 2010

Die Diplomarbeit wurde eingereicht am:  
Institut für Nachrichtentechnik  
der

Technischen Universität Braunschweig

**bei:** Prof. Dr.-Ing. U. Reimers

**Fachlicher Betreuer:** Dipl.-Ing. Jörg Robert



Institut für Nachrichtentechnik

---



An dieser Stelle möchte ich meinen ganz besonderen Dank an Professor Reimers richten, der mir die Erstellung der vorliegenden Diplomarbeit am Institut für Nachrichtentechnik ermöglichte. Mein Dank geht zudem an den fachlichen Betreuer dieser Arbeit, Herrn Dipl.-Ing. Jörg Robert. Ihm verdanke ich auch viele Anregungen und Hilfestellungen, die sehr zum Gelingen dieser Arbeit beigetragen haben.

*Vielen Dank für die lehrreiche und schöne Zeit!*

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Allgemeine Grundlagen</b>	<b>2</b>
2.1	Die OFDM-Mehrfachträger-Modulation . . . . .	2
2.1.1	Effekte bei fehlerhafter Synchronisation in OFDM-Systemen . . . . .	3
2.2	Funktionsweise eines Gleichwellennetzes (SFN) . . . . .	6
2.3	Einfluss eines Gleichwellennetzes auf OFDM-Signale . . . . .	7
2.4	Das DVB-T-Signal . . . . .	7
2.4.1	Basisbandschnittstelle . . . . .	8
2.4.2	Sync. Invertierung, Energieverwischung . . . . .	8
2.4.3	Äußerer Fehlerschutz: Reed-Solomon-Blockcode . . . . .	9
2.4.4	Äußerer Interleaver . . . . .	10
2.4.5	Innerer Fehlerschutz: Punktierter Faltungscodes . . . . .	11
2.4.6	Innerer Interleaver . . . . .	14
2.4.7	Symbolmapping . . . . .	14
2.4.8	Anpassung des Übertragungsrahmens . . . . .	16
2.4.9	Die Pilotstruktur . . . . .	17
2.4.10	OFDM-Subträger . . . . .	18
2.4.11	Guard Interval . . . . .	19
2.4.12	Guard Band . . . . .	19
2.5	Der Phasenregelkreis (PLL) . . . . .	19
<b>3</b>	<b>Kanalschätzung</b>	<b>20</b>
3.1	Der Übertragungskanal . . . . .	20
3.2	Kanalmodelle und Störungen . . . . .	21
3.2.1	Gauß'scher Kanal . . . . .	22
3.2.2	Rice-Kanal . . . . .	22
3.2.3	Rayleigh-Kanal . . . . .	23
3.3	Übertragungsfunktion und Impulsantwort . . . . .	24
3.4	Schätzung einer Kanalübertragungsfunktion . . . . .	24
3.5	Der CD3-Ansatz und der modifizierte CD3-Algorithmus . . . . .	25
3.5.1	Nutzung nur der äußeren Konstellationspunkte . . . . .	27
3.6	Filterung der Kanalübertragungsfunktion . . . . .	27
3.7	Probleme der linearen Interpolation . . . . .	27
3.8	Die Restträgerproblematik . . . . .	28

<b>4</b>	<b>Implementierung</b>	<b>30</b>
4.1	Erstellung der MATLAB-Software . . . . .	30
4.2	Hauptprogramm . . . . .	30
4.3	Die Datenstruktur <code>symb</code> . . . . .	31
4.4	Programmablauf . . . . .	34
4.4.1	Programmstart . . . . .	34
4.4.2	Grobsynchronisation mittels eines gleitenden Fensters . . . . .	36
4.4.3	Kompensation des Fractional Frequency Offsets . . . . .	36
4.4.4	Kompensation des Coarse Frequency Offsets . . . . .	37
4.4.5	Ermittlung der Scattered Pilots Konfiguration . . . . .	38
4.4.6	Pilotbasierte Kanalschätzung . . . . .	39
4.4.7	Interpolation durch Filterung der Kanalübertragungsfunktion . . . . .	39
4.4.8	Verarbeite TPS Pilots . . . . .	40
4.4.9	Restträgerlöschung . . . . .	41
4.4.10	Descrambling der Subträger und der Übertragungsfunktion . . . . .	42
4.4.11	Demapping . . . . .	42
4.4.12	Deinterleaving, Demultiplexing, Depunktierung . . . . .	44
4.4.13	Viterbi-Decodierung und Faltungscodierung . . . . .	44
4.4.14	Neucodierung des Datenstroms . . . . .	44
4.4.15	Kanalschätzung mittels des modifizierten CD3-Algorithmus . . . . .	44
4.4.16	Nachfilterung der Kanalübertragungsfunktion . . . . .	46
4.4.17	Implementierung des Phasenregelkreises . . . . .	50
4.4.18	Dateiausgabe, Initialisierung des nächsten Schleifendurchlaufs und Programmabschluss . . . . .	51
4.5	Codeoptimierungen . . . . .	51
4.6	Bekannte Fehler . . . . .	52
<b>5</b>	<b>Nutzung der Software</b>	<b>53</b>
5.1	Programmparameter . . . . .	53
5.1.1	Übersicht Parametereinstellungen . . . . .	58
5.2	Häufige Fehlerquellen . . . . .	64
<b>6</b>	<b>Messungen und Leistungsfähigkeit</b>	<b>69</b>
6.1	Der Messaufbau . . . . .	69
6.2	Auswertung . . . . .	70
6.2.1	Demonstration der Funktionstüchtigkeit . . . . .	71
6.2.2	Einfluss der Viterbi-Decodierung . . . . .	74
6.2.3	Messungen in einem realen DVB-T-SFN . . . . .	76
6.2.4	Zeitliche Filterung . . . . .	78
6.2.5	Einfluss der inneren Konstellationspunkte . . . . .	80
6.2.6	Der zeitvariante TU6-Kanal . . . . .	80
6.2.7	Effekte bei vergleichbaren Pegeln verschiedener Sender . . . . .	83
6.2.8	Messungen während einer Straßenbahnfahrt – 2. Versuch . . . . .	86
6.2.9	Messungen hinter Windkraftanlagen . . . . .	86

6.3	Grenzen . . . . .	87
6.3.1	Messungen während einer Straßenbahnfahrt – 1. Versuch . . .	87
6.3.2	Das Konstellationsdiagramm und zu geringer Signal-Rauschabstand	89
<b>7</b>	<b>Zusammenfassung</b>	<b>90</b>
	<b>Literaturverzeichnis</b>	<b>92</b>
	<b>Abkürzungsverzeichnis</b>	<b>94</b>
	<b>Abbildungsverzeichnis</b>	<b>96</b>
	<b>Codelistingverzeichnis</b>	<b>99</b>
	<b>Stichwortverzeichnis</b>	<b>100</b>

# 1 Einleitung

In modernen und zukünftigen Systemen wie beispielsweise dem digitalen terrestrischen Fernsehstandard DVB-T2<sup>1</sup>, dem WLAN-Standard<sup>2</sup> IEEE<sup>3</sup> 802.11n oder dem Mobilkommunikationssystem der Zukunft LTE<sup>4</sup> kann die spektrale Effizienz verbessert werden, indem Mehrantennensysteme (Techniken wie MISO<sup>5</sup> oder MIMO<sup>6</sup>) eingesetzt werden. Daten, die auf einem Übertragungsweg nicht hinreichend gut empfangen werden konnten, sind möglicherweise über einen anderen Kanal in guter Qualität am Empfänger angekommen. Dieser Effekt soll in neuen Systemen genutzt werden. So soll insbesondere die portable und mobile Empfangbarkeit verbessert werden [DVB09; RM10].

Bei terrestrischen Übertragungen treten spezielle Probleme auf, die auf anderen Verbreitungswegen wie der Kabelübertragung oder der satellitengestützten Ausstrahlung nicht vorkommen. Es entstehen durch Mehrwegeausbreitung und Reflexionen, aber auch durch die Sendeantennen in einem Gleichwellennetz selber, Störungen mit der Charakteristik von Echos. Diese treffen mit gedämpfter Amplitude und einer zeitlichen Verzögerung am Empfänger ein. Das derzeit beste bekannte Mehrträgerverfahren, welches speziell die Anforderungen in Echokanälen berücksichtigt, ist das OFDM-System<sup>7</sup>. Ein solches System benötigt zur Entzerrung des Empfangssignals zumeist (z. B. bei Verwendung einer QPSK oder QAM) eine Schätzung des Übertragungskanals. Bisher liegen jedoch keine Kanalmodelle vor, die den zu erwartenden Gewinn für Mehrantennensysteme abschätzen lassen und Störeinflüsse realer Umgebungen (hinreichend genau) abbilden können. Deshalb sollen im Rahmen dieser Diplomarbeit Kanalmessungen in einem real existierenden DVB-T-System durchgeführt werden, welches aufgrund seiner Auslegung als Gleichwellennetz bereits MISO-Technik einsetzt und deshalb hervorragend als Messsystem geeignet ist. Schätzungen des Übertragungskanals sollen unter Zuhilfenahme der Software MATLAB unter besonderer Berücksichtigung des iterativen CD3-Algorithmus errechnet werden [MM96].

Im gesamten Dokument wird zugunsten eines besseren Leseflusses ausschließlich die männliche Form von Personenbezeichnungen Verwendung finden.

---

<sup>1</sup>*Digital Video Broadcasting – Terrestrial, Second Generation*

<sup>2</sup>*Wireless Local Area Network*

<sup>3</sup>*Institute of Electrical and Electronics Engineers*

<sup>4</sup>*Long Term Evolution*

<sup>5</sup>*Multiple Input, Single Output*

<sup>6</sup>*Multiple Input, Multiple Output*

<sup>7</sup>*Orthogonal Frequency Division Multiplex*

## 2 Allgemeine Grundlagen

In diesem Kapitel werden allgemeine Grundlagen dargelegt, die in die verschiedenen Disziplinen einführen, die zum Verständnis einer Kanalschätzung benötigt werden. Dabei sollen die Themengebiete jeweils nur soweit behandelt werden, wie es für das Verständnis und die Durchführung der Aufgabenstellung notwendig ist. Nach einer Einführung in die Mehrfachträger-Modulation OFDM und in das Prinzip eines Gleichwellennetzes, wird ein Überblick über den Aufbau eines DVB-T-Signals vermittelt. Die in kohärenten OFDM-Systemen notwendige Kanalschätzung wird in Kapitel 3 besprochen. Da zur Schätzung der Kanalübertragungsfunktion lediglich eine Decodierung bis einschließlich des inneren Interleavers unerlässlich ist, werden sich folgende Abschnitte hauptsächlich auf die in der Encodierkette nachfolgenden Komponenten – also diejenigen, die in umgekehrter Reihenfolge ausgehend vom Empfangssignal im Decoder des Empfängers durchlaufen werden müssen – beziehen.

### 2.1 Die OFDM-Mehrfachträger-Modulation

Für digitale Signalübertragungen ist eine *Mehrfachträger-Modulation* wie das in [OL07] beschriebene *Orthogonal Frequency Division Multiplex* (OFDM) besonders geeignet. Sie überträgt innerhalb einer Symbolperiode mehrere Bits. Das Prinzip lässt sich am einfachsten anhand der aus [Rei08b] entnommenen Abbildung 2.1 erläutern.

In diesem Beispiel werden fünf Bits (11011) codiert. Jedes Bit wird dazu mit Sinusschwingungen unterschiedlicher Frequenzen multipliziert, das erste mit dem Wert *eins* wird mit einer Schwingung der Frequenz 0 Hz (Gleichspannung) multipliziert, das zweite (ebenfalls Wert *eins*) mit einer Sinusschwingung der Frequenz  $\frac{1}{T_{Nutz}}$  Hz. Mit den restlichen Bits wird ebenso verfahren. Alle modulierten Teilsignale  $s_0(t)$  bis  $s_4(t)$  werden addiert und ergeben das Zeitsignal  $s_{OFDM}(t)$  (unten links in der Abbildung), welches als Sendesignal ausgestrahlt wird. Der Empfänger erhält dieses Signal und muss eine Fouriertransformation durchführen, um das Signal  $S_{OFDM}(f)$  (unten rechts in der Abbildung) im Frequenzbereich zu erhalten. Dass eine schnelle und kostengünstige Implementierungen zur Durchführung einer *Diskreten Fourier Transformation* (DFT) in Form der *Fast Fourier Transformation* (FFT) verfügbar ist, macht das OFDM-System für den Massenmarkt tauglich. In der Praxis kommen natürlich Systeme zum Einsatz, die weit mehr als fünf Bits modulieren. Bei DVB-T werden beispielsweise in dem in Deutschland verwendeten 8K-FFT-Modus in einem 8 MHz breiten Kanal  $K = 6817$  aktive Trägerfrequenzen moduliert. Genauere Ausführungen dazu finden sich im Kapitel 2.4.10 auf Seite 18.

Die Vorteile des OFDM-Systems sind einerseits ein schonender Umgang mit den

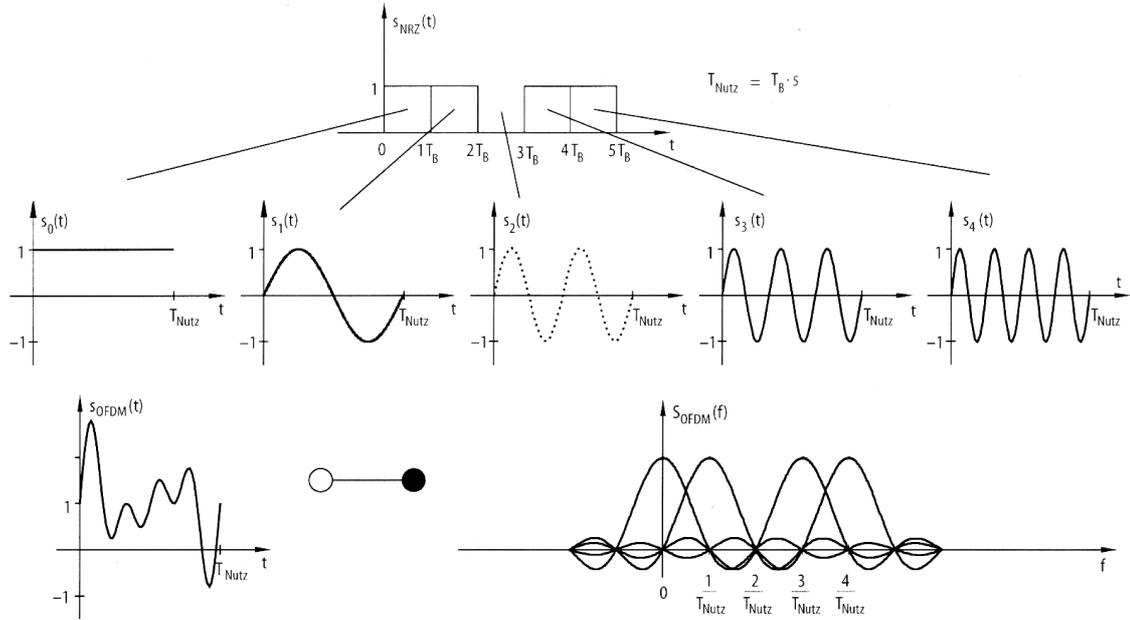


Abbildung 2.1: Prinzip eines OFDM-Modulators (aus [Rei08b])

Frequenzressourcen, andererseits vor allem aber eine besondere Robustheit gegenüber schmalbandigen Störungen. Besonders wichtig ist hierbei die Robustheit bei Frequenzselektivität wie z. B. in einem Gleichwellennetz (siehe auch Kapitel 2.2) [Rob09]. Das Prinzip dabei ist, dass ein breitbandiger frequenzselektiver Kanal in viele schmalbandige Kanäle, die OFDM-Subträger, aufgeteilt wird. Diese können jeder für sich als *nicht* mehr frequenzselektiv angesehen werden. Während der Dauer eines kompletten OFDM-Symbols, welches in der Praxis einige hundert Mikrosekunden lang ist, überträgt der Sender die ganze Zeit das gleiche OFDM-Signal, dessen Zusammensetzung oben beschrieben wurde. Treten währenddessen schmalbandige Störungen auf, werden dadurch einzelne Träger nicht mehr korrekt vom Empfänger decodiert, was aber durch die zum Einsatz kommenden Fehlerschutzmechanismen in vielen Fällen kompensiert werden kann. Natürlich sinkt auch die Bandbreite und die damit einhergehende Datenrate im Störfall. Durch geeignete Wahl der Übertragungsparameter lässt sich jedoch eine robuste und hochratige Datenübertragung (nach [Rei08b] zirka  $36 \frac{\text{Mbit}}{\text{s}}$ ) realisieren. Nach [Sch03] sinken zudem die Störemissionen in benachbarte Frequenzbänder desto mehr, je mehr Subträger in ein OFDM-Symbol verpackt werden.

### 2.1.1 Effekte bei fehlerhafter Synchronisation in OFDM-Systemen

Eine technische Herausforderung bei einem OFDM-modulierten Signal besteht darin, dass die Abtastung des Signals zu exakt definierten Zeitpunkten stattfinden muss,

damit im Frequenzbereich alle Trägerfrequenzen außer der dieser Frequenz zugeordneten null sind. Die Orthogonalität der Subträger muss gewahrt bleiben. Das Signal  $S_{OFDM}(f)$  im Frequenzbereich in der bereits bekannten Abbildung 2.1 verdeutlicht diesen Sachverhalt noch einmal. Schon eine leichte Abweichung vom idealen Abtastzeitpunkt führt zu einem Verlust der Orthogonalität und tritt als deutlicher Fehler hervor, der sich nach der inversen Fourier Transformation (IFFT) dem Nutzsignal in Form einer si-Funktion als Fehler überlagert. Aus diesem Grund kommt der Synchronisation in OFDM-basierten Systemen eine wesentlich zentralere Bedeutung als in Einträgersystem (*Singlecarrier*-Systemen) zu.

Die Synchronisation bei OFDM-Systemen lässt sich in (mindestens) zwei Teile untergliedern, die grundsätzlich andere Anforderungen mit sich bringen und dennoch beide unverzichtbar sind. Als unbedingt notwendiger erster Schritt muss eine derart genaue Synchronisation auf ein im Zeitbereich empfangenes OFDM-Signal erfolgen, dass nach einer Fouriertransformation das daraufhin im Frequenzbereich vorliegende Signal eine eindeutige Charakteristik, wie in Abbildung 2.1 unten rechts dargestellt, aufweist, also alle OFDM-Subträger außer dem aktiven an einer bestimmten Frequenz im Idealfall null sind oder aber der aktive Frequenzträger wenigstens deutlich stärker als die Nachbarfrequenzträger an der ihm zugeordneten Frequenz ausfällt. Die zeitliche Synchronisation lässt sich mittels einer *Guard Interval*-Korrelation<sup>1</sup> bewerkstelligen. Bei OFDM-Systemen hat es sich nach [SFFM01] als nützlich erwiesen, entgegen der Intuition erst eine Feinsynchronisation im Zeitbereich durchzuführen und erst anschließend eine Grobsynchronisation (im Frequenzbereich). Nachfolgende Synchronisationsschritte werden nur noch im Frequenzbereich durchgeführt.

### 2.1.1.1 Zeitsynchronisation

[SFFM99] stellt eine Methode zur zeitlichen Synchronisation mit Hilfe des Guard Intervals (siehe auch Kapitel 2.4.11) vor. Aufgrund der Beschaffenheit eines DVB-T-Signals, welches eine Kopie des hintersten Teils des codierten Nutzsignals im Zeitbereich am Anfang eines jeweiligen OFDM-Symbols enthält, kann mit Hilfe eines sogenannten *gleitenden Fensters* (*gliding window*) die Position eines OFDM-Symbols im Zeitbereich durch Korrelation ermittelt werden. Wenn der Empfänger ein OFDM-Symbol genau an der Stelle sucht, wo es tatsächlich beginnt, wird sich das Guard Intervall sowohl am Ende wie auch am Anfang des OFDM-Symbols (als Kopie) finden lassen. Wenn dieser Fall zutrifft, wird die Korrelationsfunktion ein ausgeprägtes Maximum im Vergleich zu allen anderen Fällen aufweisen, bei denen beliebige Signalteile miteinander korreliert werden, die *keine* Kopien voneinander sind (vergleiche auch Kapitel 4.4.2). Die Abbildung 2.2 skizziert das Prinzipschaltbild des gleitenden Fensters. Realisiert werden kann diese Anordnung durch zwei Fenster, die jeweils die Länge des Guard Intervals haben und um die Anzahl der Nutzsamples eines OFDM-Symbols voneinander entfernt liegen. Werden beide Fenster entlang des Datenstroms über die

---

<sup>1</sup>Um Begriffsverwirrungen zu vermeiden, werden in dieser Arbeit durchgängig die originalen Fachtermini verwendet und keine eingedeutschten Varianten. Wenn eine adäquate Übersetzung existiert, wird diese im entsprechenden Vertiefungskapitel zu den Begriffen aufgeführt.

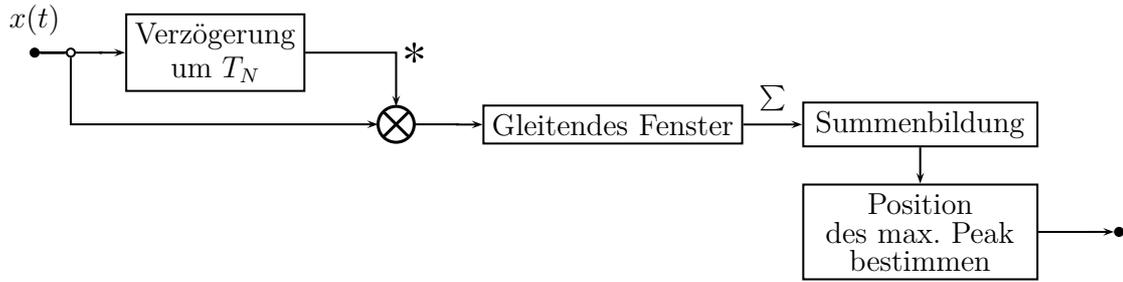


Abbildung 2.2: GI Korrelation mittels eines gleitenden Fensters

Länge eines kompletten OFDM-Symbols inklusive des Guard Intervals geschoben und die jeweilige Summe zu jedem Verschiebeschritt berechnet, so ist ein eindeutiges Maximum in der Summenfunktion erkennbar. Wird fehlerhaft und unexakt in *zeitlicher* Richtung synchronisiert, tritt nach [SFFM99] Intersymbolinterferenz (ISI) auf, also eine Überschneidung eines OFDM-Symbols mit dem nachfolgenden (bzw. vorangegangenen) OFDM-Symbol. Darüber hinaus ist es notwendig, den eventuell durch den Kanal bedingten Phasenversatz auf Basis einer Schätzung zu kompensieren. Wenn der zeitliche Versatz  $\tau$  ein ganzzahliges Vielfaches  $m$  der Samplingzeit  $T_S$  ist und  $N$  die verwendete FFT-Länge, entsteht gemäß [H<sup>+</sup>03] ein Phasenfehler  $\delta_\Phi$  entsprechend Gleichung 2.1 zwischen zwei benachbarten Frequenzsubträgern. Zur Kompensation dieses Fehlers wird der mittlere Phasenversatz aller komplexen Empfangsdaten durch Mittelwertbildung über alle Frequenzträger berechnet. Sämtliche OFDM-Subträger werden mit diesem Mittelwert korrigiert.

$$\delta_\Phi = 2\pi \frac{m}{N} \quad (2.1)$$

Des Weiteren kann ein sogenannter *Fractional Frequency Offsets* (FFO)  $\Delta f_F$  auftreten, welcher nach [SFFM01] durch Drehung des kompletten OFDM-Symbols im Frequenzbereich mit dem Winkel des Mittelwerts aller Subträger des gleitenden Fensters korrigiert werden kann.

Die Summe des gesamten Phasenversatzes ( $\delta_\Phi + \Delta f_F$ ) hat einen beträchtlichen Einfluss auf die resultierende Bitfehlerrate (BER) nach der Decodierung und sollte daher bestmöglich kompensiert werden.

### 2.1.1.2 Frequenzsynchronisation

Nachdem durch die Korrektur des *Fractional Frequency Offsets* bereits eine Feinsynchronisation in Frequenzrichtung durchgeführt wurde, wird nachfolgend die Grobsynchronisation ebenfalls nach [SFFM99] durchgeführt. Das die Feinsynchronisation *vor* der Grobsynchronisation durchgeführt wird, ist typisch für OFDM-basierte Systeme. Unter Zuhilfenahme von Trainingssymbolen – den Piloten, von denen sowohl die Position innerhalb eines OFDM-Symbols wie auch die Phasenlage<sup>2</sup> bekannt sind – kann

<sup>2</sup>Der Imaginärteil der Pilotträger ist laut Definition in [DVB06] immer gleich null.

nun die grobe Frequenzsynchronisation (*Coarse Frequency Correction*) vorgenommen werden. Dazu wird die Pilotkonstellation schrittweise über das Empfangssymbol im Frequenzbereich geschoben und ermittelt, an welcher Position die Übereinstimmung maximal ist. An dieser Stelle muss der Beginn eines OFDM-Symbols im Frequenzbereich liegen. Wird keine hinreichend genaue Synchronisation vorgenommen, tritt nach [SFFM99] das Phänomen des „Übersprechens“ in andere Frequenzbänder auf, welches als *Intercarrier interference* (ICI, „Intercarrier-Interferenz“) bekannt ist.

### 2.1.1.3 Nichtlineare Verzerrungen in OFDM-Systemen

Natürlich treten in jedem realen System Störungen auf. Genauere Untersuchungen der unterschiedlichen Zusammensetzung dieser Störgrößen sollen an dieser Stelle nicht durchgeführt werden. Wichtig hingegen ist der Gesamteinfluss aller Störungen im System und sein Einfluss auf die Leistungsfähigkeit der Decodierung. Einfache Kanalmodelle zur systemtheoretischen Modellierung dieser Einflüsse werden in Kapitel 3.2 ab Seite 21 vorgestellt. Dennoch gibt es zwei für OFDM-Systeme besonders wichtige nichtlineare Phänomene, die beim Design eines Empfängers berücksichtigt werden müssen und in [SFFM99] beschrieben sind.

**Phase Noise und der Common Phase Error** *Phasenrauschen* (PN, *Phase Noise*) entsteht durch Instabilitäten des Empfängeroszillators. Diese Nichtlinearität ist durch das Phasenrauschspektrum  $L_{\Phi_N}(f)$  charakterisiert. Phasenrauschen kann zum Verlust der Orthogonalität des empfangenen OFDM-Symbols führen, was wiederum zu Intercarrier-Interferenz führt. Daher können schon kleine Amplituden des Phase Noise zu signifikanten Einbußen der Leistungsfähigkeit führen. Darüber hinaus entsteht eine modulationsbedingte Phasendifferenz namens *Common Phase Error* (CPE), welche sich auf alle Subträger gleichermaßen auswirkt. Anhand der Pilote lässt sich der CPE berechnen und komplett korrigieren. Dazu wird die Lage des vordersten Peaks der Impulsantwort ermittelt. Der Abtastzeitpunkt (im Zeitbereich) nachfolgender OFDM-Symbole wird rekursiv durch eine Drehung im Frequenzbereich mit dem Phasenversatz zum Vorgängersymbol auf die gleiche Stelle geschoben.

## 2.2 Funktionsweise eines Gleichwellennetzes (SFN)

Die herkömmliche Ausstrahlung von Rundfunksignalen geschieht auf der Basis einzelner Sender, die eine bestimmte Region versorgen, in der für den jeweils genutzten Frequenzbereich eine hinreichende Wellenausbreitung gewährleistet ist. Benachbarte Sendestationen müssen immer unterschiedliche Frequenzen nutzen, selbst wenn sie in ihren Versorgungsgebieten das gleiche Programm ausstrahlen sollen, da es ansonsten bei Nutzung derselben Frequenz in den gemeinsam bestrahlten Randbereichen zu Störungen (Interferenzen) durch das Signal des Nachbarsenders kommen würde.

Bei einem *Gleichwellennetz* oder *Single Frequency Network* (SFN) senden mehrere exakt aufeinander synchronisierte Sender das gleiche Signal aus, was es einem

Empfänger, der Signale verschiedener Sender empfängt, ermöglicht, solche OFDM-Trägerfrequenzen, die in einem Signalpfad sehr stark abgeschwächt sind, aus einem anderen Signalpfad zu erhalten, bei welchem diese Frequenz nicht durch Störungen beeinträchtigt ist. Auf diese Art kann insbesondere in Randgebieten einzelner Senderegionen der Empfang verbessert werden. Eine Schlüsselrolle kommt dabei der Synchronisation der Sender zu, damit die Signale aller benachbarten Sender, welche im Sinne des Empfängers erst einmal als additive Störgröße auftreten, innerhalb des sogenannten *Guard Intervals* liegen. Deshalb ist die Topographie eines Gleichwellennetzes durch den maximalen Abstand benachbarter Sender begrenzt, letzterer wiederum durch die Länge des Guard Intervals (vergleiche Kapitel 2.3 und 2.4.11). Wird diese Bedingung eingehalten, treten keine Intersymbolinterferenzen auf [Rei08b].

## 2.3 Einfluss eines Gleichwellennetzes auf OFDM-Signale

Durch Mehrwegeempfang, wie er bei der terrestrischen Übertragung unvermeidlich auftritt, ergeben sich immer frequenzselektive Kanäle. Bei OFDM wird ein spektral breiter Kanal in sehr viele schmalbandige (Teil-)Kanäle aufgeteilt. Für jeden dieser OFDM-Subträger kann angenommen werden, dass er *nicht* frequenzselektiv ist. Die Empfangsgüte für jeden einzelnen OFDM-Subträger korreliert nun direkt mit der Kanalübertragungsfunktion. OFDM-Subträger, an deren Frequenz die Kanalübertragungsfunktion eine zu starke Dämpfung aufweist, sind nicht mehr auswertbar und müssen durch einen Fehlerschutz kompensiert werden. Deshalb ist für den fehlerfreien Empfang ein höheres Signal-Rauschverhältnis (SNR) notwendig als in echofreien Kanälen [Rob09].

Die Signale der Sender des SFNs, die *nicht* die maximale Feldstärke am Empfänger aufweisen, aber dennoch empfangbar sind, werden vom Empfänger mit der gleichen Charakteristik wie natürliche Echos wahrgenommen; sie kommen verzögert und gedämpft an. Die aus [Rei08a] entlehnte Abbildung 2.3 verdeutlicht, wie das Gesamtsignal am Empfänger durch Summation aller einfallenden Signale entsteht.

Zur Vermeidung von Intersymbolinterferenz muss zudem gewährleistet sein, dass ein Empfänger zu jeder Zeit von allen empfangbaren Sendestationen des Gleichwellennetzes das gleiche Symbol empfängt. Die Signale aller Sender des SFN müssen innerhalb einer definierten Zeit – dem Guard Interval (in der Abbildung durch  $T_G$  kenntlich gemacht) – am Empfänger eintreffen. Bei unexakter Synchronisation geht wiederum die Orthogonalität des OFDM-Signals verloren.

## 2.4 Das DVB-T-Signal

Wie bereits einführend erwähnt, werden für die Analyse des Kanals in dieser Arbeit DVB-T-Signale verwendet, weshalb das Verständnis der Struktur eines solchen Signals unabdingbar ist. Um ein DVB-T konformes Signal aus vorhandenen Nutzdaten

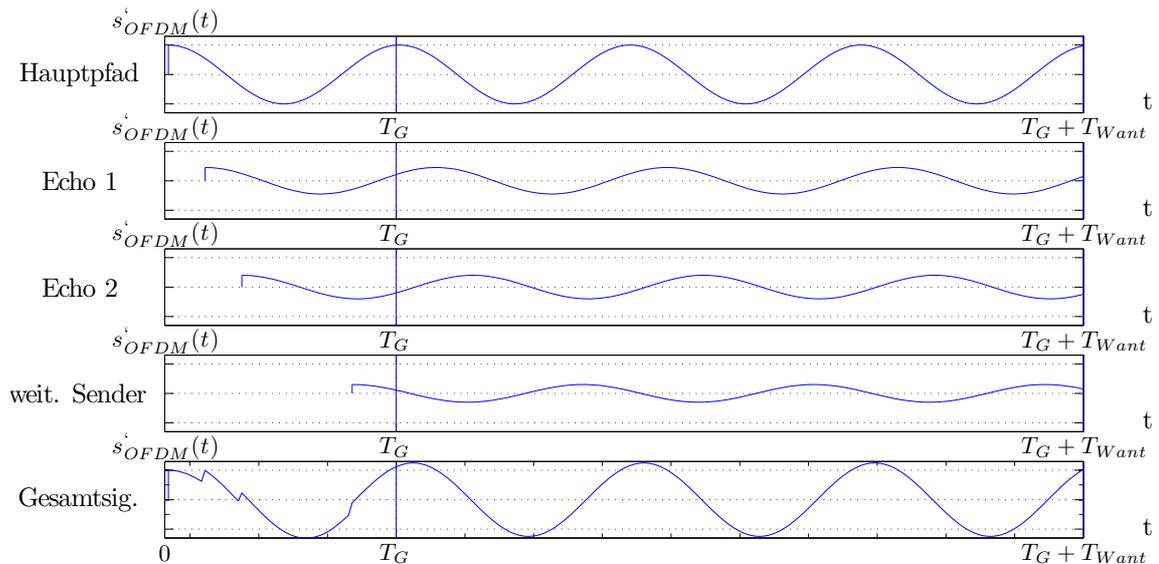


Abbildung 2.3: Mehrwegeempfang in Gleichwellennetzen und das Guard Interval (aus [Rei08a] entlehnt)

zu erstellen, ist eine Reihe von Verarbeitungsschritten notwendig, die unterschiedlichen Zwecken dienen, im Wesentlichen aber alle zusammen die Empfangbarkeit des Sendesignals auch unter nicht idealen Bedingungen gewährleisten sollen. Einen Überblick über die komplette encoderseitige Aufbereitung der zu übertragenden Nutzdaten ist aus der aus [Rei08b] entlehnten Abbildung 2.4 ersichtlich. Die umgekehrte Abarbeitung der Blöcke auf Decoderseite im Empfänger ist notwendig, um letztlich aus dem empfangenen Signal die Nutzdaten zu erhalten. Die inhaltlichen Aspekte folgen hauptsächlich [Rei08a] und [Rei08b] sowie dem DVB-T-Standard [DVB06].

## 2.4.1 Basisbandschnittstelle

Zu Beginn der Verarbeitungskette auf dem Weg vom quellencodierten Videomaterial zum sendefertigen DVB-T-Signal (in Abbildung 2.4 mit *MPEG-TS*<sup>3</sup> bezeichnet) wird das Basisbandvideosignals in den benötigten Sendefrequenzbereich angehoben. Dem Signal wird ein Takt hinzugemischt, welcher vom Decoder mit Hilfe der im MPEG-2-Transportstrom enthaltenen Synchronisationsmuster abgeleitet werden kann.

## 2.4.2 Sync. Invertierung, Energieverwischung

Zu Beginn eines jeden *MPEG-2 Transportstrom*-Pakets (MPEG-2-TS) wird ein standardisiertes Synchronisationsmuster eingefügt, welches einem Empfänger dazu dient, die Nutzpaketgrenzen zu bestimmen. Die Energieverwischung dient der gleichmäßigen Verteilung des Leistungsdichtespektrums eines Sendesignals zum Schutz von

<sup>3</sup>MPEG steht für *Moving Picture Experts Group* und TS für *Transport Stream*.

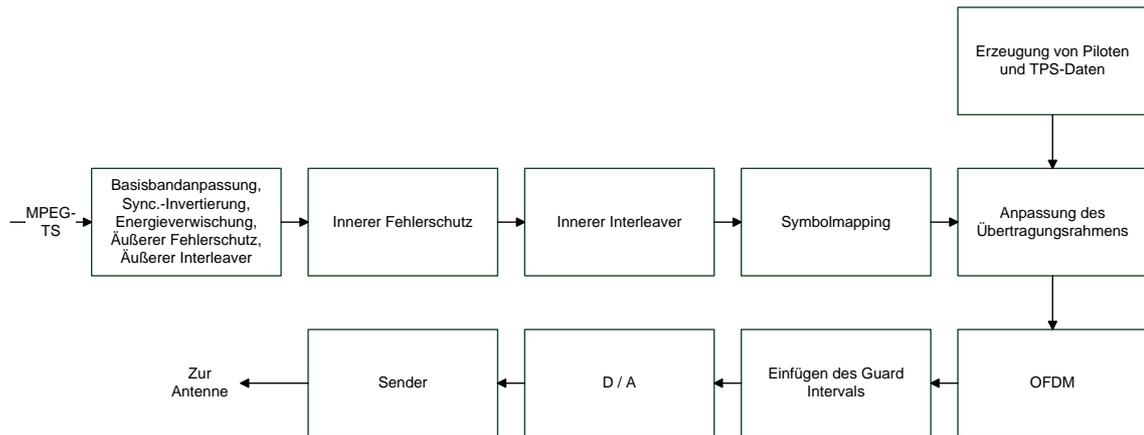


Abbildung 2.4: Encoderseitige Signalverarbeitung eines DVB-T-Signals (nach [Rei08b])

im Frequenzband benachbarten Kanälen. Die Energieverwischung wird durch sogenanntes *Scrambling* („Verwürfelung“) mit einer Pseudozufallssequenz durchgeführt. Zu Synchronisationszwecken der Energieverwischung wird jede achte Abfolge des Synchronisationsmusters nochmals invertiert, was dem Decoder signalisiert, dass die Pseudozufallssequenz von Neuem beginnt.

### 2.4.3 Äußerer Fehlerschutz: Reed-Solomon-Blockcode

Zur Korrektur von unweigerlich bei einer Datenübertragung auftretenden Übertragungsfehlern kommen zwei Vorwärtsfehlerschutzmechanismen (Forward Error Correction, FEC) zum Einsatz. Der äußere Reed-Solomon-Blockcode hat seine Stärke bei der Korrektur von sogenannten Burst- oder Büschelfehlern<sup>4</sup>. Bei Einzelfehlern arbeiten Blockcodes aufgrund ihrer Konstruktion relativ ineffizient. Zum detaillierten Verständnis der Funktionsweise der Reed-Solomon-Codes ist die Kenntnis von *Galoisfeldern*<sup>5</sup> (GF) unumgänglich [Kü08; Huf06].

Zunächst wird bei der Erstellung eines Codeworts das Nachrichtenwort  $\mathbf{u}$  als Polynom  $\mathbf{u}(x)$  dargestellt. Anschließend wird durch Multiplikation dieses Nachrichtenwortpolynoms  $\mathbf{u}(x)$  mit einem Generatorpolynom  $\mathbf{g}(x)$  ein Codewort  $\mathbf{a}(x)$  erzeugt. Gleichung 2.2 stellt den Vorgang übersichtlich zusammen.

$$\underbrace{\mathbf{a}(x)}_{\text{Codewort}} = \underbrace{\mathbf{u}(x)}_{\text{Nachrichtenwort}} \cdot \underbrace{\mathbf{g}(x)}_{\text{Generatorpolynom}} \quad (2.2)$$

<sup>4</sup>Burst- oder Büschelfehler sind Abfolgen von fehlerhaften Bits in einem Datenstrom.

<sup>5</sup>Es handelt sich dabei um mathematische Körper mit bestimmten Eigenschaften wie etwa der Zyklizität. Jedes GF besitzt ein primitives Polynom, aus welchem alle gültigen Elemente eines GF und damit alle gültigen Codeworte abgeleitet werden können. Weiterhin ist es basierend auf dem Generatorpolynom möglich, eine Prüfmatrix zur Fehlererkennung und -korrektur zu erstellen. Die in DVB-T verwendeten Reed-Solomon-Codes arbeiten auf dem Galoisfeld  $\text{GF}(2^8)$ .

Die Decodierung ist mittels eines *Syndroms* möglich, soll hier jedoch nicht weiter thematisiert werden. [Kü08] und [Rei08b] liefern weitere Ausführungen hierzu. Bei DVB-T kommt ein verkürzter RS(255, 239), nämlich ein RS(204, 188) mit einer Korrekturleistung von  $t = 8$  korrigierbaren Fehlern, zum Einsatz. Da Fehler im Datenstrom zufällig auftauchen, kann zu deren Charakterisierung nur eine zeitlich gemittelte Größe verwendet werden. So ist die Eingangsbitfehlerrate definiert als die Anzahl von Fehlern pro übertragenen Bits, die am Eingang eines Blockes – hier des Reed-Solomon-Decoders – anliegen. Entsprechend ist die Ausgangsbitfehlerrate die Anzahl von Fehlern pro Zeit, die am Ausgang eines Blocks auftreten. Um die sogenannte *Quasi Error Free Reception* (QEF), also nahezu fehlerfreien Empfang, zu garantieren, muss nach [Rei08a] bei DVB-T am Reed-Solomon-Decoder eine Ausgangsbitfehlerrate von  $\leq 1 \times 10^{-11}$  erreicht werden, wofür wiederum eine Eingangsbitfehlerrate des Reed-Solomon-Decoders von  $< 2 \times 10^{-4}$  vom Ausgang des inneren Fehlerschutzes bereitgestellt werden muss.

#### 2.4.4 Äußerer Interleaver

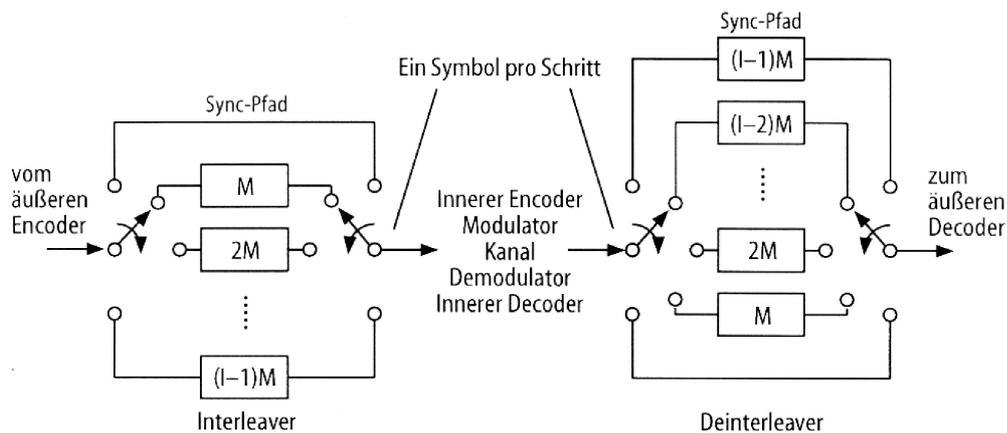


Abbildung 2.5: Prinzipschaltbild eines Faltungsinterleavers (aus [Rei08b])

Ein Interleaver dient dazu, auf dem Übertragungskanal auftretende längere Fehlerketten (Fehlerbursts) durch Umsortierung der Pakete zu kürzeren Fehlerbursts beziehungsweise im Idealfall zu Einzelfehlern umzuwandeln. Diese können durch einen auf Decoderseite dem Deinterleaver nachgeschalteten Vorwärtsfehlerschutzmechanismus korrigiert werden. Ein Interleaver dient daher in Verbindung mit einer nachgeschalteten Fehlerkorrektur<sup>6</sup> der Erhöhung der Robustheit einer Datenübertragung. Im Falle von DVB-T muss der Faltungsinterleaver gewährleisten, dass seine Ausgangsdaten vom Reed-Solomon-Blockcode korrigiert werden können.

<sup>6</sup>Anstelle einer Fehlerkorrektur kann auch eine Fehlererkennung verwendet werden. Für den Fall eines nicht korrigierbaren Fehlers müssen Strategien für die Fehlerbehandlung entwickelt werden.

Abbildung 2.5 (aus [Rei08b]) illustriert den schematischen Aufbau eines Faltungsin-terleavers. Die vom äußeren Encoder gelieferten Daten werden bitweise entlang eines bestimmten Pfades geschickt, welcher im aktuellen Taktschritt durch die Schalter aktiviert ist. Mit jedem Taktschritt werden beide Schalter um eine Stufe weiterge-schaltet. Die einzelnen Pfade verzögern einzelne Bits um ganzzahlige Vielfache der sogenannten Basisverzögerung  $M$ , welche eine Systemeigenschaft eines Faltungsin-terleavers darstellt. Die Interleavingtiefe  $I$  ist definiert als die Anzahl von Bits, die verwendet werden müssen, um alle Zustände des Interleavers einmalig zu durchlau-fen. Der Deinterleaver ist ähnlich aufgebaut wie der Interleaver, nur dass die Verzöge-rungspfade entgegengesetzt der encoderseitigen Anordnung aufgebaut sind, der Pfad mit der größten Verzögerung somit nicht an unterster, sondern an oberster Stelle zu finden ist. Somit ergibt sich nach [Rei08b] für jedes Symbol eine Gesamtverzö-gerung von  $M \cdot (I - 1) \cdot I$ . Prinzipbedingt können nur Symbole verarbeitet werden, die eine bestimmte Länge haben (1512 Bits im 2K-FFT-Modus, 6048 Bits im 8K-FFT-Modus), weshalb eine solche Anordnung auch *Symbolinterleaver* genannt wird. Bei DVB-T wird eine Interleavingtiefe von  $I = 12$  und eine Basisverzögerung von  $M = 17$  verwendet [Rei08b].

#### 2.4.5 Innerer Fehlerschutz: Punktierter Faltungscod

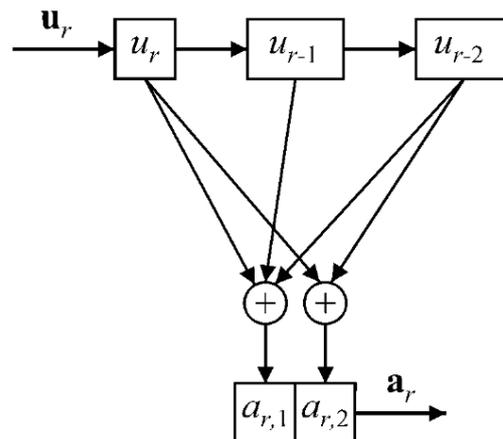


Abbildung 2.6: Beispiel eines Faltungscodierers (aus [Kü08])

Der innere Fehlerschutz ist ein Faltungscod, der besonders zur Korrektur von Ein-zelbitfehlern geeignet ist. Mathematisch wird zur Erzeugung eines Codewortes  $\mathbf{a}$  ein uncodiertes Nachrichtenwort  $\mathbf{u}$  mit einem Generatorwort  $\mathbf{g}$  gefaltet. Abbildung 2.6 aus [Kü08] zeigt einen beispielhaften *Faltungscodier*, anhand dessen sich ohne Ma-thematik die Funktionsweise einfach erklären lässt. Das Nachrichtenwort  $\mathbf{u}_r$  wird bit-weise von links nach rechts in das Register des Encoders geschoben. Mit jedem Takt-schritt werden alle Bits jeweils um ein Register weitergeschoben und das nächste Bit



ne Möglichkeiten der Decodierung, bis zwölf Jahre später<sup>8</sup> Viterbi seine Maximum-Likelihood-Decodierung (ML) vorstellte, wie sie heute noch Verwendung findet. Die Decodierung eines Faltungscodes lässt sich anhand eines Trellisdiagramms veranschaulichen. In einem Trellisdiagramm wie in Abbildung 2.7 aus [Kü08] sind alle möglichen Zustände eines Faltungscodes – folglich alle möglichen Codewörter – über der Zeit, also zu verschiedenen Taktschritten, übereinander aufgetragen. Dieses Trellisdiagramm könnte der Decodierung des Beispielfaltungscodes aus Abbildung 2.6 auf Seite 11 dienen. Unter Berücksichtigung des eingegebenen Empfangswortes, welches sich aus dem gesendeten Codewort und einem überlagerten zufälligen Fehlerwort zusammensetzt, wird nach einer gewissen Anzahl von Taktschritten der wahrscheinlichste Pfad ausgewertet und als wahr angenommen. Terminierung wie im Beispiel oben wird bei Broadcastdiensten nicht auftreten, da es keinen Anfang und kein Ende einer Übertragung gibt, so dass die Grenze, wann der Viterbi-Decoder den Trellispfad auswerten soll, künstlich gesetzt werden muss. Da ein Empfänger zu beliebigen Zeitpunkten eingeschaltet werden kann, ohne dass er Kenntnis über den aktuellen Systemzustand hat, in welchem sich der Decoder gerade befinden müsste, sind im Falle des Broadcastdiensteempfangs die ersten Ausgabewerte eines Viterbi-Decoders unzuverlässig. Dieser Umstand sollte bei Implementierungen bedacht werden [Kü08].

**Soft-Bits** Um die Leistungsfähigkeit des Viterbi-Decoders zu verbessern, kann an Stelle einer *Hard Decision*-Decodierung, die dem Decoder lediglich binäre Werte liefert, eine *Soft Decision*-Decodierung verwendet werden. Dabei wird basierend auf dem ermittelten Abstand eines Konstellationspunktes von den zugehörigen Entscheidungsschwellen eine Wahrscheinlichkeit für die Korrektheit der Entscheidung des Demappers ermittelt, welche an den Viterbi-Decoder weitergeleitet wird. Dieser Wert wird als *Soft-Bit* bezeichnet, da nicht nur *entweder* der Wert „0“ *oder* der Wert „1“ gültige Ergebnisse sind, sondern alle Zahlen innerhalb des Wertebereiches der verwendeten Variablen. Der Viterbi-Decoder kann die Wahrscheinlichkeiten entsprechend bei der Decodierung berücksichtigen. Somit kann nach [Rei08b] ein Gewinn von bis zu 2 dB Signal-Rauschabstand (SNR) im Vergleich zur Hard-Decision-Decodierung erzielt werden.

**Log-Likelihood Ratio-Werte** Zur Vermeidung numerischer Probleme wird anstelle der Wahrscheinlichkeitswerte der Soft-Bits häufig mit dem *Log-Likelihood Ratio* (LLR) eine logarithmische Größe verwendet. Es steht ein LLR-Wert größer als null für eine decodierte „1“ und ein negativer LLR-Wert für den decodierten Bitwert „0“. Der Betrag des LLR-Wertes gibt eine Größe für die Wahrscheinlichkeit der Korrektheit der Entscheidung an. Auf diese Weise enthält ein LLR-Wert sowohl den Bitwert wie auch die Wahrscheinlichkeit innerhalb *einer* Variablen. Ein LLR-Wert von null steht für eine Wahrscheinlichkeit von 50 % für eine „0“ und 50 % für eine „1“, es ist dementsprechend keine Aussage über den Wert des Bits an dieser Stelle möglich, da

---

<sup>8</sup>Bereits 1963 hat Fano seine suboptimale sequentielle Decodierung vorgestellt, die allerdings den Faltungscodes nicht zum Durchbruch verhelfen konnte.

der Konstellationspunkt genau auf einer Entscheidungsschwelle empfangen wurde. Nach [Xia03] ist das Likelihood Ratio  $L_i$  entsprechend Gleichung 2.3 definiert, wobei  $y$  das zugrunde liegende Entscheidungsmodell ist und  $P(x(i) = N)$  die Wahrscheinlichkeit dafür ist, dass der Wert  $N$  mit  $N \in \{0; 1\}$  empfangen wurde. Der Logarithmus von  $L_i$  entspricht dem LLR-Wert.

$$L_i = \frac{P(x(i) = 1|y)}{P(x(i) = 0|y)} \quad (2.3)$$

## 2.4.6 Innerer Interleaver

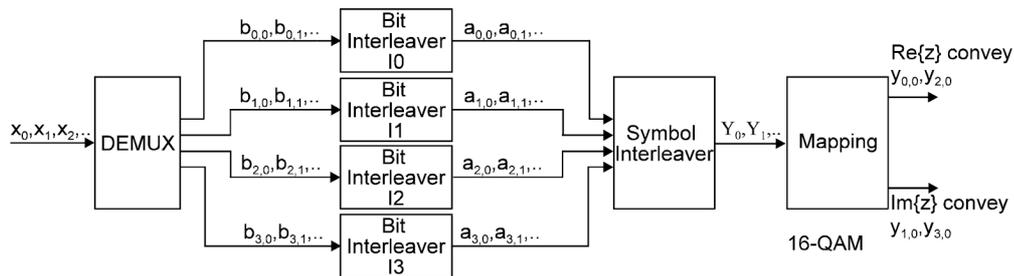


Abbildung 2.8: Prinzipschaltbild des inneren DVB-T-Interleavers für 16-QAM (aus [DVB06])

Der *innere Interleaver* ist als Blockinterleaver realisiert und dient dazu, aus eventuell aufgetretenen Burstfehlern Einzelfehler zu erzeugen, die auf Empfängerseite mit dem Faltungscoder (siehe Abschnitt 2.4.5) sehr gut korrigiert werden können. Abbildung 2.8 aus [DVB06] zeigt ihn für die (in Deutschland eingesetzte) 16-QAM. Zunächst wird der serielle Eingangsdatenstrom in mehrere Pfade aufgeteilt, im Beispiel in die Pfade  $b_{0,0}, b_{0,1}$  bis  $b_{3,0}, b_{3,1}$ , bevor die Bits eines jeden einzelnen Pfades mit einer im Standard [DVB06] definierten Interleavingsequenz  $H_n$  verwürfelt werden. Die derart gescrambelten Bits werden in einem Symbolinterleaver entsprechend einer standardisierten Permutation wieder zu einem seriellen Strom zusammengefügt.

## 2.4.7 Symbolmapping

Das bis hierhin aufbereitete Signal wird einem digitalen Modulationsverfahren unterzogen. Wahlweise stehen bei DVB-T die Quadraturphasenumtastung (QPSK) oder die Quadratur-Amplitudenmodulation (QAM) mit 16 beziehungsweise 64 Konstellationspunkten zur Verfügung. Es ist nunmehr möglich, zwei, vier oder sechs Bits auf jeden OFDM-Subträger aufzumodulieren. Die [DVB06] entnommenen Konstellationsdiagramme in Abbildung 2.9 zeigen das Mapping der einzelnen Bits bei DVB-T.

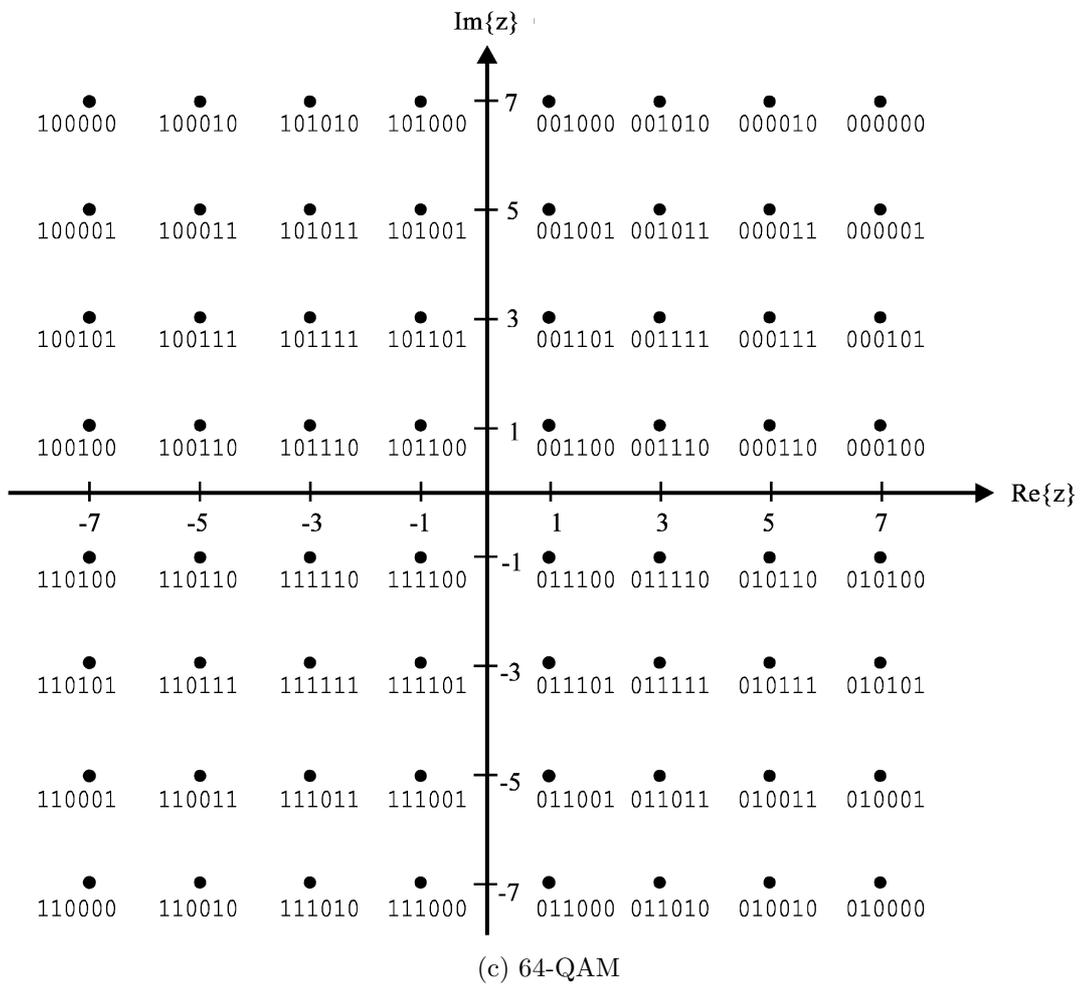
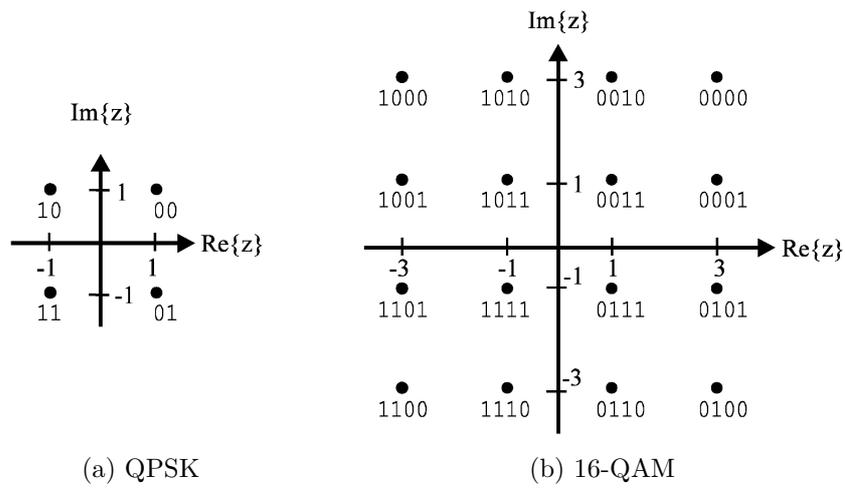


Abbildung 2.9: Konstellationsdiagramme digitaler Modulationen (aus [DVB06])

Es ist darauf zu achten, dass das *Most Significant Bit* (höchstwertiges Bit) und das *Least Significant Bit* (niedrigstwertiges Bit) korrekt zugeordnet werden, das heißt, dass die empfangenen Bits in der vom Standard vorgeschriebenen Reihenfolge ausgewertet werden. Da die 16-QAM für den deutschen DVB-T-Sendebetrieb gewählt wurde, kommt ihr hierzulande die größte Bedeutung zu [Rei08b].

### 2.4.7.1 Symboldeinterleaver

Durch den Einsatz von OFDM kann ein Empfänger frequenzselektive Störungen gut kompensieren. Die Daten sollten innerhalb des OFDM-Symbols möglichst so verteilt werden, dass zeitlich aufeinander folgende Daten auf möglichst weit voneinander entfernt liegende Subträger verteilt werden, damit Übertragungsstörungen als Einzelfehler durch den Vorwärtsfehlerschutz kompensiert werden können. Dieses leistet der in [DVB06] standardisierte *Symboldeinterleaver* [Rei08b].

### 2.4.8 Anpassung des Übertragungsrahmens

Um den für Synchronisationszwecke benötigten Anteil am Gesamtübertragungsvolumen in Form der Pilote, die in den nächsten Abschnitten genauer erklärt werden, möglichst gering zu halten, werden 68 OFDM-Symbole, welche im 2K-FFT-Modus aus jeweils 1705 Subträgern und im 8K-FFT-Modus aus jeweils 6817 Subträgern bestehen, zu einem *Übertragungsrahmen* zusammengefasst. Vier Übertragungsrahmen bilden einen *Überrahmen*. Abbildung 2.10 (nach [Rei08b]) zeigt das Resultat. In horizontaler Richtung ist die Frequenz mit den einzelnen Subträgern dargestellt, in vertikaler Richtung verläuft die Zeitachse von oben nach unten [Rei08b].

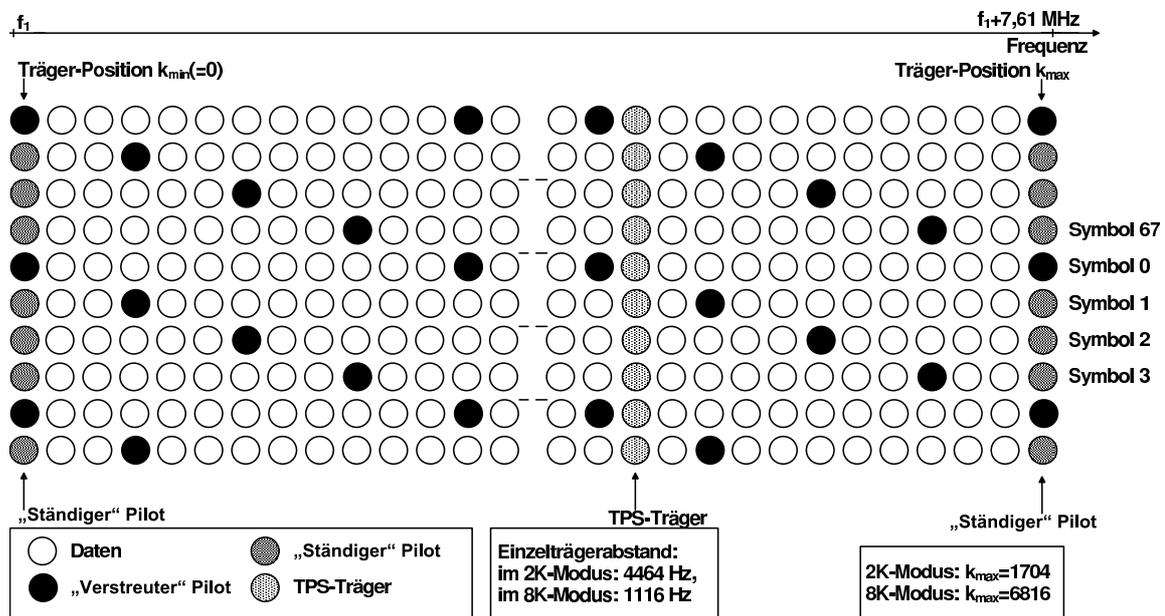


Abbildung 2.10: DVB-T-Übertragungsrahmen (nach [Rei08b])

## 2.4.9 Die Pilotstruktur

Pilote sind OFDM-Subträger, die der Synchronisation dienen und zusätzliche Übertragungsparameter enthalten können. Die Pilotstruktur ist im DVB-T-Standard [DVB06] festgelegt und soll nachfolgend erklärt werden, da die Pilote eine zentrale Rolle bei der Kanalschätzung, wie sie im Kapitel 3 beschrieben wird, einnehmen. Für alle Pilote wird zwar eine an [Rei08b] angelehnte deutsche Übersetzung aufgeführt, in den folgenden Kapiteln wird jedoch weiterhin die Originalbezeichnung aus dem Standard verwendet.

### 2.4.9.1 Continual Pilots

Die *Continual Pilots* („*Ständige Pilote*“) befinden sich immer auf den gleichen Subträgern innerhalb eines OFDM-Symbols. Ihre Lage ist so gewählt, dass innerhalb des Kanals keine Periodizitäten auftreten können, weshalb sie zur Frequenzsynchronisation und damit zur Grobeinstellung der Frequenz des Lokaloszillators des Empfängers dienen können. Sie werden mit erhöhter Amplitude übertragen und sind deshalb *Boosted Pilots* (siehe Abschnitt 2.4.9.4) [Rei08b].

### 2.4.9.2 Scattered Pilots

Die *Scattered Pilots* oder „*Verstreute Pilote*“ dienen hauptsächlich der Kanalschätzung. Sie sind in großer Anzahl in einem OFDM-Symbol vorhanden und können deshalb nach [Rei08b] zur Feineinstellung der Frequenz genutzt werden. Sie sind ebenso wie die Continual Pilots *geboostet*. Es gibt vier verschiedene Muster, die auftreten können und sich in der Startposition (in Frequenzrichtung) unterscheiden. Der erste Scattered Pilot in einem Symbol kann je nach vorliegender Konfiguration auf dem nullten, dem dritten, dem sechsten oder dem neunten OFDM-Subträger liegen. In allen möglichen Mustern liegen Scattered Pilots auf von der Startposition gerechnet jedem zwölften OFDM-Subträger. Die verschiedenen Konstellationen dienen der besseren Interpolation in Zeitrichtung, da durch die spezielle Anordnung mehr unterschiedliche Stützstellen in einem Übertragungsrahmen bereit stehen.

### 2.4.9.3 TPS Pilots

Die *Transmission Parameter Signalling-Pilots* (TPS Pilots) mit der sperrigen deutschen Übersetzung *Übertragungsdatensignalisierungspilote* enthalten wichtige Kenngrößen über das ausgestrahlte Signal. Sie enthalten die Information, mit welcher Modulation (QPSK, 16-QAM, 64-QAM) der Datenstrom versehen ist und geben im Falle der sogenannten *hierarchischen Modulation* das verwendete Konstellationsmuster an. Die hierarchische Modulation ist eine spezifizierte Variante der DVB-T-Übertragung, welche dafür sorgen soll, dass auch unter schlechten Empfangsbedingungen wenigstens ein grob aufgelöstes Bild decodiert werden kann. Diese Möglichkeit hat jedoch bislang keine praktische Bedeutung, weshalb sie auch nicht näher betrachtet werden

soll. Dennoch sind die TPS Pilots aufgrund der in ihnen enthaltenen wichtigen Informationen mit einem gesonderten BCH-Code<sup>9</sup> (67, 53) versehen. Sie liegen in jedem OFDM-Symbol auf den gleichen Subträgern und sind keine geboosteten Pilote. Es gibt allerdings im Vergleich zu den zahlreichen Scattered Pilots nur verhältnismäßig wenige TPS-Pilote (17 im 2K-FFT-Modus, 68 im 8K-FFT-Modus) [Rei08b; DVB06].

#### 2.4.9.4 Boosted Pilots

*Boosted Pilots* („Pilote erhöhter Amplitude“) sind Pilote, die mit einer im Vergleich zu der mittleren Leistung aller Datensubträger um  $\frac{4}{3}$  erhöhten Amplitude übertragen werden [Rei08b]. Da sie besonderen Zwecken wie der Frequenzsynchronisation des Empfängers oder der Kanalschätzung dienen, müssen sie gegenüber äußeren Störeinflüssen robuster sein als normale Datenträger. Durch die erhöhte Leistung der geboosteten Pilote können sie auch bei schlechteren SNR-Werten noch empfangen werden.

#### 2.4.10 OFDM-Subträger

Da DVB-T schon in der Entwurfsphase für den Betrieb in Gleichwellennetzen ausgelegt wurde, musste nach [Rei08b] als Modulationsart OFDM verwendet werden, denn in Gleichwellennetzen treten typischerweise sehr große Laufzeitdifferenzen zwischen den am Empfänger einfallenden Signalen auf. Bei Verwendung einer Einträgermodulation und – für hohe Datenraten benötigten – kurzen Symbolauern würden ansonsten Intersymbolinterferenzen auftreten. Die Verbindung mit den bereits eingeführten Methoden des Fehlerschutzes ist als *Coded Orthogonal Frequency Division Multiplex* (COFDM) bekannt. Es gibt zwei im Standard [DVB06] festgelegte Modi für die Signalverarbeitung, speziell für die Fast Fourier Transformation (FFT) und die inverse Fast Fourier Transformation (IFFT), welche jeweils mit 2K ( $2 \times 1024 = 2048$ ), bzw. 8K ( $8 \times 1024 = 8192$ ), Subträgern in einem OFDM-Symbol ausgeführt werden können. Der 2K-FFT-Modus wurde insbesondere standardisiert, da zum geplanten Einführungszeitpunkt im Jahre 1997 die Rechenleistung auf Endanwendergeräten für den technisch überlegeneren 8K-FFT-Modus noch nicht zu erschwinglichen Preisen zur Verfügung stand. An den Frequenzbandgrenzen muss nach [DVB06] der Pegel um mindestens 30 dB unterhalb des mittleren Sendepiegels des OFDM-Symbols liegen. Einerseits werden dadurch Störemissionen in Nachbarbänder möglichst gering gehalten, andererseits wird durch die starke Pegelabsenkung ein nahezu zyklischer Kanal erreicht, da beide Seiten des Spektrums auf vergleichbaren Pegeln liegen. Aufbauend auf diesen Vorgaben wurden für den 2K-FFT-Modus mit theoretisch 2048 Subträgern in jedem OFDM-Symbol 1705 aktive Subträger standardisiert, von denen 1512 Nutzdaten enthalten. Im 8K-FFT-Modus können von den 8192 Subträgern 6817 aktive genutzt werden, von denen 6048 als Nutzdatenträger Verwendung finden.

---

<sup>9</sup>*Bose-Chaudhuri-Hocquenghem-Codes* (BCH-Codes) sind zyklische lineare Blockcodes.

Der Standard sieht Kanalbandbreiten von 6, 7 oder 8 MHz vor<sup>10</sup>. In Deutschland wird der 8K-FFT-Modus in 8 MHz breiten Kanälen verwendet. Die Symboldauer eines kompletten OFDM-Symbols liegt in Abhängigkeit der verwendeten Guard Interval-Länge nach [DVB06] für den 2K-FFT-Modus zwischen 231 und 280  $\mu\text{s}$  und im 8K-FFT-Modus zwischen 924 und 1120  $\mu\text{s}$ .

### 2.4.11 Guard Interval

Das *Guard Interval* (GI) ist ein Schutzintervall vor Intersymbolinterferenz. Als Guard Interval wird der erste zeitliche Anteil eines OFDM-Symbols bezeichnet. Innerhalb des Guard Intervals müssen alle Echos, die durch Mehrwegeausbreitung oder andere Sender des Gleichwellennetzes entstehen, am Empfänger angekommen sein. Um diese Bedingung zu gewährleisten, muss die Impulsantwort des Kanals kürzer sein als das Guard Interval lang ist. Letzteres dient somit dem Einschwingen des Kanals [Rei08b]. Das Guard Interval wird üblicherweise als Bruchteil der Nutzdatenzeit angegeben. Zur Erzeugung wird der hintere Bruchteil des codierten Nutzsymbols im Zeitbereich noch einmal vor den Anfang des Nutzdatenteils kopiert. Dadurch werden durch Erhaltung der Stetigkeit des Zeitsignals breitbandige Störungen im Frequenzbereich verhindert. Standardisierte Werte für das Guard Interval bei DVB-T liegen laut [DVB06] zwischen  $\frac{1}{4}$  bis  $\frac{1}{32}$  der Zeitdauer der Nutzdaten.

### 2.4.12 Guard Band

Das *Guard Band* (GB) ist ein Schutzfrequenzband, welches einen Frequenzabstand zu den spektral angrenzenden Kanälen herstellt und so Nachbarkanalinterferenz verhindert. Aufgrund nicht-idealer Filter ist die Verwendung eines Guard Bands in realen Systemen unverzichtbar.

## 2.5 Der Phasenregelkreis (PLL)

Ein Phasenregelkreis (oder englisch *Phase-Locked Loop*, PLL) ist ein Regelkreis, bei dem basierend auf dem aktuellen Wert unter Einbeziehung eines Referenzwertes eine Korrektur durchgeführt wird. Eine Regelung mittels eines PLL eignet sich zur fehlertoleranten Nachführung eines langsam veränderlichen Wertes. Beispielsweise kann ausgehend von der vorherigen Position im Datenstrom eine leichte durch Oszillatorungenauigkeiten des Empfängers bedingte Verschiebung des Abtastzeitpunktes korrigiert werden. Der Einsatz eines Phasenregelkreises ist in diesem Falle sinnvoll, damit nicht OFDM-Symbole, auf die nicht korrekt synchronisiert werden konnte, zu starkem Einfluss auf den Abtastzeitpunkt von nachfolgenden Symbolen haben, sondern lediglich eine geringfügige Korrektur in die Richtung der Abweichung durchgeführt wird [Bar99].

---

<sup>10</sup>Im Anhang G des Standards [DVB06] ist zusätzlich der Betrieb in einem nur 5 MHz breiten Kanal abseits der üblichen Rundfunkfrequenzen spezifiziert.

# 3 Kanalschätzung

Wie bereits im Abschnitt 2.1.1 (siehe Seite 3) hervorgehoben wurde, ist die Orthogonalität des Signals in OFDM-basierten Systemen unbedingt notwendig. Durch verschiedene Einflüsse kann diese Orthogonalität jedoch verloren gehen. Letztlich müssen alle Störeinflüsse soweit kompensiert werden, dass eine hinreichende Signalqualität für eine erfolgreiche Decodierung erreicht wird. Dem Empfänger steht als einzige Information das Empfangssignal zur Verfügung. Dieses setzt sich jedoch aus dem Sendesignal  $\mathbf{x}$  und überlagerten Störungen zusammen. Wenn dem Empfänger diese Störungen bekannt sind, kann er das empfangene Signal  $\mathbf{y}$  derart manipulieren, dass das korrigierte Empfangssignal  $\mathbf{z}$  wieder dem Sendesignal entspricht. Da jedoch in realen Systemen die *wahren* Störeinflüsse nicht bekannt sind, muss eine Schätzung der überlagerten Störungen erfolgen. Dieser Vorgang wird als *Kanalschätzung* bezeichnet. Unter gewissen Annahmen, die die Störungen modellieren und unter Einbeziehung von Wissen über die Pilotstruktur (vergleiche Kapitel 2.4.9 auf Seite 17) kann eine solche Kanalschätzung erfolgreich durchgeführt werden. Dieses Kapitel führt zunächst den Übertragungskanal als solchen ein, bevor auf verschiedene Störmodelle eingegangen wird. Schließlich wird der klassische Ansatz zur Schätzung der Kanalübertragungsfunktion vorgestellt und der sogenannte CD3-Ansatz beschrieben, der eine pilotfreie Kanalschätzung vorschlägt.

## 3.1 Der Übertragungskanal

Der allgemeine Fall eines digitalen Übertragungssystems ist in der an [Kü08] angelehnten Abbildung 3.1 dargestellt. Die Kanalcodierung besteht aus den in Kapitel 2.4 (siehe Seite 7) näher beschriebenen Methoden (äußerer und innerer Vorwärtsfehler-schutz sowie äußerer und innerer Interleaver) und dient insgesamt der Erzeugung von Paritätsdaten, mit deren Hilfe bei einer fehlerhaften Übertragung die Originaldaten dennoch decodiert werden können. Mittels Modulation wird aus dem bisherigen seriellen Datenstrom ein je nach Wahl QPSK, 16-QAM oder 64-QAM moduliertes Signal erzeugt, welches anschließend auf den Kanal gegeben wird. *Modulation* dient dazu, die spektrale Effizienz eines Übertragungssignals zu erhöhen ohne die Übertragung dabei übermäßig störanfällig werden zu lassen. Der Übertragungskanal – im Folgenden auch einfach als Kanal bezeichnet – ist das zu untersuchende Kernobjekt dieser Diplomarbeit. Ihm kommt auch in praktischen Systemen eine zentrale Bedeutung zu, da sich auf diesem Kanal Störungen überlagern. Sowohl für das Verständnis bestehender Systeme wie auch für die Systemauslegung künftiger Mehrantennensysteme ist ein genaueres Verständnis der Vorgänge auf dem Kanal notwendig. Der unte-

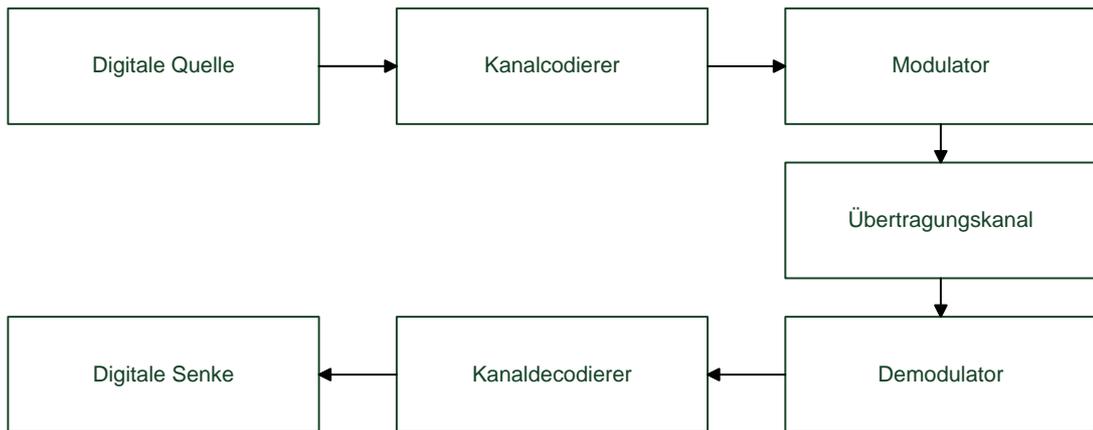


Abbildung 3.1: Digitales Übertragungssystem und Kanalmodell (frei nach [Kü08])

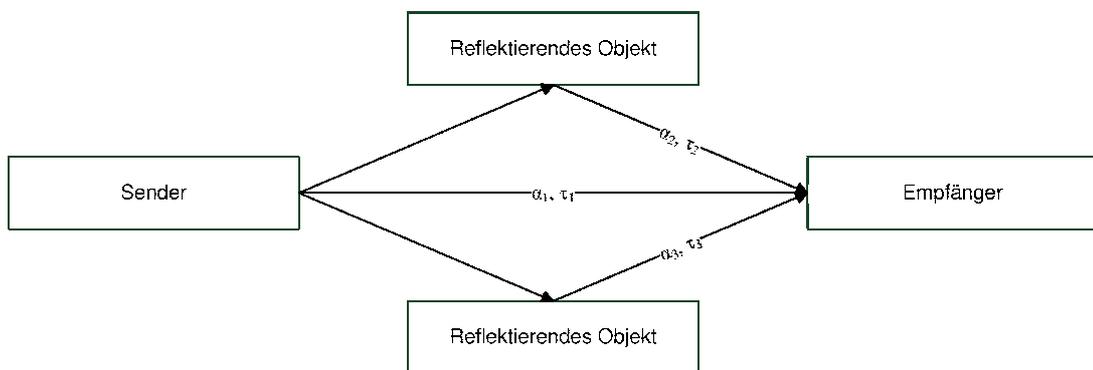


Abbildung 3.2: Mehrwegeausbreitung (angelehnt an [OL07])

re Zweig der Abbildung 3.1 stellt die Empfängerseite dar, wo oben beschriebenen Schritte in umgekehrter Reihenfolge rückgängig gemacht werden müssen, um wieder an das Nutzsignal zu gelangen.

### 3.2 Kanalmodelle und Störungen

Abbildung 3.2 (angelehnt an [OL07]) zeigt den allgemeinen Fall der Mehrwegeausbreitung. Zur Modellierung werden verschiedene Kanalmodelle genutzt, die bestimmte statistische Eigenschaften abbilden. Jedes Modell besitzt konkrete Eigenschaften, während zur Reduktion der Komplexität und damit zur Erhaltung der Berechenbarkeit darüber hinaus gehende Einflüsse bewusst vernachlässigt werden. Im Anhang B des DVB-T-Standard [DVB06] sind die nachfolgend beschriebenen Kanäle standardisiert. Die Abbildungen 3.3 bis 3.5 wurden unter Berücksichtigung der im Standard festgelegten Werte erstellt.

### 3.2.1 Gauß'scher Kanal

Bei dem einfachsten und dennoch sehr gebräuchlichen Kanalmodell, dem *Gauß-Kanal*, wird *nur* Gauß'sches weißes Rauschen der Leistungsdichte  $N_0$  modelliert. In Abbildung 3.2 entspricht dieses, dass nur der Pfad  $\alpha_1, \tau_1$  existiert. Trotz der Beschränkung wird dieses Kanalmodell sehr oft in der Praxis eingesetzt, da es mathematisch gut fassbar ist. Zwar wird das Gesamtrauschen eines Systems in der Regel von vielen verschiedenen überlagerten Rauschprozessen hervorgerufen, oft liefert jedoch die Modellierung des Gesamtrauschens mit Gauß'schem Rauschen hinreichend genaue Ergebnisse [OL07; Rei08b]. In Abbildung 3.3 sind Impulsantwort (Abbildung 3.3a) und zugehörige Kanalübertragungsfunktion (Abbildung 3.3b) dargestellt.

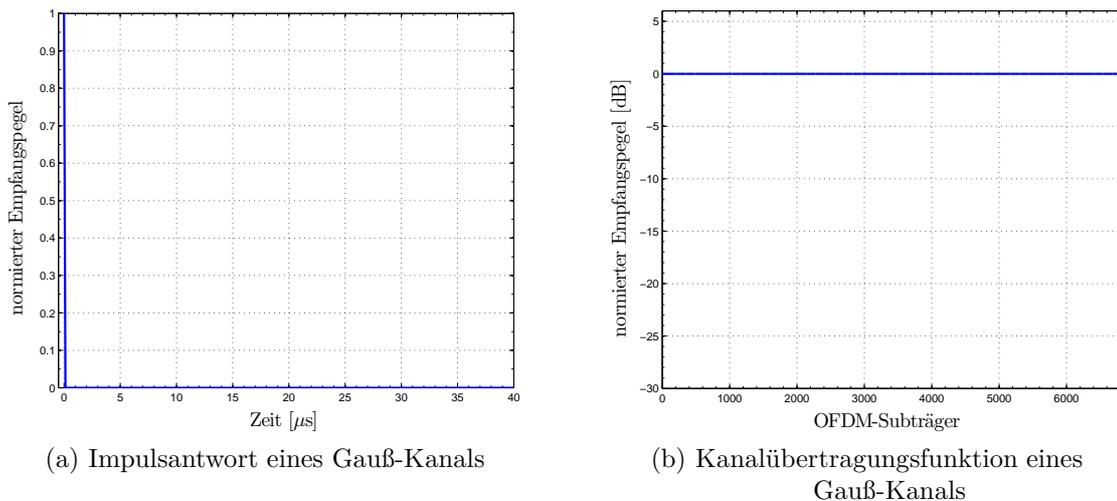
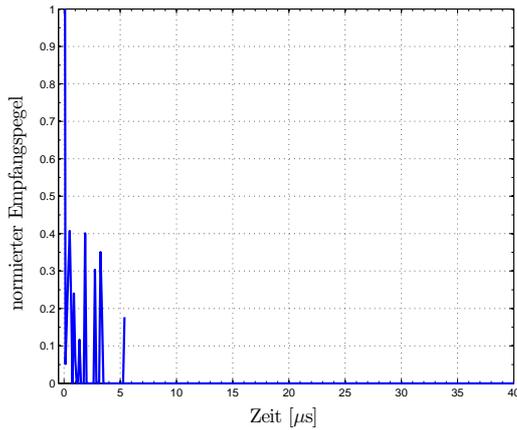


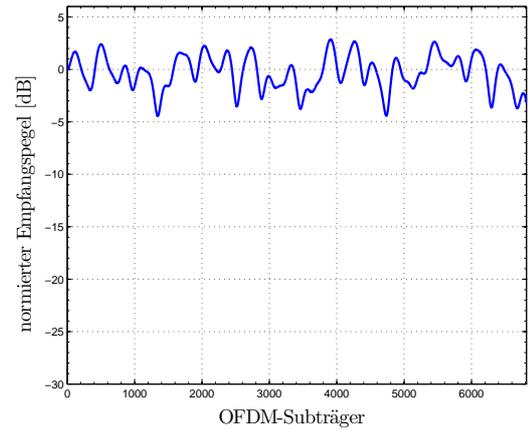
Abbildung 3.3: Gauß-Kanal

### 3.2.2 Rice-Kanal

Der *Rice-Kanal* berücksichtigt neben dem Rauschen zusätzlich Echos, die auf dem Übertragungskanal auftauchen, etwa durch Mehrwegeausbreitung in hügeligem Gelände oder durch Reflexionen an Gebäuden. Die Statistik der Mehrwegesignale folgt dabei einer Rice-Verteilung. Charakteristisch ist für dieses Modell, dass eine direkte Sichtverbindung zwischen Sender und Empfänger vorliegt und somit der Empfangspegel in der direkten Richtung deutlich stärker ist als der aller Reflexionen. Abbildung 3.2 mit allen eingezeichneten Pfaden zeigt einen Rice-Kanal. Dieses Kanalmodell wird typischerweise eingesetzt, wenn eine Dachantenne mit ausgeprägter Haupteinfallrichtung modelliert werden soll [Rei08b], wie auch aus der Impulsantwort in Abbildung 3.4a ersichtlich ist. Abbildung 3.4b zeigt die zugehörige Kanalübertragungsfunktion.



(a) Impulsantwort eines Rice-Kanals

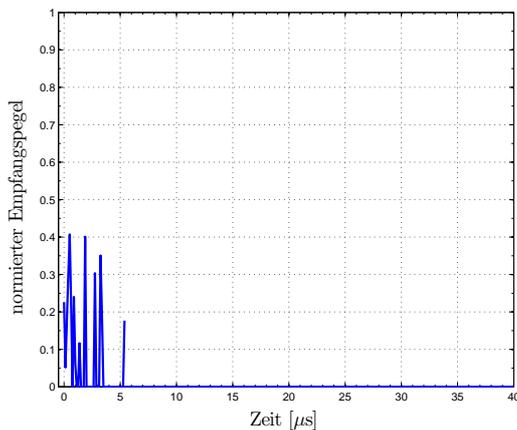


(b) Kanalübertragungsfunktion eines Rice-Kanals

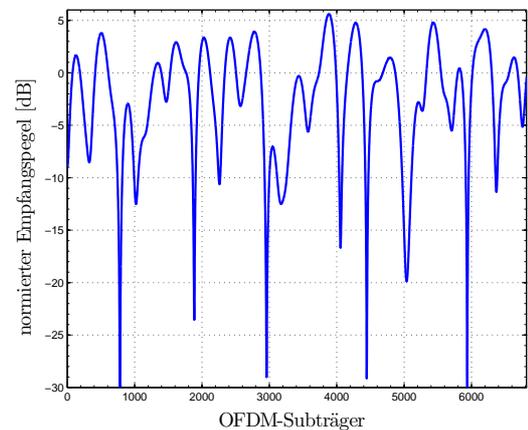
Abbildung 3.4: Rice-Kanal

### 3.2.3 Rayleigh-Kanal

Wird angenommen, dass keine direkte Sichtverbindung zwischen Sender und Empfänger besteht, wodurch die Empfangspegel aller am Empfänger einfallenden Signale in der gleichen Größenordnung liegen, wird vom *Rayleigh-Kanal* gesprochen. Im Modell in Abbildung 3.2 würde entsprechend der Pfad  $\alpha_1, \tau_1$  nicht existieren. Impulsantwort und Übertragungsfunktion sind qualitativ in Abbildung 3.5 dargestellt. Der Rayleigh-Kanal modelliert Umgebungen wie den mobilen Empfang mit dem Handy, welches sich häufig innerhalb von Gebäuden (starke Dämpfung), zwischen Häuserschluchten (Reflexionen, Mehrwegeempfang) oder in Bewegung (Dopplerverschiebung) befindet [Rei08b].



(a) Impulsantwort eines Rayleigh-Kanals



(b) Kanalübertragungsfunktion eines Rayleigh-Kanals

Abbildung 3.5: Rayleigh-Kanal

### 3.3 Übertragungsfunktion und Impulsantwort

Die *Impulsantwort*  $\mathbf{h}(t)$  eines linearen, zeitinvarianten Systems ist der zeitliche Amplituden- und Phasenverlauf, den ein System bei Anregung mit einem *Dirac-Impuls*  $\delta(t)$  zurückgibt. Sie ist hinreichend zur *vollständigen* Charakterisierung eines Systems und insofern eine gute Messgröße für ein digitales Übertragungssystem gemäß Abbildung 3.1 (siehe Seite 21) [OL07]. Die Fouriertransformierte der Impulsantwort ist die Kanalübertragungsfunktion  $\mathbf{H}(f)$ , welche ebenfalls zur Kanalbeschreibung herangezogen werden kann und darüber hinaus aufgrund der Beschaffenheit der Fouriertransformation das frequenzabhängige Verhalten des Übertragungskanals als komplexe Werte abbildet. Aufgrund der verwendeten Mehrfachträger-Modulation OFDM ist insbesondere dieses frequenzselektive Verhalten des Kanals interessant.

Während anhand der Übertragungsfunktion das Verhalten einzelner OFDM-Subträger abschätzbar ist, liefert die inverse Fouriertransformation wieder die Impulsantwort des Kanals. Diese ist für eine zeitliche Analyse des Signals interessant. Beispielsweise lassen sich anhand der Impulsantwort Phänomene wie Echos, Preechos und weitere Sender eines Gleichwellennetzes inklusive der zeitlichen Verzögerungen, mit der dessen Signale eintreffen, auswerten. Aus der Differenz zweier Peaks in der Impulsantwort lässt sich unter Berücksichtigung der Wellenausbreitungsgeschwindigkeit direkt die Entfernung der Ursachen der Ausschläge (Hindernisse etc.) zueinander errechnen. Beispielsweise lassen sich auf beschriebene Weise die Abstände der verschiedenen Sender vom Ort der Messung ermitteln.

### 3.4 Schätzung einer Kanalübertragungsfunktion

Das am Empfänger einfallende Signal  $\mathbf{y}(n, k)$ <sup>1</sup> ergibt sich im Frequenzbereich gemäß [MM96] aus der komplexen Multiplikation des Sendesignals  $\mathbf{x}(n, k)$  mit der Kanalübertragungsfunktion  $\mathbf{H}(n, k)$  und einem additiven Term  $\mathbf{n}(n, k)$ , der weißes, Gauß'sches Rauschen im betrachteten Frequenzband repräsentiert, nach Gleichung 3.1.  $n$  ist der Zeitindex eines OFDM-Symbols,  $k$  bezeichnet die Subträgerposition innerhalb eines Symbols.

$$\mathbf{y}(n, k) = \mathbf{x}(n, k) \cdot \mathbf{H}(n, k) + \mathbf{n}(n, k) \quad (3.1)$$

Um das komplexe Empfangssignal entzerren zu können, wird eine Schätzung der Kanalübertragungsfunktion  $\hat{\mathbf{H}}(n, k)$  benötigt. Schätzungen sind durch das  $\hat{\cdot}$ -Symbol über dem entsprechenden Formelzeichen kenntlich gemacht. Die eigentliche Schätzung der Kanalübertragungsfunktion geschieht klassischerweise durch Auswertung der Pilotsubträger des OFDM-Symbols. Im Frequenzbereich liefert Gleichung 3.2<sup>2</sup>

---

<sup>1</sup>Die Darstellung (Groß- und Kleinschreibung) der Formelzeichen in diesem Unterkapitel folgt der primären Quelle [MM96] dieses Abschnitts.

<sup>2</sup>Variablenamen und Indizes sind aus Konsistenzgründen angepasst, die Ursprungsgleichung ist in [H<sup>+</sup>03], Kapitel 4.3 zu finden.

nach [H<sup>+</sup>03] an den Pilotpositionen eine Kanalschätzung, wobei  $\mathbf{x}(n, k)$  die empfangenen Samples des  $n$ -ten OFDM-Symbols sind und  $\mathbf{P}_i$  die Pilote auf den OFDM-Subträgern  $k, i = 1, \dots, n_k$ .

$$\hat{\mathbf{H}}(n, k) = \frac{\mathbf{x}(n, k)}{\mathbf{P}_i} \quad (3.2)$$

Der Vektor  $\hat{\mathbf{H}}(n, k)$  enthält bisher lediglich Datenwerte ungleich null an den Pilotpositionen, weshalb eine Interpolation zwischen den Pilotsubträgern nötig ist, um die fehlenden Datenwerte zu ermitteln. Dazu können verschiedene Interpolationsverfahren angewendet werden. In dieser Arbeit wird die *lineare Interpolation* verwendet, da sie gute Ergebnisse bei relativ geringem Implementierungsaufwand liefert. Das Problem dieser Kanalschätzung besteht darin, dass die Pilotdichte für eine fehlerfreie Rekonstruktion der *wahren* Übertragungsfunktion gemäß des Shannonschen Abtasttheorems nicht ausreichend hoch ist. Dadurch entstehen laut [OL07] Aliaseffekte im rekonstruierten Signal, welche durch eine nachfolgende, angepasste Tiefpassfilterung entfernt werden müssen. Geeignet ist ein Tiefpass, wenn er alle Aliaseffekte unterdrückt und gleichzeitig keine relevanten Signalanteile zerstört. Die maximale Länge eines solchen Tiefpasses entspricht der Anzahl der Samples, die innerhalb des Guard Intervals liegen, da nach dessen Definition spätestens danach keine Echos mehr auftreten dürfen (vergleiche Abschnitt 2.4.11 auf Seite 19).

Durch die Interpolation ist eine Schätzung der Kanalübertragungsfunktion entstanden, sodass nun die Entzerrung des Empfangssymbols entsprechend Gleichung 3.3 nach [MM96] durchgeführt werden kann.

$$\mathbf{z}(n, k) = \frac{\mathbf{y}(n, k)}{\hat{\mathbf{H}}(n, k)} \quad (3.3)$$

Getrennt nach Betrag und Phase, wobei  $x(n, k)$  die Amplitudenverteilung und  $\Phi(n, k)$  die Phase ist, ergibt sich Gleichung 3.4 mit  $\boldsymbol{\nu}(n, k) = \frac{\mathbf{n}(n, k)}{\hat{\mathbf{H}}(n, k)}$ .

$$\mathbf{z}(n, k) \approx x(n, k) \cdot e^{j[\Phi(n, k)]} + \boldsymbol{\nu}(n, k) \quad (3.4)$$

Das in  $\mathbf{z}(n, k)$  enthaltene, entzerrte Empfangssignal sollte eine problemlose OFDM-Demodulation ermöglichen.

### 3.5 Der CD3-Ansatz und der modifizierte CD3-Algorithmus

*CD3* steht für *Coded Decision Directed Demodulation* und ist ein in [MM96] beschriebener Ansatz von Vittoria Mignone und Alberto Morello zur pilotfreien Kanalschätzung. Da bisherige Verfahren auf einer pilotbasierten Verarbeitung fußen, könnte im Fall von DVB-T eine Einsparung von rund 11 % aller OFDM-Subträger

erreicht werden. Die Idee zur Reduktion des Overheads<sup>3</sup> einer OFDM-Übertragung besteht darin, dass die in Mehrträgerverfahren obligatorische Kanalschätzung nicht mehr symbolweise anhand von bekannten Piloten im Datenstrom durchgeführt wird, sondern unter gewissen Stationaritätsannahmen rekursiv auf dem vorherigen Symbol basiert. Es ist ein bekanntes Initialsymbol  $\mathbf{x}(n = 0, k)$  notwendig, um die Kanaleigenschaften zu einem definierten Zeitpunkt zu ermitteln. Gemäß Gleichung 3.5 nach [MM96] wird eine Übertragungsfunktion  $\hat{\mathbf{H}}_{CD3}(n - 1, k)$  geschätzt.  $\mathbf{x}(n - 1, k)$  ist das vorherige Sendesymbol,  $\mathbf{y}(n - 1, k)$  die vorangegangene Empfangssequenz.

$$\hat{\mathbf{H}}_{CD3}(n - 1, k) = \frac{\mathbf{y}(n - 1, k)}{\mathbf{x}(n - 1, k)} \quad (3.5)$$

Die Schätzung  $\hat{\mathbf{H}}_{CD3}(n - 1, k)$  kann nun unter Verwendung der Gleichung 3.6 (ebenefalls nach [MM96]) zur Normalisierung des OFDM-Symbols  $\mathbf{z}(n, k)$  verwendet werden. Diese Methode ergibt eine bessere Schätzung der Kanalübertragungsfunktion, weil jedem Nutzdatenträger nun die Rolle eines zur Schätzung herangezogenen Subträgers zukommt, was einer zwölfmal höheren Pilotdichte entspricht.

$$\mathbf{z}(n, k) = \frac{\mathbf{y}(n, k)}{\hat{\mathbf{H}}_{CD3}(n - 1, k)} \quad (3.6)$$

Das Problem bei einem Rundfunkübertragungssystem ohne Rückkanal ist jedoch, dass das Initialsymbol zur erstmaligen Kanalschätzung nicht bereitgestellt werden kann, da es keinen definierten Startzeitpunkt der Übertragung gibt. Deshalb wird in dieser Arbeit ein modifizierter Ansatz verwendet. Die Analyse des Kanals findet ohnehin mittels DVB-T konformer Signale statt, die also auch Pilote enthalten. Das Wissen um ein komplettes, definiertes Referenzsymbol kann also getauscht werden gegen die aus den Piloten gewonnene Referenzinformation. Die Empfangssequenz des aktuellen Symbols  $\mathbf{y}(n, k)$  ist ohnehin bekannt, die Referenzsequenz  $\mathbf{x}(n, k)$  wird durch nachfolgend erläuterte Schritte erzeugt. Zunächst wird eine klassische Kanalschätzung anhand der Pilote durchgeführt, wie im Abschnitt 3.4 beschrieben. Die Pilotdichte ist hinreichend groß, um durch einfache lineare Interpolation eine brauchbare Schätzung der Kanalübertragungsfunktion ermitteln zu können, mittels derer das empfangene OFDM-Symbol normalisiert und (bis einschließlich des inneren Fehlerschutzes) decodiert werden kann. Die Nutzdaten werden in  $\mathbf{u}(n, k)$  reencodiert und eine verbesserte Kanalübertragungsfunktion mittels Gleichung 3.7 geschätzt.

$$\hat{\mathbf{H}}(n, k) = \frac{\mathbf{y}(n, k)}{\mathbf{u}(n, k)} \quad (3.7)$$

Durch diesen zweistufigen Algorithmus kann insbesondere durch das stark reduzierte Rauschen eine sehr viel genauere Kanalanalyse erfolgen.

---

<sup>3</sup>Als *Overhead* werden Daten bezeichnet, die für eine Datenübertragung notwendig sind (Übertragungsparameter etc.), jedoch keine Nutzdaten sind. Ein gewisser Overhead ist in der Regel notwendig, jedoch unerwünscht. Deshalb ist es das Ziel, ihn so weit wie möglich zu reduzieren.

### 3.5.1 Verbesserung der CD3 geschätzten Kanalschätzung durch Nutzung nur der äußeren Konstellationspunkte

Zur Verbesserung der CD3-geschätzten Übertragungsfunktion wird ein in [MRM<sup>+</sup>10] vorgeschlagener Ansatz verwendet. Die Idee besteht darin, für die Schätzung der Kanalübertragungsfunktion bei 16-QAM und 64-QAM nur die äußeren Konstellationspunkte zu verwenden, die eine minimale Amplitude überschreiten. Da alle Konstellationspunkte ohnehin für die Reencodierung des OFDM-Symbols bekannt sein müssen, entsteht zusätzlicher Rechenaufwand nur durch die Interpolation der Datenwerte, deren Konstellationspunkte eine kleine Amplitude aufweisen. Die inneren Konstellationspunkte werden mit geringerer Leistung als die weiter außen liegenden ausgestrahlt und sind somit entsprechend störanfälliger. Zu Problemen kommt es dabei insbesondere, wenn ein innerer Konstellationspunkt durch Rauschen in einen falschen Quadranten rutscht und folglich die Interpolation gravierende Schätzfehler erzeugt. Durch Vernachlässigung dieser kritischen Punkte sinkt zwar die Pilotdichte, die zur Schätzung herangezogen werden kann, allerdings kann praktisch ausgeschlossen werden, dass die Interpolation auf Piloten basiert, die fälschlicherweise einem Quadranten zugeordnet wurden, wo sie nicht liegen sollten. In [MRM<sup>+</sup>10] wird gezeigt, dass bei DVB-T2 alleine durch diese Maßnahme in Abhängigkeit von der Modulation und dem Kanalmodell ein Gewinn von 0,5 bis 1,5 dB Signal-Störabstand im Vergleich mit dem ursprünglichen CD3-Ansatz erzielt werden kann.

## 3.6 Filterung der Kanalübertragungsfunktion

Eine lineare Interpolation kann durch Faltung mit einer Dreiecksfunktion realisiert werden. Stattdessen kann auch im Zeitbereich mit einer Rechteckfensterung gearbeitet werden, welche sich durch Nullsetzen (*zero forcing*) aller nicht benötigten oder gewollten Anteile eines Signals realisieren lässt. Theoretisch entspricht diese Multiplikation des Signals mit einem Rechteck im Zeitbereich einer Faltung mit einer *si*-Funktion im Frequenzbereich. Eingangs wurde jedoch erwähnt, dass eine lineare Interpolation durch eine Faltung mit einer Dreiecksfunktion erzeugt werden kann. Der Fehler, der entsteht, ist die Differenz zwischen der Faltung des Signals mit der *si*-Funktion und der Faltung des Signals mit einer Dreiecksfunktion. Im Vergleich zu dem Effizienzgewinn, den *zero forcing* in Verbindung mit einer IFFT und einer FFT mit sich bringt, ist der entstandene Fehler vernachlässigbar gering. In Abbildung 3.6 sind *si*- und Dreiecksfunktion übereinander gelegt, um den geringen Fehler zu veranschaulichen.

## 3.7 Probleme der linearen Interpolation

Die lineare Interpolation im Frequenzbereich, die zur Schätzung der Kanalübertragungsfunktion verwendet wird, entspricht im Zeitbereich einer Multiplikation mit einer *si*<sup>2</sup>-Funktion. [Mak09] zeigt, dass die Nulldurchgänge der *si*<sup>2</sup>-Funktion im *Single*

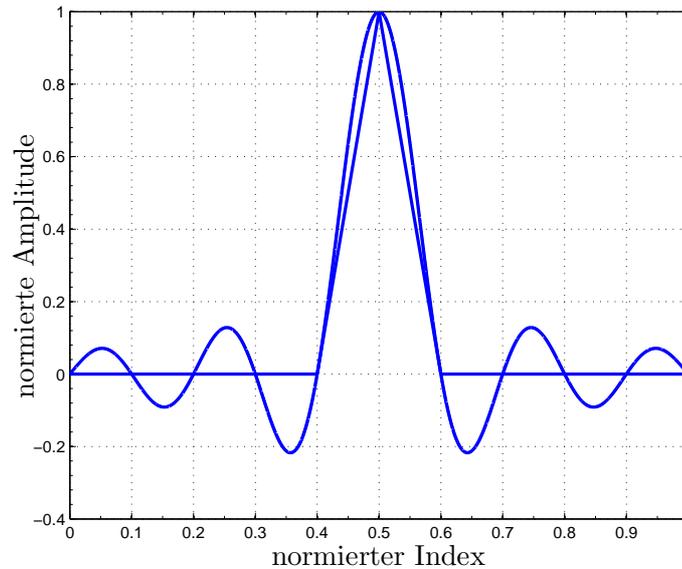


Abbildung 3.6: Vergleich si-Funktion und Dreiecksfunktion

*Input, Single Output*-Fall (SISO) genau an den Stellen der Aliasse liegen. Probleme treten jedoch bei Mehrantennensystemen (MISO) auf, da nicht alle Signale zur gleichen Zeit eintreffen und deshalb Störungen verursachen können, die nicht durch die Multiplikation mit der  $\text{si}^2$ -Funktion entfernt werden. Dennoch, konstatiert [Mak09], sei die lineare Interpolation besser geeignet als eine Konstantwertfortsetzung (Multiplikation mit einer si-Funktion im Zeitbereich), wenngleich auch weniger gut als eine FIR-Filterung<sup>4</sup>. Letztere nähert sich zwar am besten der wahren Übertragungsfunktion an, erfordert allerdings auch einen größeren Implementierungs- und Rechenaufwand. Zusammenfassend kann die lineare Interpolation also als guter Kompromiss aus Rechenaufwand und Ergebnisqualität angesehen werden, weshalb sie auch zur Kanalschätzung im verwendeten MATLAB-Code eingesetzt wird. Darüber hinaus wird nachfolgend ohnehin unter Verwendung des modifizierten CD3-Algorithmus eine verbesserte Kanalanalyse durchgeführt.

### 3.8 Die Restträgerproblematik

Ein Restträger ist ein ungewolltes Mischprodukt der analogen Modulation, die durch die verwendete Zwischenfrequenz hervorgerufen wird und auch bei digitaler Modulation als unerwünschtes Nebenprodukt entsteht. Der Restträger kann durch verschiedene Techniken unterdrückt werden, die je nach Umsetzung verschieden erfolgreich sind. Abhängig von der Güte des Empfängers kommen auf dem mittleren Träger (bzw. in ungünstigen Fällen einer minimalen Frequenzabweichung von der Sollfrequenz auch

---

<sup>4</sup>*FIR* steht für *Finite Impulse Response* („endliche Impulsantwort“) und bezeichnet eine Klasse nichtrekursiver Filter.

auf *den* mittleren Trägern) ungewollte Störeffekte hinzu. Bei einer Schätzung basierend auf den Piloten sollte sichergestellt sein, dass kein Pilot auf dem Mitterträger liegt. Leider ist bei DVB-T jedoch genau das der Fall. Je nach Übertragungsmodus liegt in jedem vierten OFDM-Symbol ein Scattered Pilot (2K-FFT-Modus) oder in jedem Symbol ein Continual Pilot (8K-FFT-Modus) auf dem Mitterträger [Gru02; DVB06].

Bei der pilotbasierten Schätzung der Kanalübertragungsfunktion muss ein eventuell vorhandener Restträger berücksichtigt werden. Beispielsweise kann der vom Restträger überlagerte Subträger *vor* der Interpolation zwischen den Piloten entfernt werden. Durch den entstehenden größeren Abstand, über den im Frequenzspektrum interpoliert werden muss, können weitere Fehler auftreten, die verglichen mit dem zerstörten Träger jedoch weniger kritisch sind. Eine Illustration findet sich im Kapitel 4.4.9 in der Abbildung 4.6 auf Seite 42.

# 4 Implementierung

## 4.1 Erstellung der MATLAB-Software

In diesem Kapitel wird die Entwicklung des MATLAB-Codes beschrieben. Es wird auf Probleme und Lösungen der verschiedenen Entwicklungsschritte (Synchronisation, Korrektur von Empfängerungenauigkeiten, ...) eingegangen und dargelegt, warum der gewählte Weg eine adäquate Verarbeitung der Daten gewährleistet. Der Anwendung der Software, die zur Kanalanalyse nach Abschluss dieser Arbeit weiterverwendet werden soll, ist das Kapitel 5 ab Seite 53 gewidmet, so dass der Fokus an dieser Stelle auf die eigentlichen Programmbestandteile und deren Verkettung zu einem lauffähigen Programm gelegt wird. Der Quellcode ist aus Konventionsgründen komplett in englischer Sprache verfasst.

## 4.2 Hauptprogramm

Der Aufbau des Hauptprogramms lässt sich in wenige Abschnitte gliedern. Im ersten werden vor Programmstart Benutzereingaben im Quelltext getätigt. In der nächsten Codesektion werden aufbauend auf den gewählten Einstellungen globale Parameter für den Programmablauf initialisiert, beispielsweise wird die Pilotstruktur passend zum FFT-Modus gewählt. Das eigentliche Hauptprogramm besteht im Wesentlichen aus einer `for`-Schleife, welche sooft durchlaufen wird, wie OFDM-Symbole verarbeitet werden sollen. Es ruft die Unterprogramme als MATLAB-Skripte<sup>1</sup> oder MATLAB-Funktionen<sup>2</sup> auf. Nach erfolgreicher Kanalschätzung werden die Ergebnisse wahlweise in Dateien geschrieben oder die im Arbeitsspeicher (RAM) berechneten Werte werden graphisch dargestellt. Zur Verarbeitung größerer Datenmengen ( $\geq 100$  OFDM-Symbole) wie sie insbesondere für zeitvariante Kanäle von Interesse sind, müssen

---

<sup>1</sup>MATLAB-Skripte sind Quellcode enthaltende `.m`-Textdateien, welche durch Aufruf ihres Namens in ein MATLAB-Programm eingebunden werden können. Der Quellcode wird von dem aufrufenden Programm so verarbeitet, als wenn er direkt in diesem stehen würde. MATLAB-Skripte dienen also nur der Strukturierung und Übersichtlichkeit eines komplexen Programms. Ihr Vorteil liegt jedoch darin, dass sie direkt auf alle im aufrufenden Programm vorhandenen Variablen zugreifen können.

<sup>2</sup>MATLAB-Funktionen beginnen mit dem Schlüsselwort `function` und benötigen Übergabewerte, wenn sie auf Daten aus dem aufrufenden Programm zurückgreifen sollen. Sie werden ebenso wie MATLAB-Skripte in `.m`-Textdateien gespeichert. Sie funktionieren wie beispielsweise C-Funktionen und stehen wie die eingebauten MATLAB-Funktionen auch über den MATLAB-Befehlsprompt zur Verfügung, sofern der sie enthaltende Ordner in den MATLAB-Pfaden vorhanden ist.

aufgrund aktueller Arbeitsspeichergrößen von wenigen Gigabyte die Kanalübertragungsfunktionen auf der Festplatte (HDD) gespeichert werden. Nach erfolgreichem Programmabschluss wird der reservierte Speicher wieder freigegeben, so dass eine Weiterverarbeitung in MATLAB oder anderen Tools möglich wird.

### 4.3 Die Datenstruktur `symb`

MATLAB stellt viele nützliche Funktionen wie eine Fast Fourier Transformation (FFT) bereits von Haus aus bereit, was es für den Einsatz in der digitalen Nachrichtentechnik geradezu prädestiniert. Leider fehlen der MATLAB-Programmiersprache einige wichtige Funktionalitäten wie beispielsweise Zeiger (Pointer). Dieser Umstand impliziert, dass sämtliche Werte, die sowohl im Hauptprogramm wie auch innerhalb einer Unterfunktion benötigt werden, entweder als Übergabeparameter hin- und wieder zurückgegeben werden müssen, oder dass eine spezielle Datenstruktur – ein sogenanntes *Structure Array* – angelegt wird, welche alle relevanten Informationen speichert. In diesem Fall muss nur diese Datenstruktur als Übergabe- und Rückgabeparameter angegeben werden. Im Hinblick auf einen strukturierteren Quelltext wird deshalb der Nachteil in Kauf genommen, dass einige Unterfunktionen aufgrund der erwarteten Structure Array-Felder nicht mehr ohne Anpassung für allgemeine Zwecke verwendet werden können, was wegen des hohen Spezialisierungsgrades der Unterfunktionen jedoch kaum negativ ins Gewicht fällt. Felder innerhalb einer Datenstruktur können durch Angabe des Strukturnamens, gefolgt von einem Punkt und dem Feldnamen adressiert werden. Die Adressierung der Elemente innerhalb eines Feldes folgt den üblichen MATLAB-Konventionen.

Es gibt zwei Datenstrukturen, auf die das Hauptprogramm aufbaut. Die erste, namens `params`, enthält statische Werte. Diese Struktur wird innerhalb des MATLAB-Skripts `global_parameters` erzeugt und enthält unveränderliche Werte wie die Pilotträgerpositionen innerhalb eines OFDM-Symbols, die verwendete Modulation, die Coderate, Informationen zur Länge des Guard Intervals, die Scrambling-Sequenzen etc.. Abgesehen von einem lokalen Zähler werden die Werte zur Programmlaufzeit nicht mehr verändert.

Die eigentlich interessierenden Daten werden in der Datenstruktur `symb` gespeichert. Sie ist graphisch in Abbildung 4.1 auf Seite 32 dargestellt. Die Bezeichner sollten soweit selbsterklärend sein, dennoch soll die Bedeutung einzelner Felder hier im Detail besprochen werden. Die Länge der Datenfelder variiert in Abhängigkeit des gewählten Modus (2K-FFT oder 8K-FFT), angegeben ist jeweils der reservierte Speicher für den 8K-FFT-Modus, da dieser in Deutschland Verwendung findet. Die Werte in Klammern geben die Speicherlänge für den 2K-FFT-Modus an. Wenn alle Daten im Arbeitsspeicher verarbeitet werden sollen, werden auch *alle* Zwischenergebnisse im RAM verarbeitet. Ansonsten werden jeweils zehn Symbole verarbeitet, bevor die Zwischenergebnisse auf die Festplatte geschrieben werden. Die blockweise Verarbeitung wird verwendet, weil Operationen wie eine zweidimensionale Filterung in MATLAB sehr viel effektiver berechnet werden als entsprechend häufige Anwen-

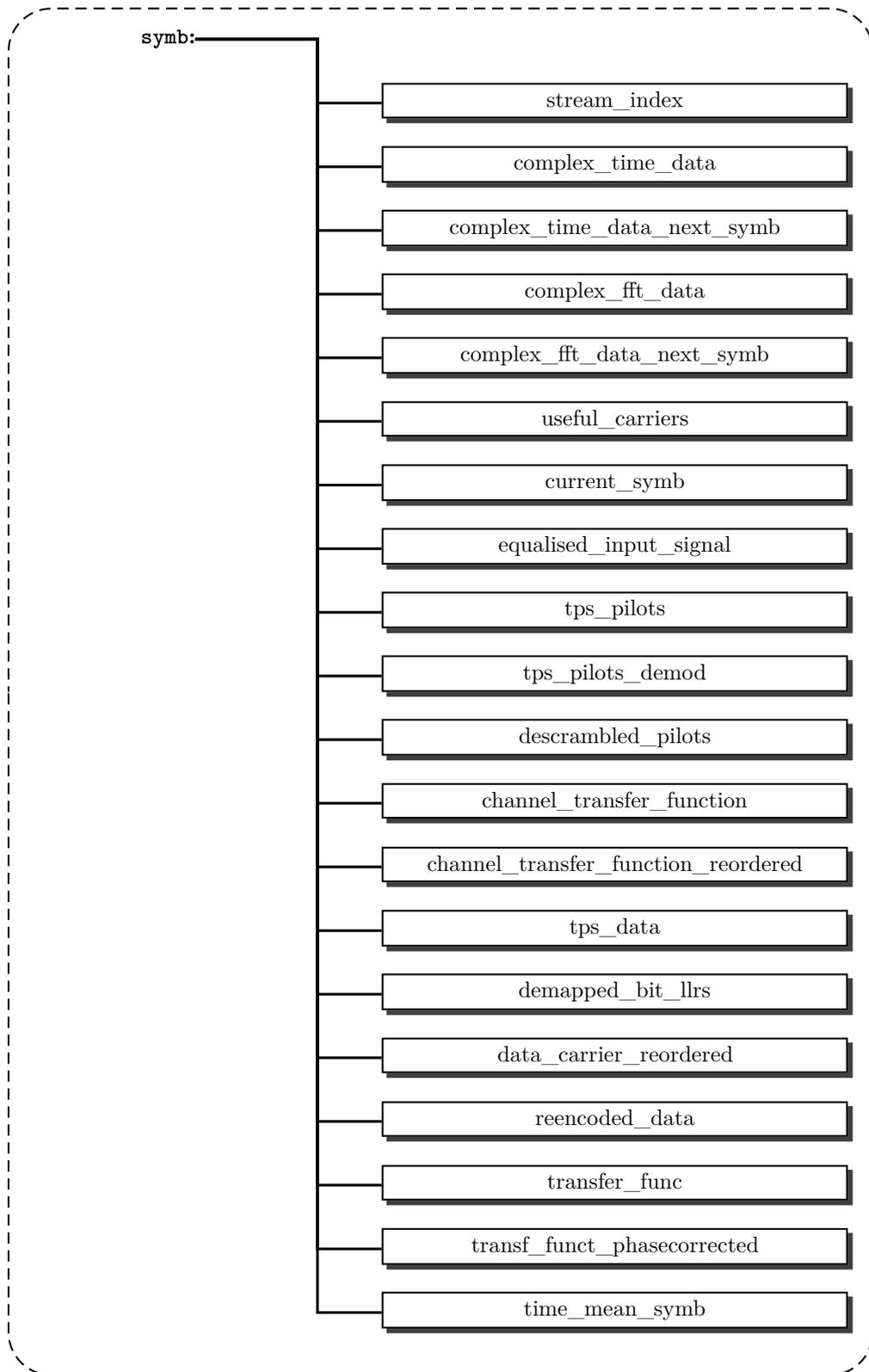


Abbildung 4.1: Die Datenstruktur des Structs `symb`

dung einer eindimensionalen Filterung zur Glättung der Kanalübertragungsfunktion. Darüberhinaus wird die Anzahl der im Vergleich zur Arbeitsspeichergeschwindigkeit langsamen Schreibzugriffe auf die Festplatte reduziert.

Der `stream_index` enthält die globale Startposition der OFDM-Symbole innerhalb des Eingabedatenstroms. Da zum Programmstart die Startposition des ersten OFDM-Symbols noch nicht bekannt ist, kann hier ein unregelmäßiger Sprung auftreten. Davon abgesehen liegen die Startpositionen jeweils  $T_N + T_G \pm PLL\text{-Ausgleich}$  Samples auseinander, wobei  $T_N + T_G$  die Anzahl der Eingangsdatenstromsamples in zeitlicher Richtung darstellt und  $PLL\text{-Ausgleich}$  einen Korrekturterm.  $T_N$  entspricht  $T_{Nutz}$  aus Abbildung 2.1 auf Seite 3 und  $T_G$  ist eine davon und von der Länge des Guard Intervals abhängige Zeitspanne. Im Falle des in Deutschland verwendeten Guard Intervals der Länge  $\frac{1}{4}$  ergibt sich also ohne  $PLL\text{-Ausgleich}$  eine Differenz von  $8192 + \frac{1}{4} \cdot 8192 = 10240$  ( $2048 + \frac{1}{4} \cdot 2048 = 2560$ ) Samples zwischen den Startpositionen. Es werden die Startpositionen aller Symbole – bzw. von bis zu zehn verarbeiteten Symbolen – in der Struktur gespeichert. Die Felder `complex_time_data` und `complex_time_data_next_symb` speichern das komplexe Signal im Zeitbereich. Ersteres enthält alle bzw. bis zu zehn Datensätze der Länge  $T_N + T_G$ , also 10240 (2560). Es wird insgesamt Speicher für  $n \times (T_N + T_G)$  komplexe Werte benötigt, wobei  $n$  die Anzahl aller zu verarbeitenden Symbole bzw. 10 ist. `complex_time_data_next_symb` wird in jedem Schleifendurchlauf überschrieben, so dass nur Platz für  $1 \times 10240$  respektive  $1 \times 2560$  komplexe Datensätze vorgehalten werden muss. Die Felder `complex_fft_data` sowie `complex_fft_data_next_symb` beinhalten die Fouriertransformierten der eben besprochenen Felder, die MATLAB mittels des `fft`-Befehls nahezu instantan berechnen kann. Dementsprechend sind sie genauso groß. Die komplette weitere Signalverarbeitung wird ausschließlich im Frequenzbereich durchgeführt, weshalb es durchaus sinnvoll ist, diese äquivalente Repräsentation des Signals erneut abzuspeichern. Zudem wird die Leserlichkeit des Quellcodes durch den direkten Zugriff auf die Fouriertransformierte deutlich erhöht. In der Variablen `useful_carriers` werden die 6817 (1705) aktiven OFDM-Subträger gespeichert, also laut [DVB06] alle Träger inklusive der Pilote, die verwertbare Informationen enthalten und nicht als Schutzintervall o. ä. dienen. Das Datenfeld `current_symb` speichert die aktuelle Symbolkonstellation der Scattered Pilots, also pro OFDM-Symbol nur eine Zahl zwischen 1 und 4. Das `equalised_input_signal` enthält das mit der geschätzten Übertragungsfunktion aus `channel_transfer_function` gemäß Gleichung 4.1 entzerrte Eingangssignal<sup>3</sup>.

$$\text{entzerrtes Signal} = \text{Eingangssignal} / \text{Kanalübertragungsfunktion} \quad (4.1)$$

Während in `tps_pilots` die empfangenen und normalisierten Werte aller TPS Pilots gespeichert werden, enthält das Feld `tps_pilots_demod` alle demodulierten TPS-Bits. Da TPS Daten innerhalb eines OFDM-Symbols immer alle gleich sein müssen, genügt eine Mehrheitsentscheidung zur Auswertung des TPS-Bits, deren Ergebnis in

---

<sup>3</sup>Der MATLAB-Operator „./“ bedeutet „punktweise Division“ (ohne mathematische Entsprechung).

`tps_data` gespeichert wird. Im Datenfeld `descrambled_pilots` werden die descrambelten Scattered Pilots gespeichert, also die Pilote, so wie sie zur direkten Weiterverarbeitung genutzt werden können. Da das DVB-T-Signal vor der Ausstrahlung zur Erhöhung der Robustheit gegen Fehler verwürfelt worden ist, muss im Empfänger diese Verwürfelung wieder rückgängig gemacht werden. Das Ergebnis wird in den beiden Variablen `data_carrier_reordered` und `channel_transfer_function_reordered` festgehalten. Beide können jeweils alle Nutzdatenträger aufnehmen, haben also eine Länge von 6048 (1512) Datenwerten. Die Größe der Variable `demapped_bit_llrs` ist abhängig von der verwendeten Modulationsart, da sie einen *Log-Likelihood Ratio*-Wert für jedes Nutzbit enthält. Sie berechnet sich gemäß Gleichung 4.2 und ist demnach bei der in Deutschland eingesetzten 16-QAM 24192 (6048) LLR-Werte lang.

$$\text{Länge}_{\text{demapped\_bit\_llrs}} = \text{Anzahl Nutzdatenträger} \cdot \frac{\text{Bits}}{\text{OFDM-Subträger}} \quad (4.2)$$

Die erneut codierten Daten müssen natürlich die gleiche Länge aufweisen wie die Empfangsdaten, weshalb das Feld `reencoded_data`, genauso wie die mittels des modifizierten CD3-Algorithmus geschätzte Übertragungsfunktion `transfer_func`, erneut eine Länge von 6817 (1705) aufweist. Die anschließende Phasenkorrektur, deren Ergebnis in `transf_func_phasecorrected` zu finden ist, sowie auch Filterung in Zeitrichtung, deren Ergebnis in `time_mean_symb` steht, ändert ebenfalls nichts an der Länge. Letzteres Feld wird auch in die Ausgabedatei geschrieben.

## 4.4 Programmablauf

Der komplette Programmablauf ist in der Abbildung 4.2 auf Seite 35 graphisch dargestellt. Die Darstellung dient als Übersicht für die folgende Programmbeschreibung. In diesem Abschnitt soll nicht jede Codezeile erklärt werden, sondern es sollen die Funktionsweisen der einzelnen Unterprogramme (MATLAB-Skripte und MATLAB-Funktionen) erläutert werden. Auf theoretische Aspekte soll an dieser Stelle so weit wie möglich verzichtet werden, da jene bereits in den vorigen Kapiteln dargelegt wurden.

### 4.4.1 Programmstart

Zu Beginn eines jeden Programmablaufes werden die im Quellcode eingestellten Benutzereingaben verarbeitet (Initialisierung) und soweit wie möglich auf Plausibilität überprüft. Der Eingangsdatenstrom wird geladen, die ersten Samples wahlweise verworfen (beispielsweise wenn nicht anders sichergestellt werden kann, dass zu Beginn der Aufzeichnung des Datenstroms das Messgerät bereits eingeschwungen ist) und Ausgabedateien – so gewählt – angelegt. Im nächsten Schritt werden die im MATLAB-Skript `global_parameters` definierten Vorgaben geladen, welche direkt anhand des Standards [DVB06] implementiert wurden.

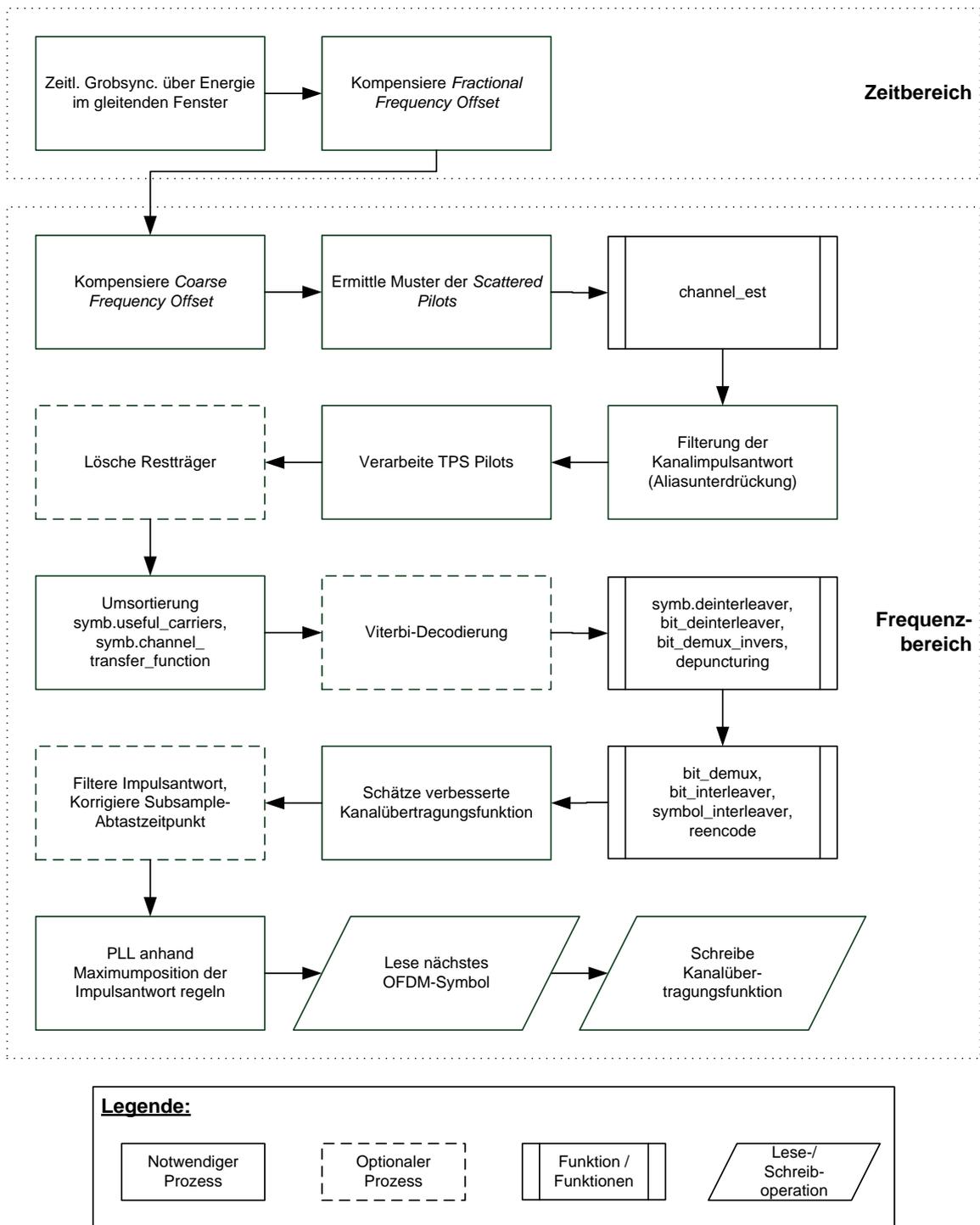


Abbildung 4.2: Blockdiagramm des Hauptprogramms

## 4.4.2 Grobsynchronisation mittels eines gleitenden Fensters

Die schwierige initiale Aufgabe besteht darin, dass der Empfänger, in diesem Fall die Analysesoftware, auf den Datenstrom synchronisiert werden muss. Eine exakte Synchronisierung ist wegen der verwendeten OFDM-Modulation notwendig, die voraussetzt, dass zu den Abtastzeitpunkten alle Signale außer dem aktiven Subträger im Frequenzbereich null sind (vergleiche Kapitel 2.1 ab Seite 2). Schon kleine Synchronisationsabweichungen führen zu verschiedenen Störeffekten, die später näher betrachtet werden sollen, da sie auf dem Weg zu einer hinreichend genauen Synchronisation auf den Datenstrom auftraten. Es werden mindestens zwei Synchronisationsstufen benötigt.

Da zum Einschaltzeitpunkt keine Information über die Startposition eines OFDM-Symbols im empfangenen Datenstrom vorliegt, muss als erstes eine zeitliche Synchronisation erfolgen. Eine mögliche Implementierung in MATLAB ist die Berechnung mittels zweier Fenster, die aus dem Datenstrom jeweils  $T_G$  Samples in konstantem Abstand der Länge  $T_N$  ausschneiden, wie es bereits im Prinzipschaubild im Kapitel „2.1.1.1 Zeitsynchronisation“ (insbesondere Abbildung 2.2 zum gleitenden Fenster auf Seite 5) verdeutlicht worden ist. Die Summe der punktweisen Multiplikation der Werte des einen Fensters mit den konjugiert komplexen Werten des zweiten Fensters ergibt die Kreuzkorrelation der beiden Fenster. Zur Erhöhung der Robustheit kann eine Korrelation über mehrere OFDM-Symbole durchgeführt werden. Die Anzahl der zu verwendenden Symbole kann dem Programm als Parameter mitgeteilt werden, voreingestellt ist eine Mittelung über vier Symbole. Es ergibt sich ein Bild wie in Abbildung 4.3 gezeigt. Deutlich zu erkennen sind die vier Ausschläge der Korrelationsfunktion. Dargestellt ist ein in einem realen DVB-T-Netz gemessenes Signal (8K-FFT-Modus, 8 MHz-Kanal, 16-QAM, Guard Interval-Länge  $\frac{1}{4}$ , Coderate  $\frac{2}{3}$ ) [H<sup>+</sup>03].

## 4.4.3 Kompensation des Fractional Frequency Offsets

Die im Kapitel 4.4.2 beschriebene Methode bietet eine Synchronisation auf einen Subträger  $\Delta f_I$  genau, die nach [SFFM01] ein ganzzahliges Vielfaches des Subträgerabstandes  $\frac{1}{T_u}$  ist. Es verbleibt ein gebrochen rationaler Frequenzversatz  $\Delta f_F$ , welcher zum Verrutschen der korrekten Abtastfrequenz und damit zu Intercarrier-Interferenz (ICI) führen kann. Dieser Versatz wird durch Phasenrauschen verursacht und kann dadurch korrigiert werden, dass der Winkel des (komplexen) Mittelwertes  $\delta_F$  aller Werte innerhalb des gleitenden Fensters an der Stelle des maximalen Peaks ermittelt wird und das komplette Symbol im Frequenzbereich verschoben wird. Programmtechnisch ist diese Verschiebung als Multiplikation mit  $e^{j\delta_F}$  realisiert, so dass sich folgende Korrekturgleichung als MATLAB-Code ergibt:

$$\begin{aligned} & \text{symb.complex\_time\_data}(params.k, 1 : params.T_N) \\ & = \text{symb.complex\_time\_data}(params.k, 1 : params.T_N) \\ & \quad \cdot e^{(1j \cdot \delta_F ./ params.T_N .* (1:params.T_N))} \end{aligned} \quad (4.3)$$

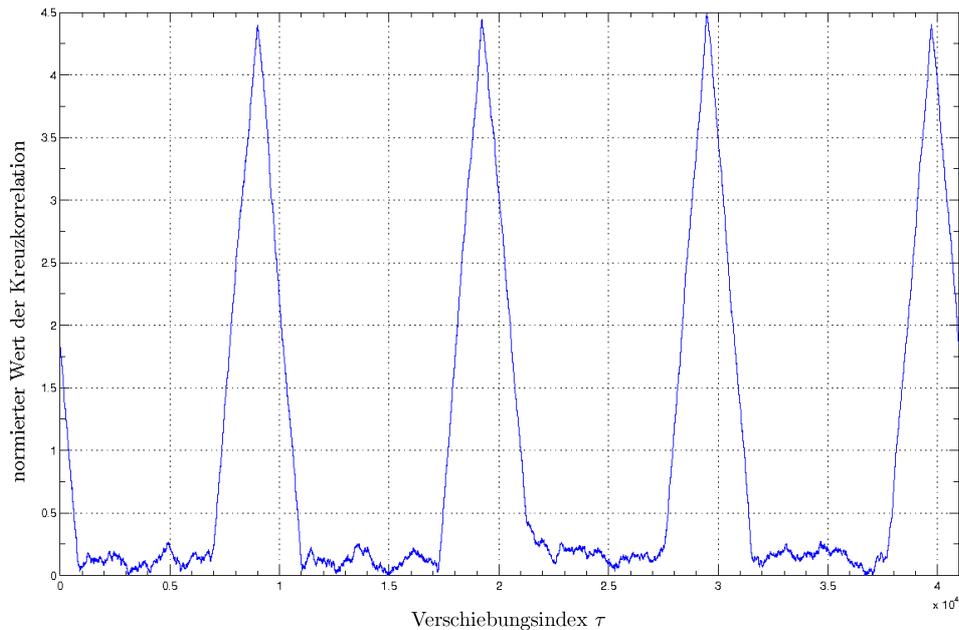


Abbildung 4.3: Korrelationsfunktion der gleitenden Fenster für Guard Interval-Synchronisation (Freifeldmessung im Säulengang am Audimax der TU, 8K-FFT-Modus, 8 MHz-Kanal, 16-QAM, GI  $\frac{1}{4}$ , CR  $\frac{2}{3}$ ).

Eine exaktere Synchronisation ist an dieser Stelle des Programms nicht notwendig.

#### 4.4.4 Kompensation des Coarse Frequency Offsets

Zur Frequenzsynchronisation werden die Continual Pilots herangezogen. Die MATLAB-Funktion `frequency_sync` in der ersten Version (v1.0) multipliziert dazu das aktuelle OFDM-Symbol im Frequenzbereich mit dem konjugiert komplexen nächsten OFDM-Symbol und speichert das Ergebnis in einer lokalen Variable namens `correl`. Der Start eines OFDM-Symbols im Frequenzbereich wird nun durch „Versuch und Irrtum“ (*trial and error*) gefunden. Es wird schrittweise für den Bereich zwischen dem ersten Frequenzträger und dem  $T_N - active\_carriers = 8192 - 6817 = 1375$  (8K-FFT-Modus) bzw.  $2048 - 1705 = 343$  (2K-FFT-Modus) Frequenzträger in `correl` angenommen, dass an der aktuellen Position der tatsächliche Start des OFDM-Symbol in Frequenzrichtung liegt. In jedem Schritt wird der verschobene Datenstrom mit der Variablen `params.continual_pilots_positions`, welche die Trägerpositionen der Continual Pilots enthält, punktweise multipliziert und die Summe über das jeweilige Ergebnis gebildet. Diejenige Position, an der die Summe maximal wird, muss den korrekten Start des OFDM-Symbols anzeigen. Auf diese Weise kann nach [SFFM01] der sogenannte *Coarse Frequency Offset* („grober Frequenzversatz“) zur Vermeidung von Intercarrier-Interferenz korrigiert werden. In der Version 1.1 der Funktion `frequency_sync` konnte die Programmausführung alleine dadurch um 30% (!) beschleunigt werden, dass Variablen, deren Feldelementanzahl innerhalb von Schleifen

wachsen, vor Beginn der Schleife allokiert werden. Dazu muss die maximale benötigte Anzahl Felder bekannt sein, was in diesem Fall auch gewährleistet ist, da der Suchbereich eine klar definierte Größe hat. Die Version 1.2 enthält eine Fehlerbereinigung, die aufgrund falscher Indizes bei bestimmten Symbolen zu einer fehlerhaften Frequenzsynchronisation geführt hat. Da MATLAB bei der Verarbeitung von `for`-Schleifen bekanntermaßen eine schlechte Performance zeigt, wird in der finalen Version 1.3 schließlich die Leistungsfähigkeit erneut optimiert, indem nur beim ersten Funktionsaufruf der gesamte mögliche Synchronisationsbereich durchlaufen wird. In den folgenden Aufrufen wird unter der Annahme der Stationarität nur noch eine lokale Umgebung um die im letzten OFDM-Symbol gefundene Synchronisierungsstelle abgesucht.

#### 4.4.4.1 Versuch der Verbesserung der Frequenzsynchronisierung durch Einbeziehung der TPS Pilots

In der Funktion `frequency_sync` sollte die Robustheit der Erkennung auch für schlechte Signalpegel dadurch gesteigert werden, dass zusätzlich zu den Continual Pilots auch die TPS Pilots für die Frequenzsynchronisation herangezogen werden. Aufgrund der Fehlerverteilung und der verhältnismäßig wenigen TPS Pilots in einem OFDM-Symbol konnte jedoch keine Verbesserung der Synchronisationsfähigkeit erreicht werden, weshalb der Ansatz zugunsten einer zügigeren Verarbeitung wieder verworfen wurde.

#### 4.4.5 Ermittlung der Scattered Pilots Konfiguration

Für die Decodierung der Nutzdaten muss der Empfänger die Konfiguration der *Scattered Pilots* des aktuellen OFDM-Symbols kennen. Dafür wird die Funktion `detect_scattered_pilot_constell` aufgerufen. Innerhalb eines OFDM-Symbols ist jeder zwölfte OFDM-Subträger ein Scattered Pilot (vergleiche Abbildung 2.10 auf Seite 16 zur DVB-T-Pilotstruktur), nur die Position des ersten Pilots innerhalb des OFDM-Symbols variiert zyklisch zwischen vier möglichen Startstellen. Zur Bestimmung der aktuellen Konfiguration genügt also schon die Summation jedes zwölften Trägers wie im Codeauszug im Listing 4.1 dargestellt. Die Bildung des Quadrates der Summen hebt den Unterschied zwischen der korrekten Konfiguration und den drei falschen stärker hervor.

Listing 4.1: Auszug aus `detect_scattered_pilot_constell.m`

```

1 | for n = 1:3:12
2 |     pilot_pos(n) = sum(abs(symb.useful_carriers(params.k,n:12:end)).^2);
3 | end;

```

Da MATLAB-Nummerierungen mit eins (und nicht mit null) beginnen, findet am Ende der Funktion noch eine Symbolumsetzung statt, so dass die Symbolnummer null des DVB-Standards [DVB06] im Programm der Konstellationsnummer eins entspricht, um MATLAB-Konformität herzustellen. Das Listing 4.2 zeigt den notwendigen Quellcode.

Listing 4.2: Auszug aus `detect_scattered_pilot_constell.m`

```

1 | symb.current_symb(params.k)...
2 | = uint32( (symb.current_symb(params.k)-1)/3+1 );

```

Es genügt eine einmalige Synchronisation, wenn sie hinreichend robust ist. Im späteren Programmablauf kann aufbauend auf der ermittelten Startkonfiguration jeweils das Konfigurationsmuster um eins weiter gezählt werden. Bei Verarbeitung der ersten vier OFDM-Symbole im Datenstrom wird wie beschrieben die Funktion `detect_scattered_pilot_constell` verwendet, bei dem fünften Symbol wird die Funktion `analyse_scattered_pilot_constellations` verwendet um aus den zuvor unabhängig ermittelten Lagen der Scattered Pilots die wahrscheinlichste Konfiguration für das aktuelle Symbol abzuleiten. Nachfolgend wird das Pilotmuster für jedes weitere OFDM-Symbol zyklisch zwischen eins und vier weitergesetzt.

#### 4.4.6 Pilotbasierte Kanalschätzung

Das Codelisting 4.3 stellt einen Auszug aus der Funktion `channel_est` dar. Die im vorigen Schritt ermittelten Scattered Pilots werden durch diese Funktion ausgeschnitten und in der Variablen `symb.descrambled_pilots` gespeichert (Zeilen eins bis vier). Das eigentliche Descrambeln wird in den Zeilen sechs bis neun mit einer vorberechneten Sequenz durchgeführt, die in der Variablen `params.scrambling_sequence` gespeichert ist; die Zeilen elf und zwölf normalisieren die Leistung und speichern das Ergebnis in der Variablen `symb.channel_transfer_function`.

Listing 4.3: Ausschneiden und Descrambling der Scattered Pilots (Auszug aus `channel_est.m`)

```

1 | symb.descrambled_pilots(...
2 | params.k,(symb.current_symb(params.k)-1)*3+1:12:end)...
3 | = symb.useful_carriers(...
4 |   params.k,(symb.current_symb(params.k)-1)*3+1:12:end);
6 | symb.descrambled_pilots(params.k,:)...
7 | = symb.descrambled_pilots(params.k,:)...
8 |   .* ((0.5 - params.scrambling_sequence(...
9 |     1:params.active_carriers)) * 2).';
11 | symb.channel_transfer_function(params.k,:)...
12 | = symb.descrambled_pilots(params.k,:) / (4/3);

```

#### 4.4.7 Interpolation durch Filterung der Kanalübertragungsfunktion

Die ausgeschnittene Scattered Pilots-Struktur wie sie zunächst in `symb.channel_transfer_function` vorliegt, enthält an jeder zwölften Stelle einen Wert ungleich null. Eine solche „Übertragungsfunktion“ ist beispielhaft in Abbildung 4.4a dargestellt. Da die Übertragungsfunktion jedoch unter anderem zur Entzerrung der Emp-

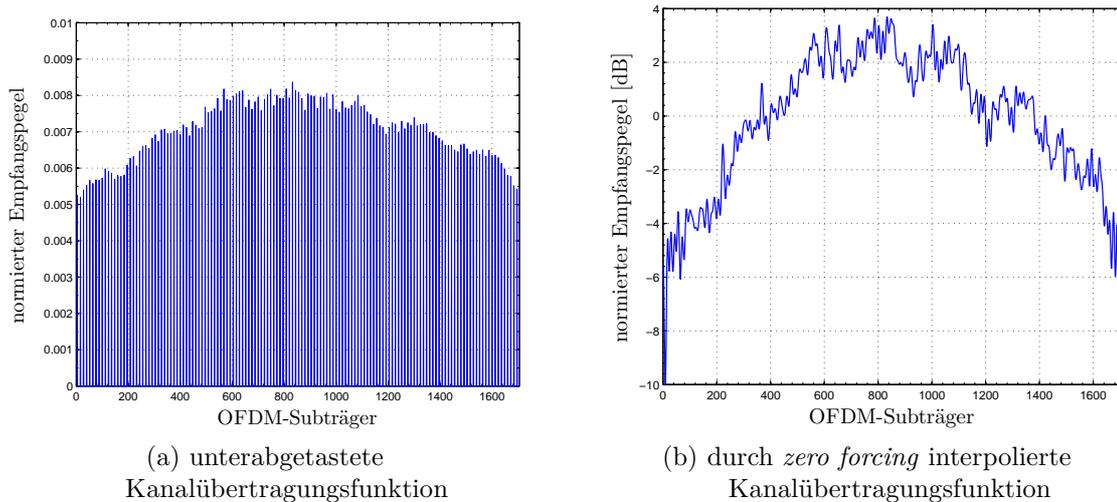


Abbildung 4.4: **Pilotbasierte Kanalschätzung** Pilotstruktur vor linearer Interpolation und interpolierte Kanalübertragungsfunktion (idealer, rauschfreier Kanal, 2K-FFT-Modus, 6 MHz, QPSK,  $GI = 1/4$ ,  $CR = 1/2$ ).

fangsdaten dienen soll, muss eine Schätzung der fehlenden Datenwerte erfolgen wie sie im Kapitel 3.6 beschrieben wurde. Mittels *zero forcing* entsteht dann die interpolierte Übertragungsfunktion wie sie in Abbildung 4.4b zu sehen ist. Bei der Wahl der Filterlänge ist darauf zu achten, dass im allgemeinen Fall im Zeitbereich einerseits so viele Samples durchgelassen werden, dass alle Echos die innerhalb des Guard Intervals liegen, nicht herausgefiltert werden und andererseits keine Aliaseffekte (siehe dazu Abbildung 4.5) durch Wahl eines zu langen Filters im Signal verbleiben.

#### 4.4.8 Verarbeite TPS Pilots

Das MATLAB-Programm verwendet statische Einstellungen bezüglich Modulationsverfahren etc. wie sie dem Programm als Parameter mitgeteilt werden (siehe Kapitel "Nutzung der Software" ab Seite 53). Insofern ist der Wert der *Transmission Parameter Signalling Pilots* für den Programmablauf selbst nicht wichtig. Da allerdings für den CD3-Algorithmus eine Neucodierung der Daten erforderlich ist, muss der gesetzte Wert des TPS-Bits bekannt sein. Da die Positionen der TPS Pilots innerhalb des OFDM-Symbols bekannt sind, können diese problemlos in die Variable `symb.tps_pilots` gespeichert werden und werden dort in-place descrambelt. Aufgrund der Tatsache, dass alle TPS Pilots innerhalb eines OFDM-Symbols immer den gleichen Wert annehmen, kann mittels einer Mehrheitsentscheidung der aktuelle Wert ausgelesen werden. Einzelne fehlerhafte Bits führen somit nicht zu Fehlern. Im Feld `tps_data` der Datenstruktur `symb` werden aus programmieretechnischen Gründen die Werte „1“ und „-1“ statt „0“ und „1“ (wie im Standard [DVB06] beschrieben) gespeichert und später dementsprechend wieder reencodiert.

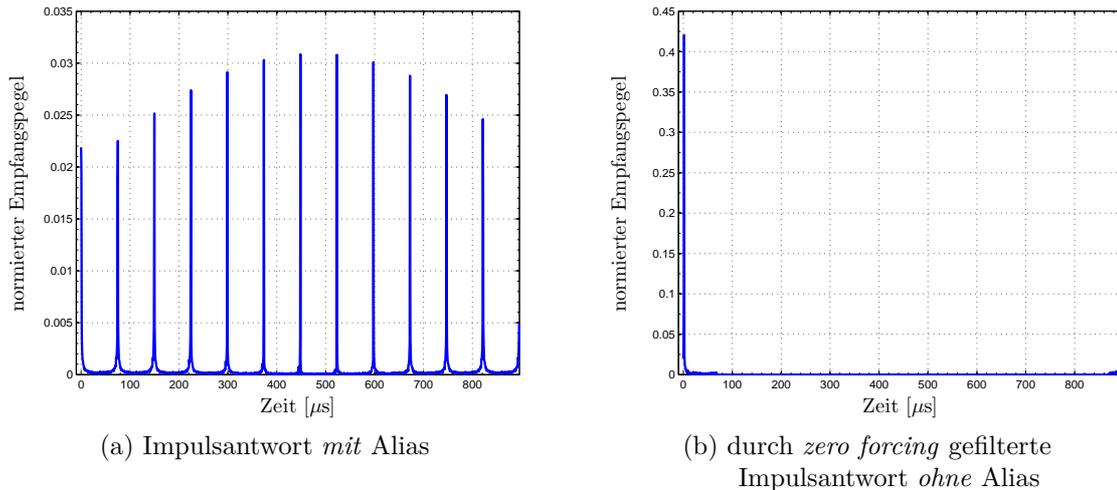


Abbildung 4.5: **Aliasunterdrückung zur linearen Interpolation:** Impulsantwort mit und ohne Aliaseffekte (idealer, rauschfreier Kanal, 2K-FFT-Modus, 6 MHz, QPSK, GI = 1/4, CR = 1/2).

#### 4.4.9 Restträgerlöschung

Die ersten geschätzten Kanalübertragungsfunktionen zeigten im Wesentlichen den erwarteten Verlauf. Jedoch trat in allen Datenströmen bei einer bestimmten Frequenz eine unerwartete Überhöhung auf dem mittleren Subträger auf, wie es in Abbildung 4.6a zu sehen ist. Zurückzuführen ist dieses Phänomen auf den *Restträger* des Messgerätes, mit dem die ersten Messaufnahmen durchgeführt wurden. Die Herkunft dieser Störung wurde bereits im Kapitel 3.8 erörtert. Im Quellcode regelt der Parameter `clear_restcarrier`, ob der Restträger gelöscht werden soll oder nicht. Das Codelistung 4.4 zeigt die Umsetzung der Maßnahme, die aus einer Löschung des mittleren Frequenzträgers und der Interpolation des fehlenden Datenwertes aus den benachbarten Subträgern besteht. Der Wert des originalen Trägers muss zwischengespeichert werden, da er später für die CD3-Schätzung benötigt wird. Die durch die Interpolation der zuerst unterabgetasteten Kanalübertragungsfunktion ebenfalls angehobenen Nachbarträger sind keine Pilote und bedürfen von daher auch keiner gesonderten Behandlung wie der Mittenträger.

Listing 4.4: Restträgerlöschung (Auszug aus `main_sync_soft.m`)

```

1 | if (clear_restcarrier == 1)
2 |     restcarrier_value ...
3 |     = symb.channel_transfer_function(params.k, ...
4 |     ceil(params.active_carriers/2));
5 | symb.channel_transfer_function(params.k, ...
6 |     ceil(params.active_carriers/2))
7 | = mean([symb.channel_transfer_function(params.k, ...
8 |     ceil(params.active_carriers/2)-1), ...
9 |     symb.channel_transfer_function(params.k, ...

```

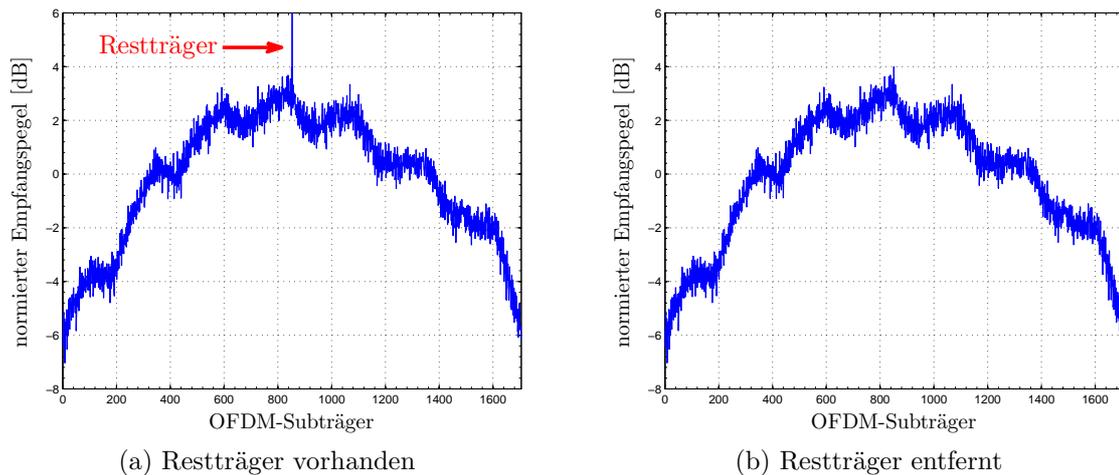


Abbildung 4.6: **Restträgerlöschung:** Kanalübertragungsfunktion mit Restträger vor und nach Entfernung mit Interpolation des fehlenden Datenwertes (idealer, rauschfreier Kanal, 2K-FFT-Modus, 6 MHz, QPSK,  $GI = 1/4$ ,  $CR = 1/2$ ).

```

10 |     ceil(params.active_carriers/2)+1));
11 | end;

```

#### 4.4.10 Descrambling der Subträger und der Übertragungsfunktion

Vor der Demodulation der Nutzdaten müssen die Datensubträger mittels einer von vier Symbolsequenzen umsortiert werden. Die vorberechneten Sequenzen enthalten die uncodierten Trägerpositionen. Da für das folgende Demapping eine Entzerrung der Daten mittels der geschätzten Kanalübertragungsfunktion notwendig ist, muss letztere mit der gleichen Umsortiersequenz descrambelt werden.

#### 4.4.11 Demapping

An dieser Stelle im Programm können die tatsächlichen Nutzdaten demappt werden. Abhängig von der Modulationsart werden aus einem Symbol zwei (QPSK), vier (16-QAM) oder sechs (64-QAM) Bits gemäß der im DVB-T-Standard [DVB06] definierten Mappingtabelle (vergleiche Abbildung 2.9 auf Seite 15) demoduliert. Es müssen zwei Fälle abhängig von der beabsichtigten Weiterverarbeitung unterschieden werden.

##### 4.4.11.1 Hard Decision-Decodierung

Im einfachen Fall genügt eine Decodierung mit Hard Decision-Werten. Es werden also binäre Werte als Ergebnis des Demappings ausgegeben, die die eigentlichen Nutzda-

tenbits repräsentieren. Es wird die MATLAB-eigene Funktion `modem.demodulate` zur Decodierung verwendet.

#### 4.4.11.2 Viterbi-Decodierung

Alternativ können für den Fall, dass eine Viterbi-Decodierung durchgeführt werden soll, vom Demapper *Soft-Bits* ausgegeben werden. Sie enthalten neben dem eigentlichen Wert auch eine Wahrscheinlichkeitsaussage über die Korrektheit des demodulierten Bits. In der Implementierung werden LLR-Werte verwendet, die als logarithmische Größe weniger numerische Probleme bereiten und durch das Hauptprogramm wie im Codelisting 4.5 auf einen Wertebereich zwischen  $-20$  und  $+20$  beschränkt werden. Dieser Beschränkung liegen die beiden Annahmen zugrunde, dass größere LLR-Werte wahrscheinlich auf numerische Probleme zurückzuführen sind und ein betragsmäßiger LLR-Wert von 20 schon eine sehr große Wahrscheinlichkeit für korrekte Decodierung darstellt.

Zunächst wurde ebenfalls die besagte `modem.demodulate`-Funktion verwendet. Durch das verwendete *zero forcing* bei der pilotbasierten Kanalschätzung wird das Rauschen im Signal verstärkt, wodurch numerische Probleme auftraten, sobald die geschätzte Kanalübertragungsfunktion einen zu kleinen Wert enthielt. Der korrigierte Nutzdatenwert war dementsprechend zu groß und führte zu falschen Ergebnissen nach der Decodierung. Durch Nutzung der Funktionen `LLRqamDemapQPSK` für QPSK, `LLRqamDemap16` für 16-QAM bzw. `LLRqamDemap64` (64-QAM) konnte dieses Problem gelöst werden. Diese Implementierungen nutzen eine Maximum Likelihood-Decodierung. Sie teilen nicht *vor* Ausführung des Demappings den Empfangswert durch die geschätzte Übertragungsfunktion wie die Funktion `modem.demodulate`, sondern multiplizieren zunächst einen (willkürlich wählbaren) Faktor an die Konstellationspunkte, der sich im Endeffekt wieder kürzen lässt. Im Code wurde als Faktor  $e^{300}$  gewählt, was einer mehrfachen Drehung des kompletten Konstellationsdiagrammes entspricht und auf diese Art numerische Probleme verhindert. Den Funktionen `LLRqamDemapQPSK`, `LLRqamDemap16` bzw. `LLRqamDemap64` müssen bei jedem Aufruf die Empfangsdaten sowie die aktuelle Kanalschätzung als Parameter übergeben werden.

Listing 4.5: Beschränkung des LLR-Wertebereichs (Auszug aus `main_sync_soft.m`)

```

1 symb.demapped_bit_llrs(params.k,:) ...
2   = (symb.demapped_bit_llrs(params.k,:) >= params.max_llr) ...
3     .* params.max_llr ...
4     + (symb.demapped_bit_llrs(params.k,:) < params.max_llr) ...
5     .* symb.demapped_bit_llrs(params.k,:);
6 symb.demapped_bit_llrs(params.k,:) ...
7   = (symb.demapped_bit_llrs(params.k,:) <= -params.max_llr) ...
8     .* (-params.max_llr) ...
9     + (symb.demapped_bit_llrs(params.k,:) > -params.max_llr) ...
10    .* symb.demapped_bit_llrs(params.k,:);

```

#### 4.4.11.3 Normalisierung

Die im Programm verwendete Normalisierung richtet sich nach dem DVB-T-Standard. Die Leistung aller Konstellationspunkte ist für alle Modulationsarten auf 1 normiert, woraus die Normalisierungsfaktoren gemäß [DVB06] resultieren. Jeweils für den Fall der nicht-hierarchischen Modulation ergeben sich Normalisierungsfaktoren für QPSK von  $c = \sqrt{10}$ , für 16-QAM von  $c = \sqrt{10}$  sowie für 64-QAM von  $c = \sqrt{42}$ . Deshalb kann im IQ-Diagramm des Programms stets der Wertebereich von  $-1,5$  bis  $+1,5$  dargestellt werden, da der Betrag der Nutzdaten gemäß obigen Ausführungen nie größer als 1 werden kann. Damit auch die in der Leistung angehobenen Boosted Pilots ebenfalls im IQ-Diagramm auftauchen, ist der Wertebereich des Diagramms geringfügig zu erweitern.

#### 4.4.12 Deinterleaving, Demultiplexing, Depunktierung

Gemäß der Abbildung 4.7 auf Seite 45 (aus [DVB06]) werden die Funktionen `symbol_deinterleaver`, `bit_deinterleaver`, `bit_demux_inverse` und `depuncturing` in umgekehrter Reihenfolge wie in der Abbildung aufgerufen, da das MATLAB-Programm den empfangenen Datenstrom decodieren muss, im Standard jedoch immer die Codierung (von links nach rechts) beschrieben ist. Bei der Implementierung müssen natürlich die Wertigkeit und die jeweilige Position der eingelesenen Bits berücksichtigt werden, damit die Bits an der korrekten Stelle im decodierten Datenstrom stehen.

#### 4.4.13 Viterbi-Decodierung und Faltungscodierung

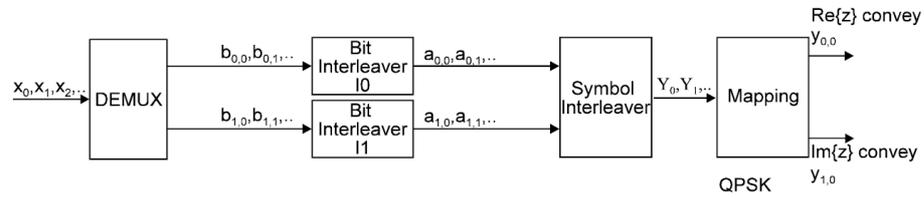
Die *eigentliche* Viterbi-Decodierung geschieht in der Funktion `viterbi_de_and_reencoding` und nutzt dazu die MATLAB-eigenen Funktionen `vitdec` für die Viterbi-Decodierung sowie `convenc` für die Faltungscodierung. Besonders behandelt werden muss die inhärente Anlaufphase eines Faltungscodes. Die ersten Werte sind prinzipbedingt unzuverlässig, da keine vorherigen Werte für die Decodierung des aktuellen Bits zur Verfügung stehen und werden deshalb verworfen und durch die Originaldaten vor der Decodierung ersetzt.

#### 4.4.14 Neucodierung des Datenstroms

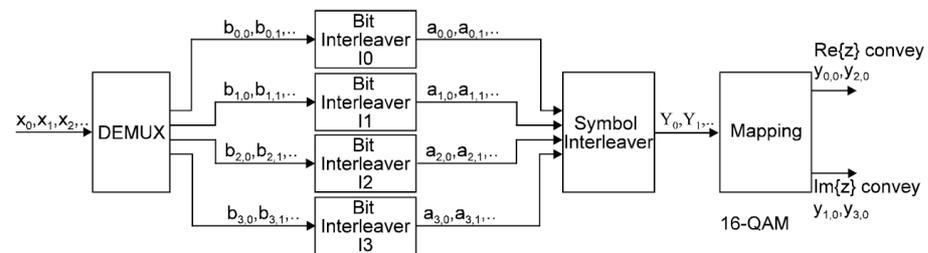
Nachdem die Nutzdaten decodiert wurden, codieren die Funktionen `bit_demux`, `bit_interleaver`, `symbol_interleaver` sowie das als Skript implementierte `reencode` die Daten in umgekehrter Reihenfolge der Decodierung.

#### 4.4.15 Kanalschätzung mittels des modifizierten CD3-Algorithmus

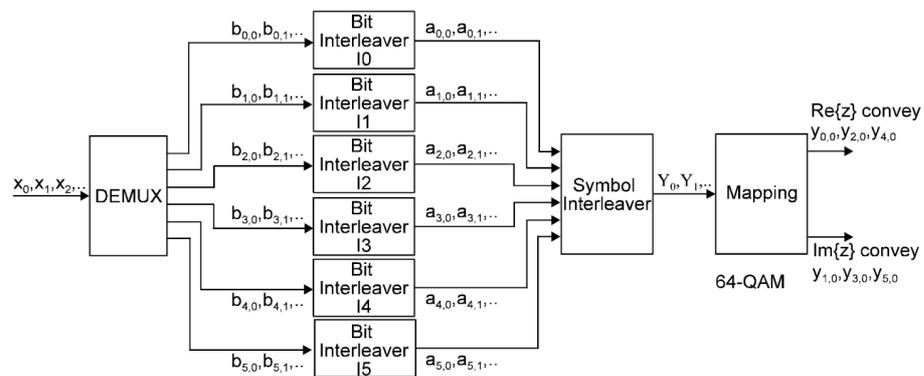
Die Kanalschätzung nach dem in 3.5 beschriebenen modifizierten CD3-Algorithmus wird durch die im Listing 4.6 abgedruckten Zeilen realisiert. Es werden die ursprüng-



(a) Encoderkette für QPSK



(b) Encoderkette für 16-QAM



(c) Encoderkette für 64-QAM

Abbildung 4.7: Encoderseitige Kette DEMUX, Bit Interleaver, Symbol Interleaver und Mapping (aus [DVB06])

lichen Empfangsdaten punktweise durch die decodierten und wieder neu codierten Daten dividiert. Dadurch entsteht eine bessere Schätzung der wahren Kanalübertragungsfunktion.

Listing 4.6: Kanalschätzung nach dem modifizierten CD3-Algorithmus (Auszug aus `main_sync_soft.m`)

```

1 | symb.transfer_func(params.k, :) ...
2 | = symb.useful_carriers(params.k, :)./ symb.reencoded_data(params.k, :);

```

## 4.4.16 Nachfilterung der Kanalübertragungsfunktion

Für die Betrachtung des Kanals über einen längeren Zeitraum hinweg müssen verschiedene Filterungen der ermittelten Kanalübertragungsfunktionen durchgeführt werden. Alle bisherigen Verarbeitungsschritte arbeiten auf Basis *einzelner* OFDM-Symbole, beziehen jedoch keine Kanaländerungen zwischen *mehreren* Symbolen mit ein und berücksichtigen keine Phasendifferenzen zwischen verschiedenen Empfangs-symbolen. In mehreren Schritten werden möglicherweise auftretende Probleme behoben. Auf diese Art wird eine Glättung der Darstellung der Kanalübertragungsfunktion erreicht, ohne dass charakteristische Merkmale eliminiert werden.

### 4.4.16.1 Verbesserung der Schätzung der Kanalübertragungsfunktion durch Vernachlässigung der inneren Konstellationspunkte

Die Schätzung der mithilfe des modifizierten CD3-Algorithmus ermittelten Kanalübertragungsfunktion kann gemäß Kapitel 3.5.1 (siehe Seite 27) verbessert werden, wenn nur die äußeren Konstellationspunkte zur Schätzung der Kanalübertragungsfunktion herangezogen werden. Die Realisierung findet in der Funktion `discard_inner_const_points` statt. Die mindestens benötigte Amplitude muss der Funktion als Parameter übergeben werden, anhand derer die entsprechenden Konstellationspunkte mit geringerem Abstand zum Ursprung entfernt werden.

Die decodierten Nutzdaten stehen zu diesem Zeitpunkt bereits zur Verfügung, so dass die eigentliche Schwierigkeit bei der Implementierung in der Interpolation der Datenwerte, die zu einem inneren Konstellationspunkt gehören, besteht. Da a priori nicht bekannt ist, wieviele Werte interpoliert werden müssen, muss der implementierte Algorithmus zunächst die reencodierten Daten durchlaufen und analysieren, welche Werte geschätzt werden müssen. Die Interpolation selber wird durch eine Faltung mit einem Dreieck angepasster Breite durchgeführt. An den Rändern kann das Problem entstehen, dass ein Datenwert zwar auf einen inneren Konstellationspunkt gemappt werden müsste, dieser dann jedoch wegen seiner zu kleinen Amplitude im Konstellationsdiagramm nicht mehr verwendet werden darf. Jenseits des Randes stehen auch keine Frequenzträger mehr für eine Interpolation zur Verfügung, so dass an den Rändern bei Bedarf eine Konstantwertfortsetzung mit dem äußersten zur Verfügung stehenden Nutzwert durchgeführt wird.

#### 4.4.16.2 Korrektur der Phasenabweichung zwischen zwei aufeinanderfolgenden OFDM-Symbolen

Zunächst wird der Phasenfehler (*Common Phase Error*, CPE) zwischen jeweils zwei aufeinanderfolgenden OFDM-Symbolen korrigiert. Das erste verarbeitete Symbol wird als Referenz genommen, an welches nachfolgende Symbole angeglichen werden. Die Kompensation wurde zunächst durch die Funktion `timing_error_correction` durchgeführt, deren wichtigsten Codezeilen im Listing 4.7 abgedruckt sind. In der Zeile 14 wird eine Anpassung der Phase innerhalb des aktuellen Symbols durchgeführt. Der Phasenunterschied zwischen zwei aufeinanderfolgenden Subträgern entspricht der Phase des Mittelwerts des punktweisen Produktes des aktuellen Symbols mit dem konjugiert Komplexen des Referenzsymbols über alle Subträger. Entsprechend Gleichung 4.4 wird die Korrektur durchgeführt und das Ergebnis im Vektor `symb_curr_corr` gespeichert.

$$e^{j \cdot 2\pi \cdot \frac{\text{Phasenfehler}}{\text{Trägeranzahl}} \cdot \text{Trägerposition}} \quad (4.4)$$

Der Ausdruck `1:length(symb_curr)` (Zeile 14) erzeugt einen Vektor mit den Werten  $1, 2, \dots, \text{Länge des Vektors } \mathit{symb\_curr}$ <sup>4</sup>. Der CPE ist nun die Phase des Mittelwertes der punktweisen Multiplikation des Referenzsymbols mit dem konjugiert Komplexen des in sich phasenkorrigierten Vektors `symb_curr_corr`. Durch die komplexe Multiplikation mit diesem CPE wird eine Drehung im Frequenzbereich erreicht, die einer Verschiebung im Zeitbereich entspricht.

Listing 4.7: Auszug aus `timing_error_correction.m`

```

1 % calculate difference in angle for two consecutive OFDM symbols
2 symbs_conv = (symb_curr .* conj(symb_ref));

4 % Calculate phase gradient between two consecutive subcarriers.
5 % It is very important to compute the mean()-operation FIRST (inner
6 % bracket) and AFTER THIS the angle()-operation (outer bracket)
7 % leading to the following prototype:
8 % angle(mean(gradient)) with gradient=symbs_conv(1:end-1) .*
9 % conj(symbs_conv(2:end));
10 slope = angle(mean(symbs_conv(1:(end-1)) .* conj(symbs_conv(2:end))));

12 %% compensate phase of current symbol based on phase difference
13 % between previous and current symbol.
14 symb_curr_corr = symb_curr .* exp(1i .* slope .* (1:length(symb_curr)));

16 % determine common phase error
17 common_phase_error = angle(mean(symb_ref .* conj(symb_curr_corr)));

19 symb_curr_corr = symb_curr_corr * exp(1i .* common_phase_error);

```

Dennoch waren im zeitlichen Verlauf der Kanalübertragungsfunktion Sprünge zu sehen, die jedoch ebenso gut nachträglich entfernt werden können. Die Nachverarbei-

<sup>4</sup>Die Erzeugung von Zahlenfolgen ist in MATLAB mittels des „:“-Operators möglich.

tung der Phase ist als Funktion `correct_fractional_subsample_freq_error` in das Hauptprogramm eingebunden und kann durch den Schalter `postprocessing_bool` aktiviert werden. Da eine mehrfache Kompensation des CPE sich nicht nur als unnötig, sondern auch als kontraproduktiv herausgestellt hat, weil die Funktionen `timing_error_correction` und `correct_fractional_subsample_freq_error` beide eine Verschiebung im Zeitbereich verursachen, die in manchen Fällen zur Korrektur eines bereits kompensierten CPE führen konnte, wird die Funktion `timing_error_correction` in der aktuellen Programmversion nicht mehr verwendet. Ihr Einsatz ist durch die Nachführung des Abtastzeitpunktes durch die PLL (ganzzahlige Abtastschritte) in Verbindung mit der Nachverarbeitung durch `correct_fractional_subsample_freq_error` obsolet geworden. Der nächste Abschnitt beschreibt die Nachverarbeitung.

#### 4.4.16.3 Kompensation des Common Phase Errors im Frequenzbereich

Durch leichte Frequenzabweichungen des Empfängeroszillators kann es bei mehreren aufeinanderfolgenden OFDM-Symbolen zu einem Verrutschen des Abtastzeitpunktes kommen. Im Zeitbereich können jedoch nur ganzzahlige Verschiebungen um einzelne Empfangssymbole durchgeführt werden, weshalb eine Korrektur geringerer Abweichungen im Frequenzbereich durchgeführt werden muss (siehe auch Kapitel 2.1.1.3 auf Seite 6). Dazu wird ein Oversampling durchgeführt, welches der Funktion `correct_fractional_subsample_freq_error` als Parameter übergeben werden kann. Listing 4.8 zeigt die relevanten Quellcodezeilen. Zunächst wird ein Oversampling (in der Praxis haben sich Oversampling-Faktoren von acht oder sechzehn bewährt) durchgeführt (Zeilen 1 und 2), danach wird die Position des maximalen Peaks ermittelt (Zeilen 4 und 5). Die Zeilen 7 bis 9 verschieben die ermittelte Maximalposition in den korrekten Wertebereich. Alle nachfolgenden Zeilen dienen der eigentlichen Korrektur des Abtastzeitpunktes durch eine Multiplikation aller OFDM-Subträger nach der bekannten Gleichung 4.4.

Listing 4.8: Kompensation des Fractional Subsampling Errors (Auszug aus `correct_fractional_subsample_freq_error`)

```

1 | in_symb_ifft_oversamp...
2 |   = ifft(in_symb, oversamp*length_of_in);
4 | [max_val, max_pos]...
5 |   = max(in_symb_ifft_oversamp);
7 | if max_pos > length_of_in * 2
8 |   max_pos = max_pos - length_of_in * oversamp;
9 | end;
11 | phase_err = max_pos / oversamp;
13 | corr...
14 |   = in_symb .* exp(1i* 2*pi* phase_err/length_of_in...
15 |     .* (1:length_of_in)');

```

```

17 % compensate common phase error
18 corr = ifft(corr);
19 corr = corr ./ (corr(1) / abs(corr(1)));
20 corr = fft(corr);

```

#### 4.4.16.4 Filterung der Impulsantwort nach verbesserter Schätzung der Kanalübertragungsfunktion

Optional kann die mittels des modifizierten CD3-Algorithmus geschätzte Impulsantwort, welche die inverse Fouriertransformierte der Kanalübertragungsfunktion ist, erneut durch *zero forcing* gefiltert werden wie im Listing 4.9 dargestellt. Die Nutzung dieser Nachfilterung kann durch die globale Variable `filter_cd3_impulse_response_bool` ein- (1) oder ausgeschaltet (0) werden. Abbildung 4.8 zeigt den Einfluss der gewählten Filterlänge: Während in Abbildung 4.8a ein auf das Guard Interval angepasstes Filter verwendet wurde, zeigt Abbildung 4.8b das Signal, welches mit einem Tiefpass mit  $\frac{1}{12}$  normiertem Durchlassbereich gefiltert wurde. Durch die Verkürzung des Durchlassbereichs des Filters wird die Rauschleistung in der verbesserten Schätzung reduziert, wodurch die Qualität der Schätzung beträchtlich verbessert werden kann. Eine manuelle Anpassung der Filterlänge auf den Bereich, in dem tatsächlich Nutzsignale liegen, bewirkt eine deutliche Signalverbesserung im Vergleich zur automatischen Filterung, die basierend auf der Länge des Guard Intervals durchgeführt werden muss, da theoretisch innerhalb des gesamten Guard Intervals Signale weiterer Sender oder Echos eintreffen könnten.

Listing 4.9: Filterung der Impulsantwort nach Kanalschätzung mittels des modifizierten CD3-Algorithmus (Auszug aus `main_sync_soft.m`)

```

1 if (filter_cd3_impulse_response_bool == 1)
2     impulse_response_after_cd3...
3     = ifft(symb.transf_funct_phasecorrected(params.k,:));
4     impulse_response_after_cd3(filter_width_cd3:end-50) = 0;
5     symb.transf_funct_phasecorrected(params.k,:)...
6     = fft(impulse_response_after_cd3) * 12;
7 end;

```

#### 4.4.16.5 Filterung in Zeitrichtung

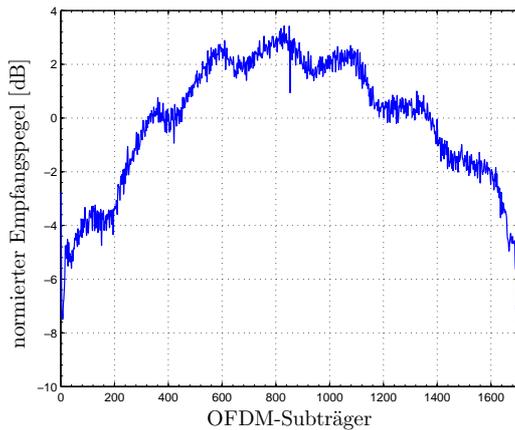
Wahlweise kann direkt im Hauptprogramm `main_sync_soft` eine Filterung über mehrere Symbole in Zeitrichtung durchgeführt werden. Diese wird aus Effizienzgründen über alle analysierten Symbole hinweg mit der MATLAB-internen zweidimensionalen Faltung `conv2` realisiert. Diese Option ist nur anwählbar, wenn die Verarbeitung im RAM aktiviert ist. Listing 4.10 stellt die Faltung dar.

Listing 4.10: Filterung der Kanalübertragungsfunktion in Zeitrichtung (Auszug aus `main_sync_soft`)

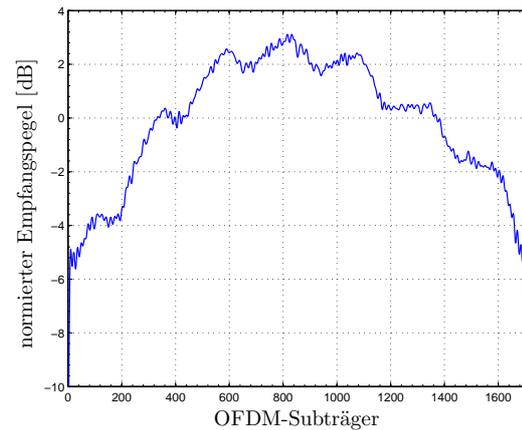
```

1 |symb.time_mean_symb...

```



(a) Tiefpassfilterung mit GI-angepasstem Filter



(b) Tiefpassfilterung mit  $1/12$  normiertem Durchlassbereich

Abbildung 4.8: **Einfluss der Filterlänge:** Mittels des modifizierten CD3-Algorithmus geschätzte Kanalübertragungsfunktion mit GI-angepasstem Tiefpassfilter und mit schmalere Tiefpassfilter zur Rauschunterdrückung. Letzteres erfordert eine manuelle Analyse der CD3-geschätzten Impulsantwort, damit keine relevanten Signalanteile entfernt werden (idealer, rauschfreier Kanal, 2K-FFT-Modus, 6 MHz, QPSK,  $GI = 1/4$ ,  $CR = 1/2$ ).

```
2 | = conv2(symb.transf_funct_phasecorrected , ...
3 |   hamm50_norm0_05', 'same');
```

#### 4.4.17 Implementierung des Phasenregelkreises

Der digitale Phasenregelkreis (PLL) sorgt für die fehlertolerante Nachführung des Abtastzeitpunktes und wird ab dem zweiten Symbol verwendet. Dementsprechend wird für das erste verarbeitete OFDM-Symbol eine perfekte Synchronisation zur Initialisierung der PLL angenommen. Es wird ein Oversampling durchgeführt, um auch geringfügige Abweichungen detektieren zu können und die Regelung auch für den Fall einer gebrochen rationalen Abweichung des realen Abtastzeitpunktes vom idealen korrekt auszuführen. In jedem Falle wird eine Nachführung um einen Abtastschritt im Zeitbereich nach links oder rechts durchgeführt, da eine *optimale* Lage des Abtastzeitpunktes unrealistisch ist. Durch diese Maßnahme können allerdings auch Abweichungen entstehen, die zu Abweichungen vom idealen Abtastzeitpunkt im Bereich von  $(-1; 1)$  Datenstromsymbolen führen können. Durch die in Abschnitt 4.4.16 beschriebenen Maßnahmen kann dieser Effekt wieder kompensiert werden.

Listing 4.11: Implementierung der digitalen PLL (Auszug aus `main_sync_soft.m`)

```
1 | pulse_resp_max_pos
2 | = mod(pulse_resp_max_pos+params.active_carriers...
```

```

3      * oversampling_factor/2, params.active_carriers...
4      * oversampling_factor) - params.active_carriers...
5      * oversampling_factor/2;

7  if ( pulse_resp_max_pos < 0 )
8      params.pll_adjust = -1;
9  else
10     params.pll_adjust = 1;
11 end;

```

#### 4.4.18 Dateiausgabe, Initialisierung des nächsten Schleifendurchlaufs und Programmabschluss

Am Ende der Berechnung eines OFDM-Symbols mit allen gewählten Filterungen werden globale Variablen wie die Startposition des nächsten OFDM-Symbols in die Variable `stream_index` geschrieben. Die digitale PLL wird ausgewertet und der Streamindex entsprechend korrigiert. Bei gewähltem sequentiellen Lesen wird das nächste Symbol aus den in der Parametersektion des Quellcodes spezifizierten Dateien gelesen.

Nach der Berechnung der voreingestellten Zahl zu verarbeitender Symbole wird, wenn eine Verarbeitung ausschließlich im Arbeitsspeicher aktiviert ist, die Kanalübertragungsfunktion aller OFDM-Symbole über der Zeit als 3D-Plot ausgegeben. Wurde das Erstellen von Ausgabedateien gewählt, so finden sich diese unter den im Quellcode angegebenen Namen auf der Festplatte wieder. Es erscheint ebenfalls ein 3D-Plot, der jedoch nur die maximal letzten zehn verarbeiteten OFDM-Symbole darstellt. Bei erfolgreichem Programmabschluss werden alle geöffneten Dateien geschlossen. Die während der Verarbeitung entstandenen Variablen werden im MATLAB-Workspace belassen um sie einer weiteren Analyse zugänglich zu machen.

### 4.5 Codeoptimierungen

In diesem Abschnitt werden ausgewählte Codeoptimierungen präsentiert, durch welche extreme Leistungssteigerungen erzielt werden konnten. Bekanntermaßen ist die Verwendung von `for`-Schleifen in MATLAB uneffizient. Wie schon im Abschnitt 4.4.4 gezeigt wurde, kann alleine durch vorherige Speicherreservierung ein enormer Laufzeitgewinn erzielt werden. Diese Maßgabe wurde bei der Entwicklung des kompletten Codes eingehalten.

Auch die bewusste Einschränkung eines Suchbereichs (insbesondere in Verbindung mit `for`-Schleifen) kann Laufzeitvorteile in nicht unerheblichem Maße mit sich bringen. Das Unterkapitel 4.4.2 hat die zeitliche Grobsynchronisation auf den Datenstrom beschrieben. Versuche haben gezeigt, dass außer bei der erstmaligen Suche nach der Lage von OFDM-Symbolen im Datenstrom bereits eine Reduzierung der Größe des Suchbereichs auf  $\frac{1}{10}$  des gleitenden Fensters die gleichen Ergebnisse liefert

wie das jeweilige Durchsuchen des kompletten gleitenden Fensters für jedes OFDM-Symbol. Diese Beobachtung lässt sich mit einer hinreichend langsamen Änderung der Oszillatordrift des Empfängers erklären. Somit kann auch diese als `for`-Schleife implementierte, zeitraubende Berechnung verkürzt werden.

Eine weitere Vorkehrung zur Effizienzsteigerung ist das Zwischenspeichern von ermittelten Synchronisationsinformationen, anstatt die jeweilige Neuberechnung, wie sie ebenfalls bereits für die Funktion `frequency_sync` beschrieben wurde. Es ist effizienter in Bezug auf die Verarbeitungsgeschwindigkeit, nur beim ersten Aufruf einer Funktion den kompletten möglichen Wertebereich zu durchsuchen, in diesem Falle das komplette gleitende Fenster. In allen folgenden Aufrufen wird nur noch in einer lokalen Umgebung um die letzte Fundstelle gesucht. Allerdings muss bei solchen Optimierungsmaßnahmen speziell darauf geachtet werden, dass der Wertebereich nicht derart eingeschränkt wird, dass in ungünstigen Fällen eine Fundstelle aus dem Suchbereich ausgeschlossen wird.

Über das gesamte Programm und alle beschriebenen Maßnahmen gemittelt konnte die Laufzeit pro berechnetem OFDM-Symbol (im 8K-FFT-Modus) um zirka den Faktor zwei reduziert werden. Dabei noch nicht berücksichtigt ist, dass die ursprünglichen Performancemessungen zu einem frühen Entwicklungsstadium stattfanden, in dem noch nicht alle Funktionen der endgültigen Version implementiert waren, sodass der reale Optimierungsgewinn sogar noch größer ausfallen dürfte. Die Laufzeit variiert natürlich in Abhängigkeit von den gewählten Einstellungen; es sollte jedoch ungefähr mit einer Sekunde Rechenzeit pro OFDM-Symbol (8K-FFT-Modus, ohne Viterbi-Decodierung) kalkuliert werden.

Eine weitere Codeoptimierung liegt in der Verwendung der Funktionen `LLRqamDemapQPSK`, `LLRqamDemap16` bzw. `LLRqamDemap64` anstelle der erwähnten `MATLAB`-eigenen `modem.demodulate`-Methoden, wodurch falsche Kanalschätzungen durch numerische Probleme verhindert werden.

## 4.6 Bekannte Fehler

Wenn in `MATLAB` die Fehlermeldung *Index exceeds matrix dimensions* unter Angabe einer `title`-Zeile als Fehlerquelle erfolgt, muss `MATLAB` neu gestartet werden. Der Quellcode wurde durchgängig mit der `MATLAB`-Version 7.7.0.471 (R2008b) unter Microsoft Windows entwickelt. Die Lauffähigkeit wurde ebenfalls unter der Linux-version mit der gleichen Buildnummer getestet. Die Fehlerquelle konnte (bisher) nicht lokalisiert werden, ein Neustart des `MATLAB`-Programms und anschließende erneute Ausführung des Codes führte jedoch reproduzierbar zur (vorübergehenden) Beseitigung der Fehlermeldung. Der Fehler trat bisher nur bei Simulationsstart und nie während einer laufenden Simulation auf.

# 5 Nutzung der Software

Beim Design der MATLAB Software wurde insbesondere auf eine klare Struktur des Programmaufbaus geachtet. Im ersten Teil des Quellcodes sind Variablen vorgegeben, deren Wert der Benutzer anpassen kann. Neben der Messdatendatei (bzw. mehreren Dateien, wenn Real- und Imaginärteil des Messsignals in separaten Dateien vorliegen) sind dies Parameter, die nicht automatisch bestimmbar sind und Eigenschaften des DVB-T-Signals sind, wie etwa die Länge des Guard Intervals oder der verwendete FFT-Modus (2K oder 8K) sowie die Kanalbreite etc.. Darüber hinaus können Einstellungen die Signalverarbeitung betreffend vorgenommen werden. Beispielsweise kann die Filterlänge zur Nachfilterung der geschätzten Kanalübertragungsfunktion manuell verändert werden. Im durchgängig englischsprachigen Quellcode sind jeweils auch kurze Hinweise zur Bedeutung und Verwendung einzelner Parameter gegeben. Eine vollständige Übersicht aller Parameter findet sich in Tabelle 5.1 ab Seite 58.

## 5.1 Programmparameter

Das Hauptprogramm heißt `main_sync_soft` und grenzt sich durch den Zusatz `soft` zu der frühen Entwicklerversion `main_sync` ab, die nur Hard Decision-Decodierung beherrschte. Vor Aufruf des Programms aus MATLAB heraus durch Eingabe von `main_sync_soft` müssen einige Einstellungen getätigt werden, die im ersten Abschnitt des Quellcodes zu finden sind. Da es sich bei dem Quellcode um eine reine Textdatei handelt, kann prinzipiell jeder Texteditor zur Modifikation der Einstellungen verwendet werden. Die Nutzung des MATLAB-eigenen Editors wird jedoch dringend empfohlen. Um die Quelldatei zu öffnen, muss sichergestellt sein, dass als MATLAB-Arbeitsordner (*Working Directory*) derjenige gewählt ist, in welchem sich die Datei `main_sync_soft.m` befindet. Durch Eingabe von `edit main_sync_soft` wird die Hauptdatei im MATLAB-Editor geöffnet. Zu Beginn sind einige Copyrighthinweise und Dokumentationen mit den benötigten Einstellungsfeldern zu finden. Die im Quellcode angegebenen Hilfen können ebenfalls bei Unsicherheiten bezüglich der korrekten Einstellungen konsultiert werden. Nachfolgend sind alle Variablen und deren Funktionen beschrieben.

Die Eingangsdaten müssen entweder im MATLAB-eigenen Format (`.mat`-Datei) oder als Binärdatenstrom vorliegen. Der Dateiname wird abhängig von der gewünschten Eingabeform entweder in der Variablen `data_file_s` (`.mat`-Datei), in den Variablen `filename_real` und `filename_imag` für Dateien, die Binärdaten jeweils für Real- und Imaginärteil getrennt enthalten, oder in der Variablen `filename_single_input_file` für den Fall, dass die Real- und Imaginärteildaten als Binärdaten in ei-

ner Datei vorliegen, gespeichert. In jedem Fall muss das gewünschte Eingabeformat dem Programm über korrekte Wahl des Wertes von `read_sequential_int` mitgeteilt werden. „0“ bedeutet, dass die Daten aus einer `mat`-Datei gelesen werden sollen, während „1“ für die separate Zuführung von Real- und Imaginärteildaten steht und „2“ für das Vorliegen einer einzelnen Datei im Binärformat. Wenn dem Programm eine `MATLAB-mat`-Datei übergeben wird, wird diese prinzipiell zum Programmstart in den Arbeitsspeicher geladen, wodurch die Anzahl der Eingangsdaten durch die Größe des zur Verfügung stehenden Arbeitsspeichers begrenzt wird. Da anhand des Binärformats keine Aussage über das Datenformat getätigt werden kann, muss dieses dem Programm über den Schalter `input_data_format` mitgeteilt werden. „2“ steht für Daten vom Typ „Short“, „4“ muss für die Verarbeitung des Datentyps „Float“ bzw. „Double“ gewählt werden. Die Einstellung von `write_sequential_bool` steuert die Form der Verarbeitung. Bei deaktiviertem Schalter („0“) wird der komplette Programmablauf inklusive der Speicherung der Ergebnisse im RAM durchgeführt, was die Anwendung bei heute üblichen Arbeitsspeichergrößen von wenigen Gigabyte auf die Verarbeitung einiger hundert Symbole einschränkt. Bei 2 GB RAM können  $\approx 100$  Symbole im 8K-FFT-Modus verarbeitet werden. Wenn `write_sequential_bool` aktiviert wird („1“) können prinzipiell beliebig lange Streams verarbeitet werden, da die Berechnung jeweils für zehn OFDM-Symbole im RAM durchgeführt wird und die Ergebnisse daraufhin auf die Festplatte in den Ordner `folder` und dort unter den Dateinamen `output_file_real` und `output_file_imag` abgelegt werden. Sofern kein absoluter Pfad für `folder` angegeben wurde (etwa `C:\MATLAB-Daten\Ausgabe` unter Windows oder `/home/benutzer/MATLAB-Daten/Ausgabe` unter GNU/Linux, `./` bezeichnet in beiden Fällen den aktuellen Ordner), wird die Einstellung als relative Pfadangabe verstanden. Unter beiden genannten Betriebssystemen funktioniert die Angabe eines Pfades sowohl mit dem Trennzeichen Slash („/“) wie auch mit dem Backslash („\“). Wenn möglich, sollte eine sequentielle Verarbeitung bevorzugt verwendet werden, wenn die Daten in einem entsprechenden Format vorliegen. Das Hauptprogramm läuft weiter, bis entweder der Arbeitsspeicher voll ist oder `No_of_ofdm_symbols_to_compute` Symbole berechnet worden sind.

Die aufgenommenen Rohdaten wurden mit der Abtastfrequenz des jeweiligen Messempfängers gesampelt und müssen deshalb auf die Symbolperiode von DVB-T normiert werden. Dazu muss unter `SW_radio_sample_freq` die bei der Aufnahme eingestellte Abtastfrequenz des Aufnahmegerätes in Megahertz (MHz) eingestellt werden. Natürlich muss auch die verwendete Modulation unter `Modulation` eingestellt werden, wobei „0“ für QPSK, „1“ für 16-QAM und „2“ für 64-QAM steht. Für die Anzahl der verwendeten Subträger pro OFDM-Symbol muss `T_N` auf 2048 für den 2K-FFT-Modus bzw. auf 8192 für den 8K-FFT-Modus gestellt werden. Andere Werte sind nicht zulässig und führen zu einer entsprechenden Fehlermeldung. Die Länge des Guard Intervals  $T_G$  wird anteilig als Länge von  $T_N$  in `GuardInterval` spezifiziert und ist beispielsweise für den in Deutschland verwendeten DVB-T-Modus „1/4“. `channel_width` enthält die Kanalbandbreite in MHz und muss 5, 6, 7 oder 8 betragen, die `CodeRate` wird **nur** benötigt, wenn eine Viterbi-Decodierung zur Verringerung der Fehlerrate gewünscht wird und ist für in Deutschland im Einsatz

befindliche DVB-T-Netze auf „2/3“ zu stellen. Aufgrund von MATLAB-internen Rundungsfehlern kommt bei der Wahl dieses Modus eine Warnung des Programms, dass die eingestellten 0,66666666 nicht exakt  $\frac{2}{3}$  entsprechen, aber dieser Modus dennoch verwendet wird.

Die Erfahrung hat gezeigt, dass viele Messempfänger nach dem Einschalten zunächst eine gewisse Zeit zum „Einschwingen“ brauchen, so dass je nach verwendeter Hardware häufig die ersten paar hunderttausend bis Millionen Samples verworfen werden müssen. Das Programm ermöglicht dieses in Abhängigkeit der in `No_of_values_to_discard` eingestellten Zahl.

Bei Aktivierung der Option `figures_on_bool` („1“) werden zur Programmlaufzeit `figures` dargestellt, die verschiedene wichtige (Zwischen-)Ergebnisse wie das Konstellationsdiagramm oder die Schätzung der Kanalübertragungsfunktion des aktuellen OFDM-Symbols anzeigen. Diese Ausgaben sind sehr nützlich, kosten aber auch einen beträchtlichen Teil Rechenzeit, sodass die Verarbeitung gerade längerer Datenströme durch Deaktivieren der Bilder („0“) beschleunigt werden kann. Es sollte jedoch sichergestellt sein, dass die Daten eine hinreichende Güte für eine erfolgreiche Decodierung aufweisen, da Fehler bei deaktivierter graphischer Ausgabe naturgemäß erst im Nachhinein bemerkt werden können. Die Qualität der decodierten Daten wird **nicht** automatisch überprüft, sondern obliegt allein dem Anwender, was diesen auch bei schlechten Eingabedaten in die Lage versetzt, die Decodierung dennoch weiterzuführen.

Die Schätzung in `SNR_est` gibt vor, von welchem Signal-zu-Rauschverhältnis bei der Decodierung mit Soft Decision-Werten beim Demappen ausgegangen werden soll. Der Wert für `SNR_est` sollte konservativ geschätzt werden, da zu hohe SNR-Annahmen zu deutlich schlechteren Ergebnissen (höheren Fehlerraten) führen, als Nachteile durch ein zu niedrig geschätztes SNR entstehen. Um numerische Probleme zu vermeiden, wird der ebenfalls nur bei der Soft Decision-Decodierung benötigte maximale LLR-Wert auf den Wert von `max_llr` begrenzt. Als praxistauglich hat sich ein LLR-Wert von 20 erwiesen, der bereits einer sehr hohen Wahrscheinlichkeit für eine korrekte Entscheidung entspricht.

Einige Messempfänger weisen unerwünschte Mischprodukte auf dem mittleren Frequenzträger auf, die als Restträger bekannt sind. Um diese zu eliminieren, kann durch Wahl der Option `clear_restcarrier` („1“) der LLR-Wert dieser Trägerposition gelöscht werden, sodass bei der Decodierung von einem unzuverlässigen Wert an dieser Stelle ausgegangen wird, wodurch Probleme durch die überhöhte Amplitude des fehlerhaften Trägers vermieden werden. Diese Option löscht nur den mittleren Frequenzträger im Spektrum. Da bei höherwertigen Empfängern in der Regel der Einfluss dieses Trägers schon hardwareseitig unterdrückt werden sollte, wird in den meisten Fällen eine Deaktivierung dieser Option („0“) bessere Decodierungsergebnisse liefern.

Durch die Einstellung für `discard_inner_const_points_bool` wird das Verwerfen von Konstellationspunkten zu geringer Amplitude aktiviert („1“) oder deaktiviert („0“). Da die inneren Konstellationspunkte aufgrund ihrer geringen Amplitude anfälliger gegen Rauschen sind als die äußeren Punkte innerhalb des Konstellations-

diagramms und dadurch das Verrutschen in einen anderen Quadranten mit schwerwiegenden Fehlern in der geschätzten CD3-Kanalübertragungsfunktion einhergehen würde, können diese Punkte für eine verbesserte Kanalschätzung bewusst durch Aktivieren dieser Option ignoriert werden. Die an den ausgeblendeten Positionen fehlenden Werte werden basierend auf den reencodierten Daten interpoliert. Die minimale Amplitude wird dem Programm über die Variable `min_amplitude_for_const_points` mitgeteilt und ist auf  $\sqrt{2}$  voreingestellt. Der Bezug sind die nicht normierten Konstellationsdiagramme im Standard [DVB06], bei denen Konstellationspunkte bei  $(\pm 1, \pm 1)$ ,  $(\pm 1, \pm 3)$ ,  $(\pm 3, \pm 1)$  etc. liegen.

Durch einen unvermeidlichen, geringen Frequenzversatz zwischen Sender und Empfänger und durch Rauschen kann der optimale Abtastzeitpunkt im Empfänger schwanken und muss nachgeregelt werden. Damit Decodierfehler bei einzelnen OFDM-Symbolen nicht zum Decodierversagen führen, ist es sinnvoll, den Abtastzeitpunkt nicht jeweils auf das aktuelle OFDM-Symbol zu setzen, sondern basierend auf der zuvor bereits ermittelten Lage eine langsame Nachführung durchzuführen. Diese ist durch einen digitalen Phasenregelkreis implementiert, dessen „Verstärkung“ durch den Parameter `PLL_Multiplier` eingestellt werden kann. Der Wert dieser Variablen entspricht der Schrittweite auf dem Datenstrom in Samples pro Korrekturschritt.

Da die Verarbeitung der OFDM-Symbole auf Symbolbasis erfolgt, kann es zu Phasensprüngen *zwischen* mehreren aufeinanderfolgenden Symbolen kommen, welche erst nach der Decodierung aller zu berücksichtigender Symbole korrigiert werden können. Bei Wahl der Nachverarbeitung durch `postprocessing_bool` („1“) wird eine nachträgliche Korrektur des Common Phase Errors durchgeführt. Wenn dieses nicht gewünscht wird, kann die Filterung mit „0“ abgewählt werden. Soll eine Filterung in zeitlicher Richtung erfolgen, werden einerseits einzelne fehlerhafte OFDM-Symbole, die etwa zu verrauscht waren, nicht mehr im zeitlichen Verlauf störend auffallen, andererseits wird durch die zeitliche Filterung auch ein gewisser Fehler in Kauf genommen, der durch den Filtereinfluss selbst entsteht. Wenn die Zeitfilterung durch Einstellen des Wertes „1“ in der Variablen `time_direction_filtering` aktiviert wurde, **mus**s auch die Nachverarbeitung (`postprocessing_bool`) aktiviert werden, da die nicht ausgeglichenen Phasendifferenzen zwischen aufeinanderfolgenden OFDM-Symbolen ansonsten zu Chaos durch die Zeitfilterung führen würden. Die Zeitfilterung wird mit einem Hammingfilter der Ordnung 50 und einem normierten Durchlassbereich von 0,05 durchgeführt. Sie kann *nicht* aktiviert werden, wenn sequentielle Verarbeitung gewählt ist, da für eine sinnvolle Zeitfilterung sehr viele OFDM-Symbole zur Verfügung stehen müssen. Bei der anschließenden Darstellung ist darauf zu achten, dass die ersten 50 und die letzten 50 Symbole durch den An- bzw. Auslaufbereich des Filters selbst korrupt sind und eine Amplitudenabschwächung in diesem Bereich nicht an den Messdaten liegt.

Durch Wahl von „1“ für `filter_impulse_response_bool` wird die Filterung der Impulsantwort der pilotbasierten Kanalschätzung aktiviert. Dieses ist die Voreinstellung und sollte außer zu Debugging-Zwecken nicht deaktiviert werden, da durch die geringe Pilotdichte in einem DVB-T-Symbol im Frequenzbereich zwangsläufig Aliaseffekte im Zeitbereich entstehen, die entfernt werden müssen, um eine konsistente

Weiterverarbeitung der Daten zu gewährleisten und die CD3-Schätzung der Kanalübertragungsfunktion zu ermöglichen. Die Filterlänge wird über `filter_width` eingestellt. Der hier spezifizierte Wert bezeichnet die Anzahl der Samples im Zeitbereich, die nicht durch die Filterung mittels *zero forcing* entfernt werden sollen. Wenn „0“ als Filterlänge eingestellt ist, wird die Filterbreite anhand der Pilotdichte automatisch gewählt. Im Einzelfall kann durch manuelle Verstellung dieses Wertes die Decodierungsleistung verbessert werden. Über `filter_cd3_impulse_response_bool` wird die entsprechende Filterung der CD3-geschätzten Kanalübertragungsfunktion aktiviert („1“) oder deaktiviert („0“). Diese Filterung ist optional, wird aber dringend angeraten. Hier kann durch manuelle Filterlängenanpassung viel Rauschen unterdrückt werden, indem anhand eines Testlaufes mit wenigen OFDM-Symbolen die Impulsantwort der CD3-geschätzten Übertragungsfunktion analysiert wird und der maximale Durchlassbereich anhand des spätesten zum Kanal gehörenden Peaks ermittelt wird. Dieser Wert wird bei `filter_width_cd3` eingetragen. Für die Einstellung „0“ wird eine automatische Ermittlung der Filterlänge durchgeführt, die basierend auf der Guard Interval-Länge zwar garantiert keine Nutzdaten vernichtet, jedoch auch viel Rauschen bei im Einzelfall gar nicht benötigter Länge der Impulsantwort durchlässt, wodurch die Qualität der Kanalschätzung leidet.

Es kann wahlweise eine Viterbi-Decodierung durchgeführt werden, die viele Fehler basierend auf den decodierten Nutzdaten erkennen und korrigieren kann. Dazu muss für `viterbi_decoding_bool` „1“ gewählt werden. Der Viterbi-Decoder verwendet Softbits, welche neben den eigentlichen Werten der Bits auch die Wahrscheinlichkeiten für die Korrektheit dieser Bits tragen. Bei deaktivierter Viterbi-Decodierung („0“) werden Hard Decision-Werte zur De- und Reencodierung verwendet.

Die nachfolgend beschriebenen Parameter sind zwar zur Erhaltung der größtmöglichen Flexibilität als Variablen implementiert, sollten unter normalen Umständen jedoch auf ihren Voreinstellungen belassen werden, die für die meisten Anwendungsfälle – zumindest solche, für welche die Analysesoftware erstellt worden ist – genügen.

Die Viterbi-Decodierung wird mit Hilfe des MATLAB-Befehls `vitdec` durchgeführt und erwartet einen Eingangsvektor, dessen Werte im Bereich zwischen 0 und  $2^n - 1$  liegen, wobei 0 der Decodierung einer 0 und  $2^n - 1$  mit höchstmöglicher Wahrscheinlichkeit einer decodierten 1 entspricht.  $n$  ist die Anzahl der vom Viterbi-Decoder verwendeten Quantisierungsstufen. „13“ ist der größte Wert, den die Funktion akzeptiert. Geringere Werte entsprächen einer gröberen Quantisierung, die in diesem Kontext nicht sinnvoll wäre, aber dennoch über den Parameter `quantizer_bits` verändert werden kann.

Wieviele Symbole für die zeitliche Synchronisation herangezogen werden sollen, lässt sich über den Wert von `no_of_ofdm_symbols_to_use_for_sync` variieren. Der voreingestellte Wert von vier OFDM-Symbolen ist eine sinnvolle und in allen dem Autor bekannten Fällen nutzbare Größe gewesen, solange decodierbare Daten vorliegen. Die initiale Zeitsynchronisation ist durch verschachtelte `for`-Schleifen realisiert, weshalb sie verhältnismäßig lange dauert. Die Verarbeitungsdauer steigt mit höheren Anzahlen an OFDM-Symbolen, die für die Zeitsynchronisation mittels Korrelation zweier gleitender Fenster über den Datenstrom verwendet werden sollen, stark an.

Bei Signalen mit einem kurzen Guard Interval kann es sinnvoll sein, mehr als die vorgeschlagenen Symbole zu verwenden. Ebenso kann durch Variation des Wertes von `time_sync_error_allowed` der Toleranzbereich vergrößert werden, der als korrekte Synchronisationsposition durch das Programm anerkannt wird. Die empirisch ermittelten 0,05 (5 %) haben sich als robuster und praxistauglicher Wert erwiesen, da einerseits leichte Schwankungen der Amplitude der Guard Interval-Korrelation schon durch den Einschwingvorgang innerhalb des Guard Intervals selber hervorgerufen werden können, andererseits aber keine Signale fälschlicherweise als Guard Interval erkannt werden, die in Wahrheit Nutzdaten sind.

Durch `block_size_bitdeinterl` wird die Größe des Bitdeinterleavers festgelegt. Der hier implementierte Bitdeinterleaver ist speziell auf DVB-T angepasst, sodass auch dieser Wert im Allgemeinen nicht verändert werden sollte. Diese Option dient insbesondere der Migration des Bitdeinterleavers in andere Programme und sollte zur Erhaltung der DVB-T-Kompatibilität fest auf „126“ eingestellt bleiben.

Der Common Phase Error setzt sich aus einem ganzzahligen und einem fraktionalen Teil zusammen. Zur Ermittlung des Fehlers, der kleiner als ein ganzes Sample ist, was der Feinsynchronisation in Zeitrichtung entspricht, wird eine Überabtastung (*Oversampling*) des Zeitsignals durchgeführt. Konkret wird die CD3-geschätzte Kanalübertragungsfunktion mit einer inversen Fast Fourier Transformation mit der eigentlichen Länge des OFDM-Symbols in Samples multipliziert mit dem Oversampling-Faktor durchgeführt. Der Multiplikator für dieses Oversampling wird durch `oversampling_factor` bestimmt.

### 5.1.1 Übersicht Parametereinstellungen

Variablenname	Vorgabewert	zulässige Werte	Erklärung
<code>data_file_s</code>		gültiger Dateiname	Dateiname der Eingabedatei (als *.mat-Datei für blockweises Laden am Programmbeginn).
<code>filename_real</code>		gültiger Dateiname	Dateiname der Eingabedatei mit Realteildaten (Binärdaten).
<code>filename_imag</code>		gültiger Dateiname	Dateiname der Eingabedatei mit Imaginärteildaten (Binärdaten).
<code>filename_single_input_file</code>		gültiger Dateiname	Dateiname der Eingabedatei mit Real- und Imaginärteildaten (Binärdaten).

*Fortsetzung auf nächster Seite*

Variablenname	Vorgabewert	zulässige Werte	Erklärung
output_file_real	'dummy'	gültiger Dateiname	Dateiname der Ausgabedatei für Realteildaten der Kanalübertragungsfunktion.
output_file_imag	'dummy'	gültiger Dateiname	Dateiname der Ausgabedatei für Imaginärteildaten der Kanalübertragungsfunktion.
folder	'output'	gültiger Speicherort	Pfadangabe für Ausgabedateien (./ bezeichnet den aktuellen Ordner).
read_sequential_int	1	0, 1, 2	Legt fest, ob sequentiell (symbolweise) gelesen werden soll (1 oder 2) oder ob der komplette Datenstrom zu Beginn des Programms in den RAM geladen werden soll (0).
input_data_format	4	2 oder 4	2: short, 4: float/double
write_sequential_bool	0	0, 1	0: Verarbeite alle Daten im RAM bis <code>No_of_ofdm_symbols_to_compute</code> Symbole berechnet worden sind oder RAM voll. 1: Verarbeite jeweils 10 Symbole, schreibe diese dann in spezifizierte Dateien im Ordner <code>folder</code> ; wiederhole bis gewünschte Anzahl Symbole verarbeitet sind.
SW_radio_sample_freq	64/8 für USRP1, 100/8, 100/10 bzw. 100/12 für USRP2	jedes Teilverhältnis	Gibt die Samplingrate des Messempfängers in MHz an.

*Fortsetzung auf nächster Seite*

Variablenname	Vorgabewert	zulässige Werte	Erklärung
Modulation	1	0, 1, 2	0: QPSK, 1: 16-QAM, 2: 64-QAM
T <sub>N</sub>	8192	2048, 8192	2048, wenn 2K-FFT-Modus verwendet werden soll, 8192, wenn 8K-FFT-Modus verwendet werden soll.
GuardIntervall	1/4	1/4, 1/8, 1/16, 1/32	Setzt $T_G$ (Länge des Guard Intervals) als Anteil der Nutzdatenzeit $T_N$ .
channel_width	8	5, 6, 7 or 8	Kanalbandbreite in MHz.
CodeRate	2/3	1/2, 2/3, 3/4, 5/6, 7/8	Angabe der Coderate. <b>Nur</b> notwendig, wenn Viterbi-Decodierung verwendet werden soll.
No_of_values_to_discard	100000	gültige Anzahl Samples, die innerhalb des Analysedatenstroms vorhanden sind	Datenwerte, die zu Beginn des Eingangsdatenstroms verworfen werden sollen.
No_of_ofdm_symbols_to_compute	50	Integerwert	Anzahl der zu verarbeitenden OFDM-Symbole. Sollte je nach RAM-Größe klein ( $\leq 100$ für 8K-FFT-Modus, $\leq 500$ für 2K-FFT-Modus) gehalten werden, wenn Verarbeitung im RAM aktiviert ist. Sonst nur durch Länge des Datenstroms und freien Speicherplatz begrenzt.

*Fortsetzung auf nächster Seite*

Variablenname	Vorgabewert	zulässige Werte	Erklärung
figures_on_bool	1	0, 1	Zeige <i>alle</i> figures an (1) oder keines (0). ( <b>Hinweis:</b> Die Verarbeitung wird <i>ohne</i> Anzeige der Figures deutlich beschleunigt.)
clear_restcarrier	0	0, 1	Gibt an, ob die mittlere Trägerposition (Restträger) gelöscht werden soll (1) oder nicht (alle anderen Werte). Im Zweifel sollte die Restträgerlöschung deaktiviert werden.
discard_inner_const_points_bool	1	0, 1	Wenn aktiviert, werden die inneren Konstellationspunkte aufgrund ihrer Störanfälligkeit durch Rauschen nicht zur CD3-Kanalschätzung herangezogen. Die Werte an den Positionen, an denen solche Konstellationspunkte kleiner Amplitude sitzen, werden interpoliert.
min_amplitude_for_const_points	$\sqrt{2}$	0, ..., 1,5	Legt minimale Amplitude eines Konstellationspunktes fest, damit Punkt für die Schätzung der Kanalübertragungsfunktion mittels des modifizierten CD3-Ansatzes verwendet wird. Ansonsten wird eine Interpolation durchgeführt.
PLL_Multiplier	1	1, ..., 15	Schrittweite für digitale PLL pro Korrekturschritt in Samples (entspricht Verstärkung der PLL).
filter_impulse_response_bool	1	0, 1	Aktiviert (1) oder deaktiviert (0) Filterung der Impulsantwort der ersten, pilotbasierten Kanalschätzung. ( <b>Warnung:</b> Diese Filterung zur Unterdrückung von Alias ist ebenso wie die Wahl einer korrekten Filterlänge extrem wichtig! Die Filterung muss für brauchbare Ergebnisse <b>immer</b> aktiviert sein.)

*Fortsetzung auf nächster Seite*

Variablenname	Vorgabewert	zulässige Werte	Erklärung
filter_width	0	0 oder ganzzahliger Wert aus Intervall $[1; \frac{1}{12} \cdot active\_carriers - 10]$	Anzahl der Samples im Zeitbereich, die <b>nicht</b> durch die Filterung ( <i>zero forcing</i> ) <b>entfernt</b> werden sollen. 0: Automatische Ermittlung der Filterlänge basierend auf der Pilotdichte.
filter_cd3_impulse_response_bool	1	0, 1	Aktiviert (1) oder deaktiviert (0) Filterung der Impulsantwort der zweiten, mittels des modifizierten CD3-Algorithmus geschätzten Kanalübertragungsfunktion.
filter_width_cd3	0	0, ganzzahliger Wert aus Intervall $[1; GuardInterval * emphactive\_carriers - 10]$	Anzahl der Samples im Zeitbereich, die nicht durch die Filterung („zero forcing“) entfernt werden sollen. 0: Automatische Ermittlung der Filterlänge basierend auf der Länge des Guard Intervals.
postprocessing_bool	1	0, 1	Aktiviert (1) oder deaktiviert (0) Nachverarbeitung zur nachträglichen Korrektur des <i>Common Phase Errors</i> . <b>Muss</b> aktiviert sein, wenn <b>zeitliche Filterung</b> durchgeführt werden soll!
time_direction_filtering	1	0, 1	Gibt an, ob eine Filterung in zeitlicher Richtung zur Glättung der Kanalübertragungsfunktion durchgeführt werden soll (1) oder nicht (0). <b>Wenn Zeitfilterung gewählt ist, muss <i>postprocessing</i> ebenfalls aktiviert werden.</b>

*Fortsetzung auf nächster Seite*

Variablenname	Vorgabewert	zulässige Werte	Erklärung
viterbi_decoding_ bool	0	0, 1	Aktiviert (1) oder deaktiviert (0) Viterbi-Decodierung mit Softbits. Bei deaktivierter Viterbi-Decodierung werden Hard Decision-Werte zur De- und Reencodierung verwendet.
SNR_est	10	-3 bis 20	Schätzung des bei der Messung vorhandenen Signal-Rauschverhältnisses. Wird für Viterbi-Decodierung benötigt. <b>SNR_est</b> sollte konservativ geschätzt werden, da zu hohe SNR-Annahmen zu deutlich schlechteren Ergebnissen (höheren Fehlerraten) führen, als Nachteile durch einen zu niedrig geschätzten SNR-Wert entstehen.
max_llr	20	0 bis 50	Maximal erlaubter LLR-Wert. Werte darüber werden künstlich beschränkt. 20 hat sich als praxistauglicher Wert bewährt.
<b>Folgende Parameter sollten nicht verändert werden!</b>			
block_size_bit- deinterl	126	126	Größe des Bitdeinterleavers. Sollte für DVB-T-Kompatibilität fest auf „126“ eingestellt bleiben.
hamm50_ norm0_05	[...]		Filterkoeffizienten <sup>1</sup> für eine FIR-Filterung mit einem Hammingfilter der Ordnung 50 und einem normierten Durchlassbereich von 0,05. <b>Nur</b> für Zeitfilterung notwendig.

*Fortsetzung auf nächster Seite*

Variablenname	Vorgabewert	zulässige Werte	Erklärung
oversampling_factor	8	1, 2, 4, 8, 16, 32, 64	Stellt das Oversampling für die Korrektur des fraktionales Anteils des Common Phase Errors ein. Sollte auf Voreinstellung belassen werden!
quantizer_bits	13	1, ..., 13	Anzahl Quantisierungsstufen für Quantisierung der LLR-Werte. Eine Reduzierung ist im Allgemeinen nicht sinnvoll.
no_of_ofdm_symbols_to_use_for_sync	4	1, ..., 16	Anzahl der für die initiale Zeitsynchronisation herangezogenen OFDM-Symbole.
time_sync_error_allowed	0.05	0, ..., 0.2	Fehler, der bei initialer Zeitsynchronisation zugelassen werden soll. 0,05 (5%) hat sich als robuster Wert herausgestellt.

Tabelle 5.1: Übersicht aller Programmparameter

## 5.2 Häufige Fehlerquellen

Im Folgenden sind einige häufige Fehlerursachen aufgeführt, die dem Autor der Software des öfteren unterliefen. Generell wird natürlich empfohlen, sämtliche Parameter noch einmal genau zu überprüfen, dennoch soll folgende Übersicht gezielte Anhaltspunkte für die Fehlersuche geben. Für bekannte Fehler sei auch der Abschnitt 4.6 auf Seite 52 erwähnt. Die unschöne Darstellung der folgenden Punkte als Aufzählungsliste wurde aufgrund einer besseren Übersicht für den Anwender gewählt.

- Wenn gleich zu Beginn des Programms Fehlermeldungen kommen, dass Scramblingsequenzen oder andere Dateien nicht gefunden werden können, stimmt wahrscheinlich das aktuelle Arbeitsverzeichnis nicht mit dem Speicherort des Hauptprogramms überein. Dieses ist jedoch notwendig, da die Pfadangaben innerhalb des Quellcodes relativ sind, damit das Hauptprogramm ohne Installation an einen bestimmten Ort (und damit auch plattformunabhängig) laufen kann, setzt aber voraus, dass beim Programmstart das aktuelle Arbeitsverzeichnis dem Speicherort des Hauptprogramms entspricht.
- Wenn MATLAB meldet, dass nicht genügend Arbeitsspeicher zur Verfügung stehen würde („out of memory“), dann ist nicht genügend RAM im Simulationsrechner verfügbar. Wenn MATLAB auch nach Löschen des Workspaces (bzw.

<sup>1</sup>Die Filterkoeffizienten sind bereits im Quellcode eingetragen, können aber auch jederzeit beispielsweise mit dem *Filter Design & Analysis*-Tool (`fdatool`) aus der *MATLAB-Signal Processing*-Toolbox erzeugt werden.

eines MATLAB-Neustarts) immer noch die gleiche Fehlermeldung beim erneuten Programmstartversuch liefert, muss entweder die Anzahl der zu verarbeitenden Symbole reduziert werden (`No_of_ofdm_symbols_to_compute`) oder die Verarbeitung kann nicht komplett im RAM durchgeführt werden. Es muss dann sequentielles Schreiben aktiviert werden, was MATLAB dazu veranlasst, jeweils nur zehn OFDM-Symbole im RAM zu behalten und die fertig berechneten Kanalübertragungsfunktion in eine Ausgabedatei zu speichern.

- Die grundlegenden Signalparameter *Modulation*, Länge des *Guard Intervals* und *Coderate* müssen korrekt sein. Häufig wird vergessen, die *Kanalbandbreite* korrekt zu wählen, da oft verschiedene Datenströme, die im gleichen Kanal gemessen wurden, hintereinander analysiert werden. Bei zu groß gewählter Bandbreite sind rechts und links schon in der pilotbasierten Übertragungsfunktion Bereiche sichtbar, die wie Rauschen aussehen oder das Programm stürzt mit der Fehlermeldung `Subscript indices must either be real positive integers or logicals`. ab. Bei zu schmal gewähltem Kanal ist im Konstellationsdiagramm eine Punktwolke ohne erkennbare Struktur einer digitalen Modulation zu sehen. Bei falsch gewählter Guard Interval-Länge sieht das IQ-Diagramm vergleichbar chaotisch aus. Die Abbildung 5.1 zeigt das Ergebnis einer Kanalschätzung mit einer falschen Einstellung für die Modulation. In Abbildung 5.1a ist das Konstellationsdiagramm einer QPSK (zuzüglich der Pilote auf der Inphase-Achse) zu sehen, obwohl „16-QAM“ eingestellt wurde (Messparameter: idealer, rauschfreier Kanal, 2K-FFT-Modus, 6 MHz, QPSK,  $GI = 1/4$ ,  $CR = 1/2$ ). Deshalb schlägt die Reencodierung und somit auch die Kanalschätzung mittels des modifizierten CD3-Algorithmus fehl (Abbildung 5.1b). Bei fehlerhaft gewählter Coderate versagt die Viterbi-Decodierung. Die verbesserte Kanalschätzung wird entweder durch die fehlerhaft reencodierten Daten versagen (Fehlermeldung `Index exceeds matrix dimensions`.) oder sehr viele Fehler im Datenstrom enthalten, die durch den Viterbi-Decoder entstehen. Bei Wahl einer Coderate von „2/3“ statt der korrekten „1/2“ für das Signal in einem zeitvarianten 6 MHz-Kanal (Tramfahrt, 2K-FFT-Modus, 16-QAM,  $GI = 1/4$ ,  $CR = 1/2$ ) werden nur noch  $\approx 84,8\%$  (statt  $\approx 94,0\%$ ) korrekte Bits vom Viterbi-Decoder erreicht, ganz zu schweigen von der fehlerhaften Nachverarbeitung und den unbrauchbaren Ergebnissen nach einer Filterung in Zeitrichtung.
- Auch das Format der Eingabedaten (`input_data_format`) wird aus ähnlichen Gründen wie oben nicht selten falsch eingestellt, wenn doch einmal ein anderes Datenformat gewählt wurde als bei der letzten Verwendung der Software.
- Selbiges gilt auch für die Samplingfrequenz des Softwareradios in der Variablen `SW_radio_sample_freq`. Die mittels des modifizierten CD3-Algorithmus geschätzte Übertragungsfunktion und Impulsantwort sehen bei falscher Samplingfrequenz („100/8“ statt korrekterweise „64/8“) ähnlich wie in Abbildung

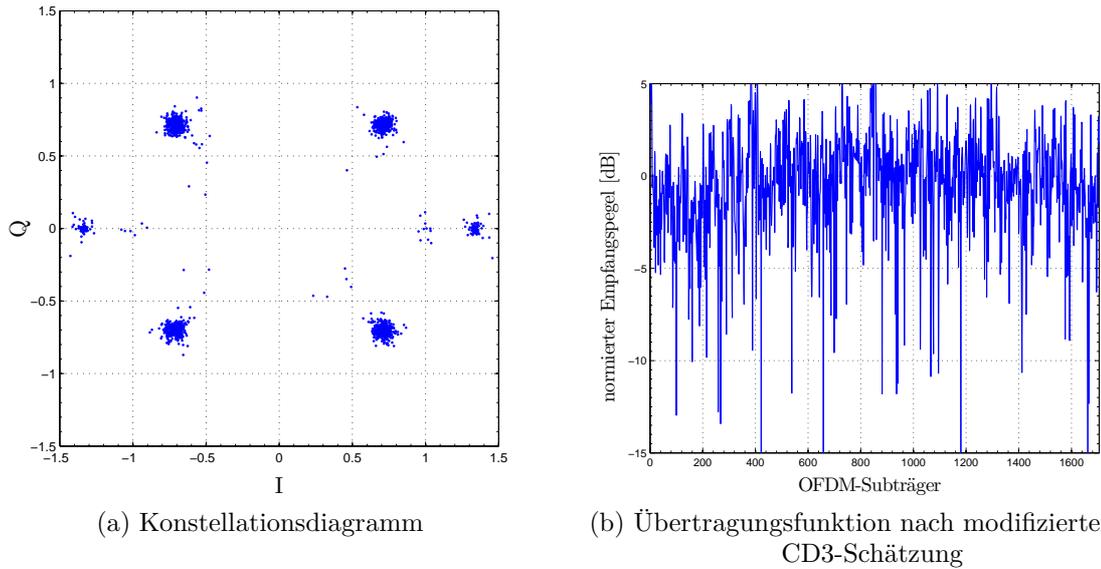


Abbildung 5.1: **Auswirkung einer falsch gewählten Modulation:** Das Konstellationsdiagramm zeigt bereits die korrekten Punkte einer QPSK (zuzüglich der Pilote auf der Inphase-Achse) an, wegen der falschen Einstellung („16-QAM“) schlägt die Kanalschätzung durch den modifizierten CD3-Algorithmus jedoch fehl (idealer, rauschfreier Kanal, 2K-FFT-Modus, 6 MHz, QPSK,  $GI = 1/4$ ,  $CR = 1/2$ ).

5.2 abgedruckt aus, da die Guard Interval-Korrelation fehlschlägt (Messparameter: idealer, rauschfreier Kanal, 2K-FFT-Modus, 6 MHz, QPSK,  $GI = 1/4$ ,  $CR = 1/2$ ). Der USRP1-Messempfänger musste auf eine Samplingfrequenz von  $\frac{64}{8}$  MHz (oder weniger) eingestellt werden, da die anfallenden Rohdaten ansonsten nicht schnell genug über den USB-Anschluss auf eine Notebookfestplatte transferiert werden konnten und bereits nach wenigen Sekunden Samples verloren gingen. Das Nachfolgegerät USRP2 kann mit höheren Abtastraten betrieben werden und gibt die Daten über eine Gigabit-Ethernet-Schnittstelle aus. Als praxistaugliche Abtastraten haben sich  $\frac{100}{12}$ ,  $\frac{100}{10}$  oder sogar  $\frac{100}{8}$  MHz herausgestellt.

- Die Verstärkung der PLL (`PLL_Multiplier`) muss eventuell angepasst werden. Je größer die Abtastfrequenz des Messgerätes ist, desto höher muss die Verstärkung gewählt werden. Versuche haben ergeben, dass eine Verstärkung von „1“ für Aufnahmen mit dem USRP1 und eine Verstärkung von „5“ für Aufnahmen mit dem USRP2 gute Ergebnisse liefern.
- Ein sehr häufiger Fehler ist, dass nicht genug Daten am Anfang eines aufgenommenen Datenstroms verworfen werden und insofern oft der Einschwingvorgang des Messsenders analysiert wird, was zu keinen sinnvollen Ergebnissen oder gar zum Programmabsturz führt. Es sollten hier, sofern der verfügbare Datenstrom

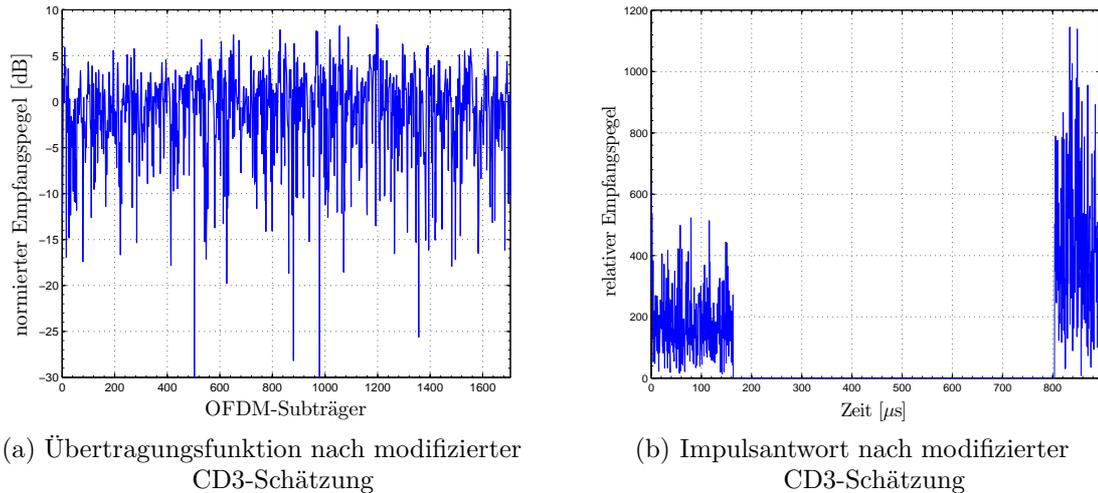


Abbildung 5.2: **Auswirkung eines falschen Resampling-Faktors:** Weder Kanalübertragungsfunktion noch Impulsantwort (hier nach CD3-Schätzung) sehen plausibel aus. Wahrscheinliche Ursache ist eine falsche Einstellung („100/8“ statt korrekterweise „64/8“) für den Resampling-Faktor, aber auch Modulation, Guard Interval oder Code-Rate können falsch sein (idealer, rauschfreier Kanal, 2K-FFT-Modus, 6 MHz, QPSK,  $GI = 1/4$ ,  $CR = 1/2$ ).

lang genug ist, ruhig einige 100.000, besser einige Millionen Samples verworfen werden.

- Der geschätzte SNR-Wert sollte eher konservativ geschätzt werden. Ein Wert von 10 für `SNR_est` hat sich als praxistauglich erwiesen. Bei zu hohen geschätzten SNR-Werten versagt der Viterbi-Decoder und liefert schlechtes Ausgangsmaterial für die Datenreencodierung.
- Die Restträgerlöschung sollte mit Vorsicht verwendet werden, weil sie Daten zerstören kann. Im Zweifel sollte sie deaktiviert werden und kann bei Auftreten einer ausgeprägten Störung auf dem mittleren Frequenzträger in der geschätzten Übertragungsfunktion wieder eingeschaltet werden.
- Eine Filterung der Kanalimpulsantwort ist *immer* notwendig, weshalb die Variable `filter_impulse_response_bool` *stets* auf „1“ gestellt sein sollte.
- Die *Länge* der Filterung kann durch die Variable `filter_width` beeinflusst werden. Zwar wird ausgehend von der Länge des Guard Intervals die maximale Filterlänge auf Wunsch (Einstellung „0“) auch automatisch berechnet, dennoch sollte anhand einzelner Testsymbole eine manuelle Korrektur der Filterlänge vorgenommen werden, damit keine Aliaseffekte die Kanalanalyse negativ beeinträchtigen.

- In Abhängigkeit der Signalqualität kann die Viterbi-Decodierung wie in Abschnitt 6.2.2 „Einfluss der Viterbi-Decodierung“ (siehe Seite 74) beschrieben sogar zusätzliche Fehler im Signal hinterlassen. Sie sollte also versuchsweise auch deaktiviert werden. Danach sollten die Ergebnisse *mit* und *ohne* Viterbi-Decodierung verglichen werden.
- Die zeitliche Filterung sollte bei unerwarteten Analyseergebnissen ebenfalls temporär deaktiviert werden.
- Eine zeitliche Filterung ist in der Regel **nur** in Verbindung mit aktiviertem *Post-processing* möglich, da ansonsten Phasensprünge *zwischen* OFDM-Symbolen nicht kompensiert werden und die Zeitfilterung somit keine sinnvollen Ergebnisse liefern kann.
- Der Fehler, der für die Zeitsynchronisation zugelassen ist, kann über die Variable `time_sync_error_allowed` verändert werden. Bei allen Versuchen des Autors hat sich ein Wert von 0,05 als robust erwiesen, aber eventuell kann die Synchronisation auch für schlechte Signale durch Veränderung dieses Wertes erzwungen werden.

# 6 Messungen und Leistungsfähigkeit

Vorliegendes Kapitel widmet sich der Aufnahme von DVB-T-Signalen, der MATLAB-gestützten Verarbeitung und letztlich der Analyse der ermittelten Kanalübertragungsfunktionen. Nach einem Überblick über notwendige vorbereitende Schritte und die Messeinrichtung wird die Funktionstüchtigkeit des im Kapitel 4 implementierten MATLAB-Programms demonstriert. Anschließend werden weitere Messungen beschrieben und analysiert sowie interessante Aspekte von DVB-T-Kanälen beleuchtet.

## 6.1 Der Messaufbau

Für die Aufnahmen der Testsequenzen kommt als Hardware das *Universal Software Radio Peripheral* (USRP) des Herstellers *Ettus Research LLC* zum Einsatz. Es kann nur in Verbindung mit einem *Daughterboard* – einem Modul, welches bestimmte Funktionalitäten enthält – verwendet werden. Das *TVRX*-Modul stellt die benötigte Ausstattung für die Vermessungen von DVB-T-Systemen bereit. Es bietet einen VHF-<sup>1</sup> und UHF-Tuner<sup>2</sup> und kann Frequenzen zwischen 50 und 860 MHz verarbeiten. In Kombination mit dem vom Hersteller empfohlenen Open Source Software Tool *The GNU Software Radio*, kurz *GNU Radio*, stellt es ein sogenanntes *Software Radio* zur Analyse beliebiger Frequenzen innerhalb des Arbeitsfrequenzbereichs des Tuners bereit. Leider konnten aufgrund der Beschränkung der Übertragungsdatenrate des verbauten USB-Ports in Verbindung mit dem verwendeten Notebook keine längeren ( $\gg 3$  s) Datenströme mit mehr als 12 Bit Quantisierung aufgezeichnet werden. Da das Gerät für den amerikanischen Markt entwickelt wurde, unterstützt der verbaute Tuner zudem nur eine maximale Kanalbandbreite von 6 MHz. Wie sich jedoch gezeigt hat, genügt bereits ein derartiges Signal für eine hervorragende Kanalanalyse. Natürlich müssen hohe Fehlerquoten im decodierten Signal in Kauf genommen werden, da die äußeren Randbereiche des Spektrums bei der Vermessung von 8 MHz breiten Kanälen schon tunerseitig sehr stark gedämpft sind. Dieses Phänomen liegt am zur Verfügung stehenden Messtuner und ist daher in allen Übertragungsfunktionen in 8 MHz-Kanälen zu finden. Sehr schön tritt der Effekt beispielsweise in Abbildung 6.7b auf Seite 79 hervor, bei dem die Randbereiche nach außen hin um bis zu 50 dB abfallen. Spätere Messungen konnten mit dem USRP1-Nachfolger,

---

<sup>1</sup> *VHF* steht für *Very high frequency* („Ultrakurzwellen“) und bezeichnet den Frequenzbereich von 30 bis 300 MHz.

<sup>2</sup> *UHF* steht für *Ultra high frequency* („Dezimeterwellenbereich“) bezeichnet den Frequenzbereich von 0,3 bis 3 GHz.

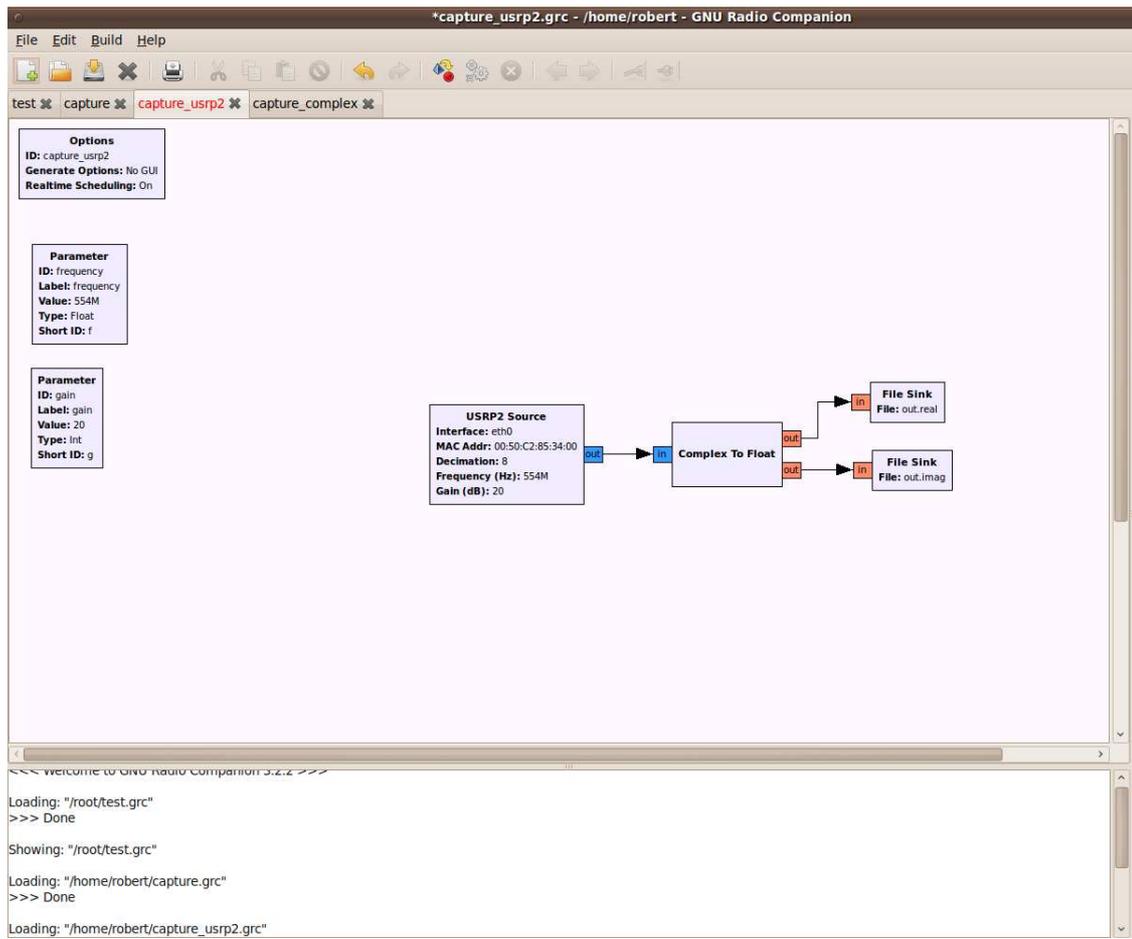


Abbildung 6.1: Screenshot des GNU Radio-Modells zur Aufnahme von DVB-T-Signalen

dem USRP2, durchgeführt werden. Der Vorteil des neuen Messempfängers besteht in einem größeren Dynamikumfang aufgrund seiner 14 Bit-Auflösung, welche sich insbesondere bei mobilen Messungen positiv bemerkbar machen. Die für die Aufnahme der Datenströme verwendete Anordnung ist als Screenshot in Abbildung 6.1 zu sehen, Abbildung 6.2 zeigt das Messequipment (mit USRP2) für die Aufnahme von Außenmessungen.

## 6.2 Auswertung

Dieser Abschnitt stellt diverse Messungen und deren Auswertungen vor und diskutiert interessante Effekte. Zunächst wird die Funktionstüchtigkeit des erstellten Programms anhand eines idealen Testsignals demonstriert.

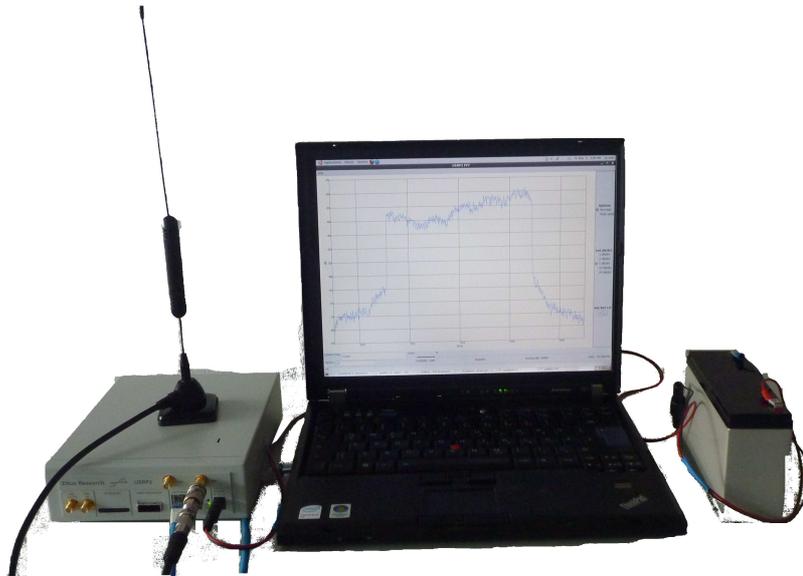


Abbildung 6.2: Messaufbau zur Durchführung von DVB-T-Messungen mit dem USRP2 und GNU Radio

### 6.2.1 Demonstration der Funktionstüchtigkeit des verwendeten Algorithmus anhand eines idealen Signals

Bevor Messungen unter schwierigen Empfangsbedingungen besprochen werden, soll die Funktionstüchtigkeit des Programms anhand eines idealen Testsignals demonstriert werden. Dazu wurden diejenigen DVB-T-konformen Einstellungen für das Testsignal verwendet, die am wenigsten kompliziert zu decodieren sind. Es wurde ein QPSK-moduliertes Signal in einem 2K-FFT-Modus und einem 6 MHz breiten Kanal mit dem Rohde & Schwarz (R&S) SFU<sup>3</sup> erzeugt. Als Guard Interval (GI) Länge wurde  $\frac{1}{4}$  verwendet, die Coderate (CR) betrug  $\frac{1}{2}$ . Ausgesendet wurde eine *Pseudo-random binary sequence* (PRBS), also ein Stream von Transportstrompaketen ohne sinnvollen Inhalt.

Abbildung 6.3 zeigt die Ergebnisse, nachdem die Analysesoftware die Testdaten decodiert hat. In Abbildung 6.3a ist das Konstellationsdiagramm dargestellt. Bei einem rauschfreien Testsignal wäre eigentlich keine Punktwolke an den Konstellationspunkten zu erwarten gewesen, sondern ein einzelner Punkt, der sich aus der Überlagerung exakt übereinanderliegender Punkte zusammensetzt. Aufgrund des hohen Phasenrauschens im Tuner des Empfängers schwanken die decodierten Punkte im IQ-Diagramm jedoch leicht. Trotzdem ist das typische Konstellationsmuster einer QPSK-Modulation sichtbar. Darüber hinaus sind um die Koordinaten  $(-1; 0)$  und  $(1; 0)$  die TPS-Pilote sichtbar, sowie bei  $(\pm\sqrt{2}; 0)$  die geboosteten Pilote. Die mittels des modifizierten CD3-Algorithmus geschätzte Kanalübertragungsfunktion ist in Abbildung 6.3b abgedruckt und offenbart ebenfalls einige interessante Aspekte.

<sup>3</sup>Das R&S SFU ist ein Broadcast Testsystem.

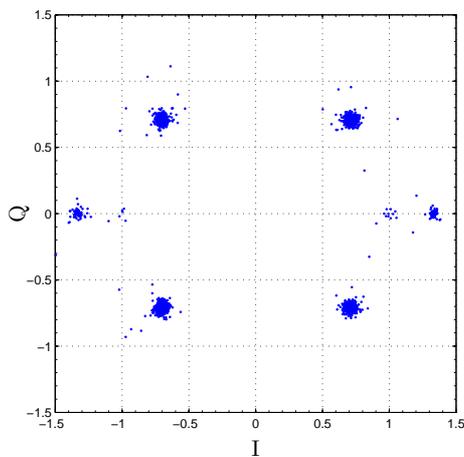
te. Zunächst fällt die stark frequenzselektive Übertragungsfunktion auf, welche den Frequenzgang des Tuners widerspiegelt. Da das R&S SFU-System auf 0,1 dB genau genormt ist, sind senderseitige Pegelschwankungen auszuschließen. Auffällig ist der hohe Rauschanteil, welcher erneut durch Phasenrauschen des Empfängers erklärt werden kann. Zudem wurde die Nachfilterung basierend auf der Länge des Guard Intervals durchgeführt. Ohne genauere Kenntnis eines Signals *könnten* schließlich innerhalb des Guard Intervals Echos oder Signale anderer Sender eintreffen. Wegen des langen Schutzintervalls liegt in diesem Fall ein hoher Rauschanteil im Durchlassbereich des Filters<sup>4</sup>. Ebenfalls ist in der Übertragungsfunktion ein sehr starker Restträger, welches ein unerwünschtes Mischprodukt des Empfängertuners ist, zu erkennen. Wie bereits in Kapitel 3.8 (Seite 28) erörtert, verlangt ein derart stark ausgeprägter Restträger eine spezielle Behandlung. Die Umsetzung im Programm ist im Unterabschnitt 4.4.9 (siehe Seite 41) dargelegt worden. Die zeitliche Darstellung der Impulsantwort ist in Abbildung 6.3d zu finden. An ihr ist der ideale Kanal am besten zu erkennen.

In Abbildung 6.4 ist der Vergleich der GI-angepassten Filterung (Abbildung 6.4a) mit einer an das Signal angepassten Filterlänge basierend auf dem spätesten relevanten Peak in der Impulsantwort (Abbildung 6.4b) zu erkennen. Durch die Verkürzung des Durchlassbereichs ist weniger Rauschen auf letzterer Übertragungsfunktion vorhanden. Manuelle Verkürzung der Breite des Nachfilters (in Frequenzrichtung) bringt also signifikant verbesserte Ergebnisse, erfordert allerdings eine manuelle Analyse und ist in zeitvarianten Kanälen mit Vorsicht zu verwenden, da sich das Eintreffen aller Echosignale verschieben kann.

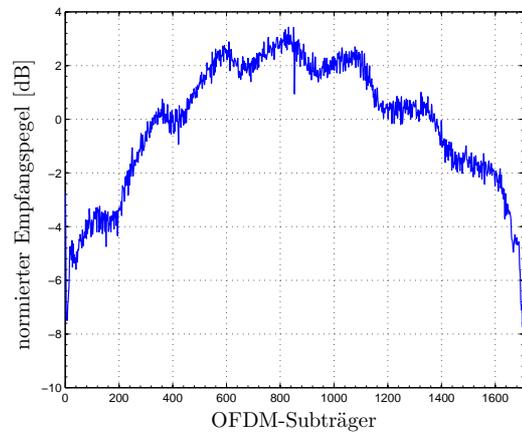
Da die Verarbeitung der Daten im Hauptprogramm auf Basis einzelner OFDM-Symbole durchgeführt wird, kann es zu Phasensprüngen *zwischen* aufeinanderfolgenden Symbolen kommen. Diese müssen vor Durchführung einer Filterung der über mehrere OFDM-Symbole geschätzten Übertragungsfunktion kompensiert werden, da die Zeitfilterung ansonsten die Übertragungsfunktion unbrauchbar machen würde. Die Anpassung der Phase zwischen verschiedenen Symbolen kann nachträglich durchgeführt werden, sie kann jedoch ebenso direkt durch das Hauptprogramm erledigt werden, wenn die entsprechende Einstellung gewählt wird. Aus Abbildung 6.5a ist die Phase über 300 OFDM-Symbole ersichtlich, bevor eine Korrektur durchgeführt wurde. Abbildung 6.5b stellt den gleichen Ausschnitt aus dem Datenstrom *nach* der Phasenanpassung dar. Auf der Abszisse sind die OFDM-Subträger aufgetragen, was einer Frequenzdarstellung entspricht, während auf der Ordinate die zeitliche Abfolge von OFDM-Symbolen abgetragen ist. Die Abbildungen 6.5c und 6.5d stellen die mittels des modifizierten CD3-Algorithmus geschätzten Übertragungsfunktionen *vor* und *nach* Filterung in Zeitrichtung über einen Zeitraum von etwa einer zehntel Sekunde (300 OFDM-Symbole im 2K-FFT-Modus) dar.

---

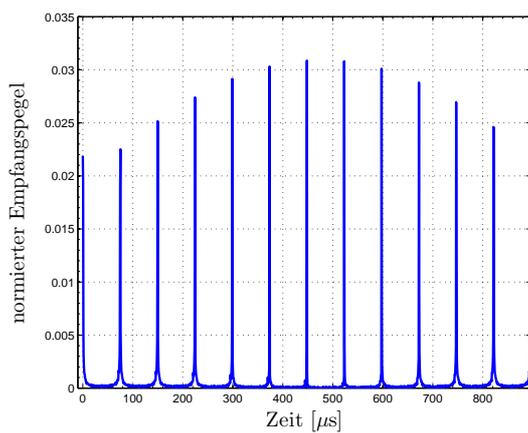
<sup>4</sup>Die Begrifflichkeit einer „Tiefpassfilterung im Zeitbereich“ verwirrt vielleicht auf den ersten Blick. Wie jedoch bereits im Abschnitt 3.6 (siehe Seite 27) beschrieben worden ist, wird die „Tiefpassfilterung“ in Wahrheit durch *zero forcing* realisiert.



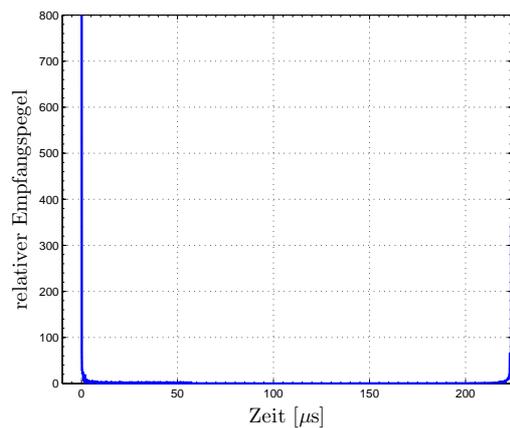
(a) Konstellationsdiagramm vor der Decodierung



(b) Kanalübertragungsfunktion nach CD3-Schätzung



(c) Impulsantwort mit Alias (vor CD3-Schätzung)



(d) Impulsantwort nach CD3-Schätzung

Abbildung 6.3: **Prinzip der Kanalschätzung:** Konstellationsdiagramm *vor*, Kanalübertragungsfunktion *nach* CD3-Schätzung sowie Impulsantwort *mit* Alias (*vor* CD3-Kanalschätzung) und *ohne* Alias nach (*nach* CD3-Kanalschätzung). Deutlich zu sehen ist der nicht kompensierte Restträger auf dem mittleren Subträger der Kanalübertragungsfunktion (idealer, rauschfreier Kanal, 2K-FFT-Modus, 6 MHz, QPSK,  $GI = 1/4$ ,  $CR = 1/2$ ).

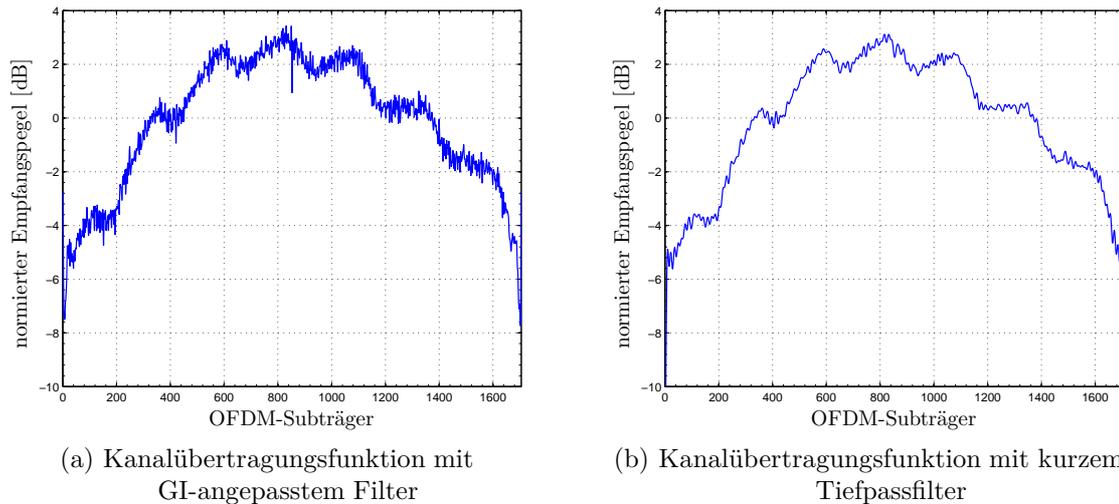
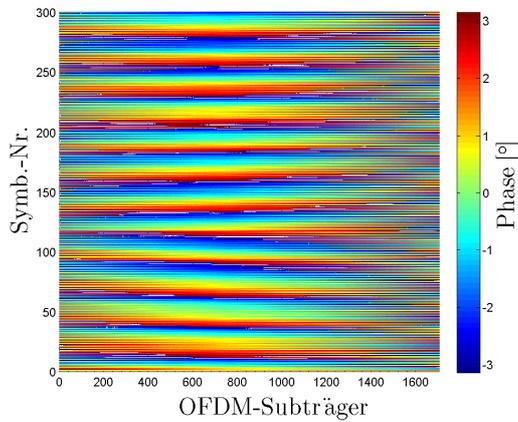


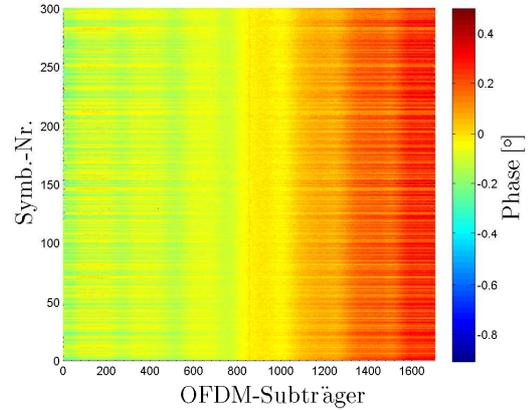
Abbildung 6.4: **Pilotbasierte Kanalschätzung im Vergleich zur Schätzung mittels des modifizierten CD3-Algorithmus:** Verbesserung der durch modifizierte CD3-Kanalschätzung erhaltenen Übertragungsfunktion (rechts) im Vergleich zur Kanalübertragungsfunktion mit GI-angepasstem Filter (idealer, rauschfreier Kanal, 2K-FFT-Modus, 6 MHz, QPSK,  $GI = 1/4$ ,  $CR = 1/2$ ).

## 6.2.2 Einfluss der Viterbi-Decodierung

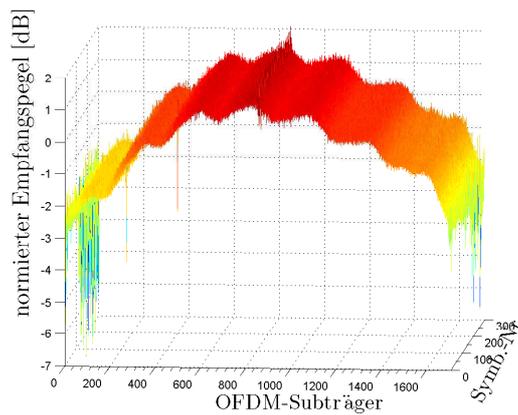
Am Beispiel von auf dem Forumsplatz der TU Braunschweig durchgeführten Messungen (Eingangsdaten `out.outdoorforumsplatz.short.real` und `out.outdoorforumsplatz.short.imag`) soll der Einfluss der Viterbi-Decodierung auf die Schätzung der Kanalübertragungsfunktion gezeigt werden. Die Parameter der Messung sind im Gleichwellennetz (SFN) Braunschweig-Hannover: 8K-FFT-Modus, 8 MHz-Kanal, 16-QAM,  $GI = 1/4$ ,  $CR = 2/3$ . Sofern nicht anders angegeben, sind alle Messungen bei einer Mittenfrequenz von 490 MHz aufgezeichnet worden. Der innere Faltungscode ist für die Korrektur von Einzelfehlern optimiert, weshalb es bei geringer Signalqualität verhältnismäßig schnell zum Decodierversagen durch Fehlerverschleppung kommen kann. Dadurch enthalten die Ausgangsdaten mehr Fehler als die Eingangsdaten. Das Problem bei den erwähnten Messungen war die maximale Kanalbandbreite des Messempfängers von 6 MHz und die daraus resultierende zusätzliche Punktierung des Signals. Der Anteil an Bits, der vor und nach der Viterbi-Decodierung und Recodierung übereinstimmt, liegt bei ungefähr 89-93 % und damit weit entfernt von der geforderten QEF-Grenze des inneren Interleavers von  $< 2 \times 10^{-4}$  fehlerhaften Bits. Zur Kanalanalyse ist das Decodierungsergebnis jedoch ausreichend. Zusätzlich muss berücksichtigt werden, dass anhand der decodierten Daten nicht entschieden werden kann, ob der Viterbi-Decoder *tatsächlich* die fehlerhaften (punktieren) Bits korrigieren konnte oder ob die Unterschiede zum Eingangsdatenstrom des Decoders durch ihn selber verursacht worden sind. Wahr-



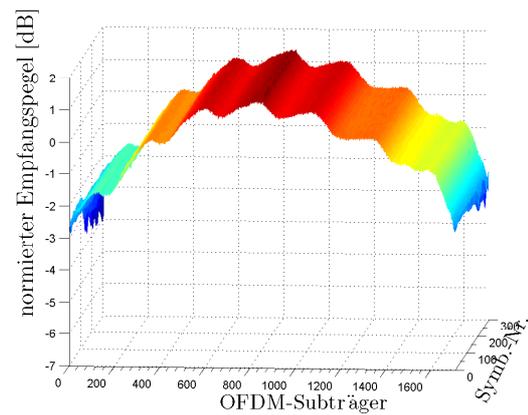
(a) Phase *ohne* Korrektur zwischen verschiedenen OFDM-Symbolen



(b) Phase *mit* Korrektur zwischen verschiedenen OFDM-Symbolen



(c) Betrag der Kanalübertragungsfunktion nach CD3-Schätzung *vor* Zeitfilterung



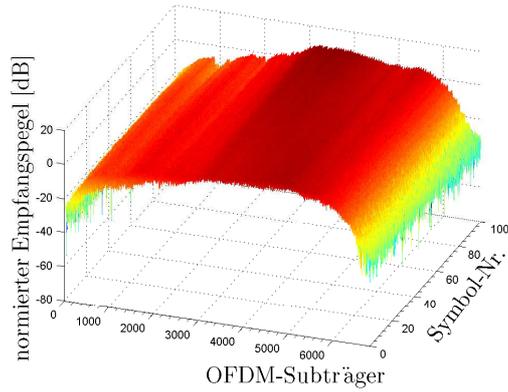
(d) Betrag der Kanalübertragungsfunktion nach CD3-Schätzung *nach* Zeitfilterung

Abbildung 6.5: **Auswirkung der Phasenkorrektur zwischen OFDM-Symbolen und Einfluss der Zeitfilterung:** Phase *vor* und *nach* Korrektur zwischen OFDM-Symbolen, Betrag der Kanalübertragungsfunktion *vor* und *nach* Filterung in Zeitrichtung. Dargestellt ist zirka  $\frac{1}{10}$  s Echtzeit (idealer, rauschfreier Kanal, 2K-FFT-Modus, 6 MHz, QPSK, GI = 1/4, CR = 1/2).

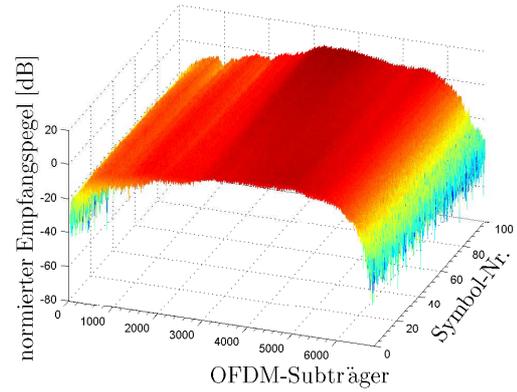
scheinlicher ist der letztgenannte Fall, da durch das eingeschränkte Frequenzspektrum des Messempfängers bereits viel Information als unsicher einzustufen ist. Es sind auf beiden Seiten des DVB-T-Kanals zusammen genommen etwas weniger als 1000 OFDM-Subträger stark gedämpft, was fast einem Achtel der Gesamtanzahl an Subträgern eines DVB-T-Signals im 8K-FFT-Modus entspricht. Bei der vereinfachenden Annahme einer Gleichverteilung der Pilote innerhalb des OFDM-Symbols sind  $1/8$  aller Pilote korrupt, was ohne Berücksichtigung weiterer Fehler im Signal zu einer Fehlerquote von zirka 12% führen müsste. Die Simulationsergebnisse belegen diese Analyse. Aus obigen Gründen muss also basierend auf den Rahmenbedingungen der Aufnahme eine Einzelfallentscheidung getroffen werden, ob die Viterbi-Decodierung verwendet werden soll oder nicht. Wenn keine sichtbaren Verbesserungen durch den zeitraubenden Prozess zu erwarten sind, genügt eine Decodierung *ohne* Auswertung des inneren Faltungscodes. Bei guten Aufnahmen bringt die Viterbi-Decodierung im Allgemeinen Vorteile. Die Abbildung 6.6 (siehe Seite 77) veranschaulicht den Einfluss einer Viterbi-Decodierung. Es wurden einhundert OFDM-Symbole analysiert, was im 8K-FFT-Modus etwa einer zehntel Sekunde Echtzeit entspricht. Die Abbildung 6.6a zeigt den zeitlichen Verlauf dieses zeitinvarianten Kanals *ohne* Viterbi-Decodierung. Abbildung 6.6b enthält den gleichen Signalausschnitt *mit* aktivierter Viterbi-Decodierung. Das 40. Symbol ist exemplarisch in den Abbildungen 6.6c und 6.6d verglichen worden, der Betrag der Differenz ist in Abbildung 6.6e verdeutlicht. Besonders anhand der Einzelsymbole ist ersichtlich, dass der Viterbi-Decoder durchaus einige Fehler in der geschätzten Kanalübertragungsfunktion korrigieren konnte, wenngleich er besonders am rechten Rand in diesem Fall aufgrund der zu geringen Tunerbandbreite Fehler „erzeugt“. Die mittlere quadratische Abweichung (MSE) liegt mit ungefähr  $-41,4$  dB auf sehr niedrigem Niveau.

### 6.2.3 Messungen auf dem Campus in einem realen DVB-T-SFN

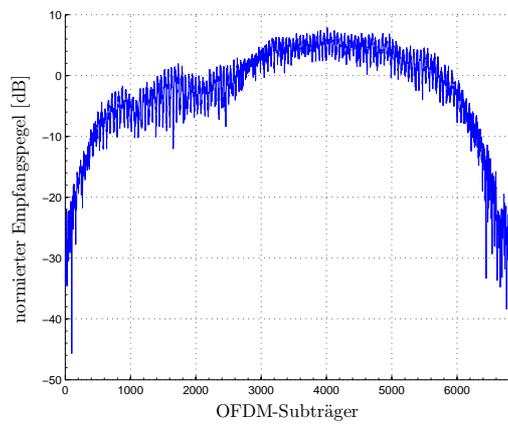
Die in Abbildung 6.7 zu findenden Messungen sind wiederum im DVB-T-SFN Braunschweig-Hannover aufgenommen worden. Damit liegt ein Signal im 8K-FFT-Modus in einem 8 MHz-Kanal vor mit den Parametern 16-QAM,  $GI = 1/4$ ,  $CR = 2/3$ . Die Messungen wurden im Säulengang am Audimax der TU Braunschweig durchgeführt. Für die Aufnahme der Daten wurde der USRP1-Messempfänger verwendet, während der Messung wurde der Empfänger im Fußgängertempo ( $\approx 5$  km/h) bewegt. Neben dem Konstellationsdiagramm (Abbildung 6.7a) sind die mittels des modifizierten CD3-Algorithmus geschätzte Kanalübertragungsfunktion (Abbildung 6.7b) und die Impulsantwort (Abbildung 6.7c) jeweils eines OFDM-Symbols dargestellt. Sehr schön zu erkennen ist die starke Dämpfung des Tuners in der Frequenzdarstellung an den beiden Rändern des Kanals, die durch die Begrenzung des Tuners auf 6 MHz zurückzuführen ist. In der Zeitdarstellung sind trotz des geringen Dynamikumfangs des USRP1 fünf Sender des Gleichwellennetzes zu erkennen, welche durch Pfeile in Abbildung 6.7c kenntlich gemacht sind. Die Laufzeit stimmt überein mit den anhand von Google Earth geschätzten Entfernungen der Sender vom Aufnahmeort. Die Signale der beiden Braunschweiger Sender (Kraftwerk und Broitzem) treffen nur wenige



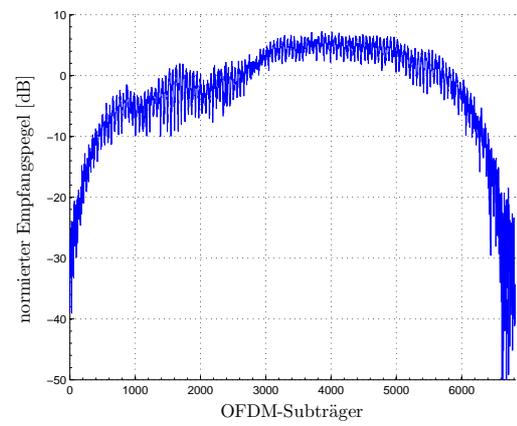
(a) Decodierung *ohne* Viterbi-Decodierung



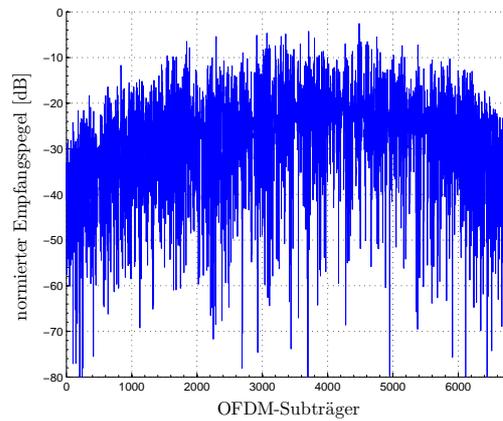
(b) Decodierung *mit* Viterbi-Decodierung



(c) Betrag der Übertragungsfunktion des 40. Symbols *ohne* Viterbi-Decodierung



(d) Betrag der Übertragungsfunktion des 40. Symbols *mit* Viterbi-Decodierung



(e) Betrag der Differenz (40. Symbol)

Abbildung 6.6: **Einfluss der Viterbi-Decodierung:** Bei zu schlechtem Signal „erzeugt“ die Viterbi-Decodierung Fehler, welche an den rechten Rändern der Beträge der Übertragungsfunktionen zu erkennen sind. MSE  $\approx -41,4$  dB (Freifeldmessung auf Forumsplatz, 8K-FFT-Modus, 8 MHz Kanal, 16-QAM, GI =  $1/4$ , CR =  $2/3$ ).

Mikrosekunden hintereinander ein, da sie nur wenige Kilometer vom Messpunkt entfernt stehen. Das Hildesheimer Signal muss eine Entfernung von ungefähr 45 km bis zum Forumsplatz der TU zurücklegen und benötigt dazu zirka 160  $\mu\text{s}$ , die Signale der Hannoveraner Sender benötigen für 54 beziehungsweise 57 km Entfernung (Luftlinie) etwa 180 und 190  $\mu\text{s}$ . Im Zeitfenster zwischen 0 und 40  $\mu\text{s}$  sind Reflexionen an den metallbedampften Scheiben und durch eine Mehrwegeausbreitung zwischen Forumsgebäude und Altgebäude ersichtlich, die jedoch aufgrund ihrer nicht senkrechten Lage zur Verbindungslinie zwischen den Sendern (speziell dem Sender Kraftwerk) und dem Empfänger nicht im Einzelnen zugeordnet werden können. Im späteren Verlauf (nach 40  $\mu\text{s}$ ) der Impulsantwort ist der Rauschteppich bis 224  $\mu\text{s}$  in der Darstellung ersichtlich, später ist das Signal durch *zero forcing* zu Null gesetzt, da per definitionem keine Echos mehr nach dem Guard Interval eintreffen dürfen.

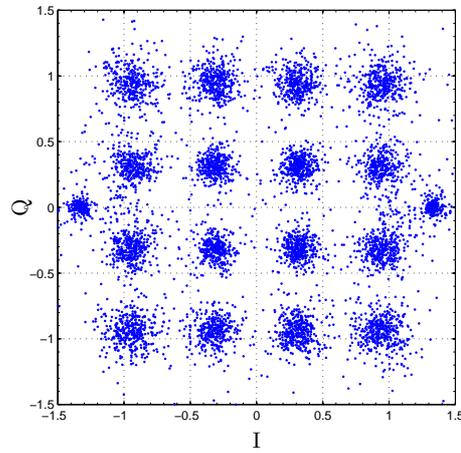
## 6.2.4 Zeitliche Filterung

Eine Filterung in Zeitrichtung kann nach Auswertung mehrerer OFDM-Symbole durchgeführt werden. Dadurch fallen einerseits komplett fehlerhafte Symbole – beispielsweise entstanden durch eine nicht zu kompensierende fehlerhafte Synchronisation – nicht mehr so stark ins Gewicht, andererseits werden ebenso *tatsächlich* in der Übertragungsfunktion einzelner OFDM-Symbole aufgetretene Effekte durch die Filterung entfernt. Die Zeitfilterung liefert also eine geglättete Darstellung auf Kosten der Genauigkeit bezüglich einzelner Symbole. Insgesamt wirkt sich für die Analyse zeitvarianter Kanäle eine zeitliche Filterung positiv aus, da es weniger auf die exakte Kenntnis des Einzelsymbols, sondern vielmehr um die Änderung der Kanalübertragungsfunktion über der Zeit ankommt. Deshalb sollte bei längeren Sequenzen die zeitliche Filterung in der Regel aktiviert werden. Bei langsam veränderlichen Kanälen kann ohne Fehler zeitlich gefiltert werden. Im Programm wird bei Aktivierung der zeitlichen Filterung ein Hamming-Filter der Ordnung 50 mit einem normierten Durchlassbereich von 0,05 verwendet. Die Filterkoeffizienten wurden mit dem MATLAB *Filter Design & Analysis Tool* (`fdatool`) erzeugt. Es ist zu berücksichtigen, dass jedes digitale Filter einen Anlaufbereich hat, sodass die ersten und letzten Symbole (entsprechend der Filterordnung) verworfen werden müssen.

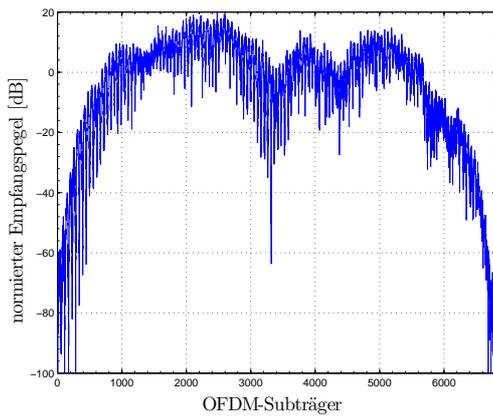
In den Abbildungen 6.8a und 6.8b ist der Einfluss der Zeitfilterung zu erkennen. Ersteres Bild zeigt den Betrag der CD3-geschätzten Kanalübertragungsfunktion<sup>5</sup> des noch nicht gefilterten Signals, letzteres den Betrag des gefilterten nach Anpassung der Phase zwischen verschiedenen Symbolen (siehe oben). Dargestellt ist ein Ausschnitt der gleichen Messung wie im vorigen Unterkapitel beschrieben (Säulengang am Audimax). Einige Beobachtungen sollen besonders hervorgehoben werden. Das ungefilterte Signal weist eine wesentlich gröbere Struktur auf, da jede kleine Abweichung der geschätzten Übertragungsfunktion dargestellt ist. Auf den mittleren Frequenzträgern ist zu erkennen, dass durch die Zeitfilterung sogar scheinbar größere

---

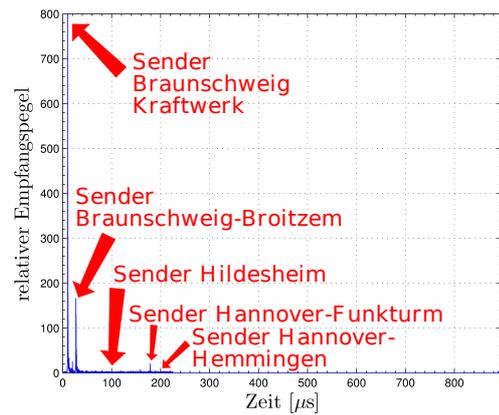
<sup>5</sup>Wenn im folgenden von der „CD3-geschätzten Übertragungsfunktion“ gesprochen wird, ist damit jeweils die „mittels des modifiziertem CD3-Algorithmus geschätzte Kanalübertragungsfunktion“ gemeint.



(a) Konstellationsdiagramm vor der Decodierung



(b) Kanalübertragungsfunktion nach CD3-Schätzung



(c) Impulsantwort nach CD3-Schätzung

Abbildung 6.7: **Messungen in einem realen DVB-T-SFN:** Konstellationsdiagramm *vor*, sowie Impulsantwort und Kanalübertragungsfunktion *nach* CD3-Schätzung (Freifeldmessung im Säulengang am Audimax, 8K-FFT-Modus, 8 MHz, 16-QAM,  $GI = 1/4$ ,  $CR = 2/3$ ).

Fehler entstehen. Unter Berücksichtigung des Pegels von ungefähr  $-10$  dB ist dieser Fehler vernachlässigbar, wie auch durch den Differenzplot in Abbildung 6.8c deutlich wird. Die Fehler um die OFDM-Subträger 2000 und 5000 sind zwar in der dreidimensionalen Ansicht weniger sichtbar, pegelmäßig aber bedeutender als diejenigen um die Mittenträger herum. Die durch den zu schmalen Tuner bedingte Randdämpfung ist erneut deutlich sichtbar und produziert ebenfalls Fehler, die zwar durch den geringen absoluten Pegel auch im Differenzplot nicht dramatisch wirken, jedoch insgesamt zu unzuverlässigen Decodierungs- und damit Schätzungsergebnissen führen. Der Einfluss wurde bereits im Abschnitt 6.2.2 besprochen und soll hier nicht erneut ausgeführt werden. Die Darstellungen zeigen jeweils 500 OFDM-Symbole und damit ungefähr eine halbe Sekunde Echtzeit. Bei Fußgängertempo hat sich der Empfänger etwa einen dreiviertel Meter weiterbewegt. Die Änderung des Kanals in dieser Zeit lässt sich in den Verlaufsdarstellungen und auch am Differenzplot bereits erkennen. Leider können mangels RAM keine längeren Sequenzen dargestellt werden, der zeitliche Einfluss auf die Kanalübertragungsfunktion ist jedoch bereits hier abzulesen.

### 6.2.5 Einfluss der inneren Konstellationspunkte

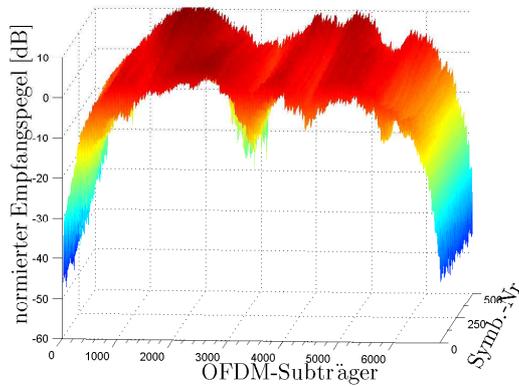
Durch Vernachlässigung von Werten der CD3-geschätzten Kanalübertragungsfunktion, deren Konstellationspunkt eine minimale Amplitude unterschreitet, können Fehlschätzungen verhindert werden. Die Verbesserung der Schätzung ist in der Abbildung 6.9 verdeutlicht. Es handelt sich um einen zeitvarianten Kanal, der während einer Tramfahrt aufgezeichnet wurde (Beschreibung siehe unten). Insbesondere bei zeitvarianten Kanälen können die inneren Konstellationspunkte durch Rauschen in falsche Quadranten abgebildet werden, was bei der Kanalschätzung zu schwerwiegenden Fehlern führen kann. Aufgrund ihrer größeren Amplitude ist dieser Effekt für die äußeren Punkte des Konstellationsdiagramms sehr viel unwahrscheinlicher. In Abbildung 6.9a ist die Kanalschätzung unter Berücksichtigung *aller* Konstellationspunkte durchgeführt worden, in Abbildung 6.9b musste der Betrag eines verwendeten Punktes bezogen auf die im Standard [DVB06] definierten Amplituden 1,5 betragen, in Abbildung 6.9b musste er 3,3 überschreiten. Deutlich zu erkennen ist der systemimmanente Rauschunterdrückungseffekt, der dadurch auftritt, dass die fehlenden Datenpunkte interpoliert werden und somit weniger Schwankungen in der Übertragungsfunktion möglich sind als ohne Interpolation.

### 6.2.6 Der zeitvariante TU6-Kanal

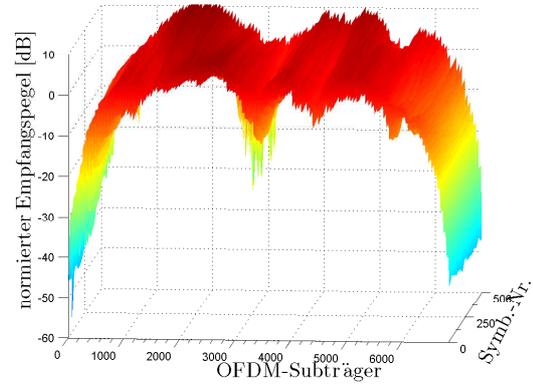
Das standardisierte Kanalmodell *TU6* (*Typical urban with 6 paths*) wird häufig zur Modellierung bewegter Empfänger verwendet. Der dargestellte Kanal wurde mit einer Dopplerfrequenz<sup>6</sup> von 25 Hz aufgenommen, was bei einer ungefähren Sendefrequenz

---

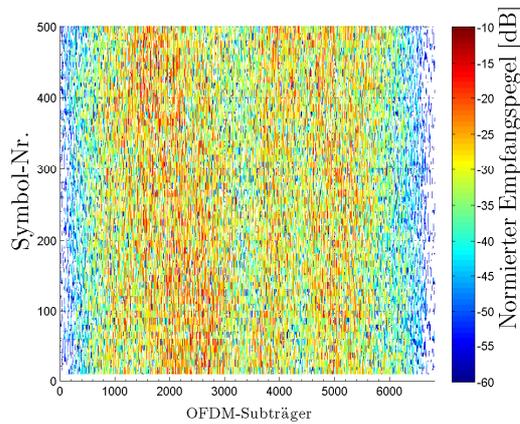
<sup>6</sup>Die Dopplerfrequenz berechnet sich bei einem festen Sender und einem beweglichen Empfänger zu  $f_B = f_S \cdot (1 + \frac{v}{c})$ , wobei  $f_B$  die vom Beobachter wahrgenommene Frequenz ist,  $f_S$  die Sendefrequenz,  $v$  die Bewegungsgeschwindigkeit des Empfängers und  $c$  die Ausbreitungsgeschwindigkeit der elektromagnetischen Wellen (Lichtgeschwindigkeit  $c \approx 300 \cdot 10^6$  m/s).



(a) *Ohne* Filterung in Zeitrichtung

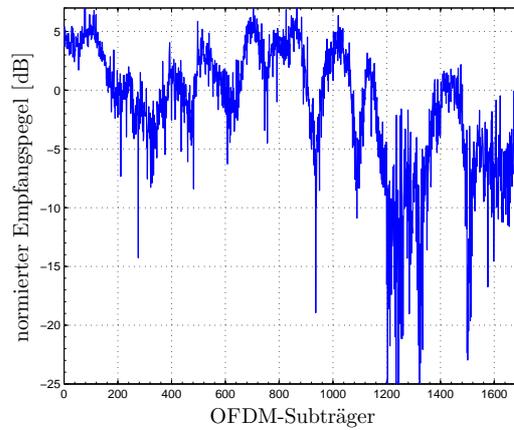


(b) *Mit* zeitlicher Filterung

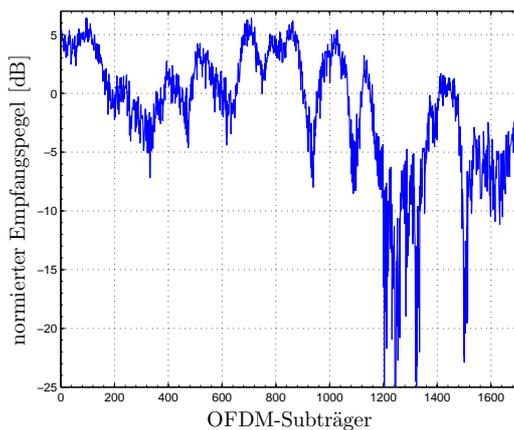


(c) Betrag der Differenz der Übertragungsfunktionen

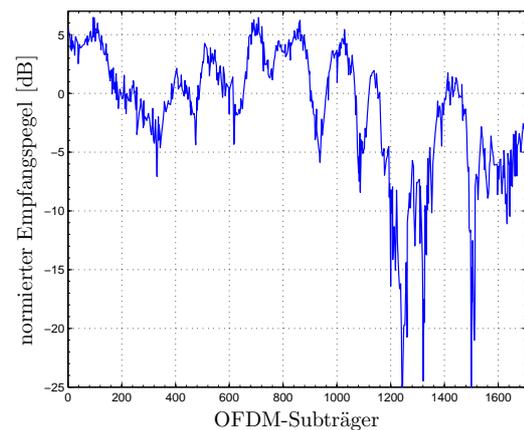
Abbildung 6.8: **Einfluss der Zeitfilterung:** Betrag der Kanalübertragungsfunktionen *ohne* und *mit* Filterung in Zeitrichtung: Eine nachträgliche Anpassung der Phase wurde durchgeführt. Zusätzlich ist das durch die Zeitfilterung entstandene Fehlersignal zu sehen, dessen mittlere quadratische Abweichung bei  $-39,3$  dB liegt. Es wurden 500 OFDM-Symbole analysiert, was in etwa einer halben Sekunde Echtzeit entspricht (Freifeldmessung im Säulengang am Audimax der TU, 8K-FFT-Modus, 8 MHz, 16-QAM,  $GI = 1/4$ ,  $CR = 2/3$ ).



(a) Mit allen reencodierten  
Konstellationspunkten geschätzte  
Kanalübertragungsfunktion



(b) Mit Konstellationspunkten mit Amplitude  
> 1,5 geschätzte  
Kanalübertragungsfunktion



(c) Mit Konstellationspunkten mit Amplitude  
> 3,3 geschätzte  
Kanalübertragungsfunktion

Abbildung 6.9: **Einfluss der inneren Konstellationspunkte auf die Schätzung der Kanalübertragungsfunktion:** Durch Interpolation von Subträgern, deren Amplitude im Konstellationsdiagramm zu gering ist, kann die Qualität der Kanalschätzung verbessert werden (Messung während Tramfahrt, 2K-FFT-Modus, 6 MHz, 16-QAM, GI =  $1/4$ , CR =  $1/2$ ).

von 600 MHz einer Bewegungsgeschwindigkeit des Empfängers von zirka 45 km/h entspricht. Die Analyse des Kanals wurde mit den aktivierten Optionen *Postprocessing* (nachträgliche Anpassung der Phase zur Vermeidung von Phasensprüngen *zwischen* zwei OFDM-Symbolen) und *Viterbi-Decodierung* in einem 6 MHz breiten Kanal im 2K-FFT-Modus mit einer Coderate  $1/2$  und einer Guard Interval-Länge von  $1/4$  durchgeführt. Das Ergebnis ist in der Abbildung 6.10 auf Seite 84 illustriert. Dabei stellt Abbildung 6.10a eine Draufsicht ohne Zeitfilterung dar, in Abbildung 6.10b ist das Ergebnis nach der Filterung in Zeitrichtung zu sehen. Abbildung 6.10c zeigt den Betrag der Differenz. Der mittlere quadratische Fehler liegt bei zirka  $-34,9$  dB. Auf der Abszisse sind jeweils die OFDM-Subträger aufgetragen, die Ordinate stellt die unterschiedlichen OFDM-Symbole in Zeitrichtung dar. Die Pegel sind durch verschiedene Farben kenntlich gemacht. Dabei symbolisieren blaue Farben einen geringen Pegel; die Farben grün, gelb bis hin zu rot kennzeichnen einen größeren Empfangspegel.

In dem zeitlich ungefilterten Signal ist bei dem 40. OFDM-Symbol eine Unstetigkeit zu erkennen, die auf einen Phasensprung zurückgeht, der durch zu großes Phasenrauschen des Empfängers zurückzuführen ist und nicht mehr durch die nachträgliche Phasenanpassung kompensiert werden konnte. Es entsteht Intercarrierinterferenz (ICI). Ebenso ist diese Stelle als starkes Fehlersignal im Differenzplot zu erkennen, da die zeitliche Filterung ohne Berücksichtigung der defekten Phase einfach über die Unstetigkeit hinweggefiltert hat. Ebenso sind große Phasensprünge in der Differenzabbildung bei den OFDM-Symbolen 120 und 200 auszumachen. Die dreihundert dargestellten Symbole entsprechen in etwa  $\frac{1}{10}$  s Echtzeit. Da die Kanalbandbreite 6 MHz beträgt, ist in den Graphiken der Frequenzbereich von 551 bis 557 MHz dargestellt (Mittelfrequenz 554 MHz). Im oberen Bereich der Differenzdarstellung um das Symbol 250 herum sind vereinzelte helle Flecken zu sehen. Diese sind durch den knapp zu eng gewählten Durchlassbereich von 0,05 des Zeitfilters entstanden. Bisherige Filterungen konnten hingegen problemlos absolviert werden, da bei ihnen zwar auch mitunter eine Zeitvarianz vorlag, jedoch nicht mit einer so hohen Dopplerfrequenz.

### 6.2.7 Effekte bei vergleichbaren Pegeln verschiedener Sender

Ebenfalls im SFN Braunschweig-Hannover wurde die im Folgenden beschriebene Messung durchgeführt. Das Besondere hierbei ist, dass die Messung im westlichen Ringgebiet in Braunschweig aufgenommen worden ist, also eigentlich inmitten des Kernversorgungsgebietes der beiden Braunschweiger Sender. Dennoch ist der DVB-T-Empfang mit kommerziellen Empfängern nur eingeschränkt möglich. Nach Analyse des Kanals wird der Grund sichtbar. Die Impulsantwort mit den vergleichbaren Pegeln der Braunschweiger Sender ist in Abbildung 6.11a zu sehen. Die Übertragungsfunktion zeigt Abbildung 6.11b. Speziell in der linearen Darstellung (Abbildung 6.11c) ist die destruktive Interferenz ersichtlich, die Einbrüche der Übertragungsfunktion bis fast auf null verursachen. Sendepegelunabhängig ist also in einem solchen Fall durch den Einfluss des Kanals kein guter Empfang möglich.

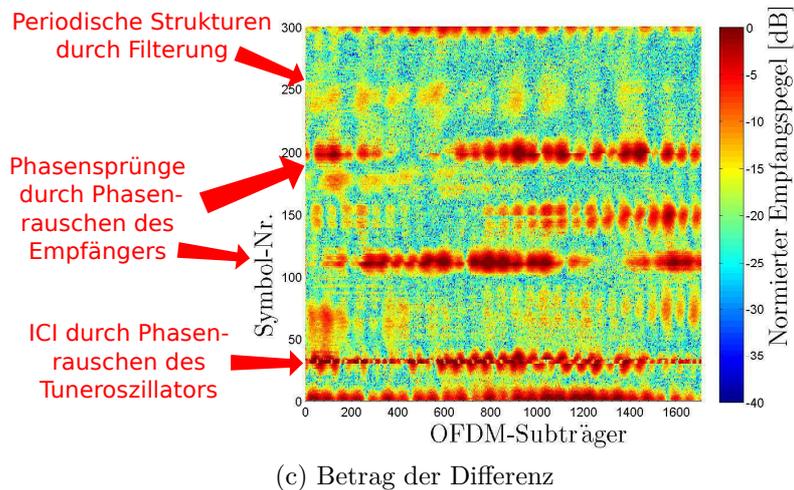
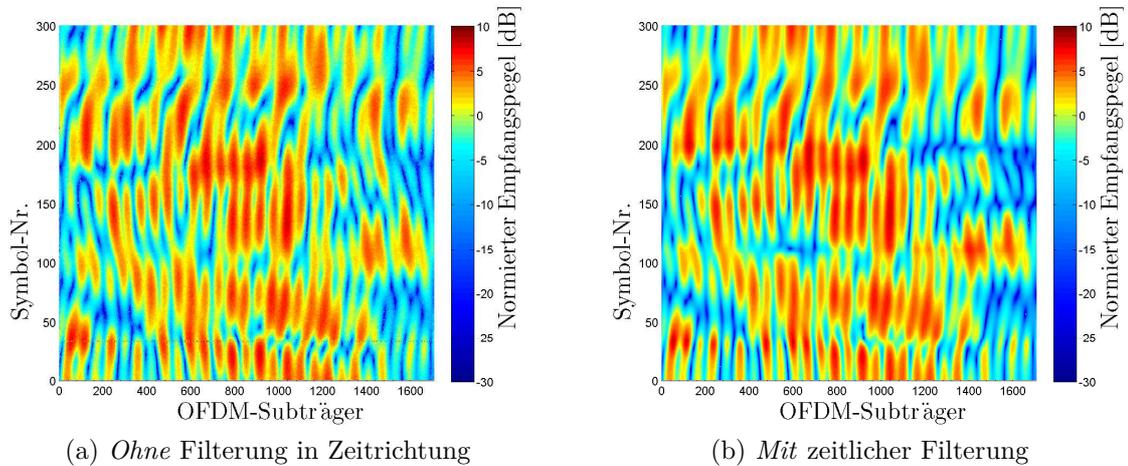
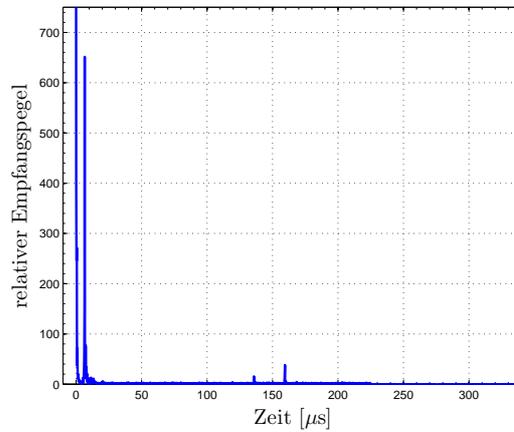
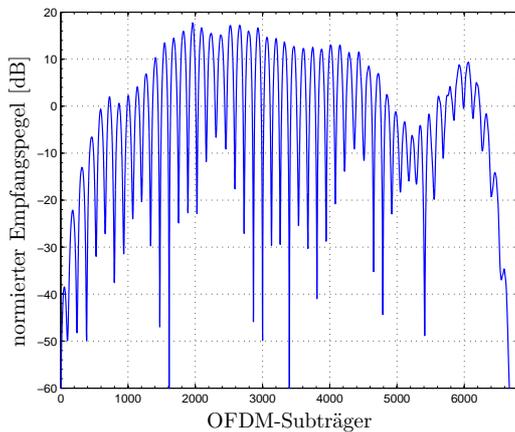


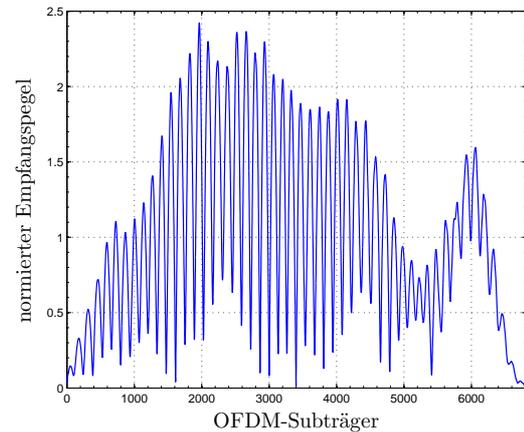
Abbildung 6.10: **Der TU6-Kanal:** Betrag der Kanalübertragungsfunktionen *ohne* und *mit* Filterung in Zeitrichtung sowie der Betrag des Differenzsignals. Eine nachträgliche Anpassung der Phase wurde durchgeführt (*Postprocessing*). Ebenso wurde eine Viterbi-Decodierung verwendet. Es wurden 300 OFDM-Symbole analysiert, was in etwa  $\frac{1}{10}$  s Echtzeit entspricht (TU6-Kanal mit 6 MHz Bandbreite im 2K-FFT-Modus mit QPSK,  $GI=1/4$ ,  $CR=1/2$ ). Der mittlere quadratische Fehler liegt bei zirka  $-34,9$  dB.



(a) Impulsantwort



(b) Kanalübertragungsfunktion (logarithmische Darstellung)



(c) Kanalübertragungsfunktion (lineare Darstellung)

Abbildung 6.11: **Vergleichbare Empfangspegel mehrerer Sender:** Impulsantwort und Übertragungsfunktion in logarithmischer und linearer Darstellung. Dargestellt ist der Fall, dass zwei verschiedene Sender eines SFN beim Empfänger in etwa mit demselben Empfangspegel eintreffen. Der Pegel der Kanalübertragungsfunktion bricht in diesem Fall an vielen Stellen (in der linearen Darstellung fast bis auf 0) stark ein, was eine Nutzdatendecodierung erschwert bzw. unmöglich macht (8K-FFT-Modus, 8 MHz, 16-QAM,  $GI = 1/4$ ,  $CR = 2/3$ ).

## 6.2.8 Messungen während einer Straßenbahnfahrt – 2. Versuch

Messungen der Kanalübertragungsfunktion während einer Straßenbahnfahrt sind besonders interessant, weil verschiedene Störeffekte gleichzeitig auf den Kanal einwirken. Die Fenster der Bahn sind aus Wärmeschutzgründen metallbedampft, was jedoch gleichzeitig das Durchdringen der Scheiben für elektromagnetische Strahlung erschwert. Zudem treten beim Übergang verschiedener Oberleitungen an den Kopplungsstellen möglicherweise impulsartige Störungen auf, die sich auf die Empfangsqualität auswirken können. Nicht zuletzt ist eine Tram mit relativ hohen Geschwindigkeiten von einigen  $10 \text{ km/h}$  unterwegs, was selbst unter Vernachlässigung aller Störeinflüsse einen zeitvarianten Kanal zur Folge hat. Die Messungen mussten mehrfach durchgeführt werden, das Misslingen der ersten Aufnahmen wird weiter unten im Kapitel „Grenzen“ ab Seite 87 erörtert. Als Stromversorgung wurde bei den hier beschriebenen Messungen wie auch bei den meisten anderen Außenaufnahmen auf eine Motoradatterie zurückgegriffen. In Abbildung 6.12 ist das Ergebnis abgedruckt, welches zwar sehr schön die Zeitvarianz des Kanals illustriert, jedoch die vermuteten Störeffekte entweder nicht oder nur in so schwacher Form enthält, dass sie keinen Einfluss auf die Kanalübertragungsfunktion haben. Als Sendeparameter wurden auf der IfN-Frequenz 554 MHz im 2K-FFT-Modus und einem 6 MHz breiten Kanal eine 16-QAM mit  $GI = 1/4$  und  $CR = 1/2$  verwendet. Der 2K-FFT-Modus wurde bewußt verwendet, weil er aufgrund der kürzeren Gesamtsymboldauer im Vergleich mit dem 8K-FFT-Modus weniger anfällig gegen Phasenrauschen ist, welches eine Hauptstörgröße speziell bei diesem Tuner darstellt.

## 6.2.9 Messungen hinter Windkraftanlagen

Die durch Windkraftanlagen in einem DVB-T-Gleichwellennetz hervorgerufenen Störungen erlangen heutzutage immer größere Bedeutung, da die Anzahl dieser Stromerzeuger in den letzten Jahren stark gestiegen ist und in den nächsten Jahren voraussichtlich noch weiter wachsen wird. Die Messung der folgenden Abbildungen erfolgte an der Kläranlage in Salzgitter an der Kreuzung zwischen K50 und K51 mit Sichtverbindung (*Line of sight*, LOS) zum Sender Braunschweig-Broitzem auf einer Frequenz von 490 MHz. In der LOS befanden sich zahlreiche Windräder. Die Darstellung der Impulsantwort über der Zeit (über verschiedene OFDM-Symbole) ist in Abbildung 6.13a zu sehen und offenbart keine Überraschungen. Deutlich ersichtlich ist das Signal des Senders in Braunschweig-Broitzem, auf den die Darstellung normiert wurde. Kurz dahinter ist mit viel geringerer Amplitude das Signal des Senders Braunschweig-Kraftwerk zu sehen. Der starke Pegelunterschied erklärt sich durch die nicht vorhandene LOS und durch die Hauptabstrahlrichtung dieses Senders in Richtung Osten. Ebenfalls eindeutig zuzuordnen ist der Hildesheimer Sender, der sich in einer Entfernung von zirka 21 km befindet. Sein Peak taucht erwartungsgemäß bei 70  $\mu\text{s}$  auf. Ein weiterer Beweis ist die Nichtexistenz dieses Signals am gleichen Standort bei einer Frequenz von 498 MHz, auf welcher die privaten Sender in Hildesheim *nicht* ausgestrahlt werden.

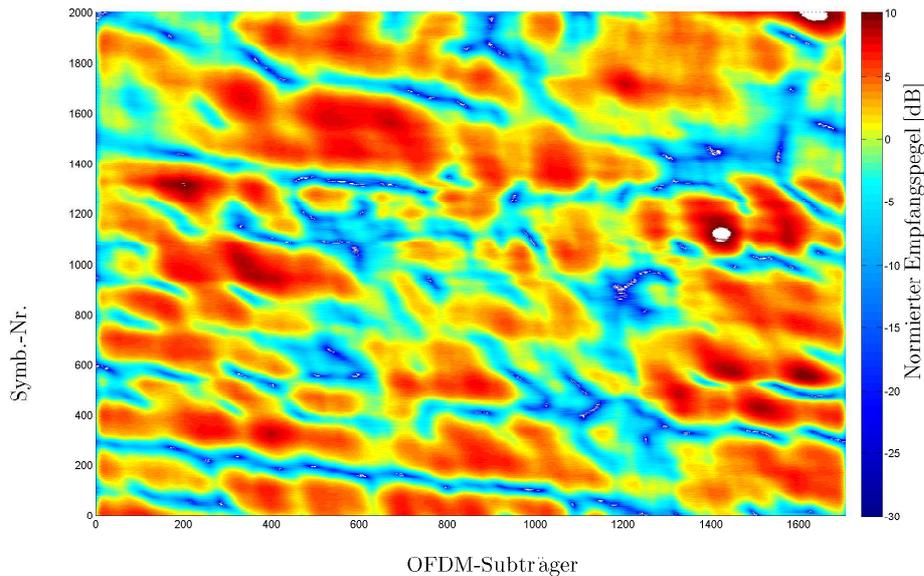


Abbildung 6.12: **Straßenbahnfahrt:** Zeitvariante Kanalübertragungsfunktion im SFN. Die Messung wurde während einer Tramfahrt (M5) vom HBF zum Rathaus durchgeführt. Dargestellt ist ungefähr eine viertel Sekunde Echtzeit (2K-FFT-Modus, 6 MHz, 16-QAM,  $GI = 1/4$ ,  $CR = 1/2$ , mit Viterbi-Decodierung, nach Filterung in Zeitrichtung).

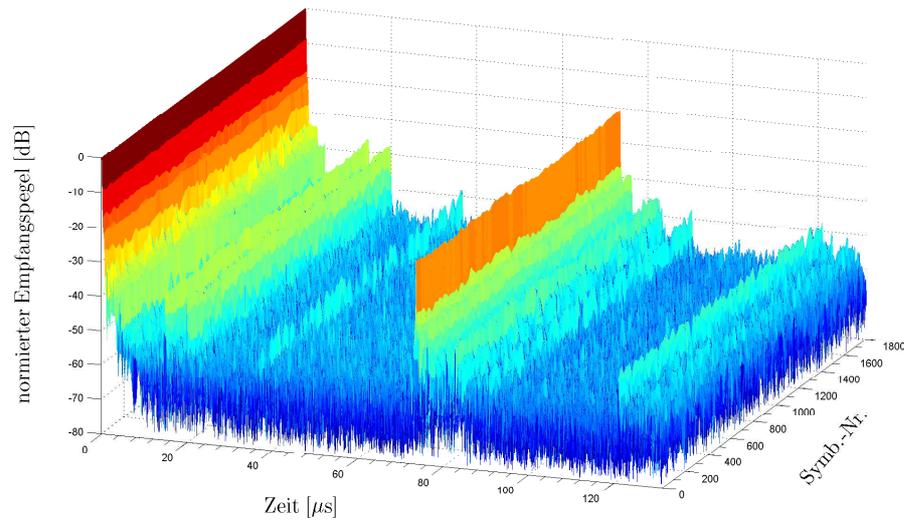
Für die Darstellung eines Teiles der Impulsantwort in Abbildung 6.13b ist der Hauptpfad ausgeblendet und eine lineare Darstellung gewählt worden. Somit werden in den abgebildeten zwei Sekunden Echtzeit sechs deutliche Amplitudeneinflüsse sichtbar, die eine Periodizität aufweisen, die einen zufälligen Störeinfluss ausschließen lassen. Jeweils jeder dritte starke Peak in dem Ausschnitt der Impulsantwort weist das gleiche Muster (Kennzeichnung in der Graphik durch Pfeile) auf und ist damit jeweils einem Windrad zuzuordnen.

## 6.3 Grenzen

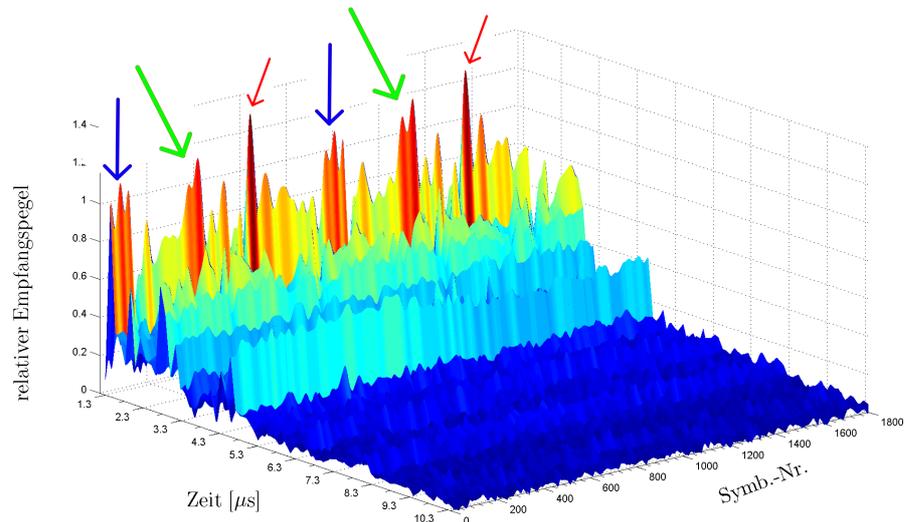
Durch die vielfältigen Möglichkeiten, Einfluss auf die Synchronisation und die Decodierung durch das Programm zu nehmen, können auch Datenströme schlechter Qualität noch decodiert werden. Gleichwohl gibt es Grenzen für die Decodierung. Diese sollen im aktuellen Abschnitt beschrieben werden.

### 6.3.1 Messungen während einer Straßenbahnfahrt – 1. Versuch

Zielsetzung und Ergebnisse dieser Messung sind bereits diskutiert worden. Hier sollen Probleme näher beleuchtet werden, die speziell während der ersten Messungen mit dem USRP1 während der Straßenbahnfahrt auftraten. Am problematischsten war die



(a) Impulsantwort mit Hauptpfad (logarithmische Darstellung)



(b) Impulsantwort ohne Hauptpfad (lineare Darstellung)

Abbildung 6.13: **Messungen hinter Windkraftanlagen:** Impulsantwort eines im SFN Braunschweig-Hannover gemessenen Signals mit Windkraftanlagen in der LOS zwischen Sender und Empfänger. Wird der Hauptpfad ausgeblendet, sind die durch verschiedene Windkrafträder periodisch erzeugten Beeinträchtigungen des Amplitudenverlaufs sichtbar. Um den Effekt hervorzuheben, wurde für letztere Darstellung eine lineare Skalierung gewählt. Messung am Klärwerk Salzgitter (K50, K51), die 1800 OFDM-Symbole entsprechen knapp zwei Sekunden Echtzeit (8K-FFT-Modus, 8 MHz, 16-QAM,  $GI = 1/4$ ,  $CR = 2/3$ ).

mobile Spannungsversorgung, die durch die Batterie eines fernsteuerbaren Autos sichergestellt werden sollte. Die Spannung sank jedoch bereits nach wenigen Sekunden auf ein Niveau ab, welches für den Betrieb des Messgerätes nicht mehr ausreichend war. Daher konnte dieses leider pro Messperiode nur sehr kurz in Betrieb genommen werden, was eine Einstellung der idealen Verstärkung unmöglich machte. Die Messungen mussten deshalb mit einem zuvor empirisch ermittelten Schätzwert für die Verstärkung durchgeführt werden. Die Analyse der Messdaten schlug schließlich wegen eines zu geringen Signal-zu-Rausch-Verhältnisses (SNR) aufgrund variierender Umgebungsverhältnisse in der sich schnell bewegenden Bahn fehl. Ebenfalls der Qualität der Messdaten abträglich war der geringe Dynamikumfang des USRP1 von lediglich 12 Bit. Die mit dem USRP2 (14 Bit Auflösung) durchgeführten Messungen sind bekanntlich geglückt.

### **6.3.2 Das Konstellationsdiagramm und zu geringer Signal-Rauschabstand**

Der Signal-Rauschabstand wurde bereits des öfteren erwähnt, wenn es um die Qualität von Signalen und deren Decodierbarkeit ging. Als geeignete Sichtprüfung zu Beginn der Analyse hat sich das Konstellationsdiagramm bewährt, was bei einem decodierbaren Signal (zumindest in der überwiegenden Anzahl an OFDM-Symbolen) deutlich erkennbare Konstellationspunkte ausprägen sollte. Sollte keine Decodierung mehr möglich sein, wird schon im Konstellationsdiagramm eine ungeordnete Punktwolke auftauchen. Der schlimmste anzunehmende Fall tritt auf, wenn die mindestens benötigte Signalqualität genau an der Einsprungstelle des Programms über mehrere Symbole hintereinander zu gering ist. Dort wird die zeitliche Synchronisation versagen, was immer einen Programmabbruch zur Folge hat. Problematisch, aber eventuell noch möglich wird die Decodierung, wenn zwar die initiale Synchronisation erfolgreich durchgeführt werden konnte, aber sich das Signal zwischenzeitlich derart verschlechtert, dass die Synchronisation verloren geht. In einigen Fällen – wenn die Frequenzdrift bei den fehlerhaften Symbolen nicht zu ausgeprägt war – gelingt eine erneute Synchronisation auf den Datenstrom. In einem solchen Fall ist allerdings zu überlegen, ob die geschätzten Kanalübertragungsfunktionen der fehlerhaften Symbole nachträglich manuell aus den Ausgabedateien entfernt werden sollten.

# 7 Zusammenfassung

In vorliegender Diplomarbeit sollten DVB-T-Übertragungskanäle mit Hilfe einer zu erstellenden MATLAB-Software analysiert werden und die Qualität der Schätzung unter besonderer Berücksichtigung iterativer Algorithmen verbessert werden, sodass aussagekräftige Kanalmodelle für MISO- und MIMO-Systeme daraus abgeleitet werden können. In den ersten Kapiteln wurden allgemeine Grundlagen der Thematik vermittelt und verschiedene Probleme behandelt. Darüber hinaus wurden unterschiedliche Möglichkeiten der Kanalschätzung beleuchtet. Die letztlich entstandene Analysesoftware für die Schätzung von Kanalübertragungsfunktionen in DVB-T-Netzen wurde detailliert beschrieben und spezieller Wert auf die Beschreibung von während der Implementierung aufgetretener Probleme und deren Beseitigung gelegt. Es wurden Messungen durchgeführt, die zunächst die Funktionstüchtigkeit der erstellten Software demonstrieren sollten. Im Kapitel 6 wurden schließlich verschiedene Szenarien untersucht und analysiert. Der Bedienung der Software für interessierte Nutzer ist das Kapitel 5 gewidmet. Die Software wie auch ausgewählte Messdaten befinden sich auf dem zu dieser Arbeit gehörenden Datenträger, weshalb von einem kompletten Codelisting abgesehen wird.

Zusammenfassend kann konstatiert werden, dass die Ergebnisse der Kanalanalyse bei bestehenden Kanalmodellen wie dem TU6-Kanal den Erwartungen entsprochen haben, während auch einige neue Erkenntnisse gewonnen werden konnten. Dazu zählen etwa das Verhalten eines zeitvarianten Kanals während einer Straßenbahnfahrt. Mittels des modifizierten CD3-Algorithmus konnte die Qualität der Kanalschätzung im Vergleich zur klassischen pilotbasierten Schätzung, wie sie in DVB-T vorgesehen ist, derart gesteigert werden, dass die Ableitung entsprechender Kanalmodelle für MISO- und MIMO-Systeme problemlos möglich ist. Dabei wurde zur zusätzlichen Qualitätssteigerung die Möglichkeit untersucht, nur äußere Konstellationspunkte, die aufgrund ihrer großen Leistung robuster gegen Rauscheinflüsse sind als innere Punkte des Konstellationsdiagramms, für die Decodierung zu verwenden. Die Trennung der Impulsantworten einzelner Sender ist ohne Schwierigkeiten auch nachträglich noch durchzuführen wie aus den Abdrucken verschiedener Impulsantworten hervorgeht. Somit ist auch eine Einzelanalyse des Kanalverhaltens in Gleichwellennetzen, die in der derzeit verwendeten Form nichts anderes als MISO-Systeme darstellen, möglich.

Aufbauend auf den Ergebnissen dieser Arbeit könnte unter Verwendung eines 8 MHz breiten Tuners mit weniger Phasenrauschen sicherlich die Empfangsqualität und somit die Genauigkeit der Kanalschätzung weiter verbessert werden. Insbesondere böte sich die Möglichkeit der Auswertung des äußeren Fehlerschutzes in Form eines Reed-Solomon-Decoders. In der hier verwendeten Versuchsanordnung konnten aufgrund der beschränkten Bandbreite des Messtuners von 6 MHz für 8 MHz brei-

te Kanäle nur Teile des Spektrums qualifiziert ausgewertet werden. Die durch diese Einschränkung entstehenden hohen Restfehlerraten nach der Viterbi-Decodierung für reale Kanäle (wie sie in Deutschland verwendet werden) verbieten den Einsatz eines Reed-Solomon-Decoders, da dieser sofortiges Decodiersversagen erlitten hätte.

# Literaturverzeichnis

- [Bar99] BARRETT, Curtis: Fractional/Integer-N PLL Basics. In: *Texas Instruments, Technical Brief SWRA029* (1999), August. <http://focus.ti.com/lit/an/swra029/swra029.pdf>
- [DVB06] DVB-T-STANDARD: *ETSI EN 300 744 v1.6.1: Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television*, September 2006
- [DVB09] DVB-T2-STANDARD: *ETSI EN 302 755 V1.1.1: Digital Video Broadcasting (DVB); Frame structure channel coding and modulation for a second generation digital terrestrial television broadcasting system (DVB-T2)*, September 2009
- [Gru02] GRUNWALD, Sigmar: Messungen an MPEG2- und DVB-T-Signalen (5). In: *Neues von Rohde & Schwarz* (2002), Februar, Nr. Heft 174, 42–46. [http://www2.rohde-schwarz.com/file\\_3978/n174\\_mpeg2\\_de.pdf](http://www2.rohde-schwarz.com/file_3978/n174_mpeg2_de.pdf)
- [H<sup>+</sup>03] HANZO, L. u. a.: *OFDM and MC-CDMA for Broadband Multi-User Communications, WLANs and Broadcasting*. John Wiley & Sons, Ltd., The Atrium, 2003
- [Huf06] HUFSCHMID, Markus: *Information und Kommunikation: Grundlagen und Verfahren der Informationsübertragung*. 1. Auflage. B. G. Teubner Verlag / GWV Fachverlage GmbH, 2006
- [Kü08] KÜRNER, Thomas: *Skript zur Vorlesung Codierungstheorie*. Wintersemester 2007/2008. – basierend auf der Version von 2004–2005 von Dipl.-Ing. Andreas Hecker
- [Mak09] MAKNI, Mariem: *Implementierung der MISO-Kanalschätzung für DVB-T2*. Braunschweig, Technische Universität Braunschweig, Institut für Nachrichtentechnik, Studienarbeit, April 2009. – Prüfer: Prof. Dr.-Ing. Ulrich Reimers
- [MM96] MIGNONE, Vittoria ; MORELLO, Alberto: CD3-OFDM: A Novel Demodulation Scheme for Fixed and Mobile Receivers. In: *IEEE Transactions on Communications* 44 (1996), September, Nr. 9

- [MRM<sup>+</sup>10] MARTÍNEZ, L. ; ROBERT, Jörg ; MEUEL, H. ; SOBRÓN, I. ; MENDICUTE, M.: Improved Robustness for Channel Estimation without Pilots for DVB-T2, 2010. – *Full Paper* zum Vortrag am 24. März 2010 auf dem *IEEE International Symposium on Broadband Multimedia Systems and Broadcasting 2010*, Paper-Nummer mm2010-70
- [OL07] OHM, Jens-Rainer ; LÜKE, Hans D.: *Signalübertragung*. 10. Auflage. Springer-Verlag, 2007
- [Rei08a] REIMERS, Ulrich: *Bildkommunikation II – Sammlung von Arbeitsblättern*. Frühjahr 2008. – Umdruck, Stand: Frühjahr 2008
- [Rei08b] REIMERS, Ulrich: *DVB – Digitale Fernsehtechnik*. 3. Auflage. Springer-Verlag, 2008
- [RM10] ROBERT, Jörg ; MEUEL, Holger: *MIMO Channel Analysis using DVB-T Signals*. <http://edas.info>. Version: Januar 2010. – *Abstract* zur Bewerbung zum *14th International Symposium on Consumer Electronics (ISCE) 2010*, Paper 1569278229, verfasst am *Institut für Nachrichtentechnik der Technischen Universität Carolo-Wilhelmina zu Braunschweig* unter Prof. Reimers
- [Rob09] ROBERT, Jörg: Anwendung von MISO in Gleichwellennetzen. In: *ITG Fachtagung für Elektronische Medien, 13. Dortmunder Fernsehseminar, 17.-18.03.2009, Tagungsband Bd. 1, 2009*, S. 86 – 90
- [Sch03] SCHÖBER, Henrik: *Breitbandige OFDM Funkübertragung bei hohen Teilnehmergegeschwindigkeiten*. Jondral, Friedrich (Hrsg.), 2003 (Forschungsberichte aus dem Institut für Nachrichtentechnik der Universität Karlsruhe (TH)). <http://digbib.ubka.uni-karlsruhe.de/volltexte/documents/2808>
- [SFFM99] SPETH, Michael ; FECHTEL, Stefan A. ; FOCK, Gunnar ; MEYR, Heinrich: Optimum Receiver Design for Wireless Broad-Band Systems Using OFDM — Part I. In: *IEEE Transactions on Communications* 47 (1999), November, Nr. 11, S. 1668–1677. <http://dx.doi.org/10.1109/26.803501>. – DOI 10.1109/26.803501. – ISSN 0090–6778
- [SFFM01] SPETH, Michael ; FECHTEL, Stefan ; FOCK, Gunnar ; MEYR, Heinrich: Optimum Receiver Design for OFDM-Based Broadband Transmission — Part II: A Case Study. In: *IEEE Transactions on Communications* 49 (2001), April, Nr. 4
- [Xia03] XIAODONG REN, SHIDONG ZHOU, YAN YAO, ZUCHENG ZHOU: A new successive interference cancellation for asynchronous CDMA, 2003, S. 252 – 256 Vol. 1

# Abkürzungsverzeichnis

A/D	.....	Analog/Digital-Umsetzer
BER	.....	Bit Error Rate (Bitfehlerrate)
Bit	.....	Binary Digit
bzw.	.....	beziehungsweise
CD3	.....	Coded Decision Directed Demodulation
CFO	.....	Coarse Frequency Offset
COFDM	.....	Coded Orthogonal Frequency Division Multiplex
CPE	.....	Common Phase Error
CR	.....	Code Rate
D/A	.....	Digital/Analog-Umsetzer
DFT	.....	Discrete Fourier Transformation (diskrete Fouriertransformation)
DVB	.....	Digital Video Broadcasting (Digitaler Videorundfunk)
DVB-T	.....	Digital Video Broadcasting – Terrestrial (Digitaler terrestrischer Videorundfunk)
DVB-T2	.....	Digital Video Broadcasting – Terrestrial, Second Generation (Digitaler Videorundfunk der zweiten Generation)
etc.	.....	et cetera (und so weiter)
FEC	.....	Forward Error Correction (Vorwärtsfehlerschutz)
FFO	.....	Fractional Frequency Offset
FFT	.....	Fast Fourier Transformation (schnelle Fouriertransformation)
FIR	.....	Finite Impulse Response („endliche Impulsantwort“)
GB	.....	Guard Band („Schutzfrequenzband“)
GF	.....	Galoisfeld
GI	.....	Guard Interval („zeitliches Schutzintervall“)
HBF	.....	Hauptbahnhof
HDD	.....	Hard Disc Drive (Festplatte)
Hz	.....	Hertz
ICI	.....	Intercarrier interference („Nachbarträgerinterferenz“)
IEEE	.....	Institute of Electrical and Electronics Engineers
IFFT	.....	Inverse Fast Fourier Transformation
IQ-Diagramm	.....	Inphase-Quadratur-Diagramm
ISI	.....	Inter Symbol Interference (Intersymbolinterferenz)
kbit	.....	Kilobit
kHz	.....	Kilohertz
LLR	.....	Log-Likelihood Ratio („logarithmierter Wahrscheinlichkeitsquotient“)

LOS	Line of sight (Sichtverbindung)
LSB	Least Significant Bit (niedrigstwertiges Bit)
LTE	Long Term Evolution
Mbit	Megabit
MHz	Megahertz
MIMO	Multiple Input, Multiple Output
MISO	Multiple Input, Single Output
ML	Maximum-Likelihood-Decodierung
MPEG	Moving Picture Experts Group
MSB	Most Significant Bit (höchstwertiges Bit)
MSE	Mean Square Error (mittlerer quadratischer Fehler)
o. ä.	oder ähnliches
OFDM	Orthogonal Frequency Division Multiplex
PLL	Phase-Locked Loop (Phasenregelkreis)
PN	Phase Noise
PRBS	Pseudorandom binary sequence
QAM	Quadrature Amplitude Modulation (Quadratur-Amplituden-Modulation)
QEF	Quasi Error Free („nahezu fehlerfrei“)
QPSK	Quadrature Phase Shift Keying (Quadraturphasenumtastung)
RAM	Random Access Memory (Arbeitsspeicher)
RS	Reed-Solomon-Code
s	Sekunde (SI-Einheit)
SFN	Single Frequency Network (Gleichwellennetz)
SI	Système international d’unités („Internationales Einheitensystem“)
si-Funktion	Sinus cardinalis-Funktion
SISO	Single Input, Single Output
SNR	Signal-to-Noise Ratio (Signal-Rauschabstand)
Sync.	Synchronisation
TPS	Transmission Parameter Signalling („Übertragungsdatensignalisierungspilote“)
TS	Transport Stream (Transportstrom)
UHF	Ultra high frequency („Dezimeterwellenbereich“, Frequenzbereich von 0,3 GHz bis 3 GHz)
VHF	Very high frequency („Ultrakurzwellen“, Frequenzbereich von 30 MHz bis 300 MHz)
WLAN	Wireless Local Area Network

# Abbildungsverzeichnis

2.1	Prinzip eines OFDM-Modulators (aus [Rei08b]) . . . . .	3
2.2	GI Korrelation mittels eines gleitenden Fensters . . . . .	5
2.3	Mehrwegeempfang in Gleichwellennetzen und das Guard Interval (aus [Rei08a] entlehnt) . . . . .	8
2.4	Encoderseitige Signalverarbeitung eines DVB-T-Signals (nach [Rei08b])	9
2.5	Prinzipschaltbild eines Faltunginterleavers (aus [Rei08b]) . . . . .	10
2.6	Beispiel eines Faltungencoders (aus [Kü08]) . . . . .	11
2.7	Beispiel eines Trellisdiagramms (aus [Kü08]) . . . . .	12
2.8	Prinzipschaltbild des inneren DVB-T-Interleavers für 16-QAM (aus [DVB06]) . . . . .	14
2.9	Konstellationsdiagramme digitaler Modulationen (aus [DVB06]) . . .	15
	(a) QPSK . . . . .	15
	(b) 16-QAM . . . . .	15
	(c) 64-QAM . . . . .	15
2.10	DVB-T-Übertragungsrahmen (nach [Rei08b]) . . . . .	16
3.1	Digitales Übertragungssystem und Kanalmodell (frei nach [Kü08]) . .	21
3.2	Mehrwegeausbreitung (angelehnt an [OL07]) . . . . .	21
3.3	Gauß-Kanal . . . . .	22
	(a) Impulsantwort eines Gauß-Kanals . . . . .	22
	(b) Kanalübertragungsfunktion eines Gauß-Kanals . . . . .	22
3.4	Rice-Kanal . . . . .	23
	(a) Impulsantwort eines Rice-Kanals . . . . .	23
	(b) Kanalübertragungsfunktion eines Rice-Kanals . . . . .	23
3.5	Rayleigh-Kanal . . . . .	23
	(a) Impulsantwort eines Rayleigh-Kanals . . . . .	23
	(b) Kanalübertragungsfunktion eines Rayleigh-Kanals . . . . .	23
3.6	Vergleich si-Funktion und Dreiecksfunktion . . . . .	28
4.1	Die Datenstruktur des Structs <i>symb</i> . . . . .	32
4.2	Blockdiagramm des Hauptprogramms . . . . .	35
4.3	Korrelationsfunktion der gleitenden Fenster . . . . .	37
4.4	Pilotstruktur vor linearer Interpolation und interpolierte Kanalübertragungsfunktion . . . . .	40
	(a) unterabgetastete Kanalübertragungsfunktion . . . . .	40
	(b) durch <i>zero forcing</i> interpolierte Kanalübertragungsfunktion . . .	40
4.5	Aliasunterdrückung zur linearen Interpolation . . . . .	41

(a)	Impulsantwort <i>mit</i> Alias . . . . .	41
(b)	durch <i>zero forcing</i> gefilterte Impulsantwort <i>ohne</i> Alias . . . . .	41
4.6	Restträgerlöschung und Interpolation . . . . .	42
(a)	Restträger vorhanden . . . . .	42
(b)	Restträger entfernt . . . . .	42
4.7	Encoderseitige Kette DEMUX, Bit Interleaver, Symbol Interleaver und Mapping (aus [DVB06]) . . . . .	45
(a)	Encoderkette für QPSK . . . . .	45
(b)	Encoderkette für 16-QAM . . . . .	45
(c)	Encoderkette für 64-QAM . . . . .	45
4.8	Einfluss der Filterlänge: Mittels des modifizierten CD3-geschätzte Kanalübertragungsfunktion mit GI-angepasstem Tiefpassfilter und mit schmalere Tiefpassfilter zur Rauschunterdrückung . . . . .	50
(a)	Tiefpassfilterung mit GI-angepasstem Filter . . . . .	50
(b)	Tiefpassfilterung mit $1/12$ normiertem Durchlassbereich . . . . .	50
5.1	Auswirkung einer falsch gewählten Modulation . . . . .	66
(a)	Konstellationsdiagramm . . . . .	66
(b)	Übertragungsfunktion nach modifizierter CD3-Schätzung . . . . .	66
5.2	Auswirkung eines falschen Resampling-Faktors . . . . .	67
(a)	Übertragungsfunktion nach modifizierter CD3-Schätzung . . . . .	67
(b)	Impulsantwort nach modifizierter CD3-Schätzung . . . . .	67
6.1	Screenshot des GNU Radio-Modells zur Aufnahme von DVB-T-Signalen	70
6.2	Messaufbau zur Durchführung von DVB-T-Messungen mit dem USRP2 und GNU Radio . . . . .	71
6.3	Konstellationsdiagramm <i>vor</i> , Kanalübertragungsfunktion <i>nach</i> CD3-Schätzung sowie Impulsantwort <i>mit</i> und <i>ohne</i> Alias (idealer 2K-FFT-Kanal) . . . . .	73
(a)	Konstellationsdiagramm vor der Decodierung . . . . .	73
(b)	Kanalübertragungsfunktion nach CD3-Schätzung . . . . .	73
(c)	Impulsantwort mit Alias ( <i>vor</i> CD3-Schätzung) . . . . .	73
(d)	Impulsantwort nach CD3-Schätzung . . . . .	73
6.4	Verbesserung der durch modifizierte CD3-Kanalschätzung erhaltenen Übertragungsfunktion (idealer 2K-Kanal) . . . . .	74
(a)	Kanalübertragungsfunktion mit GI-angepasstem Filter . . . . .	74
(b)	Kanalübertragungsfunktion mit kurzem Tiefpassfilter . . . . .	74
6.5	Phase <i>vor</i> und <i>nach</i> Korrektur zwischen OFDM-Symbolen, Betrag der Kanalübertragungsfunktion <i>vor</i> und <i>nach</i> Zeitfilterung (Idealer 2K-Kanal) . . . . .	75
(a)	Phase <i>ohne</i> Korrektur zwischen verschiedenen OFDM-Symbolen	75
(b)	Phase <i>mit</i> Korrektur zwischen verschiedenen OFDM-Symbolen .	75
(c)	Betrag der Kanalübertragungsfunktion nach CD3-Schätzung <i>vor</i> Zeitfilterung . . . . .	75

	(d) Betrag der Kanalübertragungsfunktion nach CD3-Schätzung <i>nach</i> Zeitfilterung . . . . .	75
6.6	Einfluss der Viterbi-Decodierung auf den Betrag der Übertragungsfunktion (Freifeldmessung auf dem Forumsplatz der TU) . . . . .	77
	(a) Decodierung <i>ohne</i> Viterbi-Decodierung . . . . .	77
	(b) Decodierung <i>mit</i> Viterbi-Decodierung . . . . .	77
	(c) Betrag der Übertragungsfunktion des 40. Symbols <i>ohne</i> Viterbi-Decodierung . . . . .	77
	(d) Betrag der Übertragungsfunktion des 40. Symbols <i>mit</i> Viterbi-Decodierung . . . . .	77
	(e) Betrag der Differenz (40. Symbol) . . . . .	77
6.7	Konstellationsdiagramm <i>vor</i> , sowie Impulsantwort und Kanalübertragungsfunktion <i>nach</i> CD3-Schätzung . . . . .	79
	(a) Konstellationsdiagramm vor der Decodierung . . . . .	79
	(b) Kanalübertragungsfunktion nach CD3-Schätzung . . . . .	79
	(c) Impulsantwort nach CD3-Schätzung . . . . .	79
6.8	Betrag der Kanalübertragungsfunktionen <i>ohne</i> und <i>mit</i> Filterung in Zeitrichtung sowie Differenz (Freifeldmessung im Säulengang am Audimax der TU) . . . . .	81
	(a) <i>Ohne</i> Filterung in Zeitrichtung . . . . .	81
	(b) <i>Mit</i> zeitlicher Filterung . . . . .	81
	(c) Betrag der Differenz der Übertragungsfunktionen . . . . .	81
6.9	Einfluss der inneren Konstellationspunkte auf die Schätzung der Kanalübertragungsfunktion . . . . .	82
	(a) Mit allen reencodierten Konstellationspunkten geschätzte Kanalübertragungsfunktion . . . . .	82
	(b) Mit Konstellationspunkten mit Amplitude > 1,5 geschätzte Kanalübertragungsfunktion . . . . .	82
	(c) Mit Konstellationspunkten mit Amplitude > 3,3 geschätzte Kanalübertragungsfunktion . . . . .	82
6.10	Betrag der Kanalübertragungsfunktionen <i>ohne</i> und <i>mit</i> Zeitfilterung sowie Differenz im TU6-Kanal . . . . .	84
	(a) <i>Ohne</i> Filterung in Zeitrichtung . . . . .	84
	(b) <i>Mit</i> zeitlicher Filterung . . . . .	84
	(c) Betrag der Differenz . . . . .	84
6.11	Impulsantwort und Übertragungsfunktion bei etwa gleich starken Empfangspegeln verschiedener Sender in einem realen SFN . . . . .	85
	(a) Impulsantwort . . . . .	85
	(b) Kanalübertragungsfunktion (logarithmische Darstellung) . . . . .	85
	(c) Kanalübertragungsfunktion (lineare Darstellung) . . . . .	85
6.12	Messungen während Tramfahrt . . . . .	87
6.13	Einfluss von Windkraftanlagen auf DVB-T-Signale in einem realen SFN . . . . .	88
	(a) Impulsantwort mit Hauptpfad (logarithmische Darstellung) . . . . .	88
	(b) Impulsantwort ohne Hauptpfad (lineare Darstellung) . . . . .	88

# Codelistingverzeichnis

4.1	Auszug aus <code>detect_scattered_pilot_constell.m</code> . . . . .	38
4.2	Auszug aus <code>detect_scattered_pilot_constell.m</code> . . . . .	39
4.3	Ausschneiden und Descrambling der Scattered Pilots (Auszug aus <code>channel_est.m</code> )	39
4.4	Restträgerlöschung (Auszug aus <code>main_sync_soft.m</code> ) . . . . .	41
4.5	Beschränkung des LLR-Wertebereichs (Auszug aus <code>main_sync_soft.m</code> )	43
4.6	Kanalschätzung nach dem modifizierten CD3-Algorithmus (Auszug aus <code>main_sync_soft.m</code> ) . . . . .	46
4.7	Auszug aus <code>timing_error_correction.m</code> . . . . .	47
4.8	Kompensation des Fractional Subsampling Errors (Auszug aus <code>correct_fractional_subsample_freq_error</code> ) . . . . .	48
4.9	Filterung der Impulsantwort nach Kanalschätzung mittels des modifizierten CD3-Algorithmus (Auszug aus <code>main_sync_soft.m</code> ) . . . . .	49
4.10	Filterung der Kanalübertragungsfunktion in Zeitrichtung (Auszug aus <code>main_sync_soft</code> ) . . . . .	49
4.11	Implementierung der digitalen PLL (Auszug aus <code>main_sync_soft.m</code> )	50

# Stichwortverzeichnis

**Fett** gedruckte Seitenzahlen sind Referenzen auf die Definition eines Begriffs, während normal gedruckte Seitenangaben weitere Vorkommnisse kennzeichnen.

MATLAB-Befehle sowie -variablen sind in **Typewriter** gedruckt und alphabetisch im Stichwortverzeichnis eingeordnet.

- `./` -Operator, 33
- `:` -Operator, 47
  
- Abtasttheorem, *siehe* Shannonsches Abtasttheorem
- Abtastzeitpunkt, 48
- Alias, 25, 28, 40, 67
- `analyse_scattered_pilot_constellations`, 39
- Arbeitsspeicher, 30, 31, 51
- Ausgangsbit, 12
- Ausgangsbitfehlerrate, 10
- Ausgangsrahmenbreite, 12
  
- Büschelfehler, *siehe* Burstfehler
- Basisbandschnittstelle, 8
- Basisverzögerung, 10, 11
- BCH-Code, 17
- Beeinflussungslänge, 12
- Binärformat, 53
- `bit_deinterleaver`, 44
- `bit_demux.m`, 44
- `bit_demux_inverse`, 44
- `bit_interleaver`, 44
- Bitfehlerrate (BER), 5
- `block_size_bitdeinterl`, 58
- Blockcode, *siehe* Reed-Solomon-Blockcode
- Blockdiagramm, 34
- Blockinterleaver, 14
- Boosted Pilots, 17, *siehe* Pilots
- Broadcastdienst, 13
- Burstfehler, 9
  
- CD3, *siehe* Coded Decision Directed Demodulation
- CD3-Algorithmus, 40
- `channel_est`, 39
- `channel_transfer_function`, 33, 34
- `channel_transfer_ "function_ "reordered`, 34
- `channel_transfer_function_reordered`, 34
- `channel_width`, 58
  
- `clear_restcarrier`, 41, 58
- Coarse Frequency Correction, **5**
- Coarse Frequency Offset, 37
- Coded Decision Directed Demodulation, 1, **25**, 34, 44
- Coded Orthogonal Frequency Division Multiplex, 18
- Codeoptimierung, 51
- `CodeRate`, 58
- Codewort, 11
- COFDM, *siehe* Coded Orthogonal Frequency Division Multiplex
- Common Phase Error, 6, 48
- `complex_fft_data`, 33, 34
- `complex_fft_data_next_symb`, 33, 34
- `complex_time_data`, 33, 34
- `complex_time_data_next_symb`, 33, 34
- Continual Pilots, *siehe* Pilots
- `conv2`, 49
- `convenc`, 44
- `correct_fractional_subsample_freq_error`, 48
- `current_symb`, 33, 34
  
- `data_carrier_reordered`, 34
- Dateiausgabe, 51
- Daughterboard, 69
- Deinterleaving, 44
- `demapped_bit_ "llrs`, 34
- `demapped_bit_llrs`, 34
- Demapper, 13
- Demapping, 42
- Demultiplexing, 44
- Demuxing, *siehe* Demultiplexing
- `depuncturing`, 44
- Depunktierung, 44
- `descrambled_pilots`, 34
- Descrambling, 39
- `detect_scattered_pilot_constell`, 38, 39
- DFT, *siehe* Diskrete Fourier Transformation

Dirac-Impuls, 24  
 discard\_inner\_const\_points, 46  
 discard\_inner\_const\_points\_bool, 58  
 Diskrete Fourier Transformation, 2  
 DVB-T-Signal, 2, 7  
  
 Echo, 19, 25  
 Eingangsbit, 12  
 Eingangsbitfehlerrate, 9, 10  
 Eingangsrahmenbreite, 12  
 Einträgermodulation, *siehe* Einträgersystem  
 Einträgersystem, 4, 18  
 Empfangsdaten, 5  
 Energieverwischung, 8  
 Entscheiderschwelle, 13  
 equalised\_input\_signal, 33, 34  
  
 Faltungscodierung, 11  
 Faltungscodierung, 44  
 Fast Fourier Transformation, 2, 18, 31  
 fdatool, 64  
 fdatool, 78  
 FEC, *siehe* Vorwärtsfehlerschutz  
 Fehlerburst, 10, 11  
 Fehlerkette, *siehe* Fehlerburst  
 Fehlerquellen, 64  
 Fehlerverschleppung, 74  
 Fenster, *siehe* Gleitendes Fenster  
 FFT, *siehe* Fast Fourier Transformation  
 FFT-Länge, 5  
 FFT-Modus, *siehe* Modus  
 figures\_on\_bool, 58  
 filename\_imag, 58  
 filename\_real, 58  
 filename\_single\_input\_file, 58  
 Filter Design & Analysis Tool, *siehe* fdatool  
 filter\_cd3\_impulse\_response\_bool, 58  
 filter\_impulse\_response\_bool, 67  
 filter\_impulse\_response\_bool, 58  
 filter\_width, 67  
 filter\_width, 58  
 filter\_width\_cd3, 58  
 Filterung, 27, 39, 46  
     Nach-, 46  
 folder, 58  
 Forward Error Correction, *siehe* Vorwärtsfehlerschutz  
 Fouriertransformation, 2, 4  
 Fouriertransformierte  
     inverse, 49  
 Fractional Frequency Offset, 5, **36**  
 frequency\_sync, 38, 52  
 Frequenzbereich, 4, 36  
  
 Frequenzsynchronisation, *siehe* Synchronisation, 37  
 Frequenzversatz, 36, 37  
  
 Galoisfeld, 9  
 Gauß'scher Kanal, *siehe* Gauß-Kanal  
 Gauß'sches Rauschen, *siehe* Rauschen  
 Gauß-Kanal, 22  
 Generatorpolynom, 11  
 Generatorwort, 11  
 GI, *siehe* Guard Interval  
 Gleichwellennetz, 2, **6**, 7, 24  
 Gleitendes Fenster, 4, 36  
 Gliding window, *siehe* Gleitendes Fenster  
 global\_parameters, 31  
 GNU Radio, 69  
 Guard Band, 19  
 Guard Interval, 4, 7, **19**, 25, 40  
 GuardIntervall, 58  
  
 hamm50\_norm0\_05, 49, 58  
 Hamming-Filter, 78  
 Hard Decision, 13, 53  
 Hard Decision-Decodierung, 42  
 Hauptprogramm, 53  
 hierarchische Modulation, *siehe* Modulation  
  
 ICI, *siehe* Intercarrier interference  
 IFFT, *siehe* Inverse Fast Fourier Transformation  
  
 Implementierung, 30  
 Impulsantwort, 24  
 in\_symb\_ifft\_oversamp, 48  
 Initialisierung, 34, 51  
 input\_data\_format, 65  
 input\_data\_format, 58  
 Intercarrier interference, 6, 36, 37  
 Intercarrier-Interferenz, *siehe* Intercarrier interference  
  
 Interleaver, 10  
     äußerer, 10, 20  
     innerer, 14, 20  
 Interleavingtiefe, 11  
 Interpolation, 39, 41  
     Probleme der linearen, 27  
 Intersymbolinterferenz, 5, 7  
 Inverse Fast Fourier Transformation, 4, 18  
 IQ-Diagramm, 44, 65, 71  
 ISI, *siehe* Intersymbolinterferenz  
  
 Kanal, **20**  
 Kanalübertragungsfunktion, 27, 39–42, 46  
 Kanalmodell, 1, **21**

Kanalschätzung, 20, 24, 39, 44  
 Konstellationspunkt, 13, 14  
 Korrelation, 4  
  
 Least Significant Bit, 14  
 Leistungsdichte, 22  
 Leistungsdichtespektrum, 8  
 Leistungsfähigkeit, 6  
 LLR, *siehe* Log-Likelihood Ratio  
 LLRqamDemap16, 43, 52  
 LLRqamDemap64, 43, 52  
 LLRqamDemapQPSK, 43, 52  
 Log-Likelihood Ratio, **13**, 34, 43  
  
 main\_sync\_soft, 30, 49, 53  
 MATLAB-Funktion, 30  
 MATLAB-Skript, 30  
 max\_llr, 58  
 Maximum Likelihood-Decodierung, 43  
 Maximum-Likelihood-Decodierung, 12  
 Mehrfachträger-Modulation, 2  
 Mehrwegeempfang, 7  
 Messaufbau, 69  
 Messungen, 69  
 min\_amplitude\_for\_const\_points, 58  
 MISO, *siehe* Multiple Input, Single Output  
 Mittelwertbildung, 5  
 modem.demodulate, 43  
 Modulation  
     hierarchische, 17  
     OFDM, *siehe* Orthogonal Frequency Division Multiplex  
     QAM, *siehe* Quadratur-Amplitudenmodulation  
     QPSK, *siehe* Quadraturphasenumtastung  
 Modulation, 58  
 Modus, 30  
     2K-FFT-, **18**, 29, 31, 37, 53  
     8K-FFT-, **2**, **18**, 29, 31, 37, 53  
 Most Significant Bit, 14  
 MPEG-2 Transportstrom, 8  
 Multiple Input, Single Output, 1, 28  
  
 Nachbarkanäle, 8  
 Nachbarkanalinterferenz, 19  
 Nachrichtenwort, 11  
 Nichtlineare Verzerrungen, *siehe* Verzerrungen  
 No\_of\_ofdm\_symbols\_to\_compute, 58  
 no\_of\_ofdm\_symbols\_to\_use\_for\_sync, 58  
 No\_of\_values\_to\_discard, 58  
 Normalisierung, 44  
 Nutzdatenträger, 18  
  
 OFDM, *siehe* Orthogonal Frequency Division Multiplex  
     OFDM-Subträger, 4–6, 14, 16, 18, **18**, 24, 33, 36, 38, 42, 48  
     OFDM-Symbol, 4, 5, 18, 33, 36–39, 46, 47, 51  
     OFDM-System, 3, 4  
     OFDM-Trägerfrequenz, *siehe* Trägerfrequenz  
     Orthogonal Frequency Division Multiplex, 2, **2**, 3, 7, 24  
     Orthogonalität, 4, **6**, 7  
     output\_file\_imag, 58  
     output\_file\_real, 58  
     Oversampling, 48  
     oversampling\_factor, 50, 58  
  
 Parametereinstellungen, *siehe* Programmparameter  
     params (Datenstruktur), 31  
     params.active\_carriers, 50  
     textttparams.active\_carriers, 41  
     params.k, 38, 46  
     params.max\_llr, 43  
     params.pll\_adjust, 50  
     params.scrambling\_sequence, 39  
     Permutation, 14  
     Phase Noise, *siehe* Phasenrauschen  
     Phase-Locked Loop, *siehe* Phasenregelkreis  
     Phasenfehler, 5, *siehe* Common Phase Error  
     Phasenrauschen, *siehe* Rauschen  
     Phasenrauschspektrum, 6  
     Phasenregelkreis, **19**, 33, 50  
     Phasensprung, 56  
     pilot\_pos, 38  
     Pilote, *siehe* Pilots  
     „Pilote erhöhter Amplitude“, *siehe* Boosted Pilots  
     Pilots  
         Boosted, **18**  
         Continual, **17**, 29, 37, 38  
         Scattered, **17**, 29, 33, 38, 39  
         TPS, **17**, 33, 38, 40  
     Pilotstruktur, **16**, 30  
     PLL, *siehe* Phasenregelkreis  
     PLL\_Multiplier, 58, 66  
     Pointer, *siehe* Zeiger  
     postprocessing\_bool, 48, 58  
     Preecho, 24  
     Programmablauf, 34  
     Programmabschluss, 51  
     Programmparameter, 53, 58  
     Programstart, 34  
     Pseudozufallssequenz, *siehe* Scrambling  
     pulse\_resp\_max\_pos, 50  
     Punktierung, 12, 74

QAM, *siehe* Quadratur-Amplitudenmodulation  
 QPSK, *siehe* Quadraturphasenumtastung  
 Quadratur-Amplitudenmodulation, 14, 14, 20, 42  
 Quadraturphasenumtastung, 14, 20, 42  
 quantizer\_bits, 58  
 Quasi Error Free Reception, 10  
  
 RAM, *siehe* Arbeitsspeicher  
 Rauschen  
     Gauß'sches, 22  
     Phasen-, 6  
     Phasenfehler, *siehe* Common Phase Error  
 Rayleigh-Kanal, 23  
 read\_sequential\_int, 58  
 Reed-Solomon-Blockcode, 9  
 reencode, 44  
 restcarrier\_value, 41  
 Restträger, 28, 41  
 Restträgerlöschung, 28, 41  
 Rice-Kanal, 22  
 Rice-Verteilung, *siehe* Verteilungsdichtefunktion  
  
 Sample, 33, 34, 36, 40  
 Samplingzeit, 5  
 Scattered Pilots, *siehe* Pilots  
 Scattered Pilots-Konfiguration, 38  
 Schätzung einer Kanalübertragungsfunktion, *siehe* Kanalschätzung  
 Scrambling, 8, 14, 42  
 SFN, *siehe* Gleichwellennetz  
 Shannonsches Abtasttheorem, 25  
*Signal Processing*-Toolbox, 64  
 Signal-Rauschabstand, 7, 13, 89  
 Single Frequency Networks, *siehe* Gleichwellennetz  
 Single Input, Single Output, 28  
 Singlecarrier-System, *siehe* Einträgersystem  
 Sinusschwingung, 2  
 SISO, *siehe* Single Input, Single Output  
 SNR\_äst, 67  
 SNR\_est, 58  
 Soft Decision, 13  
 Soft Decision-Decodierung, 55  
 Soft-Bit, 13, 43  
 Software Radio, 69  
 Speichertiefe, 12  
 Störgröße, 6  
 Störung, 6, 7, 21  
 „Ständige Pilote“, *siehe* Continual Pilots  
 Stationarität, 38  
 stream\_index, 33, 34  
  
 Structure Array, 31  
 Subcarrier, *siehe* OFDM-Subträger  
 Subträger, *siehe* OFDM-Subträger  
 SW\_radio\_sample\_freq, 58  
 symb, 40  
 symb (Datenstruktur), 31  
 symb.channel\_transfer\_function, 39, 41  
 symb.current\_symb, 38, 39  
 symb.demapped\_bit\_llrs, 43  
 symb.descrambled\_pilots, 39  
 symb.reencoded\_data, 46  
 symb.time\_mean\_symb, 49  
 symb.tps\_pilots, 40  
 symb.transf\_func\_phasecorrected, 49  
 symb.transfer\_func, 46  
 symb.useful\_carriers, 38, 39, 46  
 symb\_curr\_corr, 47  
 symbol\_deinterleaver, 44  
 symbol\_interleaver, 44  
 Symboldeinterleaver, 16  
 Symboldeinterleaving, 41  
 Symbolinterleaver, 10, 14  
 Symbolmapping, 14  
 Sync. Invertierung, 8  
 Synchronisation, 4, 7  
     fehlerhafte, 3  
     Frequenz-, 5  
     Zeit-, 4  
 Syndrom, 9  
  
 T\_N, 58  
 Terminierung, 13  
 Tiefpass, 25  
 Tiefpassfilterung, 25  
 time\_direction\_filtering, 58  
 time\_mean\_symb, 34  
 time\_sync\_error\_allowed, 68  
 time\_sync\_error\_allowed, 58  
 timing\_error\_correction, 47  
 TPS Pilots, *siehe* Pilots  
 tps\_data, 34, 40  
 tps\_pilots, 33, 34  
 tps\_pilots\_demod, 33, 34  
 Trägerfrequenz, 6  
 Trainingssymbol, *siehe* Pilots  
 transf\_func\_phasecorrected, 34  
 transfer\_func, 34  
 Trellisdiagramm, 12  
 trial and error, *siehe* Versuch und Irrtum  
 TU6-Kanal, 80  
  
 Überrahmen, 16

Übertragungsdatensignalisierungspilote, *siehe*  
    TPS Pilots  
Übertragungsfunktion, 24  
Übertragungsrahmen, 16  
UHF, 69  
Universal Software Radio Peripheral, 69  
Universal Software Radio Peripheral 2, 69  
Unterabtastung, 41  
Useful carrier, *siehe* Nutzdatenträger  
`useful_carriers`, 33, 34  
  
„Verstreute Pilote“, *siehe* Scattered Pilots  
Versuch und Irrtum, 37  
Verteilungsdichtefunktion  
    Rice-Verteilung, 22  
Verwürfelung, *siehe* Scrambling  
Verzerrungen  
    nichtlineare, 6  
VHF, 69  
`vitdec`, 44, 57  
Viterbi-Decoder, 13  
Viterbi-Decodierung, **12**, 43, 44, 57, 68, 74, 83  
`viterbi_de_and_reencoding`, 44  
`viterbi_decoding_bool`, 58  
Vorwärtsfehlerschutz  
    äußerer, 9, 20  
    innerer, 11, 20  
  
Wahrscheinlichkeit, 13  
`write_sequential_bool`, 58  
  
XOR-Verknüpfung, 11  
  
Zeiger, 31  
Zeitbereich, 4  
Zeitsynchronisation, *siehe* Synchronisation  
zero forcing, **27**, 39, 49, 57, 62

# Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Diplomarbeit eigenständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel angefertigt zu haben. Alle öffentlichen Quellen sind als solche kenntlich gemacht. Die vorliegende Arbeit ist in dieser oder anderer Form zuvor nicht als Prüfungsarbeit zur Begutachtung vorgelegt worden.

Braunschweig, den 29. März 2010

*Holger Meuel*