



BEUTH HOCHSCHULE FÜR TECHNIK BERLIN
University of Applied Sciences

Simulation eines DRM-Übertragungssystems mit Simulink

Masterarbeit

SS 2010

18. September 2010

Autor:

Kevin Döring

Matr.-Nr.:

760995

Fachbereich:

VI - Informatik und Medien

Studiengang:

Master Embedded Systems

Betreuer:

Beuth Hochschule für Technik Berlin
Fachbereich VII

Prof. Dr.-Ing. Marcus Purat

Prüfer:

Erstprüfer:
Zweitprüfer:

Prof. Dr.-Ing. Marcus Purat
Prof. Dipl.-Ing. Manfred Ottens

Danksagung

An dieser Stelle möchte ich allen danken, die mich im Laufe meines Studiums und bei der Anfertigung dieser Masterarbeit unterstützt haben. Ein besonderer Dank geht an meinen Betreuer Herrn Prof. Dr.-Ing. Marcus Purat für den interessanten Themenvorschlag, sowie für seine kompetente Unterstützung und stetige Motivation bei der Anfertigung dieser Arbeit.

Abstract

Die vorliegende Masterarbeit beschäftigt sich mit dem Thema DRM (Digital Radio Mondiale). Es wird eine vollständige Simulation eines DRM-Übertragungssystems mit dem Tool MATLAB/Simulink nachgebildet, welches alle Komponenten der Kanalkodierung, der Modulation und der Zwischenfrequenzverarbeitung beinhaltet. Zudem beschreibt diese Arbeit den grundlegenden Aufbau und die Algorithmen dieses digitalen Rundfunksystems basierend auf OFDM (Orthogonal Frequency Division Multiplex).

Durch die Implementierung eines Kanalmodells zwischen Sender- und Empfängerzweig soll die Notwendigkeit von Algorithmen zur Synchronisation und Kanalkompensation gezeigt werden und eine zusätzliche Umsetzung dieser im Simulinkmodell soll als mögliche Basis für einen eigenständigen DRM-Empfänger auf einer Zielplattform dienen.

The present Master's Thesis deals with the topic DRM (Digital Radio Mondiale). A complete simulation of a DRM transmission system is modeled with the tool MATLAB / Simulink, which includes all components of channel coding, modulation and intermediate frequency processing. Moreover, this work describes the basic structure and the algorithms of this digital broadcasting system based on OFDM (Orthogonal Frequency Division Multiplex).

By implementing a channel model between the transmitter and receiver branch the need of algorithms for synchronization and channel compensation are shown. An additional implementation of this components in the Simulink model serve as a possible basis for a standalone DRM receiver on a target platform.

Symbole

B_s Spektralbelegung (spectrum occupancy)

$H_c(s,k)$ diskrete zeitvariante Kanalübertragungsfunktion

K Anzahl der Träger innerhalb eines Symbols

K_{max} maximale Trägernummer

K_{min} minimale Trägernummer

N_g Anzahl der Abtastwerte im Schutzintervall

N_s Anzahl der Abtastwerte eines gesamten OFDM-Symbols ($N_u + N_g$)

N_u Anzahl der Abtastwerte im Nutzanteil eines OFDM-Symbols

T_g Dauer des Nutzanteils eines OFDM-Symbols

T_s Dauer eines gesamten OFDM-Symbols ($T_u + T_g$)

T_u Dauer des Schutzintervalls

Δf_k Subträgerabstand in einem OFDM-Symbol

f_{frac} partieller Anteil des Frequenzoffsets in Hz

f_{int} ganzzahliger Anteil des Frequenzoffsets in Hz

f_{off} gesamter Frequenzoffsets in Hz ($f_{frac} + f_{int}$)

$h_a(t)$ zeitkontinuierliche Kanalimpulsantwort

$h_c(n)$ zeitdiskrete Kanalimpulsantwort

Abkürzungen

AFS	Alternativ Frequency Switching
AM	Amplitudenmodulation
BER	Bit Error Ratio
COFDM	Coded Orthogonal Frequency Division Multiplex
CRC	Cyclic Redundancy Check
CTF	Channel Transfer Funktion
DAB	Digital Audio Broadcasting
DRM	Digital Radio Mondiale
EEP	Equal Error Protection
EVM	Error Vector Magnitude
FAC	Fast Access Channel
FDM	Frequency Division Multiplex
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FM	Frequenzmodulation
HMMix	Mixed Hierarchical Mapping
HMsym	Symmetrical Hierarchical Mapping
ICI	Intercarrier-Interferenz
IFFT	Inverse Fast Fourier Transform
IQ	Inphase/Quadrature
ISI	Intersymbol-Interferenz
KW	Kurzwelle
LW	Langwelle
MER	Modulation Error Ratio
MLC	Multilevel-Coding
MSC	Main Service Channel
MW	Mittelwelle
OFDM	Orthogonal Frequency Division Multiplex
PRBS	Pseudo Random Binary Sequence
PRNG	Pseudo Random Noise Generator
QAM	Quadrature Amplitude Modulation
SDC	Service Description Channel
SDR	Software Defined Radio

SM	Standard Mapping
SNR	Signal-to-Noise-Ratio
UEP	Unequal Error Protection
UKW	Ultrakurzwelle
ZF	Zwischenfrequenz

Inhaltsverzeichnis

Symbole	IV
Abkürzungen	V
1. Einleitung und Aufgabenstellung	1
2. Grundlagen und DRM-Systemübersicht	4
2.1. Begriffe und Grundlagen der Rundfunkübertragung	4
2.1.1. Wellenausbreitung	4
2.1.2. Frequenzbereiche des Hörfunks	5
2.1.3. Grundlagen der digitalen Nachrichtenübertragung	6
2.1.4. Der Übertragungskanal	7
2.2. DRM Systemübersicht	13
2.2.1. Senderarchitektur und Multiplexdefinition	14
2.2.2. Übertragungsmodi und Bandbreiten	16
2.2.3. Übertragungsrahmen	17
3. Algorithmen des DRM Standards	19
3.1. Kanalkodierung und Modulation	19
3.1.1. Energy dispersal	19
3.1.2. Kanalkodierung	21
3.1.3. QAM-Mapping	27
3.1.4. Cell-Interleaving	29
3.2. OFDM	32
3.2.1. Orthogonalität und das Prinzip von OFDM	32
3.2.2. Eigenschaften von OFDM	35
3.2.3. Guardintervall	37
3.2.4. OFDM-Struktur bei DRM	38
3.3. Pilotstruktur	41
3.4. Zwischenfrequenz-Verarbeitung	44
3.4.1. Über-/Unterabtastung	45
3.4.2. IQ-Modulator/Demodulator	46
4. Empfängeralgorithmen zur Synchronisation und Kanalkompensation	48
4.1. Zeitsynchronisation	49

4.1.1. Symbolsynchronisation	49
4.1.2. Framesynchronisation	52
4.2. Frequenzsynchronisation	54
4.3. Kanalschätzung und -kompensation	57
5. Umsetzung der Simulation	60
5.1. Die Entwicklungsumgebung	60
5.2. Initialisierung der Simulation	62
5.3. Sendermodell	64
5.3.1. Datengenerierung und CRC	64
5.3.2. Kanalkodierung und Modulation	66
5.3.3. OFDM	72
5.3.4. Zwischenfrequenz-Modulation	76
5.4. Verifizierung des Sendermodells	80
5.5. Empfängermodell	83
5.5.1. Zwischenfrequenz-Demodulation	83
5.5.2. OFDM	85
5.5.3. Demodulation und Kanaldekodierung	87
5.5.4. Datenempfang und Auswertung	89
5.6. Kanalmodell	94
5.6.1. Frequenz- und Phasenverschiebungen	95
5.6.2. AWGN-Kanal	95
5.6.3. Übertragungslaufzeiten und zusätzliche Verzögerungen	95
5.6.4. Mehrwegeszenario	96
5.7. Algorithmen zur Synchronisierung und Kanalschätzung	97
5.7.1. Symbolsynchronisation	97
5.7.2. Frequenzsynchronisation	99
5.7.3. Framesynchronisation	101
5.7.4. Kanalschätzung und -ausgleich	102
6. Zusammenfassung und Ausblick	105
Literaturverzeichnis	107
Abbildungsverzeichnis	110
Tabellenverzeichnis	113
A. Anhang	114
A.1. Ergänzende Tabellen und Abbildungen des DRM Standards	114
A.1.1. Arten der Multilevelkodierung	114
A.1.2. Schutzlevel und Coderaten	116

A.1.3. QAM Konstellationsdiagramme	118
A.2. Initialisierung	119
A.2.1. Dateien zur Initialisierung	119
A.2.2. Variablen der Initialisierung	121
B. Inhalt des Datenträgers	122

1. Einleitung und Aufgabenstellung

Seit der Ausstrahlung des ersten kommerziellen Radioprogramms 1920 in den USA machte der Hörrundfunk eine ständige Entwicklung mit, die durch den technologischen Fortschritt der vergangenen 90 Jahre im Bereich der Elektrotechnik geprägt war. Neben den Erkenntnissen über die Eigenschaften der verschiedenen Frequenzbereiche und die Nutzung der Frequenzmodulation als effizienteres und weniger störanfälliges Übertragungsverfahren wurde vor allem durch den Einsatz von Transistoren eine weltweite Verbreitung von kompakten und transportablen Radioempfängern ermöglicht. Die zunehmende Digitalisierung vieler technischer Verarbeitungsprozesse und die immer komplexer werdenden integrierten Schaltungen ließen Radioempfänger in kleinster Bauweise zu, so dass man diese heute z.B. als Bestandteil von Mobiltelefonen kennt.

Spricht man von der Digitalisierung des Hörfunks, so ist neben der Signalverarbeitung in digitaler Form auch der Einsatz digitaler Modulationsarten als robustere und besser anpassbare Übertragungsverfahren gemeint. Mit der Einführung des Digital Radios (DAB, engl.: Digital Audio Broadcasting) wollte man in Europa eine flächendeckende Digitalisierung des Hörrundfunks im UKW-Bereich schaffen und somit eine Abschaltung der analogen Verfahren bis 2012 ermöglichen. Durch geringe Bekanntmachung, unzureichendes Programmangebot aber auch das populärer werdende Internetradio ist bis heute kein Durchbruch des Digitalen Radios zu verzeichnen.

Das Digital Radio Mondiale (weltweites Digitalradio) ist ein digitales Hörfunksystem, welches in den Bereichen LW, MW und KW die bestehenden analogen Sendeverfahren mit Frequenzen unter 30 MHz ablösen soll. Entgegen DAB sollen damit die unpopulärer werdenden AM-Frequenzbänder schrittweise ersetzt werden, in dem zunächst ein paralleler Betrieb von Hörfunkprogrammen in analoger und digitaler Form stattfindet. Die Grundsteinlegung für das DRM-System war 1998 in China, wo ein Konsortium aus mehreren Sendebetreibern, Systemtechnik-Herstellern und Codec-Softwarefirmen sich die Digitalisierung der AM-Frequenzbereiche als Ziel setzte. Eine Erweiterung des Standards auf Frequenzen über 30 MHz wird unter dem Namen DRM+ gehandelt, welches auch den bestehenden UKW-Hörfunk (Frequenzen bis 174 MHz) umfassen soll und damit einen direkten Konkurrenten zu DAB darstellt.

Es ist in Zukunft von keinem festen Standard beim digitalen Radio auszugehen wie er beim analogen Hörrundfunk zu finden war. Wahrscheinlich werden in verschiedenen Regionen auch unterschiedliche Systeme zur Rundfunkübertragung genutzt. DRM als digitale Alternative zu den AM-Frequenzbereichen stellt durch den hohen Qualitätsgewinn jedoch ein globales Radiosystem dar, welches möglicherweise neben der Nutzung im Ama-

teurfunkbereich auch an Popularität für viele Hörfunkanbieter gewinnen wird.

Aufgabenstellung

Die vorliegende Masterarbeit soll ein gesamtes DRM-Übertragungssystem mit Hilfe des Simulationstools Simulink nachbilden. Die Simulation soll dabei nach der Quellenkodierung ansetzen und alle auf einem PC oder einer Zielplattform umsetzbaren Algorithmen der DRM-Systemspezifikation [Eur09] möglichst vollständig und bitgenau beinhalten. Komponenten die MATLAB und Simulink bereits als vordefinierte Blöcke in ihren Blocksets und Toolboxen bereithalten, sollen wenn möglich genutzt und dem Standard entsprechend initialisiert werden. Alle für den Empfang notwendigen Systemkomponenten müssen für den standardgetreuen Sender so angepasst werden, dass mit Hilfe der Simulation eine fehlerfreie Nachbildung des Systems ohne Übertragungskanal ermöglicht wird. Zusätzlich sollen aber auch Komponenten zur Synchronisation und Kanalkompensation im Empfänger vorhanden sein, so dass ebenfalls die Nachbildung eines Übertragungssystems mit einem selbst definierten Kanal, einen Empfang mit einer möglichst geringen Bitfehlerrate bietet.

Die Hauptziele der Masterarbeit lauten dabei wie folgt:

- Sie soll einen Überblick über das neuartige Themengebiet des digitalen Hörfunks bieten und anhand von DRM das Prinzip des SDR darstellen.
- Die Simulation kann Verwendung in verschiedenen Lehrveranstaltungen finden, da eine schrittweise Erklärung der einzelnen Komponenten des gesamten DRM-Übertragungssystems ermöglicht wird und damit auch auf das Prinzip weiterer digitaler Übertragungssysteme (wie z.B. DVB, DAB, WLAN) geschlossen werden kann.
- Durch die umfangreiche Variationsmöglichkeit bei der Initialisierung können diverse Algorithmen zur Kanalkodierung, Kanalschätzung und Empfangssynchronisation überprüft und optimiert werden.
- Der Aufbau der Simulation mit MATLAB/Simulink und die Verwendung möglichst vieler interner Blöcke und Funktionen dieser Tools soll eine spätere Umsetzung einzelner Komponenten auf einer Zielplattform mit Hilfe der Codegenerierung des Real-Time-Workshops ermöglichen.

Optimierungen und Untersuchungen zu den Algorithmen des DRM-Standards im Empfänger, sowie zu Synchronisations- und Kanalschätzmethoden sollen nicht Inhalt dieser Arbeit sein. Zudem soll die Arbeit kein eigenständiges Empfangssystem für DRM-Signale darstellen, sondern lediglich für die im Sender umgesetzten Komponenten einen simulativen Empfangsweg bieten.

Gliederung der Arbeit

Die Arbeit ist neben der Einleitung in 5 Kapitel gegliedert. Kapitel 2 soll dem Leser eine kurze Einführung in die digitale Nachrichtentechnik geben und grundlegende Begriffe dieses Themengebietes klären. Zudem wird ein Überblick über den DRM-Systemstandard gegeben.

Die in dem Standard verwendeten Algorithmen und Verfahren zur Umsetzung der Übertragung werden in Kapitel 3 beschrieben. Neben den definierten Sendeverfahren werden zusätzlich auch die im Empfänger für die Simulation verwendeten Algorithmen allgemein erläutert und für DRM spezifiziert.

Kapitel 4 umfasst alle nicht im Standard festgelegten Verfahren zur Synchronisation des Empfängers, sowie zur Kanalschätzung und -kompensation. Hierbei soll nur ein Überblick über die Notwendigkeit dieser Verfahren und die möglichen Algorithmen aufgezeigt werden.

Die Simulation selbst wird im Kapitel 5 beschrieben. Neben der Initialisierung des Modells, der Entwicklung des Senders und des Empfängers wird mit der Verifizierung des Sendermodells der Nachweis über eine Standard gerechte Umsetzung gegeben. Zudem wird das zur Untersuchung benötigte Kanalmodell beschrieben, sowie eine Spezifizierung der in Kapitel 4 beschriebenen Algorithmen zur Synchronisation, wie sie in der Simulation umgesetzt wurden, vorgenommen.

Im letzten Kapitel erfolgt die Zusammenfassung und es werden mögliche Ausblicke für eine Erweiterung bzw. Optimierung des Modells gegeben.

2. Grundlagen und DRM-Systemübersicht

In diesem Kapitel sollen einige Grundlagen und Begriffe der Nachrichtenübertragung erläutert werden, da sie in den folgenden Kapiteln Verwendung finden. Zudem wird ein erster Überblick über den DRM Systemstandard gegeben, wobei auch hier zunächst eine Klärung von Grundlagen und Begriffe erfolgt, so dass diese bei der vertiefenden Beschreibung als bekannt vorausgesetzt werden können.

2.1. Begriffe und Grundlagen der Rundfunkübertragung

2.1.1. Wellenausbreitung

Die Ausbreitung der elektromagnetischen Wellen bei nicht-leitungsgebundenen Übertragungen weisen unterschiedliche physikalische Effekte auf, die vor allem durch die Frequenz bzw. die Wellenlänge bestimmt werden. Grundsätzlich wird die Wellenausbreitung bei der Funkübertragung in drei Kategorien von Wellenausbreitungstypen eingeteilt, welche in der Tabelle 2.1 dargestellt sind.

Wellentyp	Frequenzbereich	Reichweite
Bodenwellen	3 kHz - 3 MHz	< 1000 km
Raumwellen	0,3 MHz - 30 MHz	> 1000 km
Sichtwellen (direkte Welle)	30 MHz - 30 GHz	< 150 km

Tabelle 2.1.: Kategorien der Wellenausbreitung

Die Bodenwelle breitet sich quasi parallel zum Erdboden aus, indem sie aufgrund von Druck- und Temperaturschwankungen innerhalb der Troposphäre reflektiert wird. Die Reichweite von Bodenwellen hängt sehr stark von der Beschaffenheit des Erdbodens ab, von welchem sie Absorptionen erfährt. So sind Gebiete mit hohem Wasseranteil (Sumpfgebiete, Seen, Meeresoberfläche) besser zur Übertragung geeignet, da hier weniger Dämpfungen aber mehr Reflexionen an der Oberfläche stattfinden.

Von Raumwellen hingegen spricht man, wenn die abgestrahlte elektromagnetische Welle an den höheren Luftsichten (Ionosphäre) und eventuell anschließend nochmal an der Erdoberfläche reflektiert wird. Durch mehrmalige Reflexionen können sehr hohe Reichweiten erzielt werden, so dass im optimalen Fall eine weltweite Übertragung ermöglicht wird.

Dadurch spielt vor allem die Ausbreitung über die Raumwelle eine große Rolle für den Hör- und Amateurfunk. [Wit10]

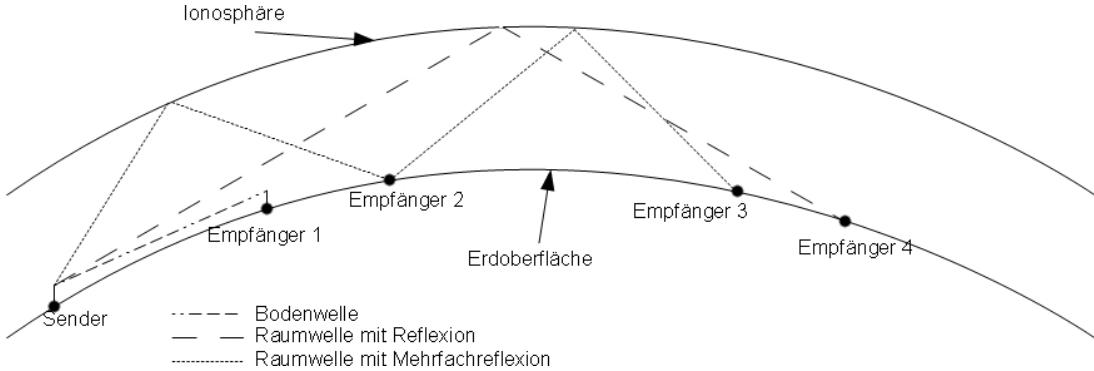


Abbildung 2.1.: Wellenausbreitung von Radiowellen

Bei sehr hohen Frequenzen > 30 MHz kann keine Übertragung über die zuvor beschriebenen Wellentypen stattfinden, da diese zu stark gedämpft werden. Man spricht in diesem Fall von Sichtwellen oder direkten Wellen, da die Antennen von Sender und Empfänger quasi in Sichtweite sein müssen.

2.1.2. Frequenzbereiche des Hörfunks

Der klassische Hörrundfunk stellt einen Teil der gesamten Rundfunkübertragung dar und wird in Deutschland in die in Tabelle 2.2 dargestellten Frequenzbänder bzw. Frequenzbereiche eingeteilt. Die Bereiche LW, MW und KW benutzen die Amplitudenmodulation (AM) und die UKW-Übertragung macht sich die Frequenzmodulation (FM) zu nutze. Der Gesamtbereich der Hörfunkübertragung reicht dabei von 150 kHz bis 108 MHz, so dass alle im vorherigen Abschnitt beschriebenen Wellentypen auftreten. [Kam04]

Frequenzbereich	Bezeichnung	Wellenausbreitungstyp
150 ... 285 kHz	Langwelle (LW)	Bodenwellen
525 ... 1605 kHz	Mittelwelle (MW)	Boden- und Raumwellen
6 ... 26 MHz	Kurzwelle (KW)	Raumwellen
87 ... 108 MHz	Ultrakurzwelle (UKW)	Sichtwellen

Tabelle 2.2.: Genutzte Frequenzbänder für den Hörrundfunk in Deutschland

Wie in der Einleitung angegeben wird bei DRM das Ziel verfolgt den klassischen Hörrundfunk zu digitalisieren. Der DRM-Standard definiert hierzu die Nutzung jeglicher Frequenzbereiche bis 174 MHz, wobei vor allem die im analogen Hörfunk genutzten Frequenzbänder als Übertragungsbereich dienen sollen. Darüber hinaus gibt es weltweit eine

Vielzahl an weiteren Frequenzbändern, die ebenfalls in das Einsatzgebiet von DRM fallen. Es wird ersichtlich, dass ein gemeinsamer Standard für die Nutzung des gesamten Wellenbereichs von LW bis UKW besondere Anforderungen an das Übertragungssystem stellt, da die Übertragungseigenschaften durch die verschiedenen Wellenausbreitungsarten in dieser großen Frequenzspanne stark variieren.

2.1.3. Grundlagen der digitalen Nachrichtenübertragung

Ein digitales Übertragungssystem wird über die in der Abbildung 2.2 dargestellten Komponenten gekennzeichnet. [Pro01]

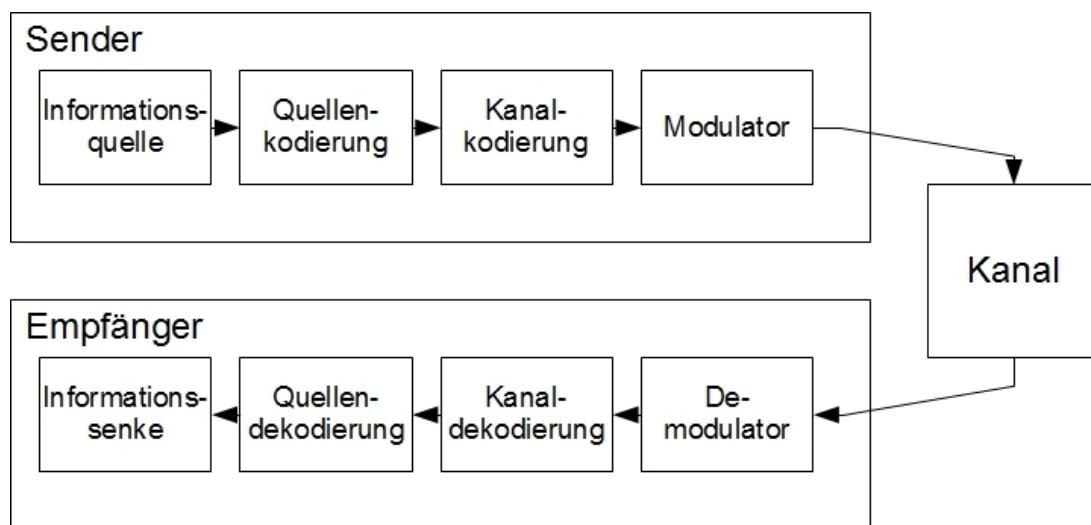


Abbildung 2.2.: Grundelemente eines digitalen Übertragungssystems

Im folgenden sollen die Funktionen der einzelnen Komponenten erläutert werden, wobei die Blöcke des Empfängers einfach als Umkehrung bzw. Wiederherstellung der im Sender verwendeten Algorithmen betrachtet werden können. Der Kanal wird aufgrund seiner Komplexität und seiner hohen Bedeutung bei der Betrachtung eines Übertragungssystems im Abschnitt 2.1.4 gesondert behandelt.

Quellenkodierung

Die Quellenkodierung dient der Konvertierung und Komprimierung der Daten der Informationsquelle. Die in analoger Form (Audio, Video) oder digitaler Form (Textnachrichten) vorliegenden Quelldaten werden mittels spezieller Kodierungsverfahren in Binärsequenzen umgewandelt und so weit komprimiert, wie es die erforderliche Qualität zulässt. Vornehmlich werden die binären Daten hier schon in eine Rahmenstruktur gepackt, was eine Ver-

arbeitung über die folgenden Komponenten des digitalen Übertragungssystems aufgrund fester Zeitintervalle vereinfacht.

Kanalkodierung

Durch Rauschen und weitere Effekte können die übertragenen Daten bitweise gestört werden, was ohne zusätzlichen Fehlerschutz zu Informationsverlust führen kann. Um zusätzliche Redundanz zu erhalten und somit mögliche Fehler nach der Übertragung zu erkennen und zu beheben, werden Verfahren zur Kanalkodierung eingesetzt. Diese führen durch bestimmte Algorithmen zu einer bekannten Verwürfelung und zu einer Vervielfachung der Daten, so dass zum einen Störungen der Übertragung weniger Einfluss haben und zum anderen Fehler im Empfänger korrigiert werden können.

Modulator

Der Modulator kann als Schnittstelle zwischen der binären Datenverarbeitung und dem Übertragungskanal betrachtet werden. Er konvertiert die binären Informationen in elektrische Signale, welche dann über Leitungen oder die Luft übertragen werden können. Im einfachsten Fall benutzt man eine Binärmودulation, bei der jedes Bit je nach Wert 1 oder 0 einem Signal $s_1(t)$ oder $s_2(t)$ zugewiesen wird. Andere Verfahren nutzen m Bits, um diese auf ein bestimmtes Signal zu legen. Hierdurch erreicht man eine höhere Datenrate bei der Übertragung.

Bitfehlerverhältnis

Bei der digitalen Nachrichtenübertragung dient das Bitfehlerverhältnis (BER, engl.: Bit Error Ratio) als ein Kriterium der Sicherheit bzw. Qualität des Übertragungssystems. Das BER gibt an wie viele Bits im Verhältnis zu der Gesamtzahl an übertragenen Bits falsch waren.

$$BER = \frac{\text{Fehlerhafte Bits}}{\text{Gesamtzahl der übertragenen Bits}} \quad (2.1)$$

Zur Ermittlung des BER müssen die binären Datenströme im Sender und Empfänger bekannt sein.

2.1.4. Der Übertragungskanal

Der Begriff des Kanals ist in der Übertragungstechnik mehrdeutig. So kann das physikalische Medium zwischen Sende- und Empfangsanenne gemeint sein oder die gesamte Übertragungsstrecke zwischen den Kanalkodierern im Sender oder Empfänger. Zudem sind weitere Definitionen zwischen den genannten möglich.[Nus07] Die Abbildung 2.3 zeigt mögliche Kanalstrecken und ihre Bezeichnungen.

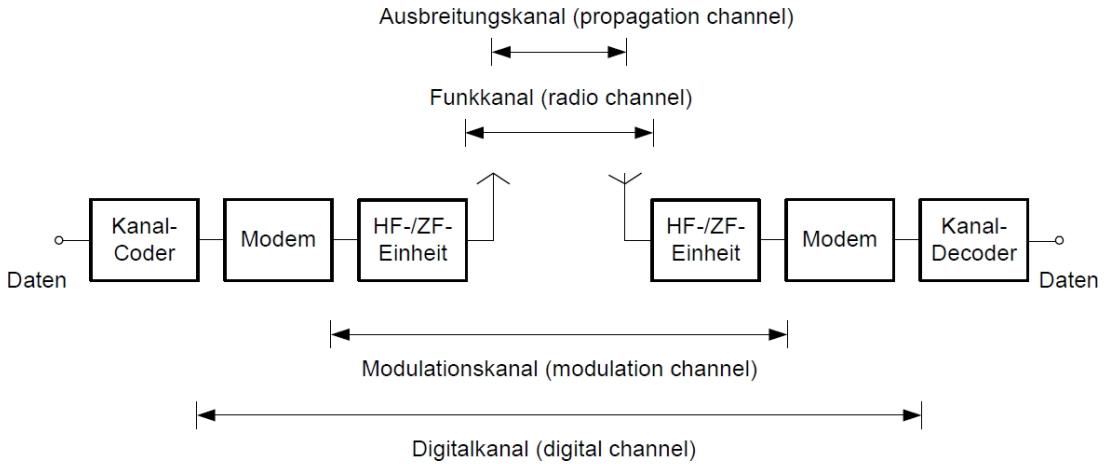


Abbildung 2.3.: Verschiedene Definitionen des Übertragungskanals [Nus07]

Für die in dieser Masterarbeit umgesetzte Simulation des DRM-Systems soll als Übertragungskanal der sogenannte Modulationskanal betrachtet werden. Im Fall der digitalen Nachrichtenübertragung beschreibt er ein System, das zwischen dem digitalen Modulator und Demodulator liegt und sämtliche analoge Komponenten, wie Antennen, Verstärker, HF-Modulatoren, u.a. beinhaltet.

Im Folgenden sollen einige Grundbegriffe und Beschreibungen eines Übertragungskanals folgen, die für die weitere Betrachtung des DRM-Systems und der Simulation von Bedeutung sind, jedoch keine umfassende Erklärung und Modellierung eines Kanals darstellen. Ausführlichere Erläuterungen zum Thema Kanal sind in [Kam04], [Hen08] und [Nus07] zu finden.

Rauschkanal

Im einfachsten Fall kann der Kanal als Rauschkanal bzw. AWGN (Additive White Gaussian Noise)-Kanal betrachtet werden. Er stellt ein idealisiertes Modell dar, wobei keine weiteren Einflüsse auf das gesendete Signal als zusätzliches gaussverteiltes weißes Rauschen auftreten. Dieses Rauschen ist in jeder realen Übertragung vorhanden und entsteht z.B. neben den atmosphärischen Einflüssen in der Luft auch bereits in den Komponenten des Übertragungssystems als thermisches Rauschen. Die mathematische Formulierung des Rauschkanals stellt das Empfangssignal $r(t)$ als Addition des Sendesignals $s(t)$ mit dem Rauschanteil $n(t)$ dar und wird mit folgender Gleichung angegeben:

$$r(t) = s(t) + n(t) \quad (2.2)$$

Der Einfluss bzw. der Anteil des zusätzlichen Rauschens wird über das Signal-Rausch-Verhältnis angegeben.

SNR Das Signal-Rausch-Verhältnis (SNR oder S/N, engl.: Signal-to-Noise-Ratio) gibt das Leistungsverhältnis von Nutzsignal $s(t)$ zum Störsignal bzw. einer Störung $n(t)$ an, wobei in den meisten Fällen eine Betrachtung der Signale als Zufallsprozesse und damit das Verhältnis der Varianzen stattfindet. [Kam04, S.75]

$$SNR = \frac{\text{Leistung des Nutzsignals}}{\text{Leistung der Störung}} = \frac{P_s}{P_n} = \frac{\sigma_s^2}{\sigma_n^2} \quad (2.3)$$

Das SNR wird häufig in logarithmischer Form angegeben.

$$SNR_{dB} = 10 \lg (SNR) \quad (2.4)$$

Bei jedem Übertragungssystem entsteht durch Bauteiltoleranzen, Quantisierung, Temperatureinflüsse, u.a. Rauschen bereits in den Komponenten der digitalen und analogen Signalverarbeitung, so dass sich hier bereits eine Verringerung des SNR ergibt. Durch zusätzliches additives Rauschen in einem AWGN-Kanal wird das Verhältnis noch weiter reduziert.

Die Betrachtung des SNR bei digitalen Systemen zeigt die Güte und Zuverlässigkeit der eingesetzten Verfahren und Komponenten zur Dekodierung des binären Datenstroms. So wird versucht bei immer kleinerem SNR die Bitfehler und somit den Informationsverlust möglichst gering zu halten.

Mehrwegeausbreitung

Der Rauschkanal reicht als Darstellung für die Übertragung von nicht leitungsgebundenen Funksignalen nicht aus. Die Abbildung 2.4 zeigt ein typisches Szenario für die Ausbreitung von Funkwellen. Neben dem direkten Pfad¹ p1 kommt es an diversen Objekten zu Reflexionen, welche verzögert beim Empfänger auftreffen und als Pfade p2 und p3 dargestellt werden.

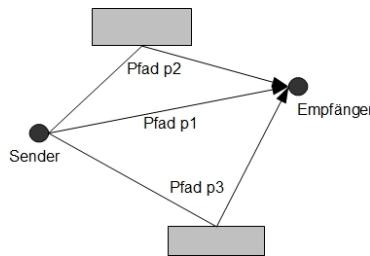


Abbildung 2.4.: Mehrwegeszenario bei der Signalübertragung

Eine solche Darstellung nennt man Mehrwegeausbreitung (engl.: multi path propagation), wobei natürlich noch weitere Szenarien mit mehreren Pfaden aber auch mit Objekten,

¹auch Sichtverbindung genannt (LOS, engl.: line-of-sight)

die in der direkten Verbindung liegen, vorstellbar sind. Diese Art der Ausbreitung führt zu einer **zeitlichen Spreizung** (engl.: **delay-spread**) des Sendesignals.

Unter der Annahme, dass Sender, Empfänger und die Objekte im Übertragungsweg als unbeweglich angesehen werden, lässt sich der Kanal über die lineare-zeitinvariante Impulsantwort $h_a(t)$ als Summe der Pfade p mit den separaten Dämpfungen a_p und Verzögerungen t_p darstellen.

$$h_a(t) = \sum_{p=1}^P a_p \delta(t - t_p) \quad (2.5)$$

Die daraus gebildete Kanalübertragungsfunktion $H_a(j\omega)$ wird auch als CTF (engl.: Channel Transfer Function) bezeichnet und lautet:

$$H_a(j\omega) = \int_{-\infty}^{\infty} h_a(t) e^{-j\omega t} dt \quad (2.6)$$

Das Empfangssignal kann nun als Faltung aus dem Sendesignal und der Kanalimpulsantwort und der Addition des zuvor beschriebenen Rauschsignals betrachtet werden.

$$r(t) = s(t) * h_a(t) + n(t) = \sum_{p=1}^P a_p s(t - t_p) + n(t) \quad (2.7)$$

Die Formulierung des Kanals als Summe der einzelnen Pfade ermöglicht die Darstellung als Verzögerungskette (engl.: Tapped-Delay-Line), wie sie in Abbildung 2.5 zu sehen ist. Das Sendesignal $s(t)$ erfährt durch die einzelnen Pfade Verzögerungen t_p und wird unterschiedlich stark gedämpft. Das Empfangssignal wird dann als Summe der Einzelpfade und dem zusätzlichen Rauschsignal gebildet. Geht man von einer existierenden Sichtverbindung aus, so kann t_1 als direkte Laufzeit der Welle zwischen Sender und Empfänger betrachtet werden.

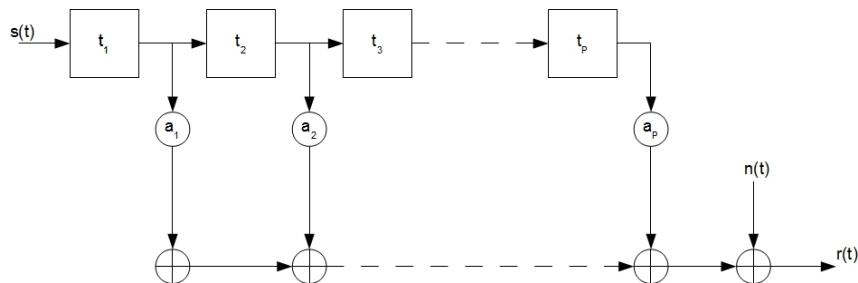


Abbildung 2.5.: Kanalmodell der Mehrwegeübertragung als Verzögerungskette

Durch den Einfluss des Kanals kommt es also zu einer Filterung des Sendesignals. Die hierdurch auftretenden Veränderungen führen zu einer Frequenzselektivität. Das Emp-

fangssignal kann also nicht mehr als konstantes Abbild des Sendesignals im Frequenzbereich betrachtet werden, so dass man von einem **frequenzselektiven Kanal** oder **frequenzselektiven Fading** spricht.[Kam04, S.92]

Der zeitvariante Übertragungskanal

Das zuvor beschriebene Mehrwegeszenario wurde für fixe Objekte betrachtet. Man muss bei einer realen Umgebungsmodellierung jedoch von nicht konstanten Bedingungen ausgehen, bei der sich Objekte bewegen und die Reflexionen an der Ionosphäre, die bei Raumwellen auftreten, nicht konstant sind. Zudem muss bei einer Radioübertragung auch ein bewegter Empfänger (z.B. im Auto) berücksichtigt werden.

Unter der Annahme dieser Einflüsse muss die mathematische Beschreibung des Übertragungskanals zu einer zeitvarianten Funktion erweitert werden.

$$h_a(t, \tau) = \sum_{p=1}^P a_p c_p(\tau) \delta(t - t_p) \quad (2.8)$$

$$H_a(j\omega, \tau) = \int_{-\infty}^{\infty} h_a(t, \tau) e^{-j\omega t} dt \quad (2.9)$$

Die Bewegung der Objekte und die zeitlichen Schwankungen in der Ionosphäre werden in den Gleichungen durch die Ergänzung des Zufallsprozesses $c_p(\tau)$ als statistisch unabhängige Änderungen der einzelnen Pfade betrachtet. [Hen08, S.18]

Man spricht in diesem Fall von einem **zeitselektiven Kanal** oder **zeitselektiven Fading**, da der Zufallsprozess über die Zeit τ hinweg die Frequenzselektivität des Kanals ändert. [Kam04, S.87]

Dopplereinflüsse

Durch die Bewegung des Empfängers kommt es nicht nur zu einer Änderung der Laufzeiten und Dämpfungsfaktoren im Kanal, sondern der Dopplereffekt führt zu einer Variation der Empfangsfrequenz.

Der Dopplereffekt beschreibt die Frequenzänderung von Wellen bei der relativen Bewegung von Sender und Empfänger zueinander. Die Änderung lässt sich wie folgt ausdrücken:

$$f_D = \frac{v_E}{c_0} f_0 \cos\alpha \quad (2.10)$$

f_0 ist die Trägerfrequenz des Signals, v_0 die Geschwindigkeit des Empfängers bei einem feststehenden Sender, c_0 ist die Lichtgeschwindigkeit und α gibt den Winkel zwischen Einfalls- und Bewegungsrichtung an. [Kam04, S.89]

Es kommt somit zu einer Spreizung der Kanalübertragungsfunktion um die Trägerfrequenz f_0 von $\pm f_D$, so dass man von **Dopplerspreizung** (engl.: **Doppler spread**) spricht. [Hen08, S.23]

Intersymbol-Interferenz

Als Intersymbol-Interferenz (ISI) oder auch Symbol-Übersprechen bezeichnet man die gegenseitige Beeinflussung aufeinander folgender Symbole bei der digitalen Nachrichtenübertragung. Symbole stellen hierbei zeitlich begrenzte abgeschlossene Informationseinheiten dar, die übertragen werden.

Zum einen kann es aufgrund der Filtereigenschaften eines Übertragungskanals zu einer zeitlichen Ausdehnung der Symbole kommen, so dass diese ineinander verlaufen können. Zum anderen kommt es im Falle der Mehrwegeausbreitung zu einem zeitlich versetzten Auftreffen der gleichen Symbole beim Empfänger, was wiederum eine Beeinflussung des Symbols mit sich selbst zur Folge hat.

Intercarrier-Interferenz

Das Träger-Übersprechen bzw. die Intercarrier-Interferenz (ICI) bezeichnet die gegenseitige Beeinflussung von Signalkomponenten benachbarter Trägerfrequenzen. Bei der Verwendung des in Abschnitt 3.2 beschriebenen Mehrträgerverfahrens kann es durch Frequenzverschiebung (z.B. durch Dopplerspreizung) zu einem Verlust der Orthogonalität kommen, woraus eine Störung angrenzender Subträger folgen würde.

2.2. DRM Systemübersicht

Die Komponenten eines DRM-Übertragungssystems lassen sich grob in die in Abbildung 2.6 gezeigten Kategorien einteilen. Der Aufbau kann ohne die Spezialisierung der Basisbandmodulation auch als Blockschaltbild der meisten digitalen Übertragungssysteme betrachtet werden. Der grau hinterlegte Teil zeigt die in digitaler Form (auf dem PC oder auf Zielpлатzformen) umgesetzten Komponenten.

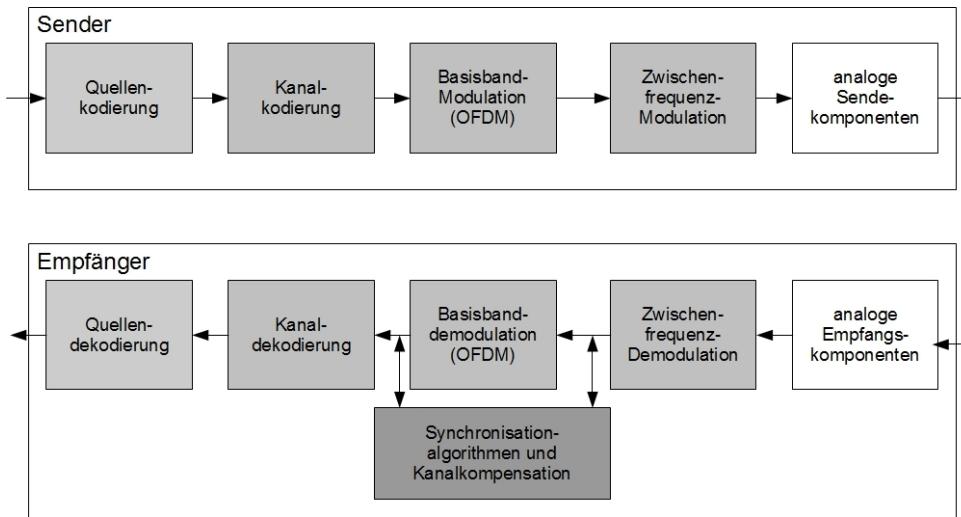


Abbildung 2.6.: Komponenten eines DRM Übertragungssystems

Man spricht bei dieser Art von Aufbau, bei dem bis auf die Umsetzung der Trägerfrequenz und zusätzlich zur Übertragung notwendigen analogen Komponenten eine Realisierung in digitaler Form stattfindet, auch von Software Defined Radio (SDR). SDR beschreibt also ein System, bei dem neben der Quellen- und Kanalkodierung auch die Modulation und die Umsetzung auf die Zwischenfrequenz in Software auf einem PC oder in programmierbarer Hardware wie DSPs und FPGAs stattfindet. Der Vorteil liegt in der Rekonfigurierbarkeit und der Kombination mehrerer Radiosysteme in einem Empfangssystem. So können bei bereits entwickelten SDR-Empfängern AM, FM, DAB und DRM mit einem Gerät empfangen werden. [Hen08]

Die Abbildung lässt erkennen, dass der Empfänger neben der Demodulation und Dekodierung, die meist gleiche oder ähnliche Komponenten mit entgegengesetzten Funktionen besitzen wie der Sender, eine zusätzliche Einheit zur Synchronisation und Kanalkompensation besitzt. Diese wird bei einer realen Übertragung notwendig, da der Kanal weder Sender noch Empfänger bekannt ist und die im Abschnitt 2.1.4 beschriebenen Störungen und Effekte auftreten können. Die Algorithmen die zum Ausgleich bzw. zur Synchronisation dienen, werden in Kapitel 4 vorgestellt.

2.2.1. Senderarchitektur und Multiplexdefinition

Die Abbildung 2.7 zeigt das in der DRM-Spezifikation definierte Blockschaltbild eines DRM-Senders. Die blau hinterlegten Funktionseinheiten stellen die in der Simulation umgesetzten Komponenten dar. D.h. die Simulation soll den gesamten Übertragungsvorgang inklusive Kanalkodierung, OFDM und der Modulation auf die Zwischenfrequenz umfassen. Hierzu werden die einzelnen Blöcke des Blockdiagramms, sowie die im Empfänger vorhandenen Systeme im folgenden Kapitel 3 vorgestellt.

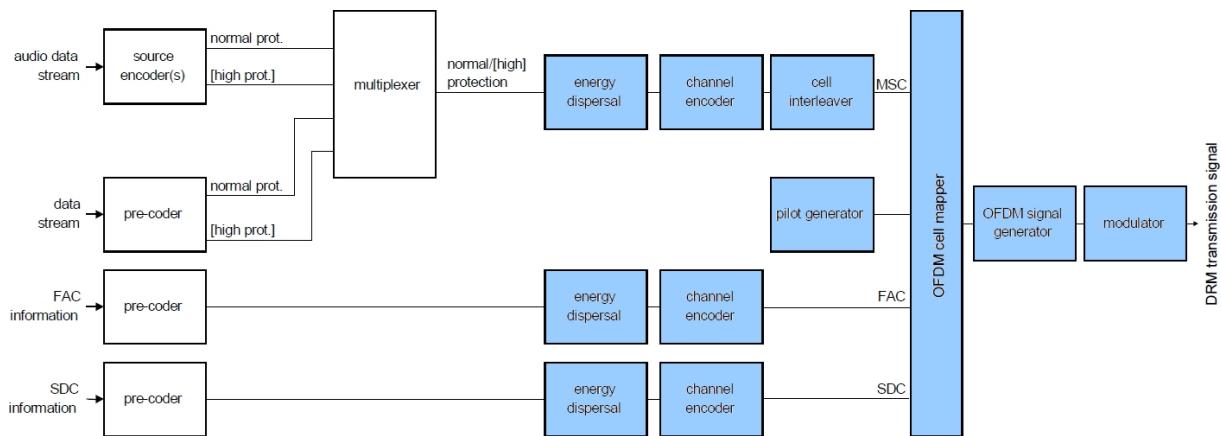


Abbildung 2.7.: Blockdiagramm des DRM-Senders [Eur09, fig.1]

In der Abbildung wird ersichtlich, dass sich der im Sender erzeugte Datenstrom aus drei logischen Kanälen und einer zusätzlichen Pilotsequenzstruktur (pilot generator), deren Aufgabe in Abschnitt 3.3 erklärt wird, zusammensetzt. Dabei stellt der oberste Zweig (MSC) den Kanal dar, der die kodierten Dienste beinhaltet. Die beiden anderen Zweige (FAC und SDC) sind sogenannte Kontrollkanäle, die dem Empfänger nötige Informationen zum Empfang und zur Dekodierung des Hauptkanals liefern. Die Bedeutung der einzelnen Kanäle soll nachfolgend erläutert werden.

Fast Access Channel

Der FAC (Fast Access Channel) stellt dem Empfänger Informationen zur Dekodierung zur Verfügung. Er beinhaltet entscheidende Parameter, die Aufschluss über die senderseitig gewählten Basisbandalgorithmen und den Übertragungsmodus (Bandbreite und Kanal) geben. [Eur09, S.72 ff.]

- Identity - Informationen zur Synchronisation eines Superframes (2.2.3)
- RM flag - Informationen zum Übertragungsmodus (2.2.2)
- Spectrum occupancy - Festlegung der Übertragungsbandbreite B_S (2.2.2)

- Interleaver depth flag - Festlegung der Tiefe des MSC Cell Interleavers (3.1.4)
- MSC mode - Festlegung der MSC QAM-Modulation (3.1.3)
- SDC mode - Festlegung der SDC QAM-Modulation (3.1.3)
- u.a.

Da der FAC grundlegende Parameter zur Initialisierung des DRM-Empfängers enthält, wird er regelmäßig in jedem Übertragungsrahmen verschickt und muss als erstes ausgewertet werden. Aufgrund der Bedeutung dieses Kanals und der Tatsache, dass ein Verlust dessen Informationen zu einer erneuten Initialisierung des Empfängers und somit zu einer Unterbrechung der Dekodierung der beiden anderen Kanäle führt, wird er mit höchster Sicherheit übertragen.

Service Description Channel

Der SDC (Service Description Channel) gibt Aufschluss über den im MSC vorhandenen Datenstrom. Er enthält Informationen über die Kodierungsart der Dienste und über mögliche Umschaltungen zu alternativen Frequenzen (AFS, engl.: Alternative Frequency Switching), auf welchen die gleichen Sendedaten empfangen werden können. Zudem werden im SDC Zusatzinformationen wie Zeit, Datum, Region und Sprache der MSC-Daten mitgeliefert. [Eur09, S.76 ff.] Der SDC muss nach dem FAC als zweiter Kanal ausgewertet werden, da ohne ihn eine Dekodierung der vom Sender angebotenen Services nicht möglich ist. Wie der FAC wird auch der SDC in regelmäßigen Abständen verschickt und sollte je nach Übertragungskanal eine höhere Sicherheitsstufe der Kodierung besitzen, als die im MSC übertragenen Dienste.

Main Service Channel

Der Hauptkanal des DRM-Systems ist der MSC (Main Service Channel). Er enthält die zu übertragenen Dienste (engl.: services), die aus Audio, Sprache und Daten bestehen können. Dabei ist es möglich jeweils bis zu vier Dienste zusammengefasst zu übertragen. Die einzelnen Dienste werden vor der Kanalkodierung zu einem Multiplexstrom zusammengefasst und der Kanalkodierung zugewiesen. Die Quellenkodierung der Audiodaten kann über diverse Verfahren (z.B. MPEG4-AAC, MPEG-CELP u.a.) umgesetzt werden. [Thi09]

Es besteht die Möglichkeit die Daten der einzelnen Dienste entweder mit einem identischen Sicherheitsniveau, dem EEP (Equal Error Protection), oder je nach Empfindlichkeit mit unterschiedlich starkem Schutz, dem UEP (Unequal Error Protection), zu kodieren. Diese unterschiedlichen Fehlerschutzmethoden haben Einfluss auf die nachfolgende Kanalkodierung und werden dem Empfänger über den SDC mitgeteilt.

2.2.2. Übertragungsmodi und Bandbreiten

Um den unterschiedlichen Bedingungen bei der Übertragung gerecht zu werden, werden bei DRM verschiedene Übertragungsmodi (engl.: robustness modes) definiert, welche in Tabelle 2.3 dargestellt sind. Grundlage bilden die in der DRM-Spezifikation definierten Kanalprofile [Eur09, S.156]. In diesen wurden verschiedene Kanalmodelle vom DRM-Konsortium festgelegt, um feste Randbedingungen für die Umsetzung der Übertragungskomponenten zu schaffen.

Mode	Kanaleigenschaften	Wellentyp	Frequenzband
A	Gaußscher Rauschkanal mit wenig Frequenzselektivität	Bodenwellen	LW,MW
B	Zeit- und Frequenzselektive Kanäle mit großen Umweglaufzeiten	Raumwellen	MW,KW
C	wie Mode B, jedoch mit höherem Doppler-spread	Raumwellen	KW
D	wie Mode C, jedoch mit noch höherem Doppler-spread und Delay	Raumwellen	KW
E	Zeit- und Frequenzselektive Kanäle im UKW-Bereich	Sichtwellen	UKW

Tabelle 2.3.: Festgelegte Übertragungsmodi bei DRM

Die definierten Modi unterscheiden sich in der Wahl der OFDM-Parameter (siehe Abschnitt 3.2.4) und in der Anzahl der Referenzsymbole (siehe Abschnitt 3.3). Damit kann je nach gewünschter Übertragungsreichweite bzw. aktuellen Kanalbedingungen vom Senderbetreiber auf die möglichen Störungen eingegangen und somit die Möglichkeit eines Empfangs optimiert werden.

Da bei der Definition von DRM eine Übergangsphase zwischen analogen und digitalen Rundfunk berücksichtigt und darauf geachtet wurde, die bestehenden Frequenzbänder und Rasterungen² des klassischen Radios zu nutzen, wurden für die weltweit unterschiedlich verwendeten Bandbreiten ebenfalls diverse Spektralbelegungen definiert. So wird beispielsweise in Deutschland im analogen Rundfunk ein Frequenzraster von 9 kHz bei LW, MW und KW genutzt. Die Tabelle 2.4 zeigt die festgelegten Bandbreiten bzw. Spektralbelegungen B_S (engl.: spectrum occupancy) des DRM-Systems.

²Trägerabstand zwischen zwei Rundfunkkanälen

	Spektralbelegung B_S					
	0	1	2	3	4	5
Bandbreite in kHz (Mode A-D)	4,5	5	9	10	18	20
Bandbreite in kHz (Mode E)	100	-	-	-	-	-

Tabelle 2.4.: Spektralbelegung bzw. Übertragungsbandbreiten in DRM

Auch hier kann der Senderbetreiber je nach Region, aber auch nach Qualitätsanforderung an das zu übertragenden Signal, ein Raster wählen.

Sowohl der Übertragungsmodus als auch die Übertragungsbandbreite haben starken Einfluss auf die Bitrate der zu übertragenden Daten. Deshalb werden die Parameter für die im Kapitel 3 beschriebenen Algorithmen stets in Abhängigkeit der genannten Übertragungsbedingungen gewählt.

2.2.3. Übertragungsrahmen

Der Abschnitt soll einen kurzen Überblick über die in DRM verwendete Rahmenstruktur geben, um eventuell auftauchende Begriffe zuvor zu klären. Die genaue Organisation der Daten wird in Abschnitt 3.2.4 näher erläutert.

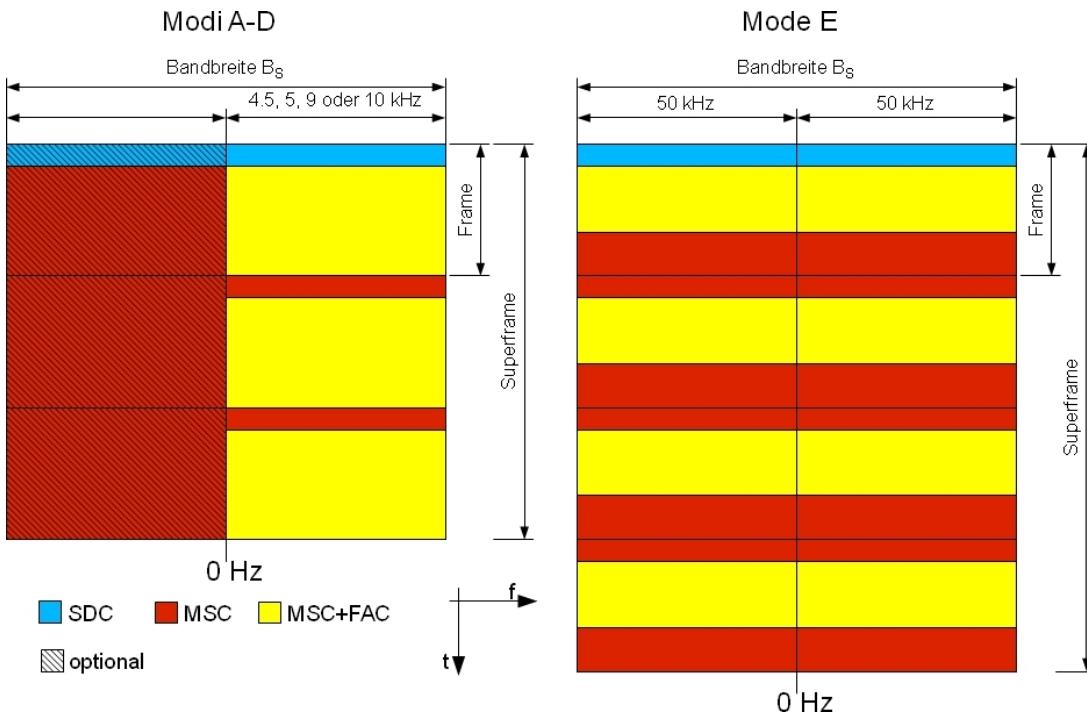


Abbildung 2.8.: Rahmenstruktur bei DRM im Basisband

Man unterscheidet in DRM zwischen einem einfachen Übertragungsrahmen (engl.: Frame) und der übergeordneten Struktur dem Superframe.

Die Abbildung 2.8 zeigt die Organisation der Übertragungsrahmen für die verschiedenen Modi und Spektralbelegungen. Es wird dargestellt, dass ein Superframe in den Modi A bis D aus drei einzelnen Frames und im Mode E aus vier einzelnen Frames gebildet wird.

Die SDC-Daten werden jeweils nur am Anfang des ersten Rahmens gesendet. In allen Übertragungsmodi liegen die FAC-Daten verteilt in allen Frames. Während sie im Mode E auf die gesamte Bandbreite verteilt sind, wird in den Modi A bis D nur das obere Seitenband bis 4,5 kHz ($0Hz < f_{FAC} < 4.5kHz$) für diese genutzt, was an der optionalen Spektralbelegung liegt und es dem Empfänger ermöglicht das Empfangsfilter auf 4,5 kHz zu begrenzen bevor die Auswertung des FAC stattgefunden hat.[Hen08, S.42] Die restlichen Daten in einem Superframe bestehen aus den MSC-Daten und den Piloten (siehe Abschnitt 3.3).

Die Synchronisation eines Superframes erfolgt mit Hilfe der Identity flag im FAC. Dieses 2 Bit Wort gibt je nach Wert die Lage der aktuellen FAC-Daten im Superframe an, so dass der Empfänger die genaue Rahmennummer kennt.[Eur09, S.73] Bei der Simulation spielt diese Synchronisierung keine Rolle, da eine Auswertung der Daten der verschiedenen Kanäle nicht stattfindet.

3. Algorithmen des DRM Standards

Das folgende Kapitel stellt die in der DRM-Spezifikation definierten und für die Umsetzung der Simulation verwendeten Algorithmen und Komponenten vor. Dabei sollen die Prinzipien der Komponenten zunächst in allgemeiner Form erläutert und anschließend für den DRM-Standard spezifiziert werden. Die Struktur eines DRM-Senders wurde in Abschnitt 2.2.1 vorgestellt. Da die Simulation ein gesamtes Übertragungssystem nachstellen soll, werden in diesem Kapitel zugleich die nicht in der Spezifikation festgehaltenen aber in der Umsetzung verwendeten Empfängeralgorithmen erklärt.

3.1. Kanalkodierung und Modulation

3.1.1. Energy dispersal

Lange Null- bzw. Einserfolgen können innerhalb eines digitalen Übertragungssystems zu Problemen führen. Um die statistischen Eigenschaften des zu übertragenden Signals zu verbessern werden die Daten vor den folgenden Prozessen mit Hilfe eines Scramblers verwürfelt.[Loc02, S.305] Diese Maßnahme wird als Datenverwürfelung (engl.: scrambling) oder Energieverstreuung (engl.: energy dispersal) bezeichnet.

Ein Scrambler basiert auf der Verwendung eines PRNG (Pseudozufalls-Rauschgenerator, engl.: Pseudo Random Noise Generator), welcher ein lineares Schieberegister mit einer festen Anzahl an Verzögerungselementen darstellt. In Abhängigkeit der Lage der Modulo-2 bzw. XOR-Verknüpfungen zwischen den Verzögerungsgliedern und der Initialisierungszuständen dieser wird eine deterministische Sequenz erzeugt, die jedoch lang genug ist, um diese als Zufallssequenz zu betrachten.

Die Abbildung 3.1 zeigt das Scramblermodell, wie es bei DRM Verwendung findet.

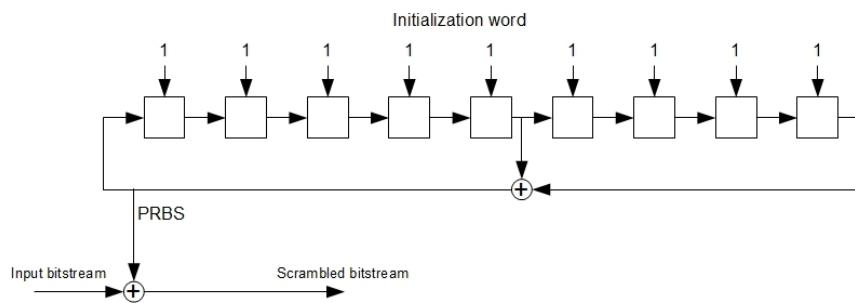


Abbildung 3.1.: PRNG als Scrambler im DRM System

3. Algorithmen des DRM Standards

Das Scramblerpolynom ist mit $P(x) = x^9 + x^5 + 1$ festgelegt und das Initialisierungswort ist so gewählt, dass alle Zustände der Verzögerungsglieder auf 1 gesetzt sind. [Eur09, 114]

Der Eingangsdatenstrom wird mit der vom Scrambler erzeugten PRBS (Pseudozufalls-Binärsequenz, engl.: Pseudo Random Binary Sequence) über eine Modulo-2-Addition verknüpft. Um einen Verlust der Synchronisation des Zufallsgenerators zu vermeiden, wird ein Reset nach jedem Frame durchgeführt. Das bedeutet, dass das Initialisierungswort die Zustände der Schieberegister zum Beginn jedes Frames auf den Ursprungszustand zurücksetzt. Die Framegröße richtet sich dabei nach den verschiedenen Kanälen, so dass jeder einzelne Scrambler der Kanäle unterschiedliche Taktraten besitzt.

Auf der Empfangsseite wird ein identischer PRNG im sogenannten Descrambler verwendet, um den verwürfelten Datenstrom wiederzugewinnen. Aufgrund der zwischen dem Scrambler-/Descrambler-Paar liegenden Verarbeitungsschritten, sowie dem Übertragungskanal entsteht eine zunächst unbekannte Verzögerung zwischen der Verwürfelung und der Wiederherstellung. Demnach ist es wichtig, dass eine Synchronisation zwischen dem Scrambling-Prozess stattfindet, wie am Beispiel ersichtlich wird.

Basierend auf der Abbildung 3.2 soll das folgende Beispiel den Scrambling-Prozess erläutern, wobei eine optimale, fehlerfreie Übertragung der Daten ohne Verzögerung angenommen wird.

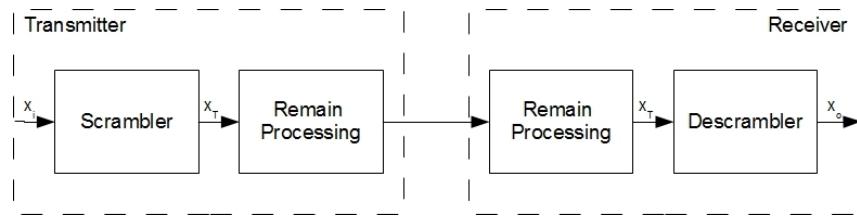


Abbildung 3.2.: Beispiel zum Scrambler/Descrambler-Prozess im DRM Übertragungssystem

Die ersten 16 Bits am Ausgang des PRNG sind wie folgt:[Eur09, S.114, Tabelle 61]:

$$pn = 0\ 0\ 0\ 0\ 0\ 1\ 1\ 1\ 1\ 0\ 1\ 1\ 1\ 1\ 1\ 0$$

Zudem wird folgende Folge von Einsen am Eingang des Scramblers angenommen:

$$x_i = 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1$$

Senderseitig wird der Eingangsdatenstrom x_i einfach mittels XOR mit dem Ausgang des PRNG im Scrambler verknüpft (siehe Abb. 3.1). Nach der Übertragung und den dazwischenliegenden Verarbeitungsschritten wird der empfangene Datenstrom x_t mit der selben PRBS im Descrambler genauso verknüpft, was zum Ausgangsdatenstrom x_o führt.

pn	0	0	0	0	0	1	1	1	1	0	1	1	1	1	1	0
x_i	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$x_t = pn \oplus x_i$	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	1
$x_o = pn \oplus x_t$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Tabelle 3.1.: Beispiel zum Scrambling-Prozess

Wie zu erkennen ist, wird der Eingangsdatenstrom x_i im Empfänger exakt zum Ausgangsdatenstrom x_o rekonstruiert. Der zu übertragende Datenstrom x_t enthält hierbei keine so lange Einserfolge wie x_i , so dass die Energie des Signals besser verteilt ist. Das Beispiel zeigt zudem die Notwendigkeit der Synchronisation zwischen den unabhängig arbeitenden Sende- und Empfangsprozessen. Sollte bei einer auftretenden Verzögerung keine Synchronisation stattfinden, so kann der Datenstrom am Ausgang des Empfängers nicht mehr korrekt wiederhergestellt werden, was somit einen Informationsverlust zur Folge hätte.

3.1.2. Kanalkodierung

Die Kanalkodierung bei DRM besteht aus mehreren Einzelkomponenten. Neben der Verwendung eines Faltungskodierers und einer anschließenden Punktierung fällt das Bit-Interleaving auch unter den Oberbegriff Kanalkodierung, da alle Komponenten dazu dienen den Einfluss von Störungen des Datenstroms während der Übertragung durch zusätzliche Redundanz und Verwürfelung zu reduzieren und im Empfänger rückgängig zu machen.

3.1.2.1. Multilevel-Kodierung

DRM benutzt zur Umsetzung der Kanalkodierung das Verfahren der Multilevel-Kodierung (MLC, engl.: Multilevel-Coding). Die Idee besteht darin, durch unterschiedlich starke Kodierung der Daten bzw. Schutzmechanismen eine optimale Übertragungsqualität zu erreichen. Dabei werden bei DRM bis zu 3 Level genutzt, die die bei der QAM (siehe Abschnitt 3.1.3) fehleranfälligeren Bitpositionen im Konstellationsdiagramm mit einem stärkeren Schutz durch größere Redundanz und Verwürfelung versehen. Dies wird zum einen durch den Faltungscoder mit unterschiedlicher Punktierung (3.1.2.2) und zum anderen durch das Bit-Interleaving (3.1.2.4) erreicht.[YSH99]

Da bei DRM verschiedene Mappingstrategien für die QAM vorgesehen sind, unterscheidet man noch zwischen dem Standard Mapping (SM), dem Mixed Hierarchical Mapping (HMMix) und dem Symmetrical Hierarchical Mapping (HMsym). Diese Verfahren unterscheiden sich in der Art der Zuweisung der Daten in das Konstellationsdiagramm und somit auch in der Aufteilung der Daten im Multilevel-Kodierer. Die genaue Aufteilung und Zuweisung wird in der DRM-Spezifikation [Eur09, S.116 ff.] und [Eur09, S.128 ff.]

erläutert. Eine Überprüfung beim Empfang verschiedener DRM-Sender und die Auswertung mehrerer im Internet verfügbarer Teststreams zeigte, dass vor allem das Standard Mapping genutzt wird, weshalb in der Simulation nur eine Umsetzung dieses Mappings erfolgt. Eine Erweiterung der Simulation auf HMMix und HMsym ist jederzeit möglich.

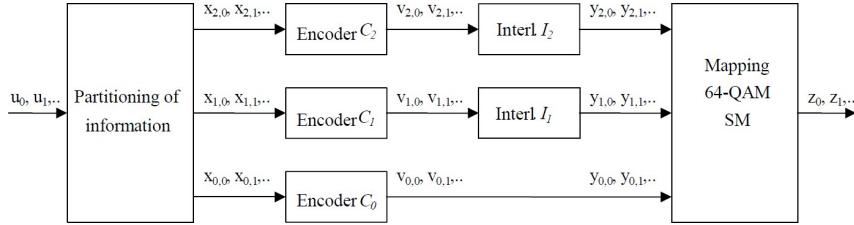


Abbildung 3.3.: Multilevel-Kodierung in DRM mit 3 Level für SM [Eur09, 115, Abb.29]

Die Abbildung 3.3 zeigt die Struktur des Multilevel-Kodierers für das SM bei einer 64-QAM. Der von der Quellenkodierung kommende Datenstrom u wird so auf die drei Level aufgeteilt, dass nach den Encodern mit entsprechender Punktierung C_2 bis C_0 die Vektoren v_2 bis v_0 die gleiche Länge besitzen und somit gleichmäßig beim QAM-Mapping verteilt werden. Jeder Zweig besitzt dabei eine andere Coderate für die Kanalkodierung. Zusätzlich bietet DRM die Möglichkeit beim SDC und MSC ein bestimmtes Schutzlevel zu wählen. Diese Schutzlevel unterscheiden sich in der Gesamtcoderate der Multilevel-Kodierung und bieten demnach verschiedene Stärken des Fehlerschutzes. Die Coderaten für die verschiedenen Kanäle, QAM-Konstellationen und Schutzlevel werden im Anhang A.1.2 dargestellt. Alle weiteren Varianten der Multilevel-Kodierung befinden sich im Anhang A.1.1.

3.1.2.2. Punktierte Faltungscodes

Um die sogenannte Vorwärtsfehlerkorrektur (FEC) durchzuführen, existieren verschiedene Ansätze der Kanalkodierung, wie z.B. Blockcodes, Faltungscodes und eine Verkettung dieser, die sich in der Einteilung der zu kodierenden Daten unterscheiden.[Loc02],[BG07] DRM benutzt das Verfahren der Faltungscodes, die im Gegensatz zu den Blockcodes die Daten kontinuierlich erweitern und nicht zuvor in Blöcke einteilen, sowie durch ein Gedächtnis eine Abhängigkeit der aktuellen Daten von den Vorherigen herstellt.

Nichtrekursive Faltungscoder sind prinzipiell eine Zusammenschaltung mehrerer FIR-Filter, die eine feste Anzahl an Verzögerungsgliedern darstellen, welche in mehreren Zweigen unterschiedlich miteinander mittels XOR verknüpft sind. Die Faltungscodes besitzen ihren Namen aufgrund der möglichen mathematischen Beschreibung durch die zeitdiskrete Faltung. [Loc02, S.525]

Geht man von dem einfachen, in Abbildung 3.4 dargestellten, Faltungscoder mit 2 Speichern aus, so kann daraus das zugehörige Zustandsdiagramm erstellt werden, indem die

3. Algorithmen des DRM Standards

Folgezustände der Speicher, ausgehend vom Zustand „00“, durch Anlegen der entsprechenden Bits am Eingang E ermittelt werden.

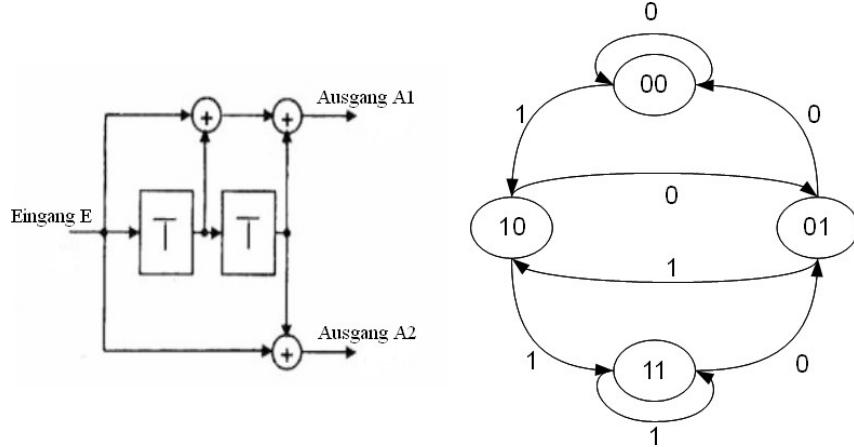


Abbildung 3.4.: Beispiel eines Faltungscodes mit 2 Speichern und das zugehörige Zustandsdiagramm

Hieraus ergibt sich eine weitere Möglichkeit der Beschreibung von Faltungscodes. Mit Hilfe eines sogenannten Trellisdiagramms, weshalb man diese Codes auch Trellis¹-Codes nennt, trägt man das Zustandsdiagramm über eine Zeitachse auf und kann das Verhalten bei einer bestimmten Eingangsfolge beobachten. Die folgende Abbildung 3.5 zeigt das Trellisdiagramm für den Faltungscoder nach Abbildung 3.4. Die Bits in Klammern stellen den Ausgangszusand (A1,A2) dar.

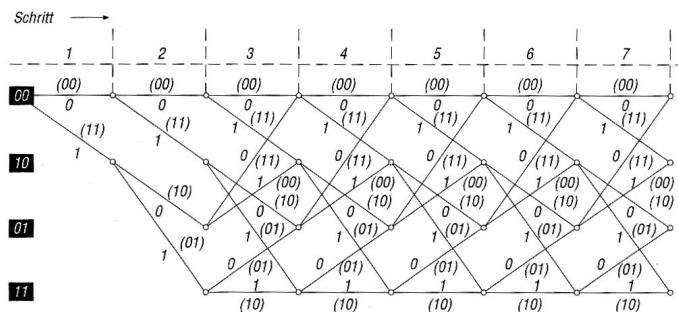


Abbildung 3.5.: Beispiel eines Trellisdiagramms [Loc02, S.523]

Weitere Darstellungsmöglichkeiten, Beispiele und exakte Erklärungen sind in [Loc02, S.519 ff.], [BG07] und [Päh98] zu finden.

¹Trellis steht für die engl. Bezeichnung „Spalier“

3. Algorithmen des DRM Standards

Die Abbildung 3.6 zeigt den Aufbau, des im DRM-Standard definierten Faltungscoders. Er besteht aus sechs Verzögerungselementen bzw. Speichern und einer festgelegten Anzahl an Verknüpfungen (engl.: constraint length) von 7. Die ankommenden Bits a_i des Eingangsframes werden über bestimmte Verknüpfungsvorschriften den sechs Ausgängen $b_{0,i}$ bis $b_{5,i}$ zugewiesen. Dabei entsteht eine Versechsfachung (ohne Punktierung) der Datenmenge, welche zum einen die geforderte Redundanz herstellt und durch die bestimmte Anordnung der Struktur eine Fehlerkorrektur im Empfänger ermöglicht.

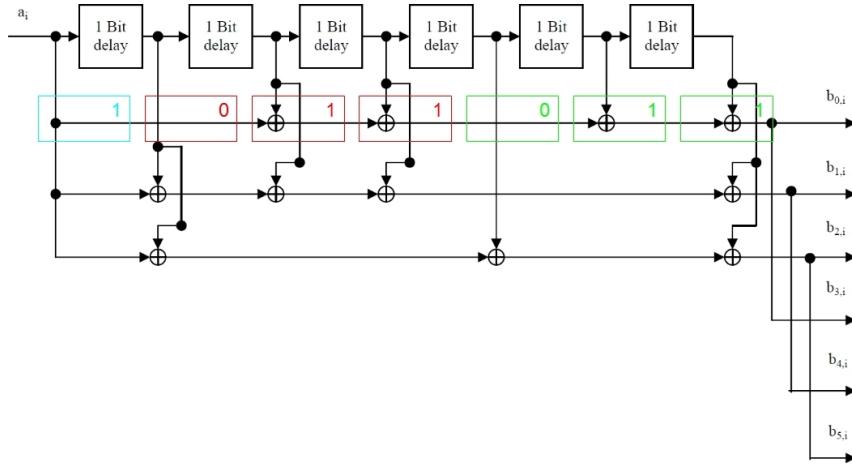


Abbildung 3.6.: Faltungscoder des DRM Systems [Eur09, Abb.34]

Man kann jeden einzelnen Ausgang $b_{a,i}$ des Faltungscoder über ein separates Generatorpolynom beschreiben, wobei man häufig zur besseren Übersicht die oktale Darstellung wählt. Diese beschreibt die Zusammensetzung des Polynoms über die Verknüpfungen indem man diese auf 3 Bit aufteilt (siehe Abbildung 3.6) und den dezimalen Wert ausliest. Bei der festgelegten Länge von 7 entfallen die beiden oberen Bits der ersten Oktalzahl und es besteht immer eine Verknüpfung, so dass sich immer eine 1 ergibt. Die beiden anderen Bitblöcke der Darstellung ergeben 3, so dass für das Beispiel $b_{0,i}$ sich ein Generatorpolynom von 133 in oktaler Form ergibt.

Ausgang $b_{a,i}$	Verknüpfungsvorschrift	Polynom (oktal)
$b_{0,i}$	$a_i \oplus a_{i-2} \oplus a_{i-3} \oplus a_{i-5} \oplus a_{i-6}$	133
$b_{1,i}$	$a_i \oplus a_{i-1} \oplus a_{i-2} \oplus a_{i-3} \oplus a_{i-6}$	171
$b_{2,i}$	$a_i \oplus a_{i-1} \oplus a_{i-4} \oplus a_{i-6}$	145
$b_{3,i}$	$a_i \oplus a_{i-2} \oplus a_{i-3} \oplus a_{i-5} \oplus a_{i-6}$	133
$b_{4,i}$	$a_i \oplus a_{i-1} \oplus a_{i-2} \oplus a_{i-3} \oplus a_{i-6}$	171
$b_{5,i}$	$a_i \oplus a_{i-1} \oplus a_{i-4} \oplus a_{i-6}$	145

Tabelle 3.2.: Zusammensetzung des Codeworts und des Polynoms beim Faltungscoder

Die Tabelle gibt die Verknüpfungen der Eingangsbits und die oktale Form des Generatorkoeffizienten-Polynoms für jeden der sechs definierten Ausgänge an, wobei die letzten drei Ausgänge nur eine Kopie der ersten drei Ausgänge darstellen.

Wie beim Scrambler wird auch hier ein Rücksetzen des Faltungscoders nach jedem Eingangsrahmen gefordert. Es müssen alle Zustände der Speicher auf Null gesetzt werden, was durch das Anhängen von einer sechs Bit langen Nullfolge an den Eingangsframe erreicht und als Terminierung bezeichnet wird. Hierdurch entstehen zusätzliche 6x6 Bits am Ausgang, welche nicht zu den eigentlichen Nutzdaten gehören und als Abschlussbits (engl.: tail bits) bezeichnet werden. Diese Bitfolge wird ebenfalls weiter verarbeitet und auch übertragen, weshalb man bei der Dekodierung darauf achten muss.

Punktierung

Der vorgestellte Faltungscoder besitzt eine feste Coderate von 1/6 und wird auch als Muttercode (engl.: mother code) bezeichnet. Um die bei der Multilevel-Kodierung geforderten unterschiedlichen Coderaten und die daraus folgenden unterschiedlich starken Redundanzen der einzelnen Level zu erreichen, wird anschließend eine Punktierung eingesetzt.

Hierzu werden die aus dem Eingangsbit a_i gewonnenen sechs Ausgangsbits $b_{0,i}$ bis $b_{6,i}$ über ein Raster je nach gewünschter Coderate selektiert. D.h. es kann durch „Weglassen“ bestimmter Ausgänge jede beliebige Coderate hergestellt werden. Die Punktierungsraster (engl.: puncturing patterns) sind in der Spezifikation [Eur09, S.120] festgehalten.

Will man aus dem Muttercode mit der Rate 1/6 beispielsweise die Coderate 1/4 erreichen, so wird folgendes Raster verwendet.

B_0	B_1	B_2	B_3	B_4	B_5
1	1	1	1	0	0

Tabelle 3.3.: Punktierungsraster für die Coderate 1/4

Bei einer Ausgangsbitfolge des Faltungskodierers

$$b_i = 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0$$

werden also die Ausgänge $b_{a,4}$ und $b_{a,5}$ weggelassen, so dass sich folgende neue Folge ergibt.

$$v_i = 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1$$

Beim FAC werden alle Bits inklusive der Abschlussbits über die angegebenen Raster punktiert. SDC und MSC weisen die Besonderheit auf, dass die Tail-Bits, also die letzten 36 Bits des Frames, separate Raster zur Punktierung verwenden, welche in [Eur09, S.122] definiert sind.

3.1.2.3. Viterbi-Decoder

Zur Dekodierung des Faltungscodes können verschiedene Methoden verwendet werden die unter der Bezeichnung Sequentielle Dekodierung zu finden sind [Päh98], wobei der 1967 von Viterbi entwickelte Algorithmus das am häufigsten angewandte und umgesetzte Dekodierverfahren darstellt.

Der Viterbi-Decoder bestimmt über seinen Algorithmus aus der Empfangsfolge (bei Fehlerfreiheit = Ausgangsfolge des Faltungscoders) die wahrscheinlichste Eingangsfolge, wenn die Struktur bzw. das Generatorpolynom des Faltungscoders bekannt ist. Der Viterbi-Algorithmus arbeitet dabei mit Metriken und bestimmt daraus den wahrscheinlichsten Weg, den eine Folge besitzt. Diese Verfahren werden auch als Maximum Likelihood² Decodierung bezeichnet.

Folgendes Beispiel soll die Funktionsweise der Viterbi-Metrik veranschaulichen. Die fehlerfreie Empfangsfolge $A = [11\ 10\ 00\ 01\ 01\ 11]$ entspricht der Eingangsfolge $E = [1\ 0\ 1\ 1\ 0\ 0]$. Die Abbildung 3.7 zeigt in Rot den Verlauf dieser Folge im Trellisdiagramm.

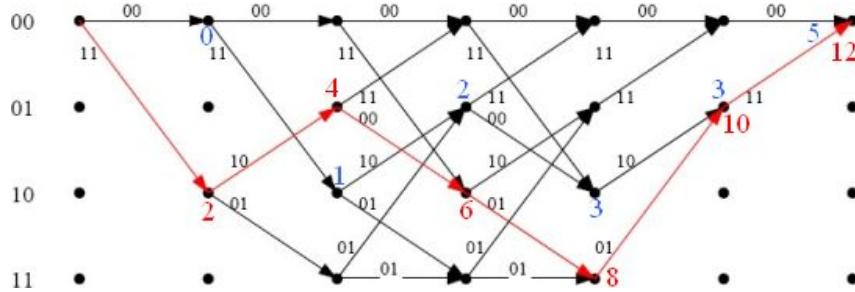


Abbildung 3.7.: Beispiel zur Metrikberechnung des Viterbi-Decoders

Ausgehend vom Zustand „00“ werden schrittweise die möglichen Ausgangszustände mit denen der Empfangsfolge A verglichen. Bei Gleichheit einzelner Bits der Zustände wird die Metrik erhöht (stimmen 2 Bits überein, so entsteht ein Metrikwert von 2, stimmt nur 1 Bit, so entsteht ein Metrik-Wert von 1 usw.). Diese Werte werden für jeden Schritt aufsummiert. Im roten Pfad stimmen alle Zustände überein, es entsteht somit der maximale Metrik-Wert von 12 für 2^6 Ausgangszustände. Hingegen wird bei der Überprüfung eines anderen Pfades eine geringer Metriksumme (blau) erzeugt. Aus den roten Pfad mit der höchsten Metrik kann nun rückwärts die Eingangsfolge E zurückgewonnen werden. Sollte eine geringe Zahl an Bits einer Empfangsfolge A (hier Metrik-Summe $\neq 12$) falsch sein, so wird über dieses Metrik-Verfahren auch eine Korrektur vorgenommen, da der Weg mit der größten Metrik der Eingangsfolge entspricht.

Da auch dieses Thema, wie schon der Faltungscoder, nicht ausführlich behandelt werden kann, wird an dieser Stelle auf die gleiche Literatur verwiesen ([Loc02], [BG07], [Päh98]), die auch die Fehler korrigierende Eigenschaft des Viterbi-Algorithmus näher erläutert.

²engl. für „Größte Wahrscheinlichkeit“

3.1.2.4. Bit-Interleaving

Als Bündelfehler (engl.: burst errors) werden Störungen des binären Datenstroms bezeichnet, welche sich über mehrere Bits (Bündel) hinweg erstrecken und somit deutliche Probleme bei der Dekodierung eines Datenframes erzeugen können. Ein weiterer Schutzmechanismus der Kanalkodierung, um den Einfluss von Bündelfehlern während der Übertragung zu verringern, ist das Interleaving.

Das Bit-Interleaving im DRM-System erfolgt über einen sogenannten Blockinterleaver, der einen Rahmen (Datenblock) eines Levels nach dem Faltungscoder mittels eines Pseudozufallsvektors erneut verwürfelt. Hierzu wird zunächst ein Vektor erstellt, der die als zufällig zu betrachtenden Positionen der Bits innerhalb eines Übertragungsrahmens enthält. Anschließend werden die Bits dieser Permutation unterzogen, indem sie nach der Reihenfolge des Permutationsvektors angeordnet werden. Da der Vertauschungsvektor im Empfänger bekannt ist, kann der Ursprungsrahmen hier durch den Bit-Deinterleaver exakt rekonstruiert werden.

Folgendes Beispiel soll das Prinzip des Bit-Interleavings verdeutlichen, wobei i für die Position des Bits innerhalb eines Frames steht und p das Level des Frames im Multilevel-Coder darstellt. Geht man von folgendem Rahmen $v_{p,i}$ als Ausgang des Faltungscoder aus:

$$v_{p,i} = 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 1$$

und ist der Permutationsvektor $\Pi_p(i)$ wie folgt definiert:

$$\Pi_p(i) = 1 \ 9 \ 5 \ 4 \ 8 \ 3 \ 10 \ 2 \ 7 \ 6$$

so ergibt sich nach dem Interleaving der neue Rahmen $y_{p,i}$:

$$y_{p,i} = v_{p,\Pi_p(i)} = 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 1$$

Der Permutationsvektor muss immer die Länge eines Rahmens des jeweiligen Kanals besitzen. Der Algorithmus zur Erstellung des Vektors $\Pi_p(i)$ befindet sich in der Spezifikation [Eur09, S.125].

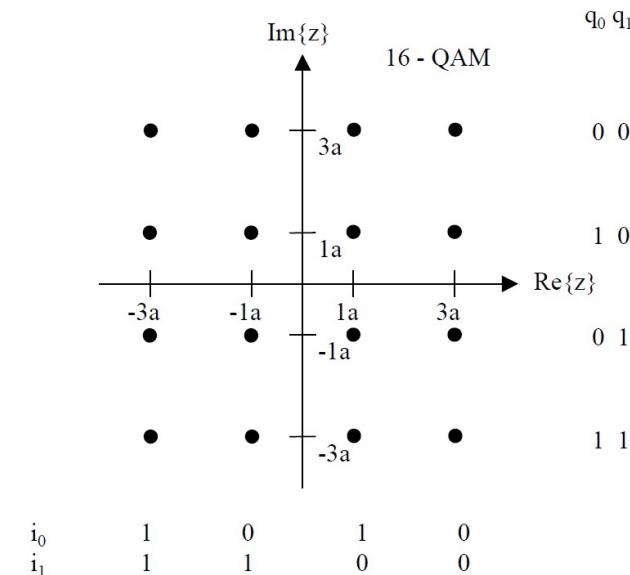
3.1.3. QAM-Mapping

Nach der Kanalkodierung werden die Daten der einzelnen Kanäle, die bis hier als Bitstrom betrachtet wurden, über eine diskrete Quadratur Amplituden Modulation (QAM, engl.: Quadrature Amplitude Modulation) zu sogenannten Symbolen zusammengefasst. Die QAM ist eine lineare Modulationsart, die sowohl Amplitude als auch Phase eines Signals beeinflusst und somit eine bessere spektrale Effizienz erzielt als die Modulation der Einzelkomponenten.[Kam04, S.281 ff.] Das Prinzip der QAM ist die Modulation der Amplitude eines Signals mit zwei zueinander um 90° verschobenen Trägerschwingungen

gleicher Frequenz und der anschließenden Addition dieser. Damit können mit einem Symbol mehrere Bits zugleich übertragen werden, was zu den verschiedenen Typen der QAM führt. Ist es möglich eine m -Bit lange Binärzahl am Eingang einem der $X = 2^m$ Symbolzustände am Ausgang zuzuweisen, so spricht man von einer X -QAM.

Im DRM-System wird eine rechteckige orthogonale QAM mit den Größen 4, 16 und 64 verwendet. Das bedeutet es können entweder 2 Bits (4-QAM), 4 Bits (16-QAM) oder 6 Bits (64-QAM) zugleich in ein Symbol gebracht werden.

Eine besonders anschauliche Weise des Prinzips einer QAM ist das Konstellationsdiagramm bzw. der Konstellationsraum. Er stellt die komplexe Ebene mit den durch die Modulation möglichen Signalzustände dar und weist jedem dieser eine bestimmte Bitfolge zu. Die folgende Abbildung 3.8 zeigt den im DRM benutzten Konstellationsraum für die 16-QAM.



Bit ordering: $\{i_0 \ i_1 \ q_0 \ q_1\} = \{y'_0 \ y'_1 \ y'_2 \ y'_3\}$

Abbildung 3.8.: Konstellationsdiagramm der 16-QAM [Eur09, 131, Abb.38]

Hierbei wird der am Eingang des QAM-Mappings liegende Bitstrom in Blöcke mit jeweils 4 Bit unterteilt und dem im Konstellationsdiagramm liegenden komplexen Symbol zugewiesen.

Sollte z.B. der eingehende Bitstrom y_i wie folgt aussehen,

$$y_i = 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0$$

so würde sich folgende Symbolfolge z nach dem QAM-Mapping ergeben

$$z = 1 + 1j \ - 3 + 1j \ - 1 - 3j \ 3 + 3j$$

3. Algorithmen des DRM Standards

Die Demodulation des QAM-Symbols geschieht nach dem genau umgekehrten Prinzip. D.h. es werden die empfangenen Symbole wieder einer der Bitfolgen zugeordnet.

Je größer der QAM-Typ ist, desto mehr Zustände existieren im Konstellationsraum und desto mehr Bits können gleichzeitig übertragen werden. Dabei ist jedoch zu beachten, dass bei gleichem Betrag und somit Leistung eines Symbols die Abstände zwischen diesen immer geringer werden. Dies kann bei reellen Übertragungskanälen mit Rauschanteil dazu führen, dass der Empfänger fehlerhafte Zuordnungen bei der Demodulation durchführt.

Der im Konstellationsdiagramm in Abbildung 3.8 dargestellte Faktor a stellt den sogenannten Normierungsfaktor dar, was bedeutet, dass die erzeugten Symbole mit diesem Faktor multipliziert werden. Für jeden QAM-Typ ist dabei ein fester Wert für a definiert:

- $a = \frac{1}{\sqrt{42}}$ für die 64-QAM
- $a = \frac{1}{\sqrt{10}}$ für die 16-QAM
- $a = \frac{1}{\sqrt{2}}$ für die 4-QAM

Die einzelnen Kanäle des DRM-Systems besitzen unterschiedliche Möglichkeiten zur Auswahl der QAM-Größe:

- Der FAC wird stets mit einer 4-QAM moduliert. Diese bietet die höchste Sicherheit der Daten, welche für den Empfang des FAC notwendig ist, da der Kanal die Grundlage für die weitere Dekodierung im Empfänger darstellt.
- Der SDC kann in den Modi A-D wahlweise mit einer 16-QAM oder 4-QAM, je nach gewünschter Sicherheit oder Datenmenge, moduliert werden. Wird eine erhöhte Anzahl an Zusatzinformationen gewünscht, so kann durch die 16-QAM eine bessere Komprimierung der Daten erreicht werden. Im Mode E wird auch hier wegen der höheren Robustheit immer die 4-QAM verwendet.
- Beim MSC kann unter den gleichen Aspekten zwischen einer 16-QAM und einer 64-QAM in den Modi A-D oder zwischen 4-QAM und 16-QAM im Mode E im Sender ausgewählt werden.

3.1.4. Cell-Interleaving

Vor der OFDM werden ausschließlich im MSC die durch das QAM-Mapping erstellten Symbole nochmals einer Permutation unterzogen. Hierzu wird immer das gleiche Prinzip wie beim Bit-Interleaving (Abschnitt 3.1.2.4) auf die Symbole in einem Rahmen angewandt, welches als „short interleaving“ (kurze Verschachtelung) bezeichnet wird. Zusätzlich können die Zellen über mehrere Frames hinweg vertauscht werden. Dieser Vorgang wird als „long interleaving“ (lange Verschachtelung) bezeichnet und führt bei schlechteren Übertragungsbedingungen zu einer erneuten Sicherheit.

3. Algorithmen des DRM Standards

Die Abbildung 3.9 zeigt das Prinzip des „long interleaving“. Die Daten der einzelnen Frames werden hierbei über mehrere Frames hinweg vertauscht.

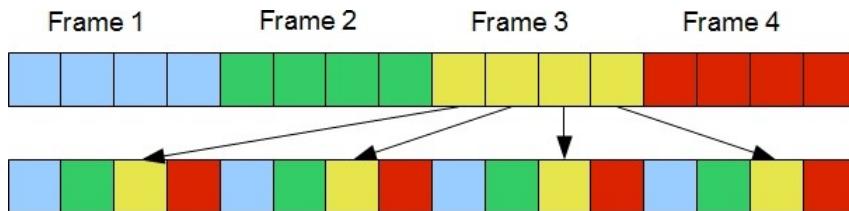


Abbildung 3.9.: Prinzip des „long interleaving“

Mit dem „long interleaving“ kann der Einfluss von größeren Burstfehlern verringert werden, indem diese Störungen nicht mehr auf den Großteil eines einzelnen Frames einwirken, sondern nur noch auf einzelne Teile mehrerer Frames.

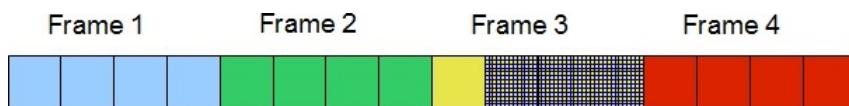


Abbildung 3.10.: Einfluss von Burstfehlern auf Daten ohne „long interleaving“

Wie in der Abbildung 3.10 zu erkennen würden die schraffiert dargestellten Bündelfehler bei einer Übertragung ohne „long interleaving“ einen starken Einfluss auf einzelne Frames nehmen, die im schlechtesten Fall vollständig nicht dekodierbar wären.

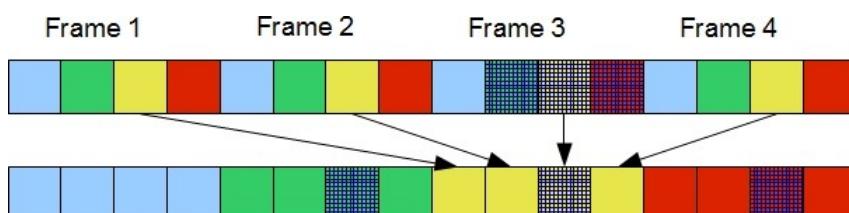


Abbildung 3.11.: Einfluss von Burstfehlern auf Daten mit „long interleaving“

Durch die Permutation der Daten über mehrere Frames hinweg kommt es wie in Abbildung 3.11 zu einer Aufteilung der Fehler auf verschiedene Frames. Hierdurch wirken sich die Fehler nicht mehr so stark aus, da in Kombination mit weiteren Fehlerschutzmaßnahmen, wie der Faltungskodierung oder dem Bit-Interleaving, kleinere Störungen auf Bitebene behoben werden können.

DRM benutzt zur Umsetzung des „long interleaving“ den sogenannten Faltungsinterleaver. Dieser besteht, wie in Abbildung 3.12 dargestellt, aus einer Reihe von Schieberegistern mit festen Verzögerungen, einem Multiplexer und einem Demultiplexer.

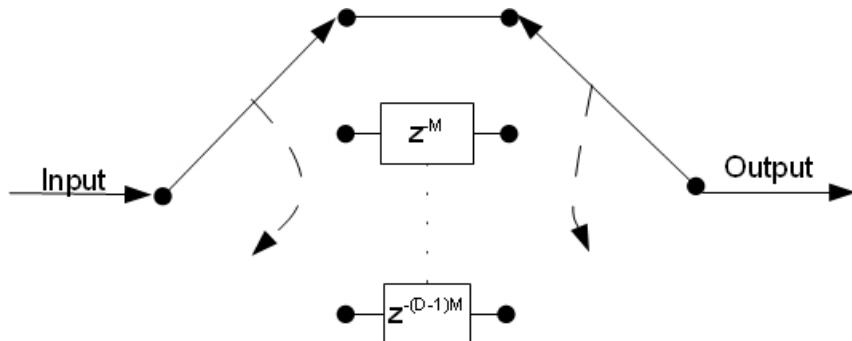


Abbildung 3.12.: Prinzip eines Faltungsinterleavers

D wird als Interleavingtiefe bezeichnet und stellt die Anzahl an Zweigen des Interleavers dar. Besteht ein durch das QAM-Mapping erzeugter Rahmen aus N Symbolen, so wird die Basisverzögerung des Faltungsinterleavers mit $M = \frac{N}{D}$ berechnet. Es werden somit pro Schritt jeweils am Eingang benachbarte Symbole um $D * M$ verzögert an den Ausgang weitergeleitet. Das bedeutet, dass ein Interleaving über $D - 1$ Frames durchgeführt wird.

Der Deinterleaver auf der Empfangsseite besteht aus der genau entgegengesetzten Struktur bezüglich Abbildung 3.12. D.h. die Verzögerung im obersten Zweig beträgt $(D - 1)M$ und im untersten Zweig 0. Zu beachten ist dabei, dass durch die Schieberegister im Deinterleaver eine feste Verzögerung des Ausgangssignals entsteht, die bei der Initialisierung des Empfängers berücksichtigt werden muss, da zunächst $D - 1$ Frames gesammelt werden müssen. [Rei08, S.173]

Die Interleavingtiefe ist bei DRM mit $D = 5$ für die Modi A-D und mit $D = 6$ für Mode E festgelegt und die Länge des verwendeten Cell-Interleavings wird dem Empfänger über den FAC signalisiert.

Die Algorithmen für das „short interleaving“ und das „long interleaving“ der MSC-QAM-Zellen befinden sich in der Spezifikation [Eur09, S.136].

3.2. OFDM

OFDM stellt eine spezielle Form des Frequenzmultiplex (FDM, engl.: Frequency Division Multiplex) dar. Dabei handelt es sich um ein Mehrträger-Modulationsverfahren, bei dem mehrere orthogonale Einzelträger zur Datenübertragung verwendet werden, um die Informationen statt auf einem Kanal in mehreren Subkanälen zu übertragen und eine Beeinflussung der Subträger untereinander zu vermeiden. Die Idee des Mehrträger-Verfahrens (MC, engl.: Multi Carrier) kam bereits in den fünfziger Jahren auf, konnte aber keinen entscheidenden Durchbruch erlangen. Auch durch die mathematische Formulierung der Schnellen Fourier-Transformation (FFT, engl.: Fast Fourier Transform) und den technologischen Fortschritt im Bereich Schaltungstechnik erlangte das Mehrträger-Verfahren keine große Popularität. Erst durch die Idee zur Digitalisierung des Hörfunks wurde mit DAB ein System entwickelt, welches ein OFDM-System standardisiert. „Heute stellt OFDM eine der wesentlichen Schlüssel-Technologien für die Mobilfunktechnik dar...“ [Kam04, S.580]. So wird diese Technik neben DAB auch bei DVB-T³, in neuen WLAN⁴-Systemen aber auch bei drahtgebundenen Systemen, wie ADSL⁵, genutzt.

3.2.1. Orthogonalität und das Prinzip von OFDM

Zwei Frequenzen f_m und f_n sind zueinander orthogonal wenn gilt:

$$\int_{-\infty}^{\infty} e^{-j2\pi f_m t} e^{-j2\pi f_n t} dt = \delta(m - n) \quad m, n \in \mathbb{N} \quad (3.1)$$

Bei OFDM wird der Datenstrom der komplexwertigen Symbole z , die durch das QAM-Mapping erzeugt wurden, in N zueinander orthogonale Frequenzen aufgeteilt. D.h. das Zeitsignal wird auf die Länge T_U begrenzt und in N Frequenzanteile f_k mit den Abständen $\Delta f_k = \frac{1}{T_U}$ zerlegt, so dass gilt:

$$f_k = k \cdot \Delta f_k = k \cdot \frac{1}{T_U} \quad 0 \leq k < N \quad k \in \mathbb{N} \quad (3.2)$$

Geht man von der Orthogonalitätsbedingung 3.1 aus und setzt die N gebildeten orthogonalen Frequenzteile aus 3.2 ein, so ergibt sich für die OFDM folgender Zusammenhang:

$$\int_0^{T_U} \prod_{k=0}^{N-1} e^{-j2\pi k \Delta f_k t} dt = 0 \quad k \in \mathbb{N} \quad (3.3)$$

Die Begrenzung des Zeitsignals auf die Symbollänge T_U kann als eine Fensterung bzw. eine Multiplikation mit einem Rechteck betrachtet werden, so dass das Zeitsignal wie folgt

³Terrestrial Digital Video Broadcasting

⁴Wireless Local Area Network

⁵Asymmetric Digital Subscriber Line

dargestellt werden kann:

$$s_{OFDM}(t) = \sum_{k=0}^{N-1} e^{-j2\pi f_k t} \cdot \text{rect}\left(\frac{t}{T_U}\right) \quad (3.4)$$

Betrachtet man das OFDM-Signal im Frequenzbereich, so ergibt sich folgende Gleichung:

$$S_{OFDM}(f) = \sum_{k=0}^{N-1} \delta(f - f_k) * T_u \frac{\sin(\pi f T_U)}{\pi f T_U} \quad (3.5)$$

Die Gleichung 3.5 bedeutet, dass das OFDM-Signal im Frequenzbereich als eine Faltung der si-Funktion mit N periodischen δ -Impulsen betrachtet werden kann. Dies soll in der folgenden Abbildung 3.13 für $N = 5$ veranschaulicht werden.

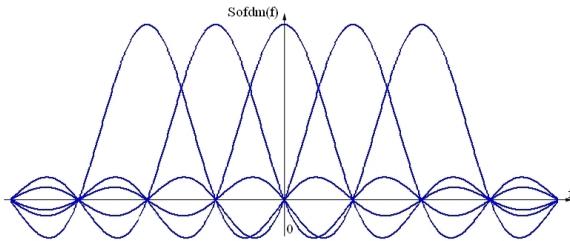


Abbildung 3.13.: Spektrum eines OFDM-Signals mit 5 Subträgern

Die Orthogonalität bewirkt, dass trotz der Überlagerung der N Träger die Information jedes Einzelnen erhalten bleibt, da bei jedem Maximum (Abtastwert) des jeweiligen Subträgers alle Anderen einen Nulldurchgang haben.

Zur Realisierung der Aufteilung der diskreten Daten in N orthogonale Subträger wird das Prinzip der IDFT (Inverse Diskrete Fourier Transformation), und dabei meist über die recheneffiziente IFFT (engl.: Inverse Fast Fourier Transform), angewandt. Als Anwendung für OFDM lässt sich die Gleichung der IFFT wie folgt ausdrücken:

$$s(l) = \sum_{k=0}^{N-1} z(k) e^{j2\pi \frac{kl}{N}} \text{ mit } l = 0, 1 \dots M-1 \quad (3.6)$$

$s(l)$ beschreibt das OFDM-Zeitsignal im Basisband, wie es durch die Erzeugung mittels IFFT gebildet wird. $z(k)$ stellt einen durch das QAM-Mapping generierten Konstellationspunkt dar. Es werden also N Konstellationspunkte auf N orthogonale Subträger mit dem Abstand k moduliert. Der Abstand k wird nach Gleichung 3.2 durch die Teilung des mit M begrenzten Signals mit der Anzahl der Subträger bestimmt.

Ein Überblick über den gesamten OFDM-Prozess im Sender und Empfänger soll durch die Abbildung 3.14 gegeben werden. Die Verwendung der Guardintervall-Blöcke wird im Abschnitt 3.2.3 erklärt.

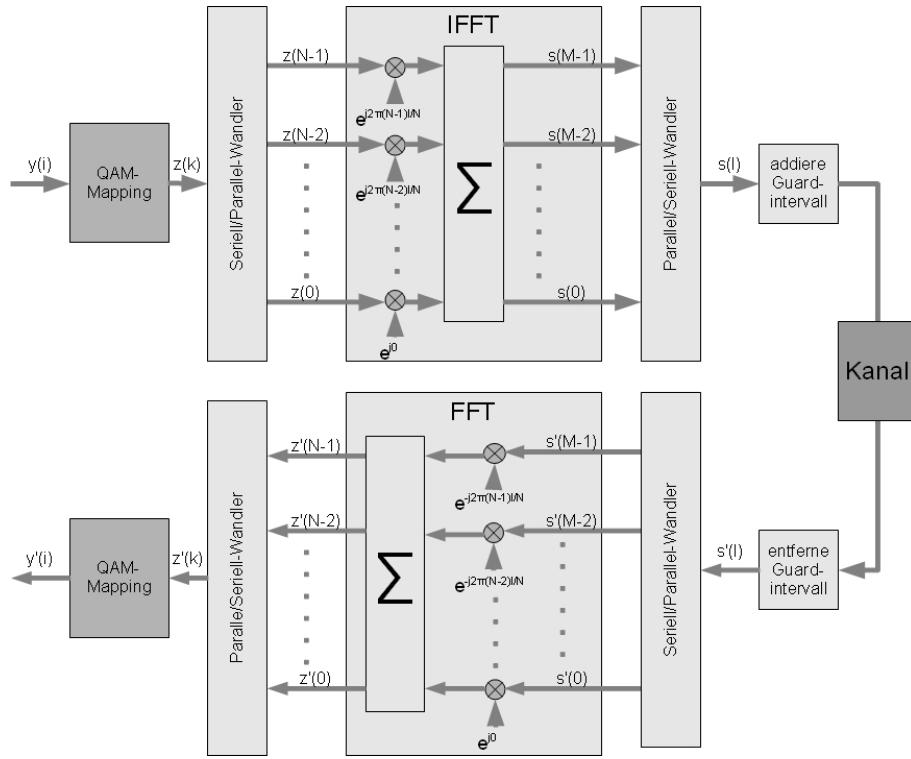


Abbildung 3.14.: Prinzip eines OFDM-Übertragungssystems im Basisband

Durch die QAM wird der binäre Datenstrom zunächst auf bestimmte Konstellationspunkte $z(k)$ verteilt. Es werden N dieser Konstellationspunkte aus dem seriellen Datenstrom ausgelesen und über die IFFT N zueinander orthogonalen Subträgern aufmoduliert. Die Summe dieser Subträger bildet dann ein OFDM-Symbol. Besteht der Rahmen, der vom QAM-Mapping erzeugt wurde, aus K Zellen, so werden diese zunächst in $M = \frac{K}{N}$ Anteile zerlegt. Dadurch entstehen am Ausgang der IFFT M OFDM-Symbole mit jeweils N Subträgern, wie in Abbildung 3.15 dargestellt.

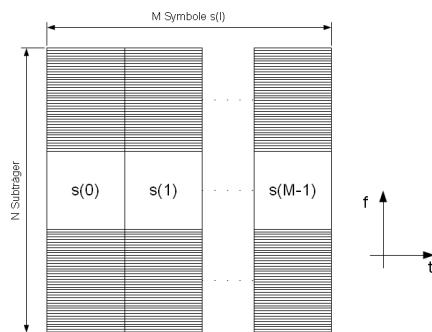


Abbildung 3.15.: Rahmenstruktur eines OFDM-Signals

Im Empfänger wird die FFT als inverse Transformation verwendet, um aus den M Symbolen wieder K Konstellationspunkte für das QAM-Demapping zu gewinnen.

3.2.2. Eigenschaften von OFDM

Die vielfältigen Anwendungen von OFDM beruhen auf den Vorteilen, die dieses Übertragungsprinzip gegenüber Einzelträgerverfahren bietet. Dabei geht man jedoch grundsätzlich von zwei Grundeigenschaften des Verfahrens aus.

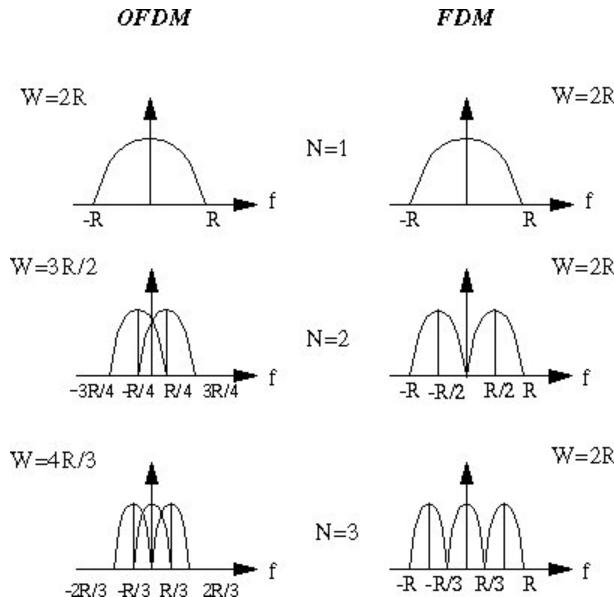


Abbildung 3.16.: Vergleich der Bandbreitennutzung zwischen OFDM und FDM [Met06]

Ein Ziel von OFDM ist es, die Ausnutzung der Bandbreite gegenüber Einzelträgerverfahren zu verbessern. Die Abbildung 3.16 zeigt den Vergleich zwischen dem Einzelträgerverfahren FDM und dem Mehrträgerverfahren OFDM. W gibt hierbei die genutzte Bandbreite und N die Anzahl der verwendeten Träger an. Da die einzelnen Subträger aufgrund der Orthogonalität näher aneinander liegen als die Einzelträger bei FDM, wird bei steigender Anzahl N eine immer bessere Ausnutzung des Spektralbereichs erzielt.

Ein weiteres Ziel ist es, Störungen, die durch frequenz- und zeitselektive Kanäle hervorgerufen werden, zu verringern. Eine Teilung eines Kanals in schmalbandigere Subkanäle hat zur Folge, dass die Kanalimpulsantwort $|h_a(t)|$ nur noch Einfluss auf die Länge eines Symbols besitzt. Im Gegensatz dazu wird beim Einträgerverfahren der Einfluss auf mehrere hintereinander folgende Symbole sichtbar, was starke Intersymbol-Interferenzen zur Folge hat. Die Abbildung 3.17 zeigt diesen Sachverhalt. Des Weiteren kann man erkennen, dass sich die Kanalübertragungsfunktion $H_a(j\omega)$ über die gesamte Bandbreite eines Einzelträgers erstreckt und hierdurch eine starke Frequenzselektivität zur Folge hat. Durch

die Aufteilung des Signals in mehrere Subträger kann die Übertragungsfunktion nun differenziert betrachtet und für jeden der Subträger als konstant und somit nicht mehr als selektiv angenommen werden.[Kam04, S.580]

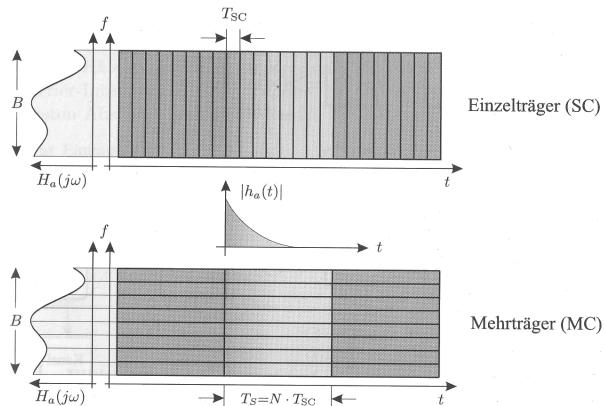


Abbildung 3.17.: Kanaleinflüsse bei Einzel- und Mehrträgerverfahren [Kam04, S.581]

Diese Ziele sind die Grundideen für die Umsetzung eines Übertragungssystems mit OFDM. Nachfolgend sollen die Vor- und Nachteile, die sich durch dieses Mehrträgerverfahren ergeben, zusammengefasst aufgelistet werden.[Hen08][Int00]

Vorteile von OFDM

- Durch die Parallelisierung und somit die Vergrößerung der Symboldauer werden ISI vermieden. Dies wird durch die Anwendung des im nächsten Abschnitt beschriebenen Guardintervalls noch verstärkt.
- Wegen der Orthogonalität kann eine Überlappung der Einzelspektren erfolgen, was eine effizientere Nutzung der zur Verfügung stehenden Bandbreite zur Folge hat.
- Schmalbandige Störungen im Frequenzbereich beeinflussen nur einzelne Subträger. In den meisten Fällen kann durch die vorher eingesetzte Kanalkodierung eine Rekonstruktion der Information erfolgen.
- Durch die Teilung des Kanals in mehrere Subkanäle kann im Empfänger eine leichtere Schätzung und Entzerrung des Übertragungskanals erfolgen, da die Algorithmen hierfür nur auf die einzelnen Subträger angewandt werden müssen.
- Es wird eine hohe spektrale Effizienz erzielt, da durch OFDM ein annähernd konstantes Leistungsdichtespektrum erreicht wird.

Nachteile von OFDM

- Verschiebungen bzw. Synchronitätsprobleme im Zeit- und Frequenzbereich führen zum Verlust der Orthogonalität der Subträger, d.h. ein OFDM-Signal reagiert sehr sensibel auf Zeit-, Frequenz und Phasenverschiebungen. Deshalb werden im Empfänger sehr hohe Ansprüche an die Synchronisation gestellt.
- Das OFDM Zeitsignal besitzt ein großes Verhältnis von Spitzenleistung zur mittleren Leistung (engl.: peak-to-average ratio). Dadurch wird ein hohes Maß an Linearität von analogen Übertragungskomponenten wie Verstärker und Modulator verlangt.
- Obwohl durch fortgeschrittene Technik eine effiziente Implementierung der FFT und der IFFT möglich ist, werden dennoch hohe Ansprüche an die Hardware gestellt.

3.2.3. Guardintervall

Eine OFDM-Übertragung besitzt durch die Aufteilung der Bandbreite in N Subträger und somit die Verlängerung der Symboldauer um N eine höhere Unempfindlichkeit gegenüber ISI. Trotzdem können Störungen bei Einfluss eines frequenzselektiven Kanals auftreten, da die Einschwingphase des Kanals Veränderungen an Amplitude und Phase am Anfang eines Symbols hervorrufen kann und damit die Orthogonalität zwischen den Subträgern zerstört, was ICI zur Folge hat. Des Weiteren bewirkt die Ausschwingphase des Kanals eine Wechselwirkung des aktuellen Symbols mit dem Nachfolgenden und erzeugt somit ISI. [Kam04, S.586]

Um diese Effekte zu vermeiden, wird dem OFDM-Symbol ein Schutzintervall bzw. Guardintervall (engl.: guard interval) hinzugefügt.

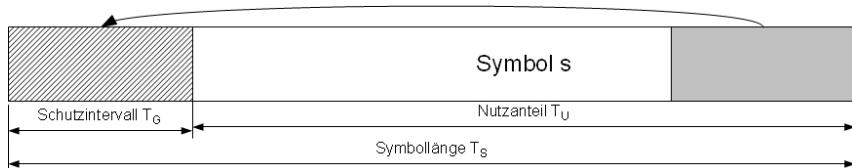


Abbildung 3.18.: Hinzufügen des Schutzintervalls eines OFDM-Symbols

Hierzu wird eine Kopie der letzten Abtastwerte des ursprünglichen Symbols diesem vorangestellt. Dieses Schutzintervall mit der zeitlichen Länge T_G wird auch als zyklische Erweiterung (engl.: Cyclic Prefix) bezeichnet. Die Abbildung 3.18 zeigt den Vorgang des Hinzufügens des Guardintervalls. Das ursprüngliche Symbol mit der Dauer T_U wird als Nutzanteil bezeichnet und die neue Symboldauer beträgt

$$T_S = T_U + T_G$$

D.h. die Anzahl der Abtastwerte in einem Symbol wird von N_U um die Abtastwerte des Schutzintervalls N_G auf

$$N_S = N_U + N_G$$

vergrößert.

Durch diese Erweiterung des Symbols fällt der Einschwing- und der Ausschwingvorgang des Kanals in das Schutzintervall und besitzt hierdurch keinen Einfluss mehr auf die eigentlichen Nutzdaten. Bei optimaler Wahl der Länge der zyklischen Erweiterung können damit ISI und ICI vollständig vermieden werden.[Kam04]

Die Länge des Schutzintervalls wird nach folgenden Kriterien ausgewählt [Roc07, S.92]:

- Es sollte so lang wie nötig sein, um den einflussreichen Anteil der Impulsantwort des zu erwartenden Kanals zu umgehen und somit eine Beeinflussung angrenzender Symbole und damit ISI und ICI zu verhindern.
- Es sollte so kurz wie möglich sein, da die Energie des Signals mit steigender Symbollänge wächst und damit eine Verringerung des SNR zur Folge hat.

3.2.4. OFDM-Struktur bei DRM

Im DRM-System wird die OFDM auch als COFDM (Coded Orthogonal Frequency Division Multiplex) bezeichnet, da eine Kombination des Mehrträgerverfahrens mit einer vorherigen FEC der Daten Verwendung findet.

Die Parameter der OFDM bei DRM sind in folgender Tabelle aufgelistet. Es werden die Länge der Nutzdaten eines Symbols T_U , die Länge des Schutzintervalls T_G und die Gesamtlänge T_S in Millisekunden angegeben. Der Trägerabstand Δf , welcher sich aus der Gleichung 3.2 zu $\frac{1}{T_U}$ bildet, ist in Hz angegeben.

Übertragungsmodus	T_U	Δf	T_G	T_S	Symbole/Rahmen
A	24	$41^{2/3}$	2,66	26,66	15
B	21,33	$46^{7/8}$	5,33	26,66	15
C	14,66	$68^{2/11}$	5,33	20	20
D	9,33	$107^{1/7}$	7,33	16,66	24
E	2,25	$444^{4/9}$	0,25	2,5	40

Tabelle 3.4.: OFDM Parameter bei DRM

Durch die unterschiedlichen Kanalbedingungen, die an jeden Übertragungsmodus geknüpft sind, werden die Schutzlevel im Verhältnis zum Nutzanteil auch immer größer, so dass die frequenz- und zeitselektiven Eigenschaften der Kanäle möglichst wenig Einfluss haben.

Will man aus der zeitlichen Länge eines Symbols die Anzahl der Subträger berechnen, so muss die Abtastfrequenz des DRM-Systems im Basisband bekannt sein bzw. legt man die mögliche Bandbreite des Nutzanteils fest, so kann daraus die Anzahl der Subträger bzw. die Anzahl der für die IFFT benötigten Abtastwerte berechnet werden. Die folgenden Gleichungen geben den Zusammenhang zwischen den zeitlichen Längen eines OFDM-Symbols und der Anzahl der möglichen Subträger bei Wahl der Abtastfrequenz f_s an.

$$N_U = T_U \cdot f_s \quad (3.7)$$

$$N_G = T_G \cdot f_s \quad (3.8)$$

$$N_S = N_U + N_G \quad (3.9)$$

N_U bestimmt die Anzahl der Abtastwerte im Nutzanteil, N_G die Anzahl der Abtastwerte im Schutzintervall und N_S die Anzahl der Abtastwerte eines gesamten OFDM-Symbols.

Die Abtastfrequenz des Systems im Basisband wird durch die in Tabelle 2.4 dargestellten Spektralbelegungen bestimmt. Um die Nyquistbedingung einzuhalten, muss die Abtastfrequenz f_s mindestens doppelt so groß sein wie der höchste verwendete Spektralan teil. Um eine Verarbeitung des Systems nach dem Prinzip des SDR allgemein zu halten, werden für Soundkarten bzw. AD/DA-Wandler gängige Abtastraten verwendet. Bei den Spektralbelegungen $B_S = 0, 1, 2$ und 3 wird hierzu eine Frequenz von 12 kHz gewählt. Da bei $B_S > 3$ das zweiseitige Spektrum ([Eur09, S.141]) über einen Anteil > 12 kHz verfügt, wird für diesen Fall eine Abtastfrequenz von 48 kHz gewählt. Die beiden folgenden Tabellen geben die Anzahl der möglichen Subträger für die gewählten Abtastfrequenzen an.

Mode	N_U	N_G	N_S
A	288	32	320
B	256	64	320
C	176	64	240
D	112	88	200

Tabelle 3.5.: Anzahl der Subträger bei einer Abtastfrequenz von 12 kHz ($B_S = 0, 1, 2, 3$)

Mode	N_U	N_G	N_S
A	1152	128	1280
B	1024	256	1280
C	704	256	960
D	448	352	800

Tabelle 3.6.: Anzahl der Subträger bei einer Abtastfrequenz von 48 kHz ($B_S = 4, 5$)

3. Algorithmen des DRM Standards

Beim Mode E, der eine Bandbreite von 100 kHz besitzt, wird für die Simulation eine Abtastfrequenz von 192 kHz festgelegt. Dies entspricht einer Trägeranzahl im Nutzanteil N_U von 432.

Es wird in unterschiedlicher Literatur auch von einer festen Abtastfrequenz von 48 kHz für die Modi A-D ausgegangen (z.B. [CPWO06]). Dies ist prinzipiell möglich, führt jedoch zu einer höheren Anforderung an das Basisbandsystem auf einer Zielplattform.

Die Berechnung der Subträger kann als die theoretisch zur Verfügung stehende Anzahl dieser betrachtet werden. Tatsächlich wird beim DRM-System nur eine begrenzte Auswahl der möglichen Subträger zur Übertragung genutzt.

Ausgehend vom Träger mit der Frequenz 0 (DC-Träger) werden die Träger mit einer Indizierung k auf die möglichen Bandbreiten der einzelnen B_S aufgeteilt. Träger mit negativer Indizierung $k < 0$ sind definiert als links vom DC-Träger ($k = 0$) und Träger mit $k > 0$ als rechts vom DC-Träger. Mode E verwendet dabei alle Träger, während die Modi A-D den DC-Träger ungenutzt lassen. Im Mode A werden zusätzlich die Träger mit $k = -1$ und $k = 1$ nicht verwendet. [Eur09, S.142]

Robustness mode	Carrier	Spectrum occupancy					
		0	1	2	3	4	5
A	K_{min}	2	2	-102	-114	-98	-110
	K_{max}	102	114	102	114	314	350
B	K_{min}	1	1	-91	-103	-87	-99
	K_{max}	91	103	91	103	279	311
C	K_{min}	-	-	-	-69	-	-67
	K_{max}	-	-	-	69	-	213
D	K_{min}	-	-	-	-44	-	-43
	K_{max}	-	-	-	44	-	135
E	K_{min}	-106	-	-	-	-	-
	K_{max}	106	-	-	-	-	-

Abbildung 3.19.: Verwendete Trägernummern für die Modi bei einer bestimmten Spektralbelegung B_S [Eur09, Tab.84]

Die Abbildung 3.19 zeigt als Tabelle den Bereich der verwendeten Trägernummern für jeden Mode bei unterschiedlichen Spektralbelegungen. Sie gibt den Bereich zwischen der minimalen Trägernummer K_{min} und der maximalen Trägernummer K_{max} an. Die Gesamtanzahl der Träger wird mit K angegeben und wird wie folgt definiert:

$$K = -K_{min} + K_{max} \text{ für alle Kombinationen mit } K_{min} < 0 \quad (3.10)$$

$$K = K_{max} \text{ für alle Kombinationen mit } K_{min} > 0 \quad (3.11)$$

Es ist zudem in der Tabelle zu erkennen, dass für einige Modi nicht alle Bandbreiten definiert sind.

Die durch die QAM erzeugten Zellen aller Kanäle werden auf alle OFDM-Symbole mit K Trägern nach der in der Abbildung 2.8 dargestellten Rahmenstruktur in Frames und Superframes in einer bestimmten Anordnung zugewiesen. Diese Anordnung wird bei der Umsetzung der Simulation 5.3.3.2 beschrieben. Zusätzlich werden die im folgenden Abschnitt dargestellten Pilotzellen auf feste Trägernummern im Rahmen verteilt.

Da für die IFFT nach Gleichung 3.6 alle Träger mit $k > 0$ berechnet werden, müssen die Träger mit negativem Index vor der Transformation zunächst in den positiven Bereich über folgende Gleichung verschoben werden. [Hen08, S.41]

$$k = k + N_U \text{ für } k < 0 \quad (3.12)$$

3.3. Pilotstruktur

Zusätzlich zu den durch die QAM erzeugten Datenzellen des MSC und die Kontrollzellen von FAC und SDC werden in einem Übertragungsrahmen Referenzzellen eingefügt. Diese sogenannten Pilotzellen dienen dem Empfänger als Referenzsignale, da sie im Sender auf festen Trägern und mit bekannten Amplituden und Phasen generiert werden.

Ein Pilot ist also eine bestimmte OFDM-Zelle, deren Position innerhalb eines Übertragungsrahmens bekannt ist. Der DRM-Standard definiert die Referenzzellen wie folgt:

$$p_{s,k} = a_{s,k} e^{j2\pi\vartheta_{s,k}} \quad (3.13)$$

s gibt die Symbolnummer an und k ist die Trägernummer. a stellt den Betrag und ϑ die definierte Phase des Piloten dar, welche unter [Eur09, S.142 ff.] dargestellt werden.

Man unterscheidet vier Typen von Pilotzellen, die aufgrund ihrer Anordnung im Übertragungsrahmen bzw. ihrer Positionen für unterschiedliche Aufgaben vorgesehen sind.

Frequenz-Referenzzellen werden in den Modi A bis D genutzt, um Frequenzverschiebungen im Empfänger festzustellen. Hierfür werden genau drei Trägerpositionen pro Symbol festgelegt, so dass diese je nach Mode exakt den Frequenzen 750 Hz, 2250 Hz und 3000 Hz entsprechen. Sie verlaufen also über den gesamten Übertragungsrahmen geradlinig und parallel. Da die Frequenzen im gesamten System bekannt sind, kann ein Frequenzoffset im Empfänger auch schon vor der FFT festgestellt werden. Die Tabelle 3.7 gibt die Trägernummer der Frequenzpiloten für die unterschiedlichen Modi an.

Übertragungsmode	Trägernummer k
A	18,54,72
B	16,48,64
C	11,33,44
D	7,21,28

Tabelle 3.7.: Trägernummer der Frequenzpiloten

Für den Mode E sind keine Frequenzpiloten vorgesehen. Der Betrag ist für alle Frequenzreferenzen mit $a = \sqrt{2}$ festgelegt.

Zeit-Referenzzellen sollen dem Empfänger zur Synchronisation des Rahmens und zur Erkennung des Zeitoffsets dienen. Hierzu werden mehrere Träger ausschließlich im ersten Symbol ($s=0$) eines Übertragungsrahmens als fest definiert. Durch die meist als paarweise gebildete Lage der Zeitpiloten kann mit Hilfe von Korrelationsmethoden eine Verschiebung im Zeitbereich bzw. die Lage des ersten Symbols erkannt werden. Zusätzlich können die Zeitpiloten aber auch zur Unterstützung der Frequenzpiloten für die Erkennung eines Frequenzoffsets angewandt werden. Auch bei den Zeitreferenzen ist der Betrag mit $a = \sqrt{2}$ festgelegt.

Gain-Referenzzellen dienen der Kanalschätzung. Durch ihre rasterförmige, regelmäßige Anordnung liefern die Gainpiloten Messwerte zu Betrag und Phase innerhalb eines gesamten Rahmens. Mit Hilfe der Messwerte können durch geeignete Interpolationsmethoden Rückschlüsse auf die eingeschlossenen Datenzellen gezogen und somit die Übertragungsfunktion und die Impulsantwort des Kanals geschätzt werden.

Die Anzahl der verwendeten Gainreferenzen richtet sich dabei stark nach dem verwendeten Übertragungsmodus. Wie in Abschnitt 2.2.2 beschrieben, sollen durch die verschiedenen Modi unterschiedlich starke Kanaleinflüsse betrachtet werden. Umso engmaschiger das Raster ist, desto genauer kann die Schätzung des Kanals erfolgen, da eine geringere Interpolation durchgeführt werden muss. Eine genaue Beschreibung der Bedingungen für die Lage der Piloten zur exakten Schätzung des Kanals erfolgt in [Chi04, S.81] und [Hen08, S.108].

Übertragungsmode	Abstand in Zeitrichtung d_s	Abstand in Frequenzrichtung d_k
A	5	20
B	3	6
C	2	4
D	3	3
E	4	16

Tabelle 3.8.: Abstände der Gain-Referenzzellen

Die Tabelle 3.8 stellt die Abstände der Gainpiloten in Zeit- und Frequenzrichtung für die verschiedenen Modi dar. Es wird ersichtlich, dass bei Mode A, der für wenig Frequenzselektivität vorgesehen ist, ein großer Trägerabstand von $d_k = 20$, während bei Mode D, definiert für große Zeit- und Frequenzselektivität, ein kleiner Abstand von $d_k = 3$ festgelegt wird.

Grundsätzlich werden auch die Gain-Referenzzellen mit einem Betrag von $a = \sqrt{2}$ definiert. Jedoch besitzen die Referenzzellen an den Bandgrenzen einen verstärkten (over-boosted) Betrag von $a = 2$.

AFS-Referenzzellen werden nur für den Mode E definiert. AFS-Piloten befinden sich im fünften Symbol ($s=4$) des ersten Frames und im vierzigsten Symbol ($s=39$) des vierten Frames eines Superframes. Ihre Hauptaufgabe besteht in der Optimierung der Kanalschätzung bei der Frequenzumschaltung und der verbesserten Prüfung zur Frequenzumschaltung auf Alternativfrequenzen. So gewährleisten die AFS-Zellen bei $s=39$ im vierten Frame die Kanalschätzung des aktuellen Frequenzbandes abzuschließen, bevor auf eine Alternativfrequenz gewechselt wird. Hingegen können die AFS-Zellen bei $s=4$ im ersten Frame als Teil der SDC-Daten betrachtet werden und verbessern die Kanalschätzung bei der Rückkehr zur Originalfrequenz.[Eur09, S.150]

Da in der Simulation keine Synchronisations- und Kanalschätzungsalgorithmen für den Mode E umgesetzt werden, spielen die AFS-Referenzzellen in der vorliegenden Arbeit keine weitere Rolle.

Beispiel zur Pilotstruktur

Die folgende Abbildung 3.20 zeigt einen Übertragungsrahmen mit der hinzugefügten Pilotstruktur. Zur Darstellung wurde der Übertragungsmodus B bei einer Spektralbelegung von $B_S = 0$ gewählt, da hier die kleinstmögliche Anzahl an OFDM-Zellen besteht.

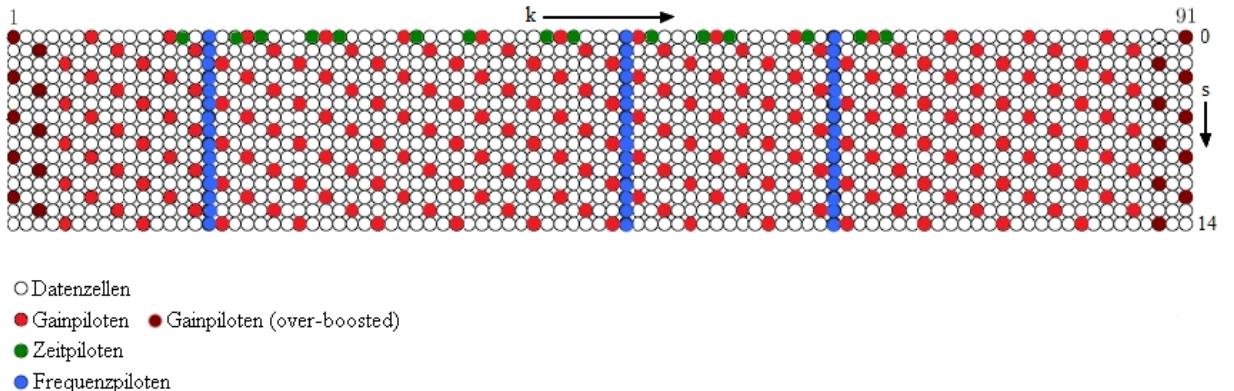


Abbildung 3.20.: Pilotstruktur für Mode B mit der Spektralbelegung $B_S=0$

Innerhalb des dargestellten Übertragungsrahmens, der aus 15 Symbolen à 91 Trägern besteht, sind die Positionen der zuvor erläuterten Referenzzellen ersichtlich. Man sieht, dass der Anteil der Piloten an der Gesamtzahl von OFDM-Zellen relativ hoch ist. Der Grund hierfür ist zum einen die hohe gewünschte Übertragungreichweite und zum anderen

die angesprochene Empfindlichkeit eines OFDM-Systems gegenüber Synchronisationsproblemen. Weitere Pilotenstrukturen werden in [Eur09, S.200 ff.] dargestellt.

3.4. Zwischenfrequenz-Verarbeitung

Um die durch die OFDM erzeugte Rahmenstruktur mittels Funkwellen zu übertragen, muss das Signal auf die entsprechenden im Abschnitt 2.1.2 gezeigten Übertragungsfrequenzen moduliert werden. Im Laufe der Zeit entwickelte sich auch bereits bei der analogen Rundfunkübertragung ein Konzept⁶, was die Verarbeitung im Empfänger vereinfacht, in dem jegliche Empfangsfrequenzen zunächst auf eine feste, meist niedrigere Frequenz demoduliert werden und somit danach ein konstantes Filter einsetzen zu können. Die Zwischenfrequenz (ZF) (engl.: Intermediate frequency (IF)) ist also die im Empfänger als fest eingestellte Frequenz, auf die das Empfangssignal herunter gemischt wird, um eine anschließende Verstärkung, Filterung und Demodulation auf dieser festen Frequenz durchzuführen. [Kam04, S.30]

Da das DRM-System auf genau den Frequenzen des analogen Hörrundfunks arbeiten soll und das in 2.2 besprochenen Prinzip des SDR meist mehrere Empfangssysteme kombiniert, wird in den DRM-Empfangsgeräten prinzipiell eine Zwischenfrequenz-Verarbeitung durchgeführt.

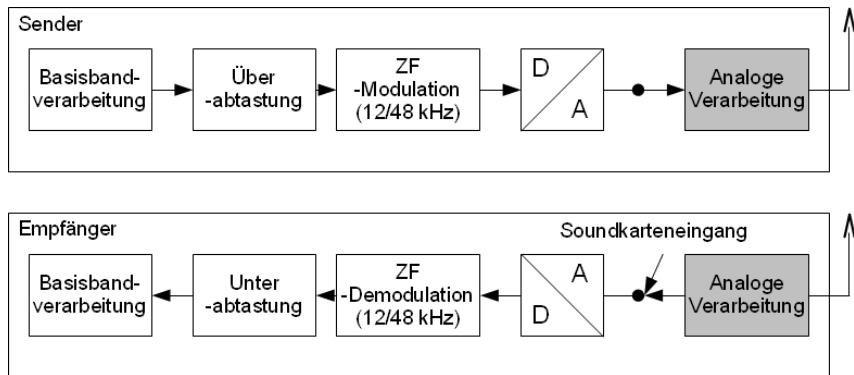


Abbildung 3.21.: Übersicht zur Zwischenfrequenz-Verarbeitung bei DRM

Die Abbildung 3.21 stellt grob das in einem SDR-System verwendete Verfahren der ZF-Modulation und ZF-Demodulation dar. Die in den vorherigen Abschnitten beschriebenen Verfahren inklusive der OFDM werden im Basisband durchgeführt, d.h. das Spektrum des Signals befindet sich in seiner Lage um den Gleichanteil ($f = 0$ Hz) herum und wird mit einer bestimmten Grundabtastfrequenz (siehe 3.2.4) verarbeitet. Bevor im Sender auf die Zwischenfrequenz moduliert werden kann, muss die Abtastfrequenz des Signals erhöht

⁶Überlagerungsempfänger bzw. Superheterodynempfänger

werden, um das Abtasttheorem einzuhalten. Bei der ZF-Modulation wird das Spektrum des Signals aus dem Basisband auf eine höhere Frequenz gebracht, um es anschließend nach der D/A-Wandlung den analogen Komponenten des Übertragungssystems zuzuführen. In der analogen Verarbeitung wird das Zwischenfrequenzsignal dann entweder direkt auf die Übertragungsfrequenz gebracht oder nochmals auf eine höhere Zwischenfrequenz gemischt und anschließend an die Antenne geführt. Im Empfänger werden diese Schritte genau umgedreht durchgeführt.

Die Zwischenfrequenzen werden nicht in der DRM-Spezifikation definiert, weshalb bei der Umsetzung eine Orientierung an den in Abschnitt 5.4 genannten existierenden Tools für den DRM-Empfang stattfand. Dazu wurden für die Simulation folgende ZF festgelegt.

Übertragungsmode	ZF in Hz
A - D	12000
E	192000

Tabelle 3.9.: Für die Simulation gewählte Zwischenfrequenzen

3.4.1. Über-/Unterabtastung

Um das Basisbandsignal auf die ZF zu modulieren, muss sichergestellt werden, dass das Spektrum des Signals auf der höheren Frequenz noch immer das Abtasttheorem erfüllt. Da dies bei einer direkten Modulation des Basisbandsignals nicht erfüllt wäre, muss zunächst die Abtastfrequenz erhöht werden. Im folgenden soll ein kurzer Überblick über das Prinzip der Überabtastung und Unterabtastung gegeben werden. Eine ausführlichere Betrachtung des Themas ist in [PM06] und [AS06] zu finden.

Die Überabtastung wird durch eine Interpolation und einer anschließenden Filterung des Signals realisiert. Es wird die Zahl der Abtastwerte in einem Intervall um I vervielfacht, in dem I-1 Nullen zwischen den ursprünglichen Abtastwerten eingefügt werden.

Die Abbildung 3.22 zeigt als Beispiel die Überabtastung mit I=2 im Frequenzbereich.

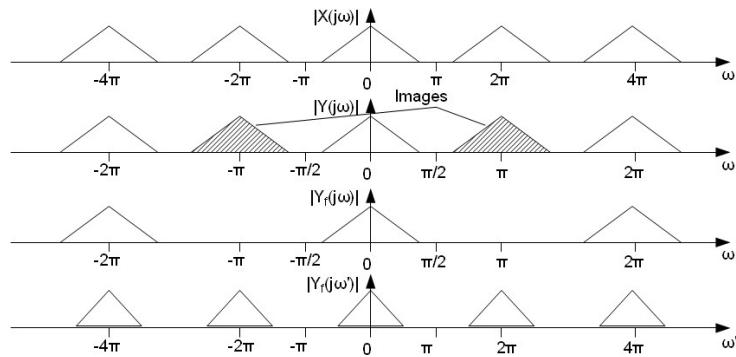


Abbildung 3.22.: Betrachtung der Überabtastung im Frequenzbereich

Das ursprüngliche Spektrum wird als $X(j\omega)$ dargestellt. Durch die Interpolation mit $I=2$ kommt es zu einer Stauchung des Spektrums, so dass das $Y(j\omega) = X(j\omega/I)$ ist. Die dabei auftretenden Spektralanteile, die nun um die Stellen $\omega = 2\pi/I$ auftreten, werden als Images bezeichnet und müssen mit einem entsprechenden Filter aus dem Spektrum entfernt werden. Vergleicht man $X(j\omega)$ und $Y(j\omega')$ so wird sichtbar, dass die normierte Bandbreite um den Faktor I verringert wurde.

Im Empfänger muss das überabgetastete Signal wieder in das ursprüngliche Spektrum zurückgeführt werden. Hierzu findet eine Unterabtastung als entgegengesetztes Prinzip statt. Um eine Überlappung der Spektralanteile zu vermeiden, die durch eine Streckung des Spektrums auftreten könnten, wird zunächst eine Filterung zur Vermeidung von Aliasing durchgeführt. Anschließend findet eine Dezimation um den Faktor $D = I$ statt, indem nur noch jeder Wert $y(n) = x(n/D)$ von den Ausgangswerten $x(n)$ genommen wird, wenn $n \in \mathbb{Z}$.

Die verwendeten Faktoren und der Filtertyp für die Umsetzung der Abtastratenkonverterung werden in der Beschreibung der Simulation 5.3.4 näher erläutert.

3.4.2. IQ-Modulator/Demodulator

Das nun mit einer höheren Abtastfrequenz vorhandene Basisbandsignal muss auf die Zwischenfrequenz moduliert werden, bevor es den analogen Komponenten des Übertragungssystems zugeführt wird.

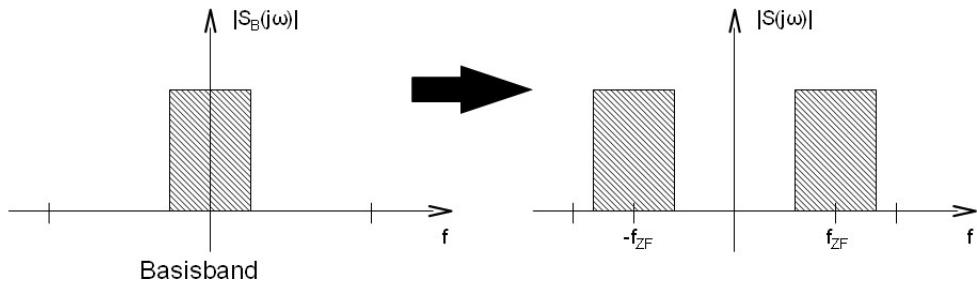


Abbildung 3.23.: Spektren bei der ZF-Modulation

Anschaulich dargestellt wird das Prinzip in Abbildung 3.23, welche lediglich zeigen soll, dass das Signal aus dem Basisband auf eine höhere Frequenz moduliert wird.

Ein für SDR-Systeme verwendetes und in der Simulation umgesetztes Verfahren der ZF-Modulation ist die IQ-Modulation. Die Prinzipschaltbilder der IQ-Modulation und Demodulation sind in Abbildung 3.24 dargestellt. IQ steht hierbei für die beiden Anteile des komplexen Signals, welche separat mit einem um 90° verschobenen Trägersignal moduliert und anschließend als reelle Signale zusammengefasst werden. Das komplexwertige diskrete Basisbandsignal $s_b(n)$ wird zunächst in Real- und Imaginärteil getrennt. Der I-Anteil (richtigphasig, engl.: Inphase) repräsentiert den Realteil des Signals und wird mit

3. Algorithmen des DRM Standards

dem reellen Kosinusträger multipliziert. Der Q-Anteil (quadratur- oder querphasig, engl.: Quadrature) stellt den Imaginäranteil dar und wird mit einem im Spektrum komplexwertigen Sinussignal multipliziert. Durch die Differenzbildung der beiden auf die ZF (f_{ZF}) modulierten Anteile des Signals wird das zur Übertragung benötigte reellwertige Signal $s(n)$ gebildet.

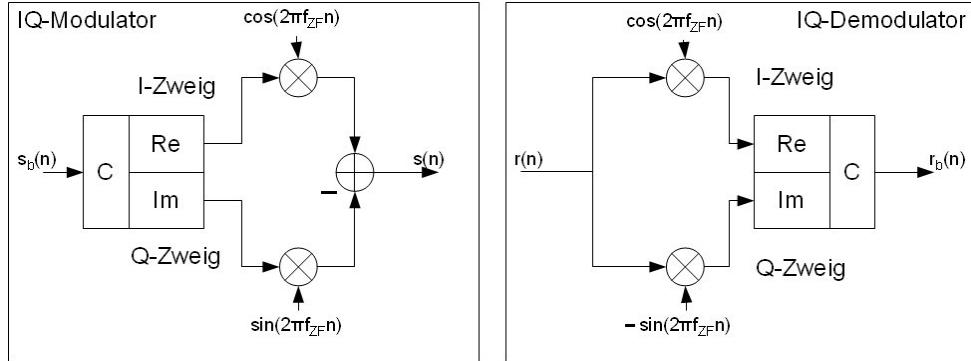


Abbildung 3.24.: Prinzip eines IQ-Modulators und IQ-Demodulators

Der Prozess der IQ-Modulation lässt sich über die folgende Gleichung ausdrücken.

$$s(n) = \operatorname{Re}\{s_b(n)\} \cos(2\pi f_{ZF}n) - \operatorname{Im}\{s_b(n)\} \sin(2\pi f_{ZF}n) = \operatorname{Re}\{s_b(n)e^{j2\pi f_{ZF}n}\} \quad (3.14)$$

Im Empfänger wird das umgedrehte Prinzip (siehe Abbildung 3.24) angewandt, so dass zunächst das reelle Signal $r(n)$ wieder mit den beiden um 90° verschobenen Trägerschwingungen multipliziert wird und anschließend durch die Zusammenfassung von I- und Q-Anteil das komplexe Basisbandsignal $r_b(n)$ zurückgewonnen wird.

4. Empfängeralgorithmen zur Synchronisation und Kanalkompensation

Durch die Eigenschaften des in Abschnitt 2.1.4 beschriebenen Übertragungskanals kommt das Empfangssignal gestört beim Empfänger an. So treten bei einem realen Übertragungssystem Veränderungen des gesendeten Signals in Frequenz und Phase auf. Zudem kommt es bei jeder Übertragung zu Laufzeiten, die dem Empfänger bekannt sein müssen, um die Rahmenstruktur der OFDM zur Dekodierung einzuhalten. Um eine Demodulation des Empfangssignals trotz dieser Einflüsse mit möglichst geringer Fehlerrate zu realisieren, müssen diese Verschiebungen des Signals in Zeit, Frequenz und Phase optimal geschätzt und kompensiert werden. Das folgende Kapitel soll die verschiedenen Mechanismen zur Synchronisation und Kanalschätzung vorstellen und ihre Notwendigkeit erläutern. Dabei soll ein Überblick über die verschiedenen Ansätze gegeben werden und eine Spezifizierung für die in der Simulation umgesetzten Algorithmen erfolgen. Die folgende Beschreibung der Synchronisationsalgorithmen und des Konzepts zur Umsetzung in einem Empfänger sind hier zwar speziell für DRM erläutert, können jedoch in angepasster Form auch für andere auf OFDM basierte Systeme mit Pilotzellen (wie z.B. DVB) übertragen werden.

Grundsätzlich lässt sich eine Empfängerstruktur mit den Synchronisationsalgorithmen wie in Abbildung 4.1 darstellen.

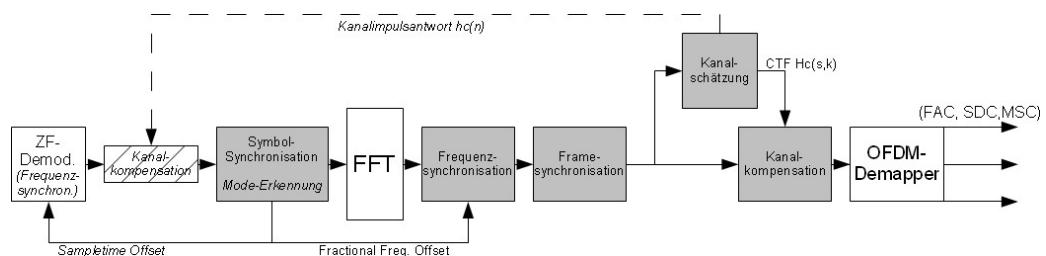


Abbildung 4.1.: Mögliche Empfängerstruktur mit Synchronisationsalgorithmen

Dabei wird ersichtlich, dass bestimmte Funktionen vor und manche nach der FFT oder z.T. auch kombiniert stattfinden. Zudem können auch unterschiedlichste Konzepte der Methoden zur Synchronisation verwendet werden, so dass die Abbildung auch optionale bzw. alternative Anordnungen zeigt. Die grau dargestellten Algorithmen zur Zeit- und Frequenzsynchronisation, sowie die Kanalschätzung und -kompensation stellen die in der Simulation verwendete Reihenfolge dar und werden in den folgenden Abschnitten beschrieben.

Die Detektion des Modes wurde in dieser Arbeit nicht betrachtet und umgesetzt, da das Simulationsmodell mit einem vom Nutzer vorgegebenen Parametersatz (Mode, B_S , u.a.) arbeiten soll und damit diese dem Empfänger von vorn herein bekannt sind. Zudem wurde keine Umsetzung der Samplerate-Korrektur vorgenommen. Um einen eigenständigen Empfänger basierend auf dem Simulationsmodell zu entwickeln, wären diese Komponenten jedoch notwendig, um beliebige DRM-Signale zu empfangen.

4.1. Zeitsynchronisation

Durch diverse Verzögerungen im Übertragungskanal und bei der Verarbeitung im Sender und Empfänger kommt es zu einer Verschiebung einzelner Träger aber auch ganzer Symbole innerhalb der OFDM-Rahmenstruktur. Die Zeitsynchronisation versucht im Empfänger diese Verzögerungen zu erkennen und die Struktur wieder so herzustellen, dass eine problemlose Dekodierung der Daten ermöglicht wird. Dazu wird die Zeitsynchronisation in zwei Teile mit unterschiedlichen Aufgaben eingeteilt.

Ein Teil ist die Symbolsynchronisation. Sie wird verwendet, um dem Empfänger exakt den Beginn, also den ersten Subträger, eines Symbols anzuzeigen. Sollte eine Verschiebung um einzelne Abtastwerte aufgetreten sein, so würden bei der FFT Teile von zwei Symbolen anliegen, was zu einer falschen Rücktransformation führt. Das Prinzip besteht darin, die Empfangsdaten so zu synchronisieren, dass exakt alle N_U Subträger genau eines Symbols für die FFT verwendet werden.

Der zweite Teil der Zeitsynchronisation dient der Synchronisation der Framestruktur. Durch größere Verzögerungen während der Übertragung, aber auch im Empfänger selbst, kommt es nach der FFT zu einer Unkenntnis über die Anordnung der s Symbole in einem Frame. Die Framesynchronisation versucht deshalb die Lage des ersten Symbols ($s=0$) innerhalb eines Rahmens zu erkennen und dies so auszugleichen, dass vor dem OFDM-Demapping eine richtige Struktur des Superframes vorliegt, damit die drei Kanäle wieder exakt in ihre Einzeldaten getrennt werden können.

4.1.1. Symbolsynchronisation

Die Synchronisation der Träger innerhalb der Symbole kann als der kompliziertere und wesentlich komplexere Teil der Zeitsynchronisation betrachtet werden. Da vor der FFT eine Anordnung der Symbole bzw. der Beginn eines Symbols genau bekannt sein muss aber die Referenzzellen noch nicht wieder in ihrer ursprünglichen, vom Sender eingestellten, Struktur und Form vorliegen, basiert das Prinzip der Symbolsynchronisation auf der Auswertung der Leistungsverteilung des OFMD-Zeitsignals.

Die verschiedenen Ausarbeitungen zur Symbolsynchronisation ([BSB97], [CPWO06] und [LLT04]) arbeiten alle nach dem gleichen Grundprinzip und unterscheiden sich nur in diversen Parametern oder Ergänzungen.

Der Ansatz besteht darin innerhalb eines bestimmten Betrachtungsfensters eine Maximierung bei der Korrelation eines OFDM-Symbols mit dem um N_U -Werte verzögerten Symbol zu erkennen. Dies wird durch die Verwendung des Schutzintervalls (Abschnitt 3.2.3) ermöglicht, was eine Kopie der letzten N_G Werte eines Symbols diesem voranstellt.

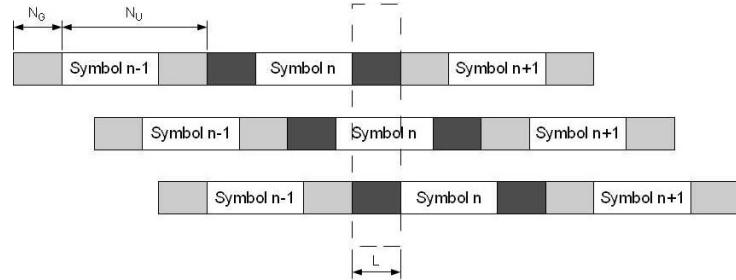


Abbildung 4.2.: Symbole mit Schutzintervall im Betrachtungsfenster

Die Abbildung 4.2 zeigt das Grundprinzip zur Erkennung des Symbolanfangs. Die Länge des Detektionsfensters L kann je nach Empfindlichkeit gewählt werden [CPWO06], wird aber meist mit der Länge des Schutzintervalls N_G gleich gesetzt. Durch Variation der Fensterlänge entsprechend der unterschiedlichen Größen der OFDM-Parameter N_G und N_U (siehe Tabelle 3.5) wird dieses Prinzip ebenfalls zur Detektion des Modes genutzt [CPWO06], was in dieser Arbeit jedoch keine Betrachtung findet.

Der in der Simulation umgesetzte Algorithmus zur Symbolsynchronisation basiert auf der Beschreibung [BSB97] und wird nach dem in Abbildung 4.3 dargestellten Prinzip implementiert.

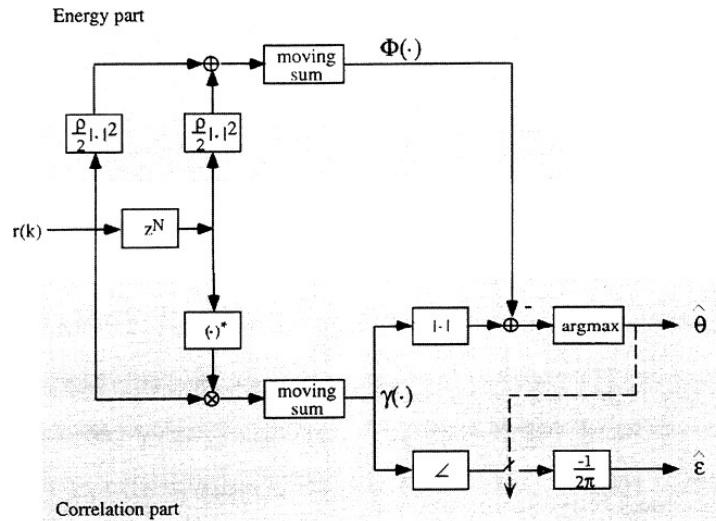


Abbildung 4.3.: Prinzip der Symbolsynchronisation nach [BSB97]

Diese Variante wird als kombinierte, grobe Schätzung des Symboltimings und der partiellen Trägerverschiebung (engl.: Combined coarse symbol timing (CST) and coarse fractional carrier frequency offset (CFCFO) estimation) bezeichnet, da neben der Ermittlung der Symbollage eine Bestimmung des partiellen Frequenzoffsets zugleich erfolgt.

Die mathematische Formulierung des Algorithmus lautet wie folgt:

$$\hat{\Theta} = \arg \max_{\Theta} \{|\gamma(\Theta)| - \rho\Phi(\Theta)\} \quad (4.1)$$

Der zu bestimmende größte Zeitoffset (Verzögerung um Θ Abtastwerte) $\hat{\Theta}$ wird aus dem Maximum der Differenz zwischen dem Korrelationsterms $\gamma(\Theta)$ und dem Energieterm $\rho\Phi(\Theta)$ innerhalb eines Symbols bestimmt.

Der Korrelationsterm stellt die Korrelation des empfangenen Symbols mit k Trägern $r(k)$ mit dem um N_U verzögerten Empfangssignal über das Fenster der Länge N_G dar und ermittelt dadurch die Lage des Schutzintervalls.

$$\gamma(\Theta) = \sum_{k=\Theta}^{\Theta+N_G-1} r(k)r^*(k+N_U) \quad (4.2)$$

Der Energieterm drückt die Signalenergie im gleichen Fenster aus und wird zur Berücksichtigung des additiven Rauschens im Empfangssignal zur verbesserten Detektion der Korrelationsmaxima genutzt.

$$\Phi(\Theta) = \frac{1}{2} \sum_{k=\Theta}^{\Theta+N_G-1} |r(k)|^2 |r^*(k+N_U)|^2 \quad (4.3)$$

Dieser Term kann noch über den zusätzlichen Faktor ρ gewichtet werden. Er ist abhängig vom SNR der Übertragung und wird bei hohem SNR fest mit 1 definiert. [CPWO06]

Neben dem Betrag des komplexen Zeitoffsets $\hat{\Theta}$ stellt die Phase an dieser Stelle den partiellen Frequenzoffset $\hat{\epsilon}$ über folgende Gleichung dar.[BSB97]

$$\hat{\epsilon} = -\frac{1}{2\pi} \angle \gamma(\hat{\Theta}_{ML}) \quad (4.4)$$

Dieser wird zusammen mit der Ermittlung der ganzzahligen Trägerverschiebung (Abschnitt 4.2) zur Frequenzsynchronisation verwendet.

Die Realisierung der Synchronisation aus der geschätzten Signalverschiebung, sowie graphische Auswertungen des Algorithmus werden bei der Umsetzung der Simulation im Abschnitt 5.7.1 beschrieben.

4.1.2. Framesynchronisation

Da im Empfänger die OFDM-Rahmenstruktur durch den OFDM-Demapper wieder aufgelöst wird und daraus eine Rückbildung der einzelnen Kanäle folgt, muss nach der FFT eine weitere Synchronisation folgen, die die Lage des ersten Symbols in einem Frame bestimmt. Diese Synchronisation wird als Schätzung der Framelage (engl.: frame timing estimation) oder Framesynchronisation bezeichnet.

Im Gegensatz zur Symbolsynchronisation werden hier die Referenzzellen, die nach der FFT wieder in ursprünglicher Form als komplexe Symbole vorliegen, genutzt. Wie im Abschnitt 3.3 angedeutet, spielen hier die Zeit-Referenzzellen, welche z.T. paarweise angeordnet sind, eine wichtige Rolle. Der folgende Algorithmus zur Framesynchronisation basiert auf der Ausarbeitung [CPWO06] zum Thema „Combined integer carrier frequency offset (ICFO) estimation and frame timing (FT) estimation“. Die für die Simulation verwendete Methode stellt die reine Umsetzung der Framesynchronisation ohne die zusätzliche ICFO dar.

Ausgehend von den Zeit-Referenzzellen, die nur im ersten Symbol ($s=0$) eines Frames liegen, können unter Hinzunahme der ebenfalls im ersten Symbol liegenden Gain- und Frequenz-Referenzzellen eine bestimmte Anzahl L_t an Pilotpaaren gebildet werden. Die Abbildung 4.4 stellt dies basierend auf der Pilotstruktur aus Abschnitt 3.3 dar.

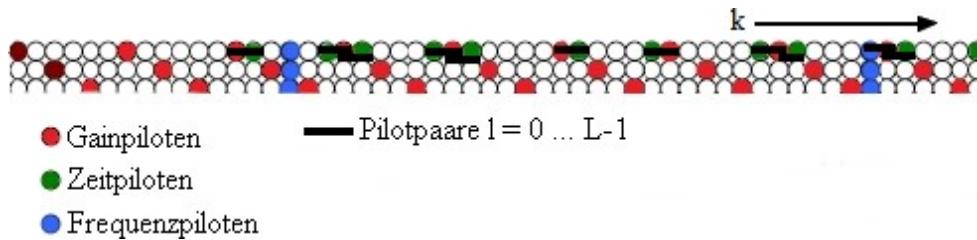


Abbildung 4.4.: Bildung von Pilotpaaren im ersten Symbol ($s=0$) eines Frames

Beispielsweise ergeben sich für den Fall Mode B mit $B_S=0$ $L_T=16$ Pilotpaare, die genutzt werden um über eine Korrelation die Lage des ersten Symbols im Frame zu bestimmen.

Nach der Bildung der Pilotpaare wird folgende Gleichung angewandt, die aus [CPWO06, Gl. 9] entnommen und so modifiziert wurde, dass der Anteil der Frequenzschätzung entfällt.

$$C(s) = \Re \left\{ \sum_{l=0}^{L_T-1} z_{s,p(l)} z_{s,q(l)}^* \cdot c_{0,p(l)} c_{0,q(l)}^* \right\} \quad (4.5)$$

$z_{s,k}$ stellen die empfangen Zellen und $c_{s,k}$ die gesendeten Zellen für das Symbols s mit den Trägern k dar. Da nur die Trägerindizes der Referenzzellen bekannt sind, wird über die gebildeten Trägerpaare l eine Korrelation aller empfangerer Symbole eines Frames mit den Trägerpaaren des ersten Symbols $c_{0,p(l)}$ und $c_{0,q(l)}$ durchgeführt. $p(l)$ gibt hierbei den

Trägerindex der ersten Referenzzelle eines Paars und $q(l)$ den nachfolgenden Trägerindex eines Paars an.

Ziel ist es die Schätzfunktion 4.5 zu maximieren, was dann der Fall ist, wenn die Korrelation ihren höchsten Wert erreicht und somit s die Lage des ersten Symbols im empfangenen Frame markiert. Durch die Ermittlung der Lage des ersten Symbols im Frame kann im Empfänger durch bestimmte Verzögerungen eine Synchronisation der Rahmenstruktur erfolgen, wie sie z.B. in der Umsetzung 5.7.3 dargestellt wird.

In [CPWO06] wird das Thema der Framesynchronisation noch ausführlicher behandelt und es werden verschiedene Kontrollstrategien genannt, um den Algorithmus zu optimieren.

4.2. Frequenzsynchronisation

Neben den zeitlichen Verschiebungen können durch verschiedene Effekte auch Störungen der Frequenz (Frequenzoffset) auftreten, die zu einer Verschiebung der einzelnen Subträger und somit zum Orthogonalitätsverlust führen. Folgende Komponenten innerhalb des gesamten Übertragungssystems können zu Verschiebungen der Frequenz beitragen:

- Durch bewegte Empfänger bzw. Sender oder durch eine Verschiebung der Objekte, an denen die Funkwellen reflektiert werden, kommt es zum Dopplereffekt, welcher einen Einfluss auf die Empfangsfrequenz nimmt.
- Durch den Einsatz von analogen Komponenten zur Verstärkung und Modulation können durch meist thermische Einflüsse Unregelmäßigkeiten bei Referenzfrequenzen auftreten.
- Die begrenzte Genauigkeit in einem digitalen System kann ebenfalls zu möglichen Verlusten der Frequenzsynchronität führen.

Der Frequenzoffset f_{off} kann wie folgt beschrieben werden:

$$f_{off} = f_{int} + f_{frac} \quad (4.6)$$

$$f_{int} = m \cdot \Delta f_k \quad m \in \mathbb{Z} \quad (4.7)$$

$$f_{frac} = c \cdot \Delta f_k \quad c \in \mathbb{R} \quad -0,5 \leq c < 0,5 \quad (4.8)$$

Er setzt sich also aus einem sogenannten ganzzahligen Anteil (engl.: integer frequency offset) f_{int} , der ein Vielfaches des Subträgerabstandes Δf_k bezeichnet, und einem partiellen Anteil f_{frac} (engl.: fractional frequency offset) zusammen.[Hen08]

Z.B. würde ein Wert von $f_{off} = 120$ Hz bei Mode B mit $\Delta f_k = 46,875$ Hz den folgenden Parametern entsprechen:

$$f_{int} = 140,625\text{Hz} \rightarrow m = 3$$

$$f_{frac} = -20,625\text{Hz} \rightarrow c = -0,44$$

Die Aufgabe der Frequenzsynchronisation ist es, die möglicher Weise aufgetretene Verschiebung der Frequenz mit geeigneten Methoden zu erkennen und zu kompensieren, wobei nach folgenden Ansätzen differenziert wird: [Hen08, S.82 ff.]

Akquisition Durch eine grobe Schätzung des Frequenzoffsets mit einer Genauigkeit $< \frac{\Delta f_k}{2}$ wird hierdurch ein Startwert für die folgende exaktere Schätzung festgelegt. Die Akquisition wird aufgrund des erhöhten Aufwands einmalig zu Beginn eines neuen Demodulationsprozesses im Empfänger durchgeführt.

Tracking Basierend auf der groben Schätzung der Akquisition erfolgt eine kontinuierliche, möglichst exakte Schätzung des verbleibenden partiellen Frequenzoffsets. Mit Hilfe diverser Regelalgorithmen findet eine ständige Korrektur der geringen Abweichungen statt. Sollte hierbei eine zu hohe Frequenzverschiebung festgestellt werden, muss erneut eine Akquisition erfolgen.

Zur Realisierung der beiden Verfahren gibt es eine Vielzahl verschiedenster, in der Literatur vorgestellter Algorithmen, die sich zunächst durch ihre Lage im Empfängersystem und dadurch bedingt auch durch ihre grundlegende Funktionsweise unterscheiden. Während die in z.B. [Hen08, S.84], [CPWO06] und [FK10] vorgestellten Akquisitions- und Trackingverfahren nach der FFT (Post-FFT) durchgeführt werden und auf den Frequenzpiloten aufbauen, werden in [BSB97] und [SFFM01] Verfahren vor der FFT (Pre-FFT) beschrieben, die auf der Korrelation mit dem Schutzintervall basieren.

Der in der Simulation umgesetzte Algorithmus zum Tracking, welcher nach [BSB97] arbeitet, wurde bereits im Abschnitt 4.1.2 vorgestellt. Der als \hat{e} ermittelte Wert zu Beginn eines jeden Symbols kann als der partielle Anteil c des Frequenzoffsets in Gleichung 4.8 betrachtet werden.

Die Ermittlung der ganzzahligen Trägerverschiebung in der Simulation wird nach dem in [SDSGU05] verwendeten Algorithmus umgesetzt, der grundsätzlich nach der FFT ansetzt und nach dem Prinzip der Korrelation angrenzender Symbole versucht die Lage der Frequenzpiloten im Empfangssignal zu detektieren. Da die Frequenzpiloten eine geradlinige Struktur mit festen Phasen über den gesamten Frame an ihren Trägerpositionen aufweisen, während an den Stellen der anderen Träger abwechselnd Daten- und Pilotzellen mit unterschiedlichen Phasen auftreten, wird versucht durch die Korrelation eine Maximierung an den Stellen der Frequenz-Referenzzellen zu erkennen.

Die Abbildung 4.5 stellt noch einmal die im Abschnitt 3.3 beschriebene Pilotstruktur dar. Zu beachten ist dabei jedoch, dass ein kompletter OFDM-Rahmen mit der für die IFFT/FFT verwendeten Länge N_U gezeigt wird.

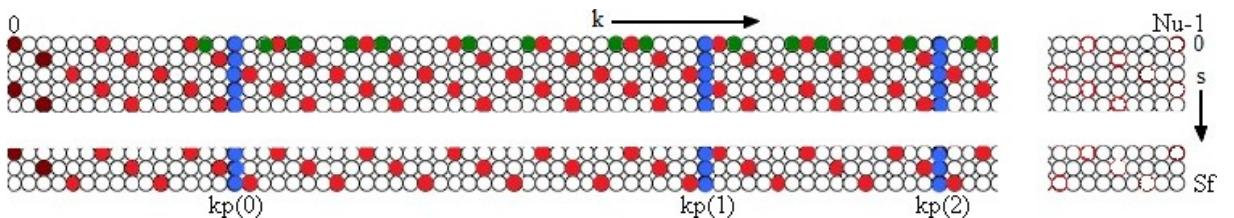


Abbildung 4.5.: Betrachtung der Frequenzpiloten in einem OFDM-Rahmen über N_U Werte

Der in [SDSGU05] in der Datei `get_frequency_offset_integer.m` implementierte Algorithmus lässt sich mathematisch wie folgt formulieren.

Es wird eine Korrelation der von der FFT kommenden Vektoren $r_{s,k}$ mit $k = N_U$ Subträgern zwischen zwei angrenzenden Symbolen s durchgeführt und die Phase des Kor-

relationsergebnisses über die Anzahl der Symbole eines gesamten Frames $s = S_f$ aufakkumuliert.

$$R_k = \left| \sum_{s=0}^{S_f} e^{j\angle(r_{s,k} r_{s-1,k}^*)} \right| \quad k = 0 \dots N_U - 1 \quad (4.9)$$

Dabei entstehen N_U Akkumulationswerte R_k , die an den Stellen $k = k_p$ der Frequenzpiloten ihre Höchstwerte erreichen.

Um den Effekt der Korrelationsspitzen zu verstärken, wird noch einmal eine Summierung über die drei ($p = 0, 1, 2$) Referenzzellen der akkumulierten Phasen der Träger mit einer Verschiebung um die Stellen an denen die Frequenzpiloten liegen ($k_p(p)$) durchgeführt und anschließend über das Maximum der N_U Akkumulationswerte die ganzzahlige Trägerverschiebung m ermittelt.

$$m = \max_k \left\{ \sum_{p=0}^2 R_{k+k_p(p)} \right\} \quad (4.10)$$

Mit dem ermittelten ganzzahligen Anteil m und dem partiellen Anteil c der Trägerverschiebung kann über die Multiplikation mit Δf_k der Frequenzoffset nach den Gleichungen 4.6, 4.7 und 4.8 bestimmt und zur Kompensation genutzt werden. Die Umsetzung der Frequenzsynchronisation in der Simulation wird im Abschnitt 5.7.2 erklärt.

4.3. Kanalschätzung und -kompensation

Die Kanalschätzung dient zur Ermittlung der Kanalübertragungsfunktion (CTF). Da durch die beschriebene Frequenz- und Zeitselektivität des Kanals (siehe Abschnitt 2.1.4) die einzelnen OFDM-Zellen in einem OFDM-Rahmen unterschiedliche, meist nicht konstante Änderungen an Betrag und Phase erfahren können, ist es notwendig diese Filtereigenschaften des Kanals zu bestimmen und durch ein geeignetes inverses Filter zu kompensieren.

Das Themengebiet der Kanalschätzung und der Kanalkompensation ist sehr umfangreich und komplex und kann unter verschiedensten Aspekten betrachtet werden. Eine ausführlichere Abhandlung hierzu ist in [Kam04, Kap. 14] zu finden, wo grundsätzlich zwischen der Referenzsignal-gestützten und blinden Kanalschätzung differenziert wird. Im folgenden soll nur die für DRM und anderer OFDM-Systeme mit Pilotsignalen verwendete Referenzsignal-gestützte Kanalschätzung betrachtet werden.

Die Schätzung der CTF basiert auf den in Zeit- und Frequenzrichtung regelmäßig angeordneten Pilotzellen. Betrachtet man Abbildung 3.20 im Abschnitt 3.3, so ist eine periodische diagonale Struktur der Gain-Piloten zu erkennen. Andere Systeme können dabei auch andere Strukturen aufweisen, wie in [Chi04, S.81 ff.] und [Pul07, S.6 ff.] dargestellt, arbeiten jedoch grundsätzlich nach dem im Folgenden beschriebenen Prinzip.

Das vom Sender kommende OFDM-Signal kann in seiner Rahmenstruktur als Matrix $S(s, k)$, bestehend aus s Symbolen mit k Trägern, angesehen werden. Ebenfalls kann die in diskreter Form vorliegende CTF $H_c(s, k)$ und das additive Rauschen $N(s, k)$ als Matrix betrachtet werden. So lässt sich das beim Empfänger ankommende Signal nach Gleichung 2.7 im Frequenzbereich wie folgt beschreiben:

$$R(s, k) = S(s, k)H_c(s, k) + N(s, k) \quad (4.11)$$

Ziel ist es also die CTF aus dem Sendesignal und dem mit möglichen Rauschen überlagerten Empfangssignal optimal zu schätzen.

$$H_{est}(s, k) = \frac{R(s, k)}{S(s, k)} \quad (4.12)$$

Da im Empfänger nur Betrag und Phase der Referenzzellen bekannt sind, kann zunächst nur eine unvollständige Ermittlung der CTF an den Stellen der Pilotzellen $S(s_p, k_p)$ erfolgen.

$$\hat{H}_{est}(s_p, k_p) = \frac{R(s_p, k_p)}{S(s_p, k_p)} \quad (4.13)$$

Mit Hilfe dieser Stützstellen kann über eine anschließende zweidimensionale Interpolation eine Schätzung von H_{est} erfolgen. Abbildung 4.6 zeigt einen Ausschnitt des Rasters der Gain-Piloten für den Mode B und deutet die Interpolation in Zeitrichtung (rot) und Frequenzrichtung (blau) an. Zudem sind die in Tabelle 3.8 dargestellten Abstände zwischen

den Referenzzellen markiert. Hieraus wird ersichtlich, dass bei steigender Anforderung an den Modus hinsichtlich Zeit- und Frequenzselektivität des Kanals die Abstände geringer gewählt werden müssen, um eine Schätzung der Einflüsse auf die dazwischen liegenden Datenzellen über den Interpolator zu vereinfachen.

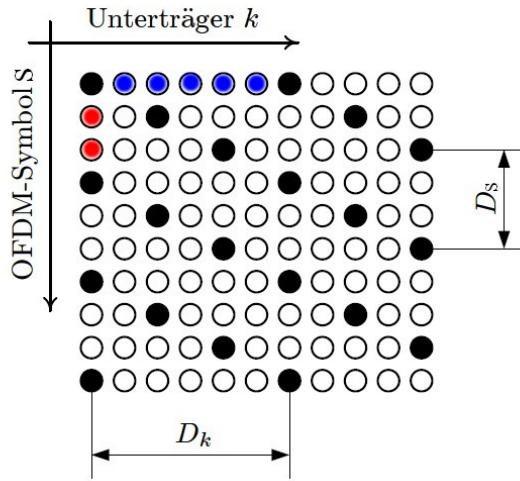


Abbildung 4.6.: Darstellung der Methode zur Interpolation bei der Kanalschätzung

Die Interpolation der CTF für die Datenzellen kann je nach Anforderung an die Genauigkeit bzw. an die Rechenleistung des Empfängersystems in unterschiedlichster Weise erfolgen. Eine Untersuchung zu diesem Thema und damit eine Gegenüberstellung der verschiedenen Verfahren wurde in [Hen08, S.113] durchgeführt. Nachfolgend soll ein kurzer Überblick dazu erfolgen:

Lineare Interpolation Die am recheneffizientesten Methode bildet eine Gerade zwischen zwei Stützwerten und übernimmt die dadurch entstehenden Zwischenwerte an den Abtaststellen der Datenzellen. Bei geringen Abständen zwischen den Piloten können damit gute Ergebnisse erzielt werden.

Spline Interpolation Durch die Unterteilung der Lücken zwischen den Abtastwerten in kleinere Intervalle und einer Polynombildung 3. Ordnung können exaktere Schätzergebnisse für die Datenzellen geliefert werden, was jedoch schon einen deutlichen Anstieg des Rechenaufwands bedeutet.

DFT Interpolation Eine Transformation der teilweise geschätzten CTF in den Zeitbereich lässt eine Veränderung der Kanalimpulsantwort zu. Durch das Einfügen von Nullen und einer anschließenden Rücktransformation erhält man die vollständige

CTF. Diese Methode erzielt wesentlich bessere Ergebnisse als die beiden zuvor. Aufgrund des sehr hohen Rechenaufwands sollte diese jedoch zum einen nur in Frequenzrichtung und zum anderen nur bei stark frequenzselektiven Kanälen eingesetzt werden.

Wiener Filter Diese Art der Interpolation ist die "...‘optimale Lösung...“[Kam04, S.607] zur Ermittlung der CTF. Es wird über stochastische Methoden versucht eine Minimierung der Schätzfehlerleistung zu erzielen.[Kam04] Diese Interpolationsmethode erzielt die besten Ergebnisse, da auch auf sich ändernde unabhängige Rauscheinflüsse eingegangen wird. Obwohl sie auch den höchsten Rechenaufwand erfordert, kann sie heutzutage bei PC-basierten SDR-Systemen ([SDSGU05],[o.A10]) problemlos verwendet werden.

Prinzipiell können all diese Methoden auch kombiniert verwendet werden, da sich die zweidimensionale Interpolation in Zeit- und Frequenzrichtung separierbar durchführen lässt.

Ist die vollständige, geschätzte CTF H_{est} vorhanden, kann mit dieser eine Kanalkompensation entweder nach der FFT durch Multiplikation der Empfängermatrix mit dem Reziproken $R(s, k) \cdot \frac{1}{H_{est}(s, k)}$ oder wie in Abbildung 4.1 dargestellt durch die Multiplikation der Empfangssignale vor der FFT mit der Kanalimpulsantwort durchgeführt werden.

5. Umsetzung der Simulation

Die Umsetzung der Simulation beinhaltet sechs grundlegende Kategorien. Neben der Umsetzung des DRM-Standards als Sender- und Empfängermodell in Simulink stellen die Initialisierung der Simulation, die Verifizierung des Sendermodells, die Entwicklung des Kanalmodells und die Realisierung der Synchronisationsalgorithmen für den Empfänger die wichtigsten Stufen der Masterarbeit dar.

Die fertige Simulation samt aller zugehörigen Dateien befindet sich auf dem Datenträger. Das Modell wird dabei über den Aufruf der Datei „drm_simu.mdl“ gestartet und ermöglicht dem Nutzer diverse Einstellungen und Möglichkeiten zur Auswertung eines DRM-Systems, wie sie in diesem Kapitel beschrieben werden.

Alle im folgenden Kapitel kursiv dargestellten Begriffe beziehen sich entweder auf Bezeichnungen und Blocknamen aus MATLAB und Simulink oder stellen in MATLAB benutzte Variablen zur Initialisierung der Simulation dar.

5.1. Die Entwicklungsumgebung

Simulink ist eine Simulationsumgebung und ein Modellierungswerkzeug, welches auf dem Programm MATLAB der Firma „The MathWorks“ aufbaut. Dabei stellt Simulink ein interaktives und grafikorientiertes Tool zur Analyse und Modellierung von linearen und nichtlinearen mathematischen Zusammenhängen dar, welches auf die numerischen Lösungsalgorithmen von MATLAB zurückgreift.

MATLAB ist eine kommerzielle, plattformunabhängige Software, welche ursprünglich zur numerischen Lösung von Differentialgleichungen mit Hilfe von Matrizen (MATrix LABoratory) entwickelt und im Laufe der Zeit durch immer mehr zusätzliche Funktionen ausgestattet wurde, so dass in fast allen technischen und naturwissenschaftlichen Bereichen ein Einsatz dieses Tools zur Lösung von mathematischen Problemen und zur Visualisierung dieser möglich ist.

Durch die Erweiterung von MATLAB mit Simulink wurde eine graphische Oberfläche geschaffen, die die Lösung mathematischer Modelle anschaulich macht und darüber hinaus eigenständige Simulationen diverser Systeme als dynamische Modelle ermöglicht.

Nachfolgend sollen einige wichtige Eigenschaften und Funktionen von Simulink genannt werden, die dieses Tool besonders auszeichnen:

- Es ermöglicht eine schnelle und einfache Modellierung und Überprüfung fast aller technischer Zusammenhänge über Drag&Drop.

5. Umsetzung der Simulation

- Eine sehr umfangreiche Funktionspalette, sowie zusätzliche Komponenten für spezielle Themengebiete beschleunigen die Modellbildung, da eine komplexe Erstellung bestimmter Systeme entfällt.
- Durch eine Vielzahl an Visualisierungsmöglichkeiten wird eine Überprüfung des Modells während der Laufzeit in einer anschaulichen Weise möglich.
- Bei immer komplexeren Simulationen kann über die hierarchische Modellierung mit Subsystemen ein Überblick gewahrt werden.
- Die Verwendung eines Workspace (Arbeitsbereich) schafft die Option der Verwendung von Variablen zur Ein- und Ausgabe der Simulationsparameter.

Die Umsetzung der Simulation des DRM-Übertragungssystems erfolgte mit MATLAB-/Simulink (Version 7.9.0 (R2009a)), da diese Tools neben vielen Funktionen und Programmierelementen der numerischen Mathematik einen breiten Umfang an zusätzlichen Funktionen und vordefinierten Simulink-Blöcken im Bereich der Signalverarbeitung und der Nachrichtentechnik bieten. So wurden für das Modell folgende zusätzlichen Funktionsbibliotheken (Blocksets) genutzt, welche im Standardumfang des Softwarepaketes nicht vorhanden sind:

- Communications Blockset
- Signal Processing Blockset

Die Blocksets bieten neben vieler fertig erstellter Funktionen und Blöcke, die für die Implementierung des DRM-Systems genutzt werden können auch diverse Möglichkeiten zur Auswertung und Analyse von Signalen und zur Darstellung von Signalverläufen im Zeit- bzw. Frequenzbereich, wie z.B. BER-Auswertung, Konstellationsdiagramme und Spektralanalysen.

Ein weiteres Argument für die Verwendung von MATLAB/Simulink zur Umsetzung des DRM-Systemmodells ist die Möglichkeit mit Hilfe des Real-Time-Workshops aus den erstellten Komponenten des Systems lauffähigen C/C++ oder VHDL-Code zu generieren, welcher sich auf diversen Zielplattformen (DSP, FPGA, Mikroprozessor) portieren lässt und somit eventuell eine Implementierung des Systems auf diesen ermöglicht.

5.2. Initialisierung der Simulation

Um dem Benutzer der Simulation die Flexibilität zu geben alle im DRM-Standard definierten Modi, Spektralbelegungen, die QAM-Typen und die Schutzlevel für den SDC und MSC, sowie die Interleavingtiefe des Cell-Interleavers frei zu wählen, muss vor dem Aufruf des eigentlichen Modells eine Initialisierung erfolgen. Hier werden basierend auf der Auswahl der genannten Parameter alle Einstellungen für die im Simulink-Modell benutzten Blöcke zuvor festgelegt. Der Hauptinitialisierungsauftrag bzw. die Nutzerschnittstelle zur Konfiguration des Modells befindet sich in der Datei *init_drm.m*, welche sich als Aufruf als *PostLoadFcn* im *Model Explorer* befindet. D.h. zum einen wird die Initialisierungsdatei automatisch beim Starten der Simulation, also des Modells *drm_simu*, aufgerufen und zum anderen kann beim geöffneten und nicht laufenden Modell eine Änderung der Parameter in *init_drm.m* vorgenommen werden, welche dann zu einer Umstellung der Blockeinstellungen führt und den Benutzer ermöglicht ständige Änderungen der variablen DRM-Parameter vorzunehmen. Neben der Festlegung der Parameter erfolgen hier die Aufrufe sämtlicher weiterer Initialisierungsdateien.

Die Datei *init_drm.m* baut sich wie folgt auf:

Zunächst muss der Nutzer die Auswahl der MATLAB/Simulink-Version treffen. Hierbei kann zwischen den Versionen 2009a und 2008a gewählt werden. Diese Einstellung wird für den MLC (siehe 5.3.2.2) benötigt, da hier der Name des Modells zur Initialisierung benötigt wird.

```
% model version for R2009a
model_type = 'drm_simu/';
% model version for R2008a
%model_type = 'drm_simu_r08/';
```

Es folgt die Auswahl des Modes und der Spektralbelegung B_S , wobei darauf geachtet werden sollte, dass nicht für alle Modi jegliche B_S vorgesehen sind.

```
% select mode and spectrum occupancy
mode = 'B';
spec_occupancy = 3;
```

Dann können die verschiedenen Schutzlevel für den MSC und den SDC, welches der Auswahl des QAM-Typs für diesen entspricht, gewählt werden.

```
sdc.protection_level = 0;
...
msc.protection_level = 0;
```

Des Weiteren wird die Auswahl des QAM-Typs für den MSC über folgenden Parameter angegeben.

```
% select mapping type of MSC
% can be 16 or 64 QAM for mode A-D and 4 or 16 for mode E
MSC_QAM = 16;
```

Als letzte Variation kann der Nutzer die Tiefe für das Cell-Interleaving (siehe 3.1.4) festlegen. Dabei steht eine '1' für den Einsatz des „long interleavings“.

5. Umsetzung der Simulation

```
% select interleaving depth
% long_interleaving = 1 (long interleaving is active)
msc.long_interleaving = 1;
```

Nach der Festlegung der Parameter erfolgt der Aufruf der weiteren Initialisierungsdateien, welche in Abhängigkeit von den selektierten Einstellungen die Variablen für die Simulationsblöcke belegen. Die Initialisierungsdateien und ihre Funktionen sind im Anhang A.2.1 aufgelistet. Eine Beschreibung der einzelnen Schritte erfolgt anhand von Kommentaren in den Dateien selbst.

Die Initialisierungsdateien erzeugen Schritt für Schritt die Variablen für die Funktionsblöcke des Simulationsmodells, welche je nach ihrer Kategorie in Strukturen oder als Einzelvariable organisiert werden und im Anhang A.2.2 mit ihren Bedeutungen aufgeführt sind.

5.3. Sendermodell

Das Simulationsmodell soll weitestgehend dem DRM-Systemstandard [Eur09] entsprechen und wird in der höchsten Ebene (Top-Level) wie in der Abbildung 2.7 aufgebaut.

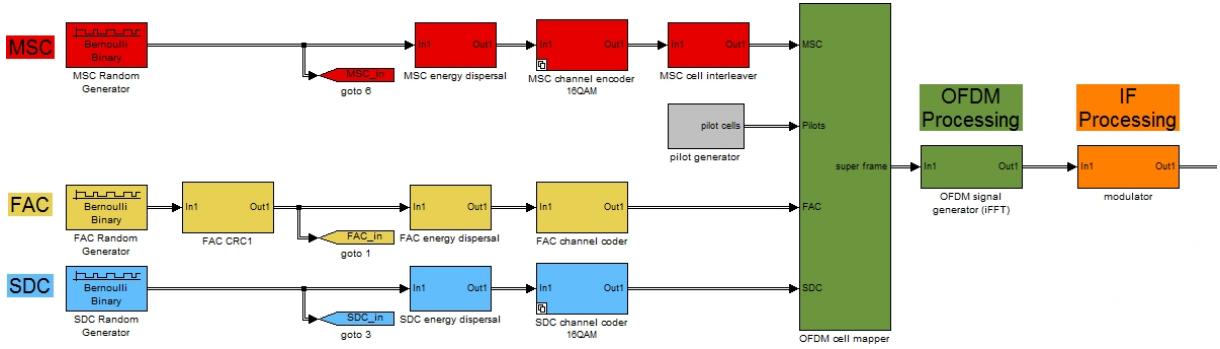


Abbildung 5.1.: Sendermodell in Simulink

Der Aufbau des in Abbildung 5.1 dargestellten Sendermodells erfolgt nach den im Kapitel 3 beschriebenen Komponenten der DRM-Spezifikation über Subsysteme. Die farbliche Kennzeichnung der einzelnen Kanäle und der Prozessschritte der Datenverarbeitung sollen dem Nutzer einen schnelleren Überblick gewähren. Die einzelnen Subsysteme werden nachfolgend in der Reihenfolge ihrer Abarbeitung beschrieben.

5.3.1. Datengenerierung und CRC

5.3.1.1. Datengenerierung

Da die Simulation nach der Quellenkodierung einsetzt, müssen die Eingangsdatenströme der einzelnen Kanäle mittels eines Signalgenerators erstellt werden. Hierzu wird der *Bernoulli Binary Generator* aus dem *Communications Blockset* genutzt, welcher framebasierte binäre Datenströme erzeugen kann. Bei allen drei Kanälen wird eine Gleichverteilung der Daten (*Probability of a zero: 0.5*) und eine dem Kanal entsprechende Rahmengröße *Samples per frame* eingestellt.

Die Rahmengröße entspricht der von der Quellenkodierung stammenden Datenmenge und ist beim SDC und MSC von den wählbaren Parametern (siehe Initialisierung 5.2) abhängig. Beim FAC wird immer die gleiche Rahmengröße mit 72 Bits für die Modi A-D und 244 Bits für den Mode E über die Variable *fac.Nmb_input_bits* genutzt, wobei hier 8 Bits für den CRC (siehe 5.3.1.2) vorgesehen sind. Die Variablen der Rahmengrößen *msc.Nmb_input_bits* und *sdc.Nmb_input_bits* für den MSC und SDC werden in den jeweiligen Initialisierungsdateien der Kanäle *init_sdc.m* und *init_msc.m* erstellt und entsprechen den Größen der in der Spezifikation befindlichen Tabellen [Eur09, S.192 ff.] festgelegten Anzahl an Eingangsbits.

Der Datengenerator für den MSC besitzt des Weiteren eine wichtige Bedeutung für das Modell, da hier die Taktrate bzw. *Sample time* für die Simulation festgelegt wird. Außer den unabhängig arbeitenden PN-Generatoren für den Scrambler und Descrambler und den für die Zwischenfrequenzmodulation genutzten Sinusgeneratoren, sind alle weiteren Blöcke des Modells von dieser Abtastzeit abhängig bzw. haben diese Abtastzeit geerbt (engl.: inherited). Die Abtastzeit t_s wird in der Datei *init_fft.m* als Variable *sFFT.input_samplerate* festgelegt und berechnet sich folgendermaßen:

$$t_s = \frac{\text{Anzahl der Symbole in einem Frame} * \text{Dauer eines Symbols } (T_S)}{\text{Anzahl der Eingangsbits des MSC}}$$

5.3.1.2. CRC

Die zyklische Redundanzprüfung (CRC, engl.: Cyclic Redundancy Check) wird bei digitalen Nachrichtensystemen eingesetzt, um die übertragenen Daten auf eventuell auftretende Fehler zu prüfen. Hierzu wird jedem Datenblock bzw. Datenrahmen im Sender eine zusätzliche Prüfsumme (engl.: checksum) angehängt, welche im Empfänger kontrolliert werden kann. Die Erstellung der Prüfsumme erfolgt durch Polynomdivision auf Basis der Modulo-2-Verknüpfung des Datenblocks mit einem festgelegten CRC-Polynom. Im Empfänger kann durch erneute Polynomdivision mit dem gleichen Generatorpolynom festgestellt werden, ob sich der Datenblock während der Übertragung geändert hat.

Die bei DRM definierten CRC-Polynome und deren Verwendung sind in der Spezifikation [Eur09, S.164] festgehalten.

Obwohl die CRC Bestandteil der Quellenkodierung ist, wird diese jedoch für bestimmte Funktionen auch im Simulationsmodell eingesetzt. Zum einen lässt sich die CRC beim FAC in einfacher Weise an den Eingangsframe anfügen und zum anderen dient die CRC-Prüfung des FAC im Empfänger als Kontrolle für die abgeschlossene Synchronisierung und damit als Ausgangspunkt für die Auswertung des Bitfehlerverhältnisses (siehe 5.5.4). Zusätzlich erfolgt eine Verwendung der CRC-Prüfung bei der Verifizierung des Sendermodells 5.4.

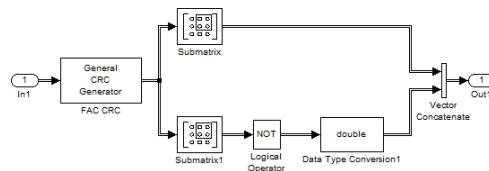


Abbildung 5.2.: Subsystem der zyklischen Redundanzprüfung beim FAC

Zur Umsetzung der CRC beim FAC wird der *General CRC Generator* aus dem *Communications Blockset* genutzt. Das Generatorpolynom und dessen Initialisierungswort sind wie folgt definiert:

5. Umsetzung der Simulation

```
sCRC_poly(1).G8 = [1 0 0 0 1 1 1 0];
sCRC_poly(2).G8 = ones(1,8);
```

Die vom CRC-Generator erstellte 8 Bit lange Prüfsumme wird anschließend invertiert und wieder an den 64 bzw. 236 Bit langen Eingangsframe des FAC angefügt, so dass dieser die für den Kanalkodierer vorgesehene definierte Größe (72 oder 244 Bits) besitzt.

5.3.2. Kanalkodierung und Modulation

5.3.2.1. Energy dispersal

Zur Umsetzung des Scramblers wäre in Simulink die Nutzung des gleichnamigen vorgefertigten Blocks aus dem *Communications Blockset* denkbar. Dieser bietet jedoch nicht die Möglichkeit den in der DRM-Spezifikation geforderten Reset des PRNG nach jedem binären Eingangsframe durchzuführen. Deshalb wird ein Aufbau über einen rücksetzbaren *PN Sequence Generator* und einer zusätzlichen XOR Verknüpfung gewählt, wie in Abbildung 5.3 dargestellt wird.

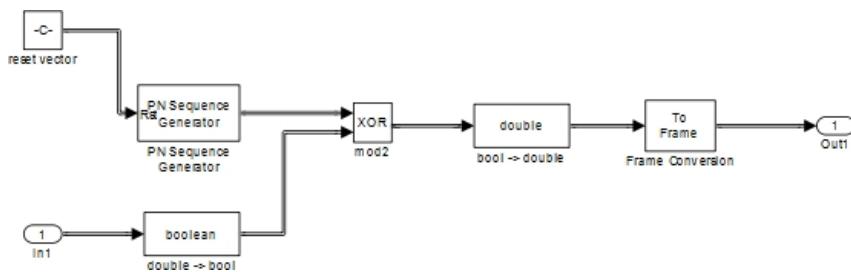


Abbildung 5.3.: Subsystem für das Energy dispersal

Der Aufbau zeigt das in Abschnitt 3.1.1 beschriebene Prinzip einer Scrambler- / De-scramblerschaltung. Die Parameter für den PRNG und den Resetvektor werden in der Datei *energy_dispersal.m* initialisiert und werden in der Struktur *scrambler* zusammengefasst. Das Scramblerpolynom und das Initialisierungswort werden wie in der Spezifikation festgelegt, mit folgenden Werten definiert:

```
scrambler.poly = [1 0 0 0 0 1 0 0 0 1];
scrambler.init = ones(1,9);
```

Für jeden Kanal wird das gleiche System verwendet, aber eine separate *Sample time* und ein separater Resetvektor eingestellt, da diese Parameter zum einen von der variablen Framegröße und zum anderen von der Taktrate des Eingangsdatenstroms abhängig sind.

5.3.2.2. Kanalkodierung

Nach der Verwürfelung des Eingangsdatenstromes durch den Scrambler erfolgt die Kanalkodierung. Hierzu werden die Daten innerhalb der in Abschnitt 3.1.2.1 beschriebenen MLC zunächst in die einzelnen Zweige aufgeteilt. Anschließend erfolgt die Verarbeitung der Daten der separaten Level über den Faltungscoder, den Bit-Interleaver und dem abschließenden QAM-Mapping, welches die Daten der Zweige wieder zusammenfasst und der OFDM zuführt.

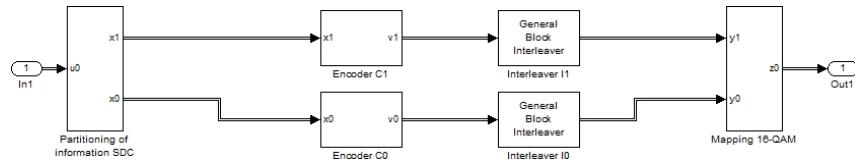


Abbildung 5.4.: Subsystem zur Kanalkodierung für den SDC mit einer 16-QAM

Die Abbildung 5.4 zeigt den Simulinkaufbau der Kanalkodierung für den SDC bei einer 16-QAM. Die Aufteilung der Daten im Subblock *Partitioning of information SDC* geschieht über die Indizierung des Eingangsstromes u_0 , welche bei der Initialisierung in den Dateien *init_sdc.m* und *init_msc.m* festgelegt wird und nach dem in der Spezifikation [Eur09, S.116] definierten Prinzip erfolgt.

Multilevel-Kodierung

Da die Simulation die Möglichkeit bieten soll unterschiedliche QAM-Typen für SDC und MSC zu nutzen, muss die Kanalkodierung variabel gestaltet werden. Hierzu werden zunächst alle für das SM definierten Multilevel-Kodierungen der möglichen QAM-Varianten für SDC und MSC in den Simulink-Libraries *SDC_coder_lib* und *MSC_coder_lib* als separate Subsysteme zusammengefasst. Über diese erstellten Bibliotheken kann mit Hilfe eines *Templates* ein variables Subsystem (*Configurable Subsystem*) im Modell eingebunden werden. Diese optionale Umsetzung des Kanalkodierers ist für den FAC nicht nötig, da stets eine 4-QAM in allen Modi genutzt wird.

Bei der Initialisierung der Simulation wird dann je nach gewünschtem QAM-Typ bzw. Schutzlevel von SDC und MSC eine Modellvariante der Bibliotheken über folgende Einstellungen in der Datei *init_msc.m*

```

if (MSC_QAM == 64)
    set_param(strcat(model_type, 'MSC channel encoder'), 'BlockChoice', '64QAM');
    set_param(strcat(model_type, 'MSC channel decoder'), 'BlockChoice', '64QAM');
elseif (MSC_QAM == 16)
    set_param(strcat(model_type, 'MSC channel encoder'), 'BlockChoice', '16QAM');
    set_param(strcat(model_type, 'MSC channel decoder'), 'BlockChoice', '16QAM');
elseif (MSC_QAM == 4)
    set_param(strcat(model_type, 'MSC channel encoder'), 'BlockChoice', '4QAM');
    set_param(strcat(model_type, 'MSC channel decoder'), 'BlockChoice', '4QAM');
else

```

5. Umsetzung der Simulation

```

error('msc:qam_type','MSC QAM type does not exist');
end
...

```

und in der Datei *init_sdc.m*

```

switch mode
case {'A', 'B', 'C', 'D'}
    if (sdc.protection_level == 0)
        SDC_QAM = 16;
        set_param(strcat(model_type, 'SDC channel coder'), 'BlockChoice', '16QAM');
        set_param(strcat(model_type, 'SDC channel decoder'), 'BlockChoice', '16QAM');
        ...
    elseif (sdc.protection_level == 1)
        SDC_QAM = 4;
        set_param(strcat(model_type, 'SDC channel coder'), 'BlockChoice', '4QAM');
        set_param(strcat(model_type, 'SDC channel decoder'), 'BlockChoice', '4QAM');
        ...
    else
        error('sdc:protection_level',...
            'Protection level is not defined for SDC');
    end
...

```

vorgenommen. Hierbei wird zunächst der Modellname, der in der *init_drm.m* festgelegt wird, mit der entsprechenden Bibliothek verknüpft und anschließend über den Parameter *BlockChoice* die gewählte Variante der Simulation zugewiesen.

Faltungscoder

In allen Zweigen der Multilevel-Kodierung der drei Kanäle wird zur Umsetzung des in Abschnitt 3.1.2.2 beschriebenen Faltungscodes eine gleiche Struktur benutzt. Diese besteht wie in Abbildung 5.5 dargestellt aus dem Faltungskodierer mit einer festen Coderate und der anschließenden Punktierung, welche die geforderten variablen Coderaten umsetzt.



Abbildung 5.5.: Subsystem des Faltungscoders

Zur Realisierung des Faltungscoders kann auf den in Simulink existierenden Block *Convolutional Encoder* zurückgegriffen werden. Hier wird zum einen über die Variable *conv_poly* aus der Initialisierungsdatei *conv_and_puncturing.m* die Struktur (Trellisstruktur) des Faltungscoders festgelegt und zum anderen kann durch die Auswahl des *Operation mode* „Terminate trellis by appending bits“ die geforderte Wiederherstellung des Grundzustands des Encoders durch Anfügen von Nullen realisiert werden.

Die Struktur des Faltungscodes wird wie folgt festgelegt:

```
conv_poly = poly2trellis(7, [133 171 145 133 171 145]);
```

Die Funktion *poly2trellis* dient der Umwandlung des in oktaler Form eingegebenen Generatorpolynoms des Faltungscoders in eine für den Block benötigte Trellisstruktur. Die 7 steht für die möglichen Verbindungen zwischen den einzelnen Verzögerungselementen und der folgende Vektor steht für den in Abbildung 3.6 dargestellten Aufbau.

Durch den gewählten *Operation mode* werden die für die Terminierung erforderlichen 6 Abschlussbits an das Ende eines Frames des Eingangsdatenstroms angefügt und somit der Grundzustand des Faltungscoders wiederhergestellt.

Punktierung

Um die für die Multilevel-Kodierung definierten unterschiedlichen Coderaten umzusetzen, erfolgt eine Punktierung des durch den Faltungscoder erweiterten Datenstroms.

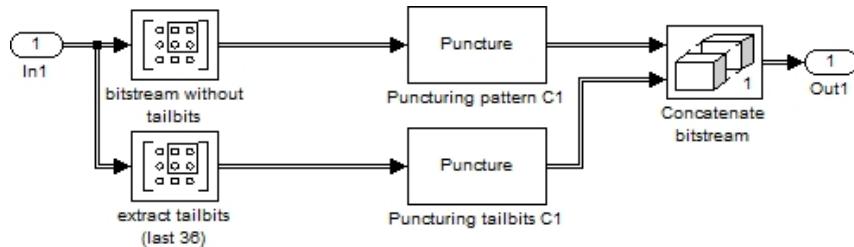


Abbildung 5.6.: Subsystem zur Punktierung

Die Abbildung 5.6 zeigt den für den SDC und den MSC verwendeten Simulinkaufbau für die Punktierung. Da eine getrennte Verarbeitung der Tail-Bits in der DRM-Spezifikation gefordert wird, werden die letzten 36 Bits (6x6 Tail-Bits) vom eigentlichen Datenstrom getrennt und einer separaten Punktierung unterzogen. Die in [Eur09, S.120] und [Eur09, S.122] befindlichen Punktierungsraster sind in den Dateien *conv_and_puncturing.m* und *tailbit_select.m* festgehalten. Bei der Initialisierung der einzelnen Kanäle findet dann eine Auswahl dieser Vektoren nach den im Anhang A.1.2 dargestellten, gewünschten Coderaten statt.

Da für den FAC keine gesonderte Punktierung der Tail-Bits vorgesehen ist, entfällt in der Simulinkstruktur die Aufteilung und es findet eine einfache Punktierung der Bits über den Block *Puncture* statt.

5.3.2.3. Bit-Interleaving

Für das Bit-Interleaving wird der Simulink-Block *General Block Interleaver* verwendet. Dieser erhält die, bei der Initialisierung in den Dateien *init_fac.m*, *init_sdc.m* und *init_msc.m* erstellten, Permutationsvektoren für die unterschiedlichen Level und Kanäle.

```

fac.interleave_P0
sdc.interleave_P0 = P1
  
```

`msc.interleave_P1 – P2`

Die Vektoren werden genutzt, um eine Vertauschung der Bitpositionen nach dem in Abschnitt 3.1.2.4 beschriebenen Prinzip durchzuführen.

5.3.2.4. QAM-Mapping

Nach der getrennten Verarbeitung der einzelnen Level der MLC werden die binären Daten durch das QAM-Mapping zusammengefasst und über festgelegte Konstellationsdiagramme nach dem in Abschnitt 3.1.3 beschriebenen Verfahren komplexen Zeigern zugewiesen, welche dann als OFDM-Zellen im Übertragungsrahmen strukturiert werden.

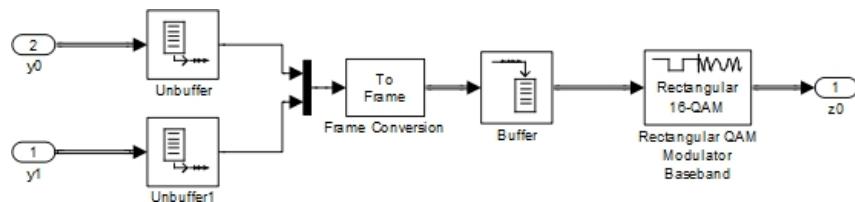


Abbildung 5.7.: Subsystem für das QAM-Mapping mit einer 16-QAM

Das Zusammenfassen der Daten erfolgt nach dem in der Spezifikation [Eur09, S.132] festgelegten Verfahren. Die Abbildung 5.7 zeigt den Aufbau des QAM-Mappings in Simulink für den SDC mit einer 16-QAM. Zur Realisierung der QAM wird der Simulink-Block *Rectangular QAM Modulator Baseband* benutzt, welcher es ermöglicht benutzerdefinierte Konstellationen für eine rechteckige, orthogonale QAM umzusetzen. Die Konstellationspunkte und die Normierungsfaktoren der QAM werden in der Datei *qam_mapping.m* definiert und als Variablen *qam_map* und *qam_norm* in den Strukturen der einzelnen Kanäle definiert.

Die Abbildung 5.8 zeigt das in Simulink vom QAM-Modulator genutzte Konstellationsdiagramm für die 16-QAM bei SM. Die Abbildung entspricht exakt dem im Abschnitt 3.1.3 dargestellten Bild 3.8, welches in der DRM-Spezifikation festgelegt ist, wobei die Normierungsfaktoren in der Darstellung bereits mit eingerechnet sind.

5. Umsetzung der Simulation

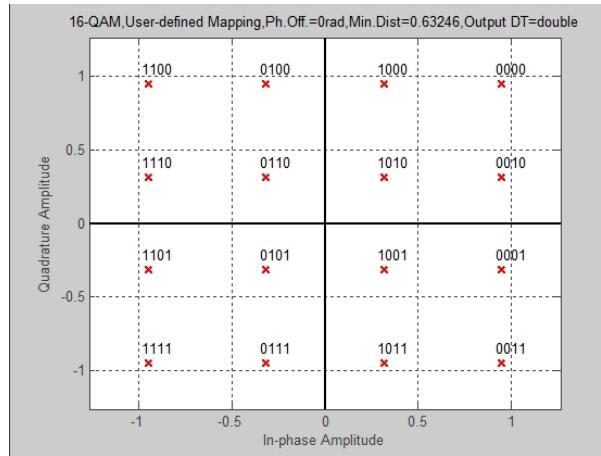


Abbildung 5.8.: Konstellationsdiagramm 16-QAM in Simulink

Durch die QAM werden in der Simulation keine binären Daten, sondern komplexe Zellen zur Weiterverarbeitung durch das Cell-Interleaving und die OFDM genutzt.

5.3.2.5. Cell-Interleaving

Das Cell-Interleaving besteht wie im Abschnitt 3.1.4 beschrieben aus zwei Stufen. Die Abbildung 5.9 zeigt den in Simulink genutzten Aufbau für die beiden unterschiedlichen Interleavingalgorithmen.



Abbildung 5.9.: Subsystem des MSC Cell-Interleavings

Für das immer stattfindende „short interleaving“ wurde wie beim Bit-Interleaving der *General Block Interleaver* verwendet. Hier wird nach dem selben Prinzip zunächst der Permutationsvektor *msc.cell_interleave_P* in der Datei *init_msc.m* nach dem in der Spezifikation beschriebenen Algorithmus erstellt und mit dem *General Block Interleaver* zur Vertauschung der durch die QAM generierten Zellen verwendet.

Der Faltungsinterleaver für das „long interleaving“ wird mit einem *General Multiplexed Interleaver* realisiert. Ist die Variable *long_interleaving* auf 1 gesetzt, so wird in der Datei *init_msc.m* der Verzögerungsvektor *msc.long_interleave_vec* erstellt. Hierfür wird ein linearer Vektor mit der Anzahl der QAM-Zellen über die dem Mode entsprechende Modulo-Operation verknüpft. Hierdurch entsteht ein Vektor, der die im Block verwendeten Schieberegister (*Interleaver delay*) als sich periodisch wiederholende Verzögerungskette

erstellt. Der für die Modi A-D bei einer Interleavingtiefe von 4 durch eine Modulo-5-Operation erstellte Vektor hat beispielsweise folgendes Aussehen:

$$\text{msc.long_interleave_vec} = [0 \ 1 \ 2 \ 3 \ 4 \ 0 \ 1 \ 2 \ 3 \ 4 \ 0 \ 1 \ 2 \ 3 \ 4 \ \dots]'$$

Jedes Element stellt einen Zweig der Verzögerungskette, wie in Abbildung 3.12 gezeigt, mit seiner entsprechenden Verzögerung dar und sorgt somit für eine Vertauschung der Zellen über die letzten 4 Frames hinweg.

Bei nicht ausgewählten „long interleaving“ ($\text{long_interleaving}=0$) wird der Permutationsvektor auf 0 gesetzt und somit die Funktion des Multiplex Interleavers deaktiviert.

5.3.3. OFDM

5.3.3.1. Pilotgenerator

Da die Pilotzellen (siehe Abschnitt 3.3) definierte Beträge, Phasen und Positionen innerhalb eines Übertragungsrahmens besitzen, können diese als konstante Variablen in die Simulation eingebbracht werden. Bei der Initialisierung werden die vier Pilottypen in der Datei *init_ofdm.m* als Matrizen mit der Größe eines Frames bzw. eines Superframes bei den AFS-Referenzzellen festgelegt und über die in Abbildung 5.10 dargestellte Struktur dem Modell hinzugefügt.

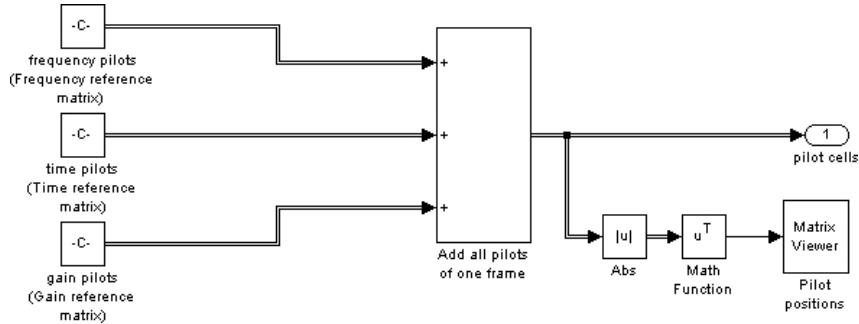


Abbildung 5.10.: Subsystem des Pilotgenerators

Die Pilot-Matrizen besitzen hierfür folgende Bezeichnungen:

```
ofdm.frequency_ref_array % Frequenzreferenzzellen
ofdm.time_ref_array % Zeitreferenzzellen
ofdm.gain_ref_array % Gainreferenzzellen
ofdm.AFS_ref_array % AFS-Referenzzellen
```

Alle nicht belegten Zellen in den Matrizen sind 0 und bei der Initialisierung muss darauf geachtet werden, dass manche Piloten zugleich als Frequenz-, Zeit und Gainreferenz

dienen, so dass bei der Erstellung der gesamten Pilotstruktur durch die Addition der Einzelmatrizen keine falschen Beträge entstehen. Die AFS-Piloten werden aufgrund ihrer Lage im Superframe erst beim nachfolgend beschriebenen Cellmapping hinzugefügt.

Die Lage und der Betrag der Pilotzellen wird über einen *Matrix Viewer* dem Nutzer der Simulation angezeigt. Die Abbildung 5.11 zeigt die Referenzstruktur, wie sie bereits theoretisch in Abbildung 3.20 erläutert wurde, für den Mode B mit der Spektralbelegung 0.

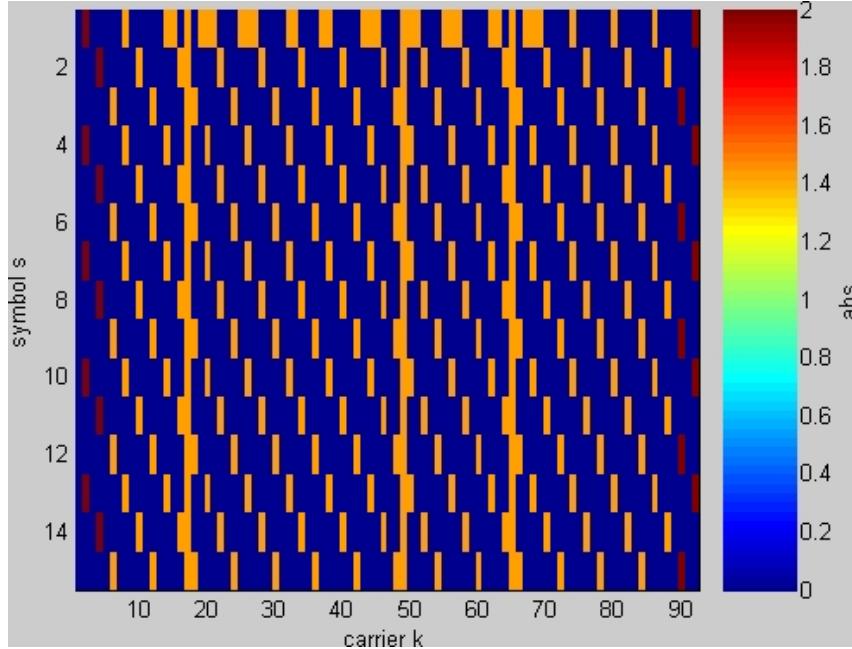


Abbildung 5.11.: Pilotstruktur in der Simulation (Mode B, $B_S=0$)

Man sieht hier die bereits in Abschnitt 3.3 beschriebene Anordnung der Piloten innerhalb eines Frames, wobei die an den Rändern liegenden verstärkten Gainreferenzzellen durch ihren Betrag von 2 deutlich erkennbar sind.

5.3.3.2. OFDM-Cellmapper

Der OFDM-Cellmapper dient der Zusammenfassung der drei Kanäle und der Pilotzellen. An dieser Stelle werden die durch das QAM erzeugten Zellen ihren Positionen innerhalb eines Frames bzw. eines Superframes zugewiesen und die in Abschnitt 2.2.3 beschriebene Rahmenstruktur gebildet. Die DRM-Spezifikation [Eur09, S.152 ff.] gibt hierfür die Lage der Kontrollzellen des FAC und SDC an. Die Datenzellen des MSC werden dann auf die noch zur Verfügung stehenden OFDM-Zellen in einem Superframe gelegt. Die Abbildung 5.12 zeigt die Simulinkstruktur des OFDM-Cellmappers. Die farbliche Kennzeichnung

5. Umsetzung der Simulation

orientiert sich dabei an der des Top-Levels, so dass die Bildung eines Übertragungsrahmens für alle einzelnen Kanäle nachvollziehbar ist.

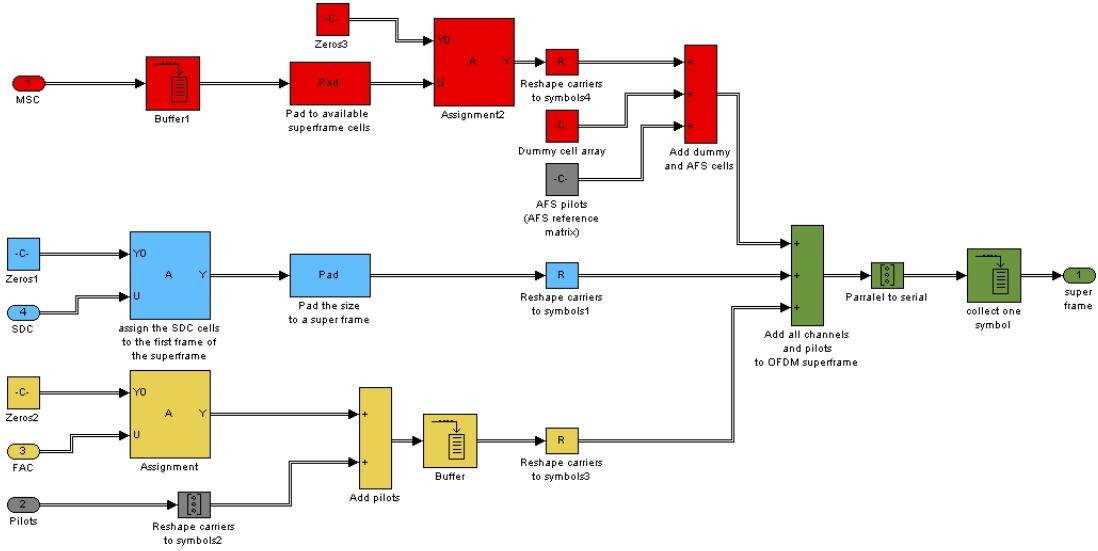


Abbildung 5.12.: Subsystem des OFDM-Cellmappings

Ein Rahmen baut sich in DRM wie in Abschnitt 3.2.4 beschrieben aus mehreren Symbolen mit einer festgelegten Anzahl aus Trägernummern auf. Die Spezifikation legt dabei fest, welche Zelle eines Kanals auf welchem Träger in einem Symbol liegt. Das Mapping, also das Anordnen der Zellen in einem Frame bzw. Superframe, findet in serieller Weise statt. D.h. ein Rahmen wird als Vektor mit nacheinander folgenden Symbolen betrachtet. Zu beachten ist bei der Zuweisung der Zellen, dass negative Trägerindizes definiert sind jedoch in MATLAB jegliche Indizierung von Elementen eines Vektors größer 0 sein müssen.

Nachfolgend sollen die Schritte bei der Umsetzung des Cellmappings, welches bei der Initialisierung in *init_ofdm.m* stattfindet, für die einzelnen Kanäle beschrieben werden.

Die FAC-Zellen werden in jedem Frame in Abhängigkeit vom Mode an festen Stellen stets oberhalb des DC-Trägers positioniert. Die Trägernummern werden im Standard [Eur09, S.153] in den Tabellen 97-101 für jeden der Modi definiert und in der Datei *fac_cells.m* festgehalten. Der Zuweisungsvektor *ofdm.FAC_cell_index* verteilt über den Block *Assignment* alle FAC-Zellen in einem Frame, welcher anschließend mit der ebenfalls als seriell umgewandelten gesamten Pilotstruktur addiert wird. Durch das *Buffer* werden alle für einen Superframe benötigten Frames gesammelt, da die beiden anderen Kanäle ausschließlich auf der Ebene eines Superframes verarbeitet werden. Anschließend werden die Symbole in die Rahmenstruktur über den Block *Reshape* parallelisiert und auf den für alle Kanäle zuständigen Addierer gegeben.

Die Zellen des SDC werden zunächst in gleicher Weise wie die FAC-Zellen über die

5. Umsetzung der Simulation

Variable `ofdm.SDC_cell_index` auf einen Frame verteilt. Die SDC-Zellen füllen dabei alle Träger der ersten Symbole auf, wie es in 3.2.4 dargestellt wird. Da der SDC nur jeweils im ersten Frame eines Superframes auftaucht, werden über den *Pad*-Block alle restlichen Zellen eines Superframes mit Nullen aufgefüllt. Auch hier wird anschließend eine Parallelisierung der Symbole und die Addition zu den anderen Kanälen vorgenommen.

Beim Mode E werden die AFS-Referenzen entsprechend ihrer definierten Position zugeordnet und füllen somit die in 3.3 beschriebenen Symbole vollständig aus.

Die Datenzellen des MSC werden letztlich auf alle restlichen, noch nicht belegten, Zellen eines Superframes nacheinander angeordnet. Der Zuweisungsvektor für die MSC-Zellen wird als `ofdm.MSC_cell_index` bei der Initialisierung erstellt.

Da ein Superframe mehr Zellenpositionen besitzen kann als durch die drei Kanäle und die Piloten entstehen, werden sogenannte Dummy-Zellen nach der in der Spezifikation dargestellten Tabelle [Eur09, S.138] am Ende eines Superframes noch hinzugefügt.

Nachdem alle OFDM-Zellen belegt sind würde nach der Theorie die IFFT über den parallelen Datenstrom folgen, wie in der Abbildung 3.14 gezeigt. Bei der Entwicklung des Empfängermodells stellte sich jedoch heraus, dass eine Verarbeitung über einzelne Symbole die Umsetzung bestimmter Algorithmen vereinfachte. Dadurch werden hier schon vor der IFFT die Zellen wieder in einen seriellen Datenstrom gewandelt und symbolweise gesammelt und verarbeitet. Dies hat keinerlei Auswirkung, da die in Gleichung 3.6 beschriebene Transformation sowieso für jedes einzelne Symbol gilt.

5.3.3.3. IFFT

Der Kern der OFDM-Verarbeitung, nämlich die Orthogonalisierung der Subträger in einem Symbol, findet in dem in Abbildung 5.13 gezeigten Subsystem statt. Alle hierfür benötigten Initialisierungen und Festlegungen werden in der Datei `init_fft.m` vorgenommen. Durch die IFFT wird die Verarbeitung des DRM-Systems im Basisband abgeschlossen.

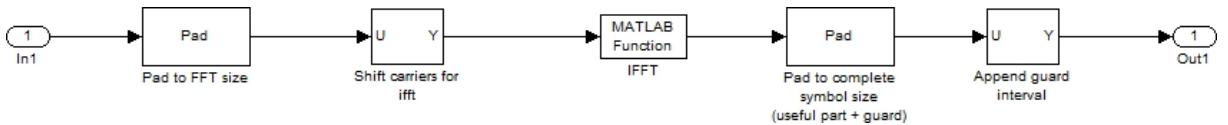


Abbildung 5.13.: Subsystem der IFFT

Zunächst wird die Größe eines Symbols mit K Subträgern auf die für die IFFT benötigten Abtastwerte N_U durch das Anfügen von Nullen erweitert. Die Größe wird dazu über die Variable `sFFT.size` bestimmt.

Über den Vektor `sFFT.shift_vec` werden mit Hilfe des *Selectors* die Träger nach der in Gleichung 3.12 angegebenen Beziehung in einem Symbol verteilt.

Die anschließende IFFT konnte nicht mit dem gleichnamigen Simulinkblock realisiert werden, da dieser nur Zweierpotenzen als Eingangsgröße verarbeiten kann. Deshalb wird

auf die externe *MATLAB Function*: *ifft* zurückgegriffen, die eine Verarbeitung jeglicher Größen gestattet.

Nachdem das Symbol damit in den Zeitbereich transformiert wurde, folgt das Anfügen des in Abschnitt 3.2.3 beschriebenen Schutzintervalls, indem zunächst auf die Gesamtgröße N_S mit Hilfe der Variablen *sFFT.symbol_length* erweitert wird und dann die letzten N_G Abtastwerte des ursprünglichen Symbols durch den Vektor *sFFT.guard_index* an den Anfang kopiert werden.

5.3.4. Zwischenfrequenz-Modulation

Nachdem die IFFT über die einzelnen Symbole durchgeführt wurde, gelangt das Signal zur ZF-Modulation. Dies stellt den letzten in der Simulation umgesetzten Teil im Sender dar. Die ZF-Modulation teilt sich in die beiden im Abschnitt 3.4 beschriebenen Vorgänge, welche in Abbildung 5.14 als separate Subsysteme dargestellt werden.

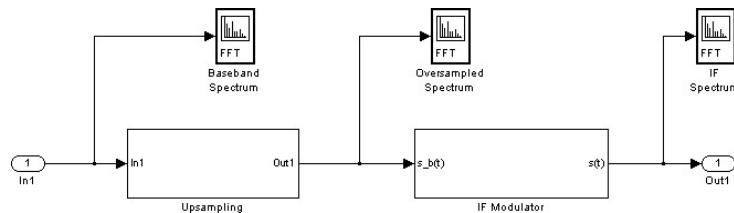


Abbildung 5.14.: Subsystem der ZF-Modulation

Die drei erkennbaren *Spectrum Scopes* dienen in der Beschreibung der Subsysteme zur beispielhaften Darstellung eines ZF-Modulationsvorganges. Alle Initialisierungsvorgänge, die für die ZF-Modulation von Bedeutung sind, werden in der Datei *if_modulation.m* vorgenommen.

5.3.4.1. Überabtastung

Vor der eigentlichen Modulation muss das Signal, wie in Abschnitt 3.4.1 erläutert, zunächst auf eine höhere Abtastrate konvertiert werden. Dieser Vorgang teilt sich in die Interpolation und die folgende Filterung zur Unterdrückung von Images und wird in Abbildung 5.15 abgebildet.

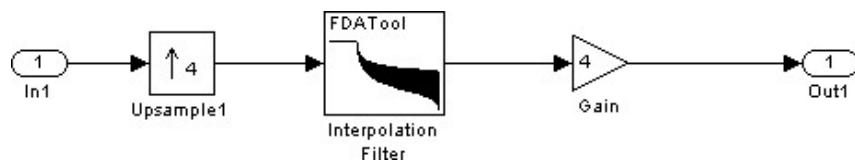


Abbildung 5.15.: Subsystem der Überabtastung

5. Umsetzung der Simulation

Die Interpolation wird mit dem im *Signal Processing Blockset* vorhandenen *Upsample*-Block durchgeführt, der je nach Faktor Nullen zwischen die Abtastwerte einfügt. Der Faktor wird als *if_mod.upsample_factor* in der Simulation zunächst immer mit 4 definiert, kann aber bei weiteren Tests geändert werden.

Zur Filterung des überabgetasteten Signals wird mit Hilfe des *FDATools* ein festes Antiimaging-Filter entworfen, dessen Parameter bei der Initialisierung festgelegt werden. Es wird ein Nyquist-Filter eingesetzt, was stets die erste Nyquistbedingung erfüllt und somit eine Überlagerung angrenzender Signale und der daraus folgenden ISI verhindert. Nach der Filterung muss der Betrag des Signals um den Interpolationsfaktor verstärkt werden,

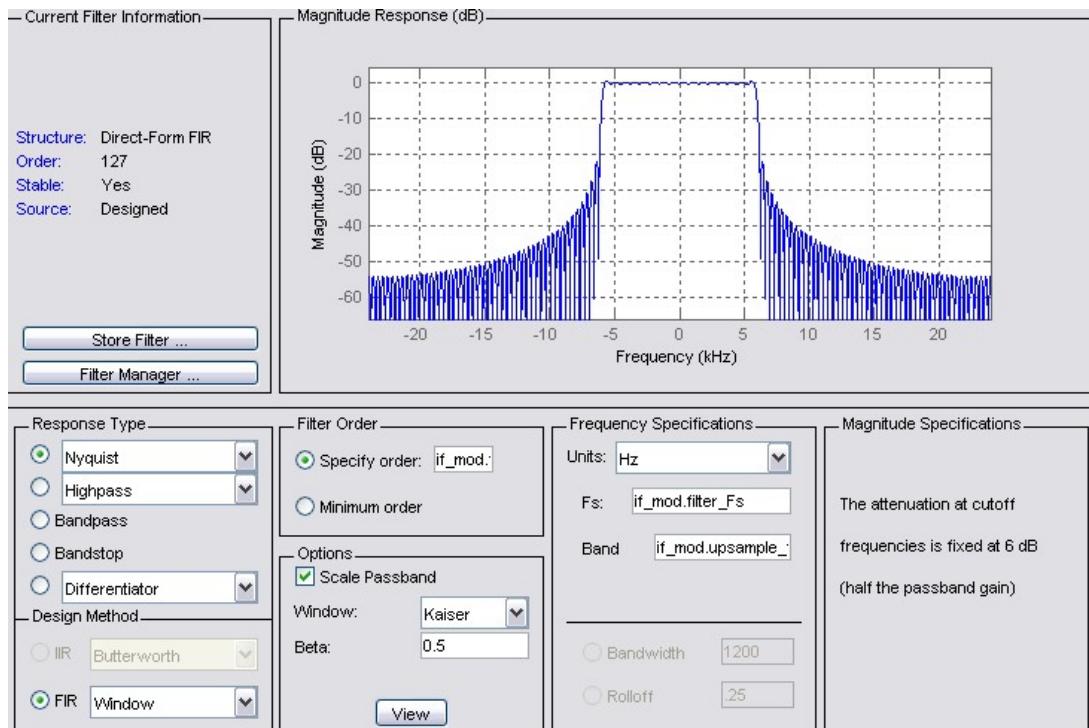


Abbildung 5.16.: Entwurf des Antiimaging-Filters

In der Abbildung 5.16 wird der Entwurf des Filters für die Modi A bis D bei einer B_S von 0 bis 3 dargestellt. Die Abtastfrequenz F_S wird als die für die Überabtastung definierte Frequenz mit der Variable *if_mod.filter_Fs* eingestellt und mit *Band* wird der Interpolationsfaktor, hier $I=4$, festgelegt. Experimentell wurde das Kaiser-Fenster und eine Filterordnung von 128 bestimmt, welche sich bei der Simulation und der anschließenden Verifizierung des Sendermodells als günstig herausstellten. Im Betragsspektrum des Filters ist ersichtlich, dass es einen Durchlassbereich bis 6 kHz besitzt, was der halben ursprünglichen Abtastrate von 12 kHz entspricht und somit das um den Gleichanteil

5. Umsetzung der Simulation

liegende Band beibehält während es alle anderen durch die Überabtastung entstandenen Bänder (Images) unterdrückt.

Die Abbildung 5.16 zeigt die beiden Spektren vor (links) und nach (rechts) der Überabtastung, wobei die stark unterdrückten Imagebänder bei den Vielfachen der ursprünglichen Abtastrate noch erkennbar sind.

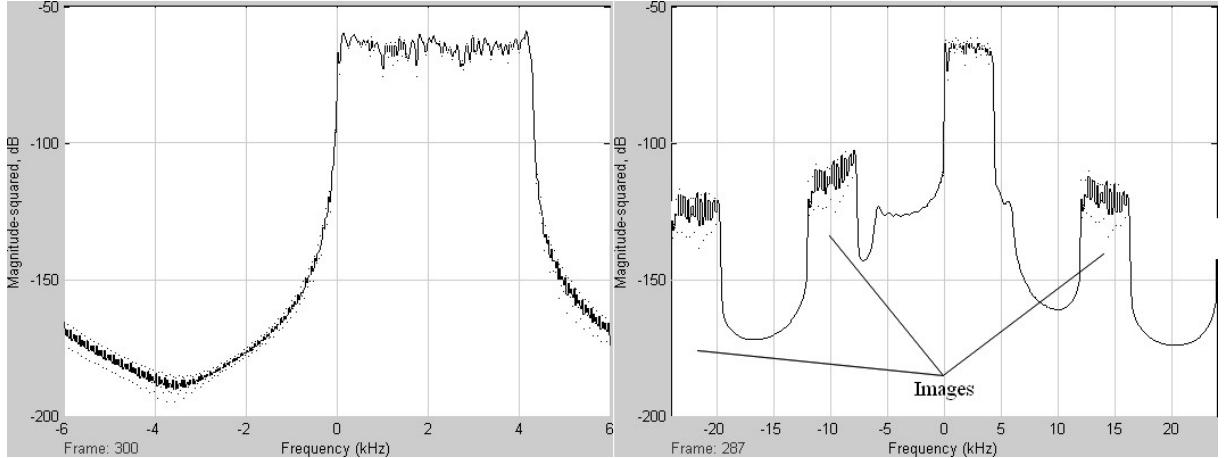


Abbildung 5.17.: Spektrum des Ursprungsignals und des überabgetasteten Signals

5.3.4.2. IQ-Modulator

Die Modulation auf die Zwischenfrequenz wird in Abbildung 5.18 gezeigt, wobei das im Abschnitt 3.4.2 beschriebenen Prinzip des IQ-Modulators verwendet wurde. Die Einstellung der beiden Signalgeneratoren erfolgt in der Initialisierung.

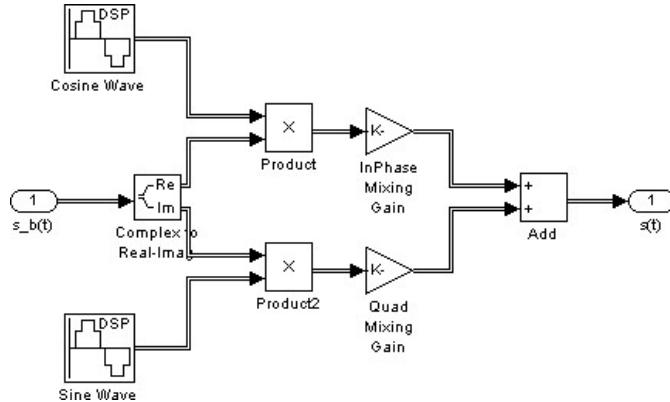


Abbildung 5.18.: Subsystem des IQ-Modulators

Mit `if_mod.if_freq` wird die ZF, mit `if_mod.sample_time` wird die neue Abtastrate und mit `if_mod.frame_size` wird die um den Interpolationsfaktor vergrößerte Symbollänge als

5. Umsetzung der Simulation

Kenngröße übergeben. Die beiden Verstärkungsfaktoren mit dem Wert $\sqrt{2}$ dienen dem Ausgleich der Beträge von Real- und Imaginärteil.

Die Abbildung 5.19 zeigt die Spektren vor (links) und nach (rechts) dem IQ-Modulator. Das zuvor überabgetastete Signal im Basisband liegt nach dem Modulator auf der Zwischenfrequenz.

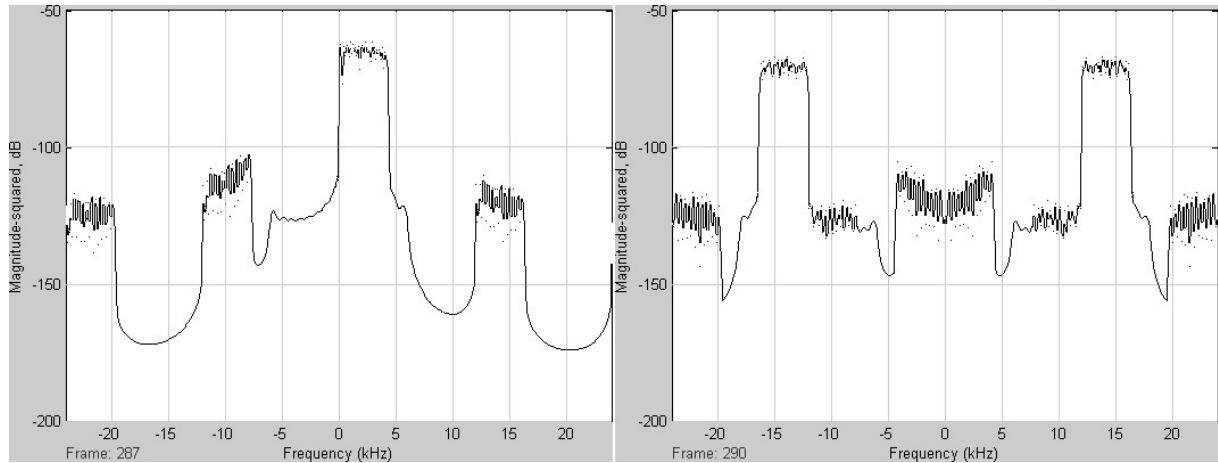


Abbildung 5.19.: Spektrum im Basisband (überabgetastet) und nach der ZF-Modulation

5.4. Verifizierung des Sendermodells

Um die korrekte Funktionsweise des Sendermodells zu überprüfen und somit die Übereinstimmung der Umsetzung zur DRM-Systemspezifikation nachzuweisen, werden Audiodateien über Simulink erzeugt und in verschiedenen für den DRM-Empfang existierenden Tools verarbeitet. Hierzu wird ein separates Modell erstellt, welches sich auf dem Datenträger unter der Bezeichnung „DRM_wav“ befindet. Folgende Programme wurden für die Überprüfung der standardgerechten Umsetzung genutzt:

- Diorama - DRM-Empfänger der Universität Kaiserslautern auf Basis von MATLAB [SDSGU05]
- Dream - freier und quelloffener DRM-Empfänger in C++ [o.A10]

Die Abbildung 5.20 zeigt das für die Verifizierung verwendete Modell. Bis auf die Änderung der Datenquellen und dem zusätzlichen Hinzufügen des Blocks zur Erzeugung der Audiodatei entspricht das Modell vollständig dem in der Simulation genutzten und im vorherigen Abschnitt beschriebenen Sendermodell des DRM-Systems.

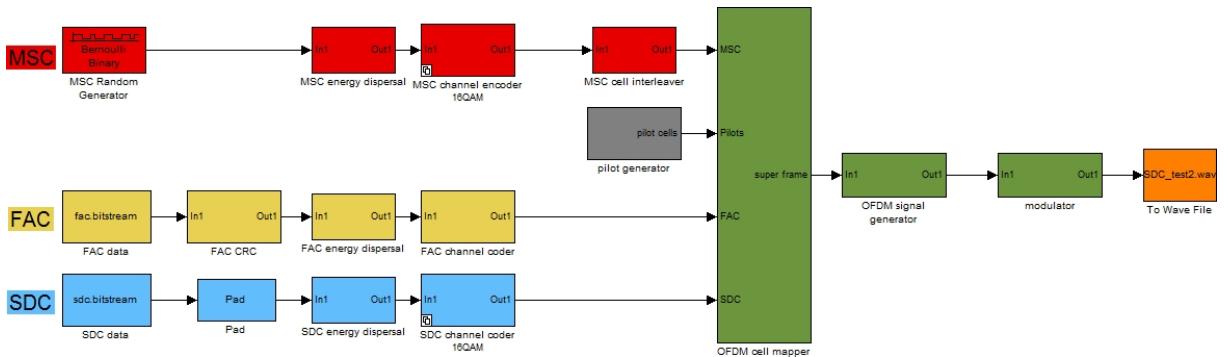


Abbildung 5.20.: Modell zur Überprüfung des Sendermodells auf standardgerechte Umsetzung

Die Datenquelle wird zur Überprüfung nicht mehr als Zufallsgenerator eingesetzt, sondern es werden konkrete Informationen als Bitstream für FAC und SDC bei der Initialisierung generiert. Für den FAC werden hierzu alle zur Überprüfung benötigten Parameter in der Datei *init_fac.m* nach der in der DRM-Spezifikation [Eur09, S.72 ff.] festgelegten Reihenfolge als Bitstream belegt. Neben den Parametern, welche vom Nutzer bei der Initialisierung 5.2 gesetzt werden, können an dieser Stelle noch weitere für den FAC definierte Zusatzinformationen angegeben werden, die für die Überprüfung des Modells nicht von Bedeutung sind. Da bei der CRC-Auswertung des SDC mit einer Empfangssoftware jedoch die Lage der einzelnen Frames im Superframe signalisiert werden muss, erfolgt eine Festlegung der im Abschnitt 2.2.3 beschriebenen „identity flag“. In der Simulation wird

5. Umsetzung der Simulation

nach dem Bitstream noch die in 5.3.1.2 erläuterte CRC für den FAC durchgeführt und an diesen angefügt.

Die Daten des SDC werden im Gegensatz zum FAC nicht direkt nach dem in der Spezifikation [Eur09, S.78 ff.] festgelegten Rahmenaufbau spezifiziert. Die SDC-Informationen sollen dem Empfänger die Quellenkodierung der Dienste des MSC signalisieren. Da in der Simulation keine Quellenkodierung umgesetzt wird, werden diese Informationen in der Datei *init_sdc.m* als Nullen abgelegt und nach der in [Eur09, S.78] definierten Anordnung, wird die CRC-Prüfsumme über diesen Nullvektor gebildet. Die Größe des Datenfeldes des SDC wird in der Spezifikation in Tabelle 56 [Eur09, S.79] dargestellt. Da der CRC-Generator des SDC sich nicht in einfacher Weise wie beim FAC im Modell implementieren lässt, wird zur Hilfe die bereits in Diorama umgesetzte Funktion *crc16(...)* in der Implementierung eingesetzt. Der fertige SDC-Bitstream wird über den *Pad-Block* noch auf die festgelegte Länge erweitert, in dem zusätzliche Nullen angefügt werden.

Die vom, in Abbildung 5.20 dargestellten, Sendermodell erzeugten Daten werden nach dem Zwischenfrequenzmodulator als wav-Datei mittels dem Simulink Block *To wave file* abgespeichert. Es werden alle Modi bis auf den Mode E, der nicht von den Tools unterstützt wird, mit allen möglichen Spektralbelegungen erzeugt und abgelegt. Das als 16 Bit festgelegte Format der wav-Datei richtet sich dabei nach den geforderten Eingabeformat der Software. Die Abtastfrequenz entspricht der des Simulationsmodells nach der Überabtastung.

Folgende Abbildungen zeigen die Anzeige von Diorama bei der Überprüfung einer aufgezeichneten Datenübertragung. Die Abbildung 5.21 wird als „Signal Information“ bezeichnet und stellt alle vom FAC und SDC übermittelten Informationen des empfangenen DRM-Signals, sowie die Korrektheit der CRC-Prüfsumme der Kanäle dar. Zusätzlich werden Synchronisations- und Kanaldaten angezeigt, die an dieser Stelle bei der Auswertung und Überprüfung keine Rolle spielen. Die dargestellten Informationen wie Mode, Bandbreite, Interleavingtiefe und QAM-Modulation entsprechen in diesem Fall denen, welche beim Modell eingestellt wurden. Zudem wird über die grünen Indikatoren „FAC CRC“ und „SDC CRC“ die CRC-Prüfsumme der beiden Kanäle als korrekt markiert, so dass die Implementierung des Sendermodells in diesem Fall als standardgerecht angesehen werden kann.

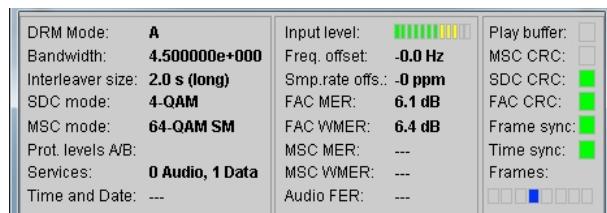


Abbildung 5.21.: Diorama: Überprüfung der CRC der Testdatei

5. Umsetzung der Simulation

Die Abbildung 5.22 stellt das bei Diorama als „Input spectrum“ bezeichnete Fenster zur Überprüfung des Spektrums des Empfangssignals dar. Man erkennt, dass für die in Darstellung 5.21 angezeigte Spektralbelegung von 4,5 kHz ein rechtseitiges Spektrum auf einer ZF von 12 kHz liegt. Damit kann die im Abschnitt 5.3.4 beschriebene Modulation ebenfalls als richtig betrachtet werden.

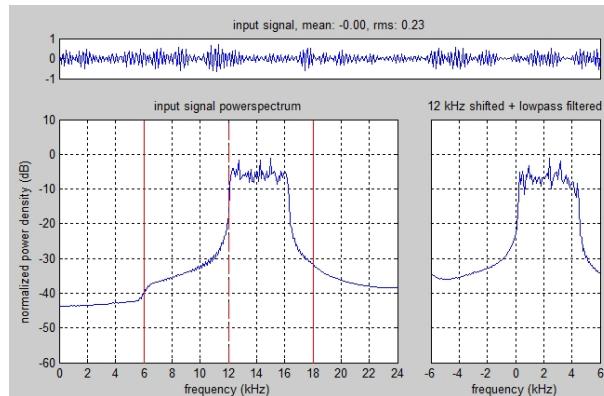


Abbildung 5.22.: Diorama: Empfangsspektrum

Es wurden für die Modi A bis D bei jeglichen Spektralbelegungen und QAM-Typen positive Testergebnisse erzielt. Somit kann die Annahme getroffen werden, dass das Sendermodell dem DRM-Standard für die umsetzbaren Parameter (SM, EEP) entspricht. Das im folgenden Abschnitt beschriebene Empfängermodell kann damit ebenfalls als standardgerecht betrachtet werden, wenn eine simulationsinterne Fehlerüberprüfung zwischen Sender und Empfänger zu keinem negativen Ergebnis führt.(siehe Abschnitt 5.5.4)

5.5. Empfängermodell

Das in Abbildung 5.23 dargestellte Empfängermodell im Top-Level stellt bis auf die in lila gekennzeichneten Synchronisationssysteme eine inverse Darstellung des Sendermodells dar.

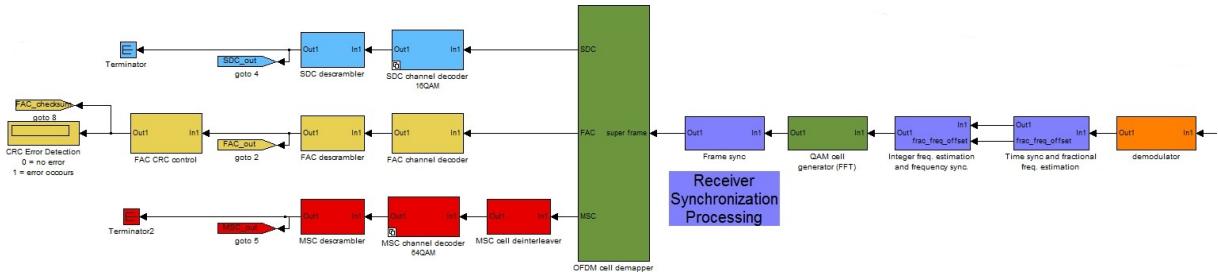


Abbildung 5.23.: Empfängermodell in Simulink

Die Entwicklung der Simulation erfolgte iterativ, indem zunächst für jede im Sender umgesetzte Komponente des DRM-Systems eine Empfangskomponente parallel dazu aufgebaut wurde. Die ständige und schrittweise Überprüfung, ob eine richtige Umsetzung der Komponenten im Empfänger erfolgt, wurde über die im Abschnitt 5.5.4 beschriebene BER-Kalkulation durchgeführt. Durch diese Vorgehensweise wird nach der Verifizierung des Sendermodells die Korrektheit und die dem DRM-Standard entsprechende Umsetzung des Empfängers gewährleistet. D.h. ohne Übertragungskanal zwischen Sendermodell und Empfängermodell ist eine standardgetreue Simulation des DRM-Übertragungssystems entstanden.

Mit der Hinzunahme des im Abschnitt 5.6 beschriebenen Kanalmodells kann keine fehlerfreie Simulation durchgeführt werden, ohne dass eine Umsetzung der lila dargestellten Komponenten zur Synchronisation und zum Kanalausgleich erfolgt.

Die einzelnen Subsysteme, die im Empfänger als Dekodier- bzw. Demodulationsalgorithmen der im Sender umgesetzten DRM-Algorithmen aufgebaut wurden, werden nachfolgend einzeln erläutert. Die Beschreibung der Komponenten zur Synchronisation findet im Abschnitt 5.7 statt.

5.5.1. Zwischenfrequenz-Demodulation

Wie bei der ZF-Modulation im Sender, besteht die Demodulation aus den in der folgenden Abbildung 5.24 dargestellten zwei Teilen. Erst wird das empfangene Signal durch den IQ-Demodulator zurück ins Basisband gebracht und anschließend wird die ursprüngliche Abtastfrequenz durch die Unterabtastung wiederhergestellt.

5. Umsetzung der Simulation

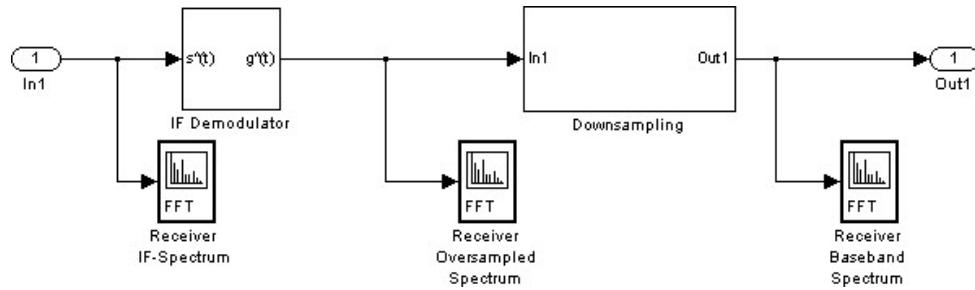


Abbildung 5.24.: Subsystem der ZF-Demodulation

5.5.1.1. IQ-Demodulator

Der IQ-Demodulator in Abbildung 5.25 entspricht bis auf die beiden Verstärkungsfaktoren der Prinzipdarstellung im Abschnitt 3.4.2.

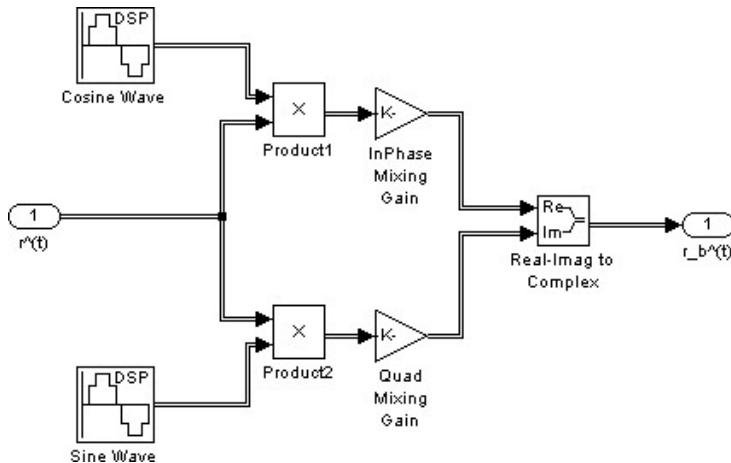


Abbildung 5.25.: Subsystem der IQ-Demodulation

Das empfangene Signal wird über die beiden Generatoren, deren Einstellungen identisch sind mit denen des IQ-Modulators, mit der ZF multipliziert. Dadurch entstehen Real- und Imaginärteil des OFDM-Signals im Basisband. Da hier die beiden Teile wieder zu einem komplexen Signal zusammengeführt werden, erfolgt auch hier ein Ausgleich der Beträge mit $\sqrt{2}$.

5.5.1.2. Unterabtastung

Das sich nun im Basisband befindliche Empfangssignal wird durch die Unterabtastung zurück zur ursprünglichen Abtastrate konvertiert. Hierzu wird zunächst eine Antialiasing-Filterung durchgeführt, wobei mit dem gleichen Filter wie bei der Überabtastung (Abschnitt 5.3.4) gearbeitet wird.

5. Umsetzung der Simulation

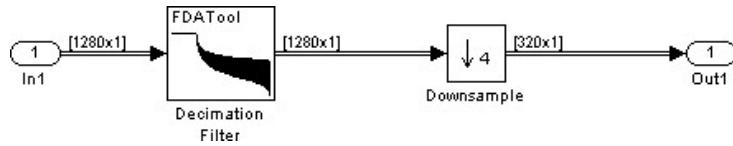


Abbildung 5.26.: Subsystem der Unterabtastung

Der in Simulink befindliche Block *Downsample* führt eine Dezimation um den Faktor D (*if_mod.upsample_factor=4*) durch und kann als Gegenstück zum Interpolator betrachtet werden. Er reduziert durch Weglassen von D-1 Samples die Framegröße auf 1/D, was zur Reduktion der Samptetime im Modell und damit der Abtastfrequenz führt.

5.5.2. OFDM

5.5.2.1. FFT

Das sich nach wie vor im Zeitbereich befindliche Signal muss durch die FFT zurück in den Frequenzbereich transformiert werden, so dass die QAM-Zellen der einzelnen Kanäle wieder in ihrer ursprünglichen Anordnung vorzufinden sind.

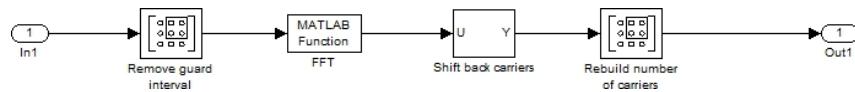


Abbildung 5.27.: FFT Subsystem

Die Abbildung 5.27 zeigt den Ablauf des FFT-Prozesses der OFDM im Empfänger, wie er auch bei der theoretischen Betrachtung im Abschnitt 3.2.1 dargestellt wurde. Als erstes wird das Schutzintervall entfernt, in dem mit dem *Submatrix*-Block nur der Nutzanteil des OFDM-Symbols selektiert wird. Die FFT selbst wird wieder über eine externe MATLAB-Funktion durchgeführt, da auch hier der gleichnamige Simulink-Block nur eine Verarbeitung über Zweierpotenzen gestattet. Nachdem das Symbol zurück transformiert wurde, wird die ursprüngliche Anordnung der Subträger über den *Selector* mit Hilfe des Vektors *sFFT.backshift_vec* wiederhergestellt und die zusätzlichen Nullen, die zum Auffüllen der IFFT-Größe benötigt wurden, entfernt. Somit werden alle Subträger f_k mit negativen Index wieder links vom DC-Träger angeordnet, so dass die Indizierung die folgende ursprüngliche Form besitzt:

$$k = [K_{min}, \dots, -1, 0, 1, \dots, K_{max}]$$

Damit liegt nun ein Symbol mit K Trägern vor, welches über das folgende OFDM-Demapping wieder in die einzelne Kanäle und Piloten zerteilt werden kann.

5.5.2.2. OFDM-Demapper

In der Abbildung 5.28 ist zu sehen, dass sich das OFDM-Demapping wesentlich leichter gestaltet als das im Sender durchgeföhrte OFDM-Mapping. Zunächst kommt es zur Parallelisierung der einzeln zurück transformierten Symbole, indem durch den *Buffer* alle für einen Superframe benötigten Symbole gesammelt werden. Es folgt die Kanalschätzung über einen Frame, welche im Abschnitt 5.7.4 behandelt wird und zunächst für die Beschreibung des Empfängermodells ohne zusätzlichen Kanal keine Bedeutung besitzt.

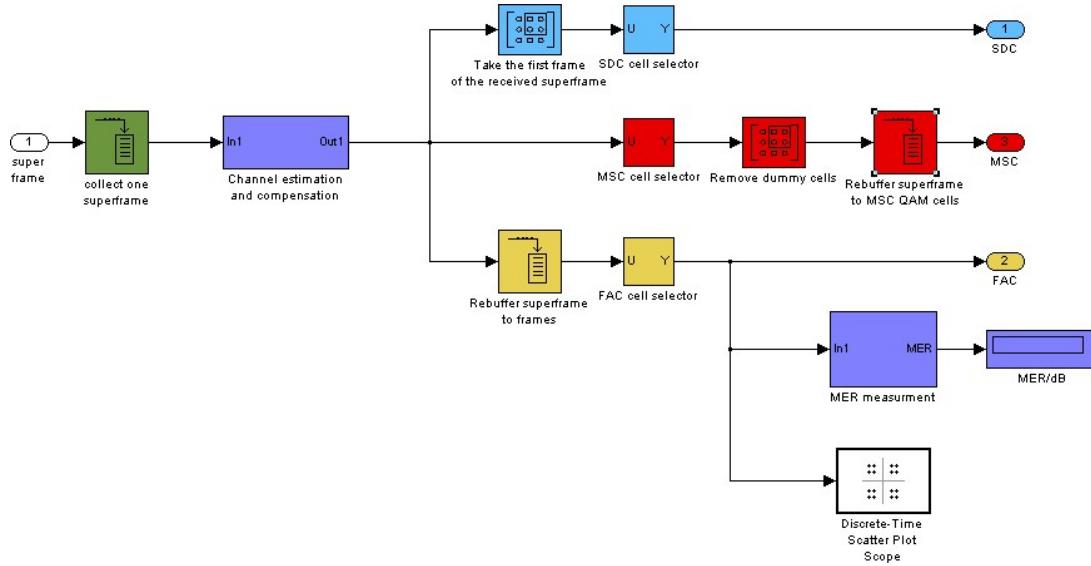


Abbildung 5.28.: Subsystem des OFDM-Demappers

Da sich die Zellen des SDC nur im ersten Frame befinden, wird dieser im SDC-Zweig selektiert. Über den Anordnungsvektor *ofdm.SDC_cell_index* werden die SDC-Zellen aus dem Frame extrahiert. Beim MSC werden zunächst alle Zellen aus dem Superframe mit Hilfe von *ofdm.MSC_cell_index* ausgewählt, die Dummy-Zellen entfernt und anschließend die Anzahl der QAM-Zellen für einen Frame ausgelesen. Die in fester Struktur angeordneten FAC-Zellen können frameweise auch hier mit dem im OFDM-Mapping verwendeten Vektor *ofdm.FAC_cell_index* ausgewählt werden. Die damit aus den Subträgern ausgelesenen Zellen der einzelnen Kanäle werden damit wieder in separaten Zweigen der Kanaldekodierung zugeführt.

Alle Pilotzellen müssen an dieser Stelle nicht zusätzlich aus der Rahmenstruktur ausgelesen werden. Sie finden Verwendung in ihren angedachten Aufgaben im Empfänger, dem Synchronisieren und der Kanalschätzung (siehe 5.7).

Die beiden zusätzlichen nach dem FAC-Demapping angefügten Systeme stellen Analysewerkzeuge zur Auswertung dar und werden im Abschnitt 5.5.4 erläutert.

5.5.3. Demodulation und Kanaldekodierung

5.5.3.1. Cell-Deinterleaving

Der Cell-Deinterleaver wird wie auch der Cell-Interleaver nur für den MSC verwendet. Die Struktur in Abbildung 5.29 zeigt, dass an dieser Stelle einfach die in Simulink vorhandenen Deinterleaver-Blöcke zu den im Sender eingesetzten Interleaver Verwendung finden.



Abbildung 5.29.: Subsystem des MSC Cell-Deinterleavers

Sowohl der *General Block Deinterleaver* als auch der *General Multiplexed Deinterleaver* erhalten die selben Initialisierungsvektoren wie ihr Pendant im Sendezweig. Hierdurch werden beide Vertauschungen („long und short interleaving“) rückgängig gemacht. Da wie im Abschnitt 3.1.4 beschrieben der Faltungsdeinterleaver eine um $D-1$ Frames verzögerte Ausgabe der MSC-Zellen verursacht, muss darauf bei der Auswertung der Daten im Abschnitt 5.5.4 geachtet werden.

5.5.3.2. Kanaldekodierung

Zur Kanaldekodierung wird, wie in Abbildung 5.30 zu sehen, eine identische Struktur wie bei der Kanalkodierung im Sender verwendet. An dieser Stelle könnten jedoch auch andere Ansätze zur Dekodierung umgesetzt werden, wie in [SDSGU05]. Jedoch wird mit der gleichen Struktur in der Simulation eine gewisse Übersicht für den Nutzer gewahrt, da jeder Schritt im Sender genau in umgekehrter Reihenfolge im Empfänger geschieht.

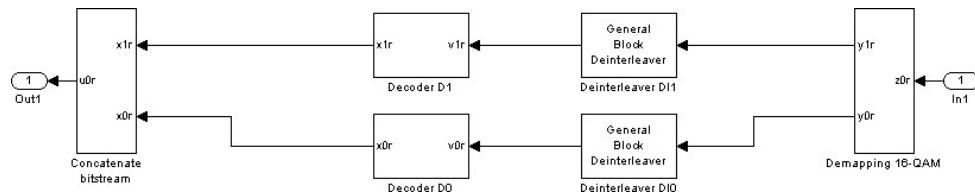


Abbildung 5.30.: Subsystem zur Kanaldekodierung für den SDC mit einer 16-QAM

Da der Decoder hierdurch ebenfalls eine Multilevel-Kodierung aufweist wie der Encoder, werden ebenfalls Bibliotheken (*SDC_decoder.lib* und *MSC_decoder.lib*) für SDC und MSC erstellt. Diese werden nach dem in Abschnitt 5.3.2.2 vorgestellten Prinzip angelegt und initialisiert.

5.5.3.3. QAM-Demapping

Beim QAM-Demapping werden die komplexen Symbole je nach Lage im Konstellationsdiagramm wieder zu binären Daten zurückgewandelt. Es wird also eine QAM-Demodulation durchgeführt.

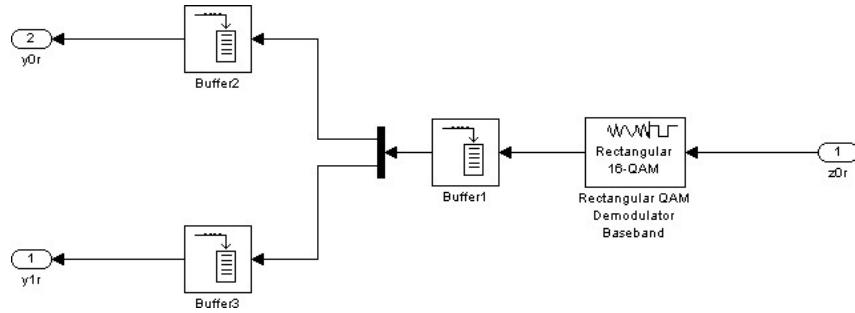


Abbildung 5.31.: Subsystem des QAM-Demapping bei einer 16-QAM

Wie in Abbildung 5.31 zu sehen, wird der Simulink-Block *Rectangular QAM Demodulator Baseband* als Umkehrung des QAM Modulators im Sender genutzt. Es werden an dieser Stelle die gleichen Parameter für das Konstellationsdiagramm und die Normierung eingesetzt, so dass das in Abbildung 5.8 dargestellte Diagramm auch hier verwendet wird. Als Entscheidungstyp (*Decision type*) wird eine einfache *Hard decision* eingesetzt, welche keine Schätzalgorithmen verwendet, sondern den Symbolräumen im Konstellationsdiagramm feste Grenzen zuweist.

Die Buffer dienen zum Wiederherstellen der Reihenfolge der Bits bevor sie durch die in der Spezifikation [Eur09, S.132] festgelegten Anordnung im QAM-Mapper aufgrund der MLC vertauscht worden sind.

5.5.3.4. Bit-Deinterleaving

Das Bit-Deinterleaving wird mit dem *General Block Deinterleaver* durchgeführt. Hierzu werden die gleichen Permutationsvektoren für der einzelnen Kanäle wie beim Bit-Interleaving des Senders verwendet. Der Deinterleaver stellt die ursprüngliche Anordnung des Bitstroms vor der Dekodierung durch den Viterbi-Dekoder wieder her.

5.5.3.5. Viterbi-Decoder

Der Viterbi-Decoder stellt wie in Abschnitt 3.1.2.3 beschrieben das Gegenstück zum Faltungscoder dar. Der Dekodierprozess unterteilt sich in drei Abschnitte, wie in Abbildung 5.32 dargestellt wird. Es erfolgt zuerst eine Depunktierung des Bitstroms, um die ursprüngliche vom Faltungsencoder erzeugte Datenmenge wiederherzustellen. Der Viterbi-Decoder

reduziert über seinen Algorithmus den Bitstrom um 1/6 und durch die anschließende Entfernung der Abschluss-Bits stehen die als Eingangsframe definierten Größen (siehe [Eur09, S.192 ff.]) der drei Kanäle wieder zur Verfügung.

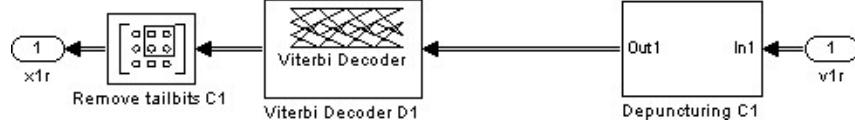


Abbildung 5.32.: Subsystem der Viterbi-Dekodierung

Das Pendant zur Punktierung ist das Einfügen von Nullen an den weggelassenen Stellen der Punktierungsraster. Wie in der folgenden Abbildung dargestellt, wird dies über den Block *Insert Zero* für die getrennte Verarbeitung von Tailbits und Nutzbits bewerkstelligt.

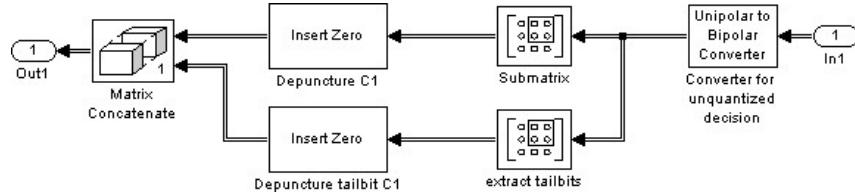


Abbildung 5.33.: Subsystem zur Depunktierung des Datenstroms

Da der Viterbi-Algorithmus wissen muss, an welchen Stellen eine 1 oder eine 0 übertragen wurde, wird mit dem *Unipolar to Bipolar Converter* sichergestellt, dass jede 1 zu -1 und jede 0 zu 1 konvertiert wird. Der Viterbi-Algorithmus wird mit dem *Unquantized*-Entscheidungstyp betrieben und kann somit zwischen den Nullen, die zum Datenstrom gehören, und den Nullen, die durch die Depunktierung eingefügt werden, unterscheiden.

Nach dem der Viterbi-Decoder den ursprünglichen Frame mit den Abschluss-Bits wieder gewonnen hat, werden diese vom eigentlichen Nutzteil über die *Submatrix* abgetrennt.

5.5.3.6. Descrambler

Der Descrambler im Empfänger wird über das gleiche Prinzip, wie das im Abschnitt 5.3.2.1 für den Scrambler vorgestellte und verwendete Subsystem umgesetzt. Die Parameter aus der Datei *energy_dispersal.m* sind dabei identisch. Nach der Rückverwürfelung sind die Daten aller Kanäle, wenn keine Fehler auftraten, identisch mit denen der Zufallsgeneratoren für die Datenerzeugung.

5.5.4. Datenempfang und Auswertung

Da keine Quellendekodierung nach dem Descrambler folgt, werden die Daten an dieser Stelle ausgewertet. Bis auf die CRC-Prüfung beim FAC werden alle Kanäle an dieser Stelle

mit den Eingangsdaten verglichen und anschließend terminiert. Dieser Abschnitt soll die BER-Prüfung der drei Kanäle darstellen und die beiden im OFDM-Demapper gezeigten Strukturen zur Analyse näher erläutern.

5.5.4.1. CRC

Wie auch im Sender wird ausschließlich beim FAC eine zusätzliche CRC-Einheit verwendet. Diese dient hier zur Auswertung des Datenstroms, indem die CRC-Prüfsumme erneut über den FAC-Frame gebildet wird und ein Vergleich mit der angehängten Prüfsumme des Senders stattfindet. Die Abbildung 5.34 zeigt die Struktur zur Auswertung der CRC.

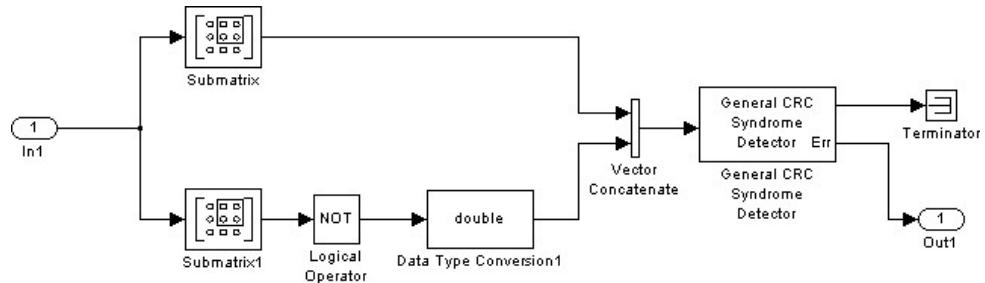


Abbildung 5.34.: Subsystem zur Auswertung der CRC-Prüfsumme beim FAC

Hierbei wird zunächst in umgekehrter Weise als beim Sender die CRC-Prüfsumme aufgetrennt, negiert und anschließend wieder an den Datenstrom angefügt. Über den *General CRC Syndrom Detector* werden die Prüfsummen des Senders und des Empfängers verglichen. Sollten diese identisch sein, so wird ein Ergebnis von 0 geliefert, was somit keinen Fehler im CRC nachweist. Eine 1 am Ausgang des Prüfsignals *FAC_checksum* würde eine Ungleichheit der beiden Prüfsummen anzeigen und somit einen fehlerhaften Empfang der Daten bestätigen.

5.5.4.2. BER-Messung

Die BER-Auswertung soll für alle drei Kanäle die Bitfehler, die zwischen Sender und Empfänger entstanden sind, ermitteln. Die Abbildung 5.35 zeigt einen Auszug aus den verwendeten Subsystemen.

5. Umsetzung der Simulation

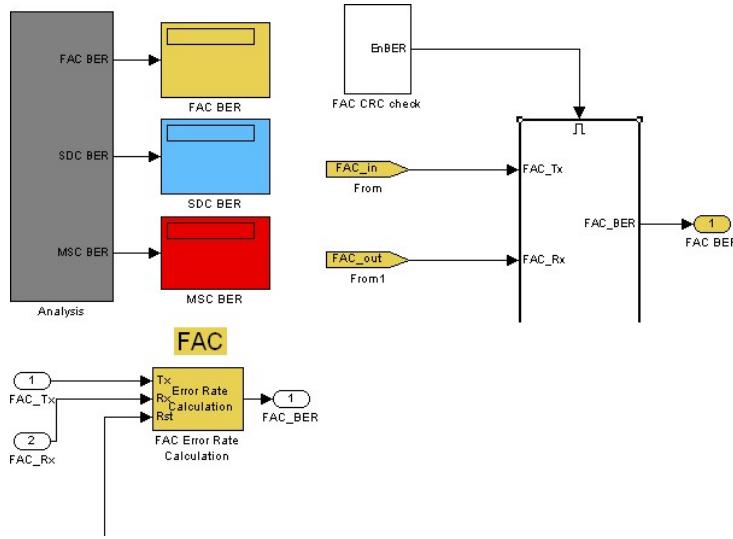


Abbildung 5.35.: Teile der Subsysteme zur BER-Auswertung

Die relativ komplizierte Struktur dieser Auswertung beruht darauf, dass zum einen die BER-Messung erst beginnen soll, wenn die im folgenden Abschnitt beschriebenen Synchronisationsvorgänge abgeschlossen sind, und zum anderen den Nutzer der Simulation die Möglichkeit eingeräumt wird, die Messung jederzeit zurückzusetzen. Das Zurücksetzen wird über den in der Abbildung gezeigten manuellen Schalter ermöglicht. Der Startzeitpunkt der BER-Messung wird über die CRC-Kontrolle bestimmt. Da davon auszugehen ist, dass es bei fehlender Synchronisation oder bei zu starkem Rauschen kein fehlerfreier Datenempfang möglich ist, wird die Messung erst gestartet, wenn die CRC-Kontrolle keine Fehler ergibt und die Prüfvariable *FAC_checksum = 0* ist.

Zur Ermittlung des BER wird der Simulink-Block *Error Rate Calculation* eingesetzt. Er vergleicht die beiden Datenframes vom Sender und Empfänger an den gleichen Stellen im System, hier nach dem Scrambler/Descrambler-Prozess, und gibt die Gesamtzahl der überprüften Bits, die Anzahl der fehlerhaften Bits und das Bitfehlerverhältnis aus.

Da zwischen den beiden zur Auswertung genutzten Stellen in der Simulation Verzögerungen durch Buffer oder durch den Cell-Deinterleaver entstehen, müssen diese natürlich bei dem Vergleich der Datenströme berücksichtigt werden. Die Berechnung der Verzögerung zwischen Sende- und Empfangsdatenstrom findet in der Datei *analysis.m* statt und wird dem Kalkulator über die Variablen *fac.tx_rx_delay*, *msc.tx_rx_delay* und *sdc.tx_rx_delay* als *Receive delay* mitgeteilt.

Bei dem am Anfang des Abschnitts 5.5 erklärten iterativen Aufbau der Simulation spielte die BER-Messung eine wichtige Rolle, da erst durch stetiges Anpassen der Verzögerung und der Überprüfung auf eine BER von 0 ein zum Sender korrespondierendes und Standard gerechtes Empfängermodell entwickelt wurde.

Die BER-Messung findet zudem eine wichtige Verwendung bei der Auswertung der

Synchronisations- und Kanalschätzmethoden.

5.5.4.3. Scatter Plot zur Fehleranalyse

Die Ermittlung des Konstellationsdiagramms im FAC wird über den Simulink-Block *Discrete-Time Scatter Plot¹ Scope* nach dem OFDM-Demapper durchgeführt. Es zeigt die Lage der QAM-Zellen im Konstellationsraum mit Störungen die während des gesamten Systems aufgetreten sind. Z.B. stellt folgende Abbildung die mit Rauschen gestörten QAM-Zellen dar, welche sich gestreut um die durch die QAM-Modulation erzeugten Konstellationspunkte aufhalten.

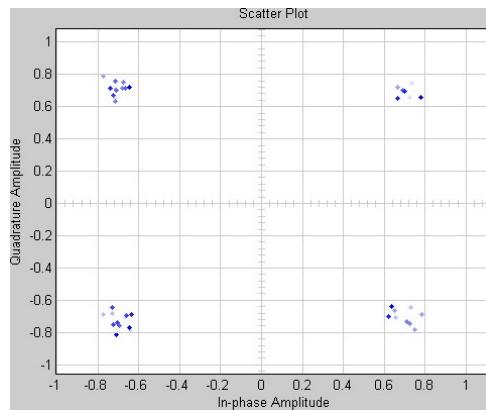


Abbildung 5.36.: Scatter Plot für den FAC bei Rauschanteil

Diese Darstellung ermöglicht dem Nutzer der Simulation einen ersten Eindruck der Störungen die durch Rauschen oder Synchronisationsprobleme auftreten. Die Bestimmung des SNR wird durch die MER-Messung im Folgenden beschrieben.

5.5.4.4. MER Messung

Das MER (Modulationfehlerverhältnis, engl.: Modulation Error Ratio) besitzt einen starken Zusammenhang zum SNR und kann über folgende Gleichung beschrieben werden:

$$MER_{dB} = 10 \lg \left(\frac{P_s}{P_e} \right) \quad (5.1)$$

Es gibt also das Verhältnis zwischen der Leistung des Nutzsignals P_s und der Leistung des Fehlersignals P_e an. Im Gegensatz zum SNR wird hier das Fehlersignal nicht als reines Rauschen sondern als alle während der Übertragung auftretenden Fehlerquellen, wie Rauschen, Unsymmetrien, Modulationsfehler u.a., ausgedrückt. Dadurch ist der Wert

¹Scatter Plot: englisch für Punktfolke

5. Umsetzung der Simulation

des MER bei einer gleichzeitigen Ermittlung immer geringer als beim SNR. Sollten nur die Rauschanteile betrachtet werden, so ist MER=SNR. [Eur01]

Durch den Zusammenhang wird in der Simulation eine Messung des MER und damit gleichzeitig eine Schätzung des SNR durchgeführt. Der Aufbau der MER-Messung wird in Abbildung 5.37 dargestellt und folgt der in [Eur01, S.43] aufgeführten Gleichung.

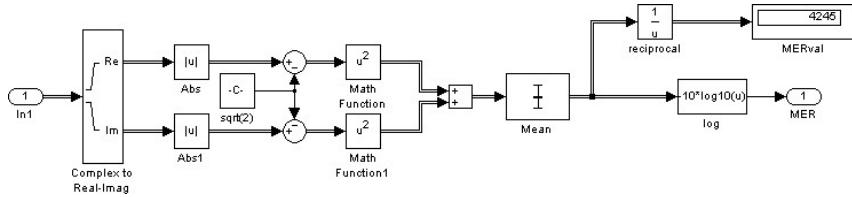


Abbildung 5.37.: Subsystem zur Messung des MER (SNR)

Es wird also der Mittelwert über das Quadrat der Differenz zwischen den empfangenen Beträgen der FAC-Zellen und dem im Sender festgelegten Betrag bei einer 4-QAM ($\sqrt{2}$) gebildet. D.h. geht man von Abbildung 5.36 aus, so werden für alle QAM-Zellen des FAC in einem Frame die Abweichungen der Beträge zum eigentlichen Konstellationspunkt als Betragsquadrat ermittelt. In diesem Kontext kann auch vom Fehlerbetragssvektor (EVM, engl.: Error Vector Magnitude) gesprochen werden, dessen Definition und dessen Zusammenhang mit dem MER ebenfalls in [Eur01] beschrieben wird.

Die ursprüngliche Verwendung der MER-Messung war für den Symbolsynchronisationsalgorithmus im Abschnitt 5.7.1 vorgesehen, wurde aber an dieser Stelle nicht umgesetzt, so dass die Messung lediglich als Auswertungselement der Simulation dient.

Die MER-Messung nach dem OFDM-Demapper beim FAC gewährleistet eine sichere Analyse der MER, da der FAC stets die robuste 4-QAM verwendet.

5.6. Kanalmodell

Das Kanalmodell dient der Simulation und Analyse von realen Übertragungsbedingungen. Die umgesetzten Teile des Übertragungskanals orientieren sich an den im Abschnitt 2.1.4 beschriebenen Kanaleigenschaften und sollen damit die Störungen und Unregelmäßigkeiten, die bei einem realen Übertragungsweg auftreten können, nachbilden. Die folgenden Störungen bzw. Szenarien können dabei manuell eingestellt und auch während der Ausführung der Simulation geändert werden.

- Hinzufügen eines Rauschanteils (AWGN-Kanal)
- Gesamtdämpfung des Kanals
- Zeitliche Verzögerung einzelner Samples bzw. zeitliche Verschiebung innerhalb der Symbole
- Zeitliche Verzögerung ganzer Symbole innerhalb eines Frames
- Frequenz- und Phasenänderungen
- Mehrwegeszenarien zur Umsetzung der Frequenzselektivität

Die Dopplerverschiebung und eine direkte Umsetzung eines zeitselektiven Kanals wurden nicht berücksichtigt. Eine Änderung der vorher aufgezählten Kanaleinflüsse während der Ausführung kann jedoch im gewissen Sinne als zeitliche Änderung des Übertragungskanals und somit als Zeitselektivität betrachtet und nachgestellt werden.

Die in der Abbildung 5.38 grün dargestellten Blöcke zeigen die Teile des in der Simulation modellierten Übertragungskanals.

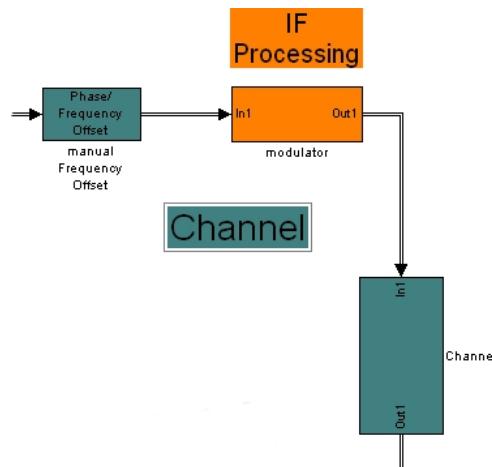


Abbildung 5.38.: Teile des Übertragungskanals im Modell

5.6.1. Frequenz- und Phasenverschiebungen

Die Frequenz- und Phasenänderungen werden über den Block *Phase/Frequency Offset* des *Communications Blockset* separat vor der ZF-Modulation eingebracht. Dies ist dadurch begründet, dass der Simulink-Block für Simulationen im Basisband konzipiert ist und eine Multiplikation der einzelnen komplexwertigen Samples mit dem entstehenden Phasenanteil vornimmt. Da hinter der ZF-Modulation reelle Signale entstehen, ist eine Umsetzung ohne vorherige Demodulation nicht möglich und würde in diesem Fall das Kanalmodell komplizierter gestalten. Dem Nutzer wird an dieser Stelle ermöglicht, ständige Änderungen der Phase bzw. der Frequenz sowohl im Positiven als auch im Negativen vorzunehmen und somit die durch die in Abschnitt 4.2 beschriebenen Störungen der Übertragungsfrequenz nachzubilden.

Alle weiteren Kanaleinflüsse werden im Block *Channel* zusammengefasst, welcher in der Abbildung 5.39 dargestellt wird und in dem erkennbar ist, dass sich das Subsystem aus vier getrennt einstellbaren, sowie zu- bzw. abschaltbaren Teilen zusammensetzt.

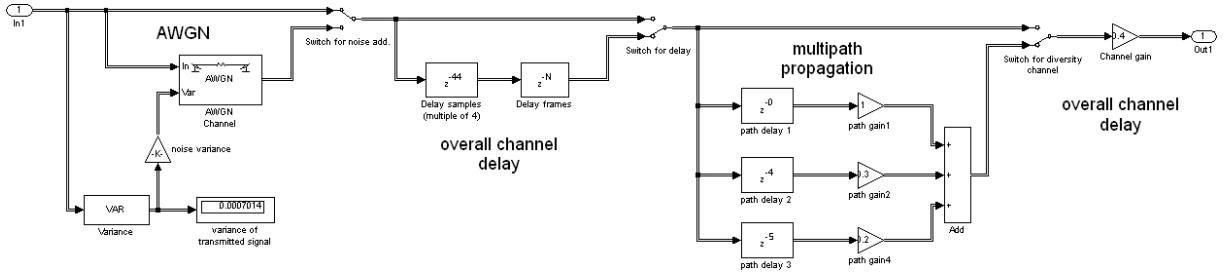


Abbildung 5.39.: Subsystem des Übertragungskanals

5.6.2. AWGN-Kanal

Der erste Teil ist die Nachbildung des AWGN-Kanalmodells. Im zugehörigen Block *AWGN Channel* kann die Option gewählt werden, die Varianz des zusätzlichen Rauschens über einen separaten Eingangsport anzugeben. Es wird zunächst die Varianz des Übertragungssignals ermittelt. Über den Verstärkungsfaktor *noise variance* kann dann der Anteil des Rauschsignals manuell eingegeben werden. Es ist somit möglich bestimmte SNR festzulegen, was aus Gleichung 2.3 im Abschnitt 2.1.4 hervorgeht. Wird z.B. ein Faktor von 0,01 gewählt, so würde das einer SNR von 20 dB entsprechen. Der Rauschanteil kann durch den Schalter während der Simulation ausgewählt werden.

5.6.3. Übertragungslaufzeiten und zusätzliche Verzögerungen

Der zweite Teil des Kanals stellt die einstellbare Übertragungslaufzeit des Signals dar. Er dient zur Simulation der im Abschnitt 4.1 behandelten Zeitsynchronisation. Dabei

wird zwischen der Verzögerung einzelner Samples, also einer Verschiebung innerhalb eines Symbols, und der Verzögerung ganzer Symbole innerhalb eines Übertragungsframes unterschieden. Diese Optionen dienen zum einen der Einstellung der Symbolsynchronisation und zum anderen der Einstellung der Framesynchronisation im Empfänger. Bei der Eingabe der Verzögerung einzelner Samples muss die Überabtastung berücksichtigt werden, so dass hier zunächst nur Vielfache von 4 eingegeben werden können, da sich sonst im Empfänger nicht ganzzahlige Verschiebungen ergeben, die mit diesem Empfängermodell nicht auszugleichen sind. Auch diese Kanaloption ist durch den Schalter während der Ausführung der Simulation aus- bzw. einschaltbar.

5.6.4. Mehrwegeszenario

Mit Hilfe der als *Multipath propagation* bezeichneten Parallelschaltung von Verzögerungsgliedern und Verstärkungsfaktoren soll die Multipfadausbreitung, wie in Abschnitt 2.1.4 dargestellt, nachgestellt werden. Die einzelnen Zweige stellen somit die Pfade dar, welche sich durch getrennte Verzögerungen und Dämpfungsfaktoren beschreiben lassen. Die Summation der Einzelzweige entspricht in diskreter Form der Gleichung 2.5, so dass über diesen Teil des Kanalmodells die diskrete Kanalimpulsantwort $h_c(n)$ manuell erstellt werden kann. In der Abbildung 5.39 lautet diese, ohne weitere Kanaleinflüsse, wie folgt:

$$h_c(n) = \delta(n) + 0,3\delta(n - 4) + 0,2\delta(n - 5)$$

Die Impulsantwort wird für den dritten Teil der im Kapitel 4 beschriebenen Empfängeralgorithmen benötigt, um eine Erstellung und Simulation von Kanalschätzungen zu ermöglichen. Es ist natürlich ebenso möglich weitere Pfade hinzuzufügen oder aber statt diesem Aufbau einfach ein digitales Filter mit der entsprechenden Impulsantwort zur Nachbildung eines frequenzselektiven Kanals zu verwenden.

Die vierte einstellbare Option ist der Gesamtverstärkungsfaktor des Kanals, welcher auch bei nicht selektierter Mehrwegeausbreitung eine Überprüfung der Kanalschätzmethoden ermöglicht. Er kann als Dämpfungsfaktor der Umgebung bei der Übertragung von Funkwellen angesehen werden.

5.7. Algorithmen zur Synchronisierung und Kanalschätzung

Der folgende Abschnitt soll die in der Simulation umgesetzten Algorithmen zur Zeit- und Frequenzsynchronisation, sowie zur Kanalschätzung und -kompensation erläutern, wie sie im Kapitel 4 vorgestellt wurden.

5.7.1. Symbolsynchronisation

Der in der Simulation umgesetzte Algorithmus zur Symbolsynchronisation basiert auf der im Abschnitt 4.1.1 vorgestellten Methode, bei dem das Guardintervall genutzt wird, um die Symbollage in einem Betrachtungsfenster zu erkennen.

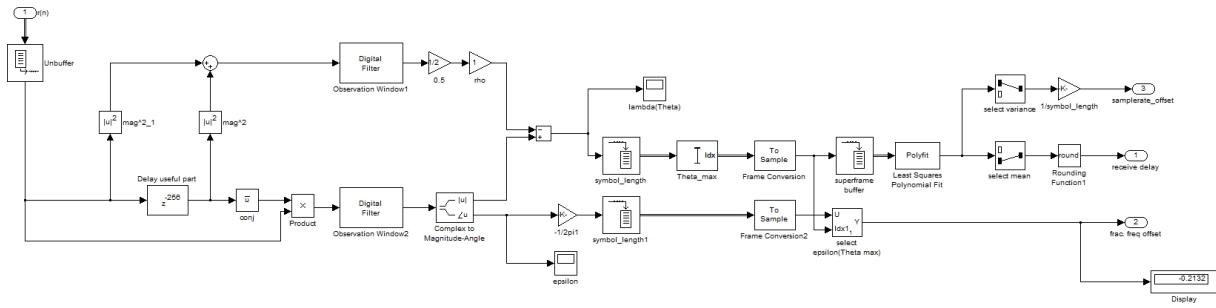


Abbildung 5.40.: Kombinierte Ermittlung der Symbollage und der partiellen Trägerverschiebung

Die Abbildung 5.40 zeigt das Subsystem zur kombinierten Schätzung der Symbollage und des partiellen Anteils der Frequenzverschiebung. Das Betrachtungsfenster wurde über ein *Digital Filter* mit den Koeffizienten `ones(1,sFFT.guard_size)` realisiert, was somit der Summenbildung der Gleichungen 4.2 und 4.3 über N_G Abtastwerte entspricht.

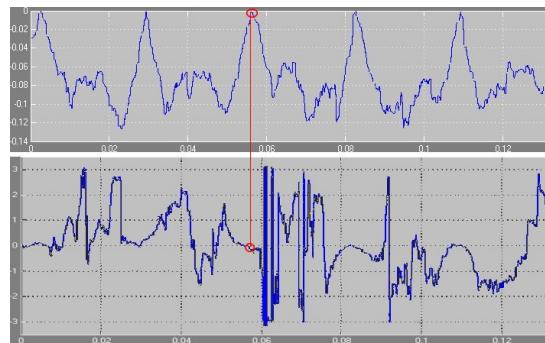


Abbildung 5.41.: Zeitlicher Verlauf der Schätzfunktionen für die Symbolsynchronisation

Die Ergebnisse der Funktionen 4.1 und 4.4 werden über *Scopes* ausgegeben. Dabei ergibt sich beispielsweise eine Darstellung nach Abbildung 5.41, in der der zeitliche Verlauf von Θ (oben) und ϵ (unten) über fünf Symbole sichtbar ist. Die Spitzen der Korrelation weisen auf die Lage des Symbolanfangs hin und die entsprechende Frequenz an dieser Stelle gibt die partielle Trägerverschiebung an. Über die Detektion des Maximums innerhalb eines Symbols kann $\hat{\Theta}$ zur Symbolsynchronisation und entsprechend $\hat{\epsilon}$ zur Frequenzsynchro-nisation verwendet werden.

Bei der Untersuchung dieser Methode mit unterschiedlichen Modi und SNR ergab sich, dass der geschätzte Wert für $\hat{\Theta}$ häufig instabil wurde und zwischen zwei Werten variierte. Um dies zu verhindern, wurde eine Mittelwertbildung über einen Superframe durchgeführt, was nicht unbedingt als optimale Lösung betrachtet werden sollte. Durch den Einsatz der Funktion *Polyfit* ergibt sich dabei nach [SDSGU05] zusätzlich die Ermittlung des Samptetime-Offsets des empfangenen Signals, welcher jedoch in der vorliegenden Arbeit keine weitere Rolle spielt.

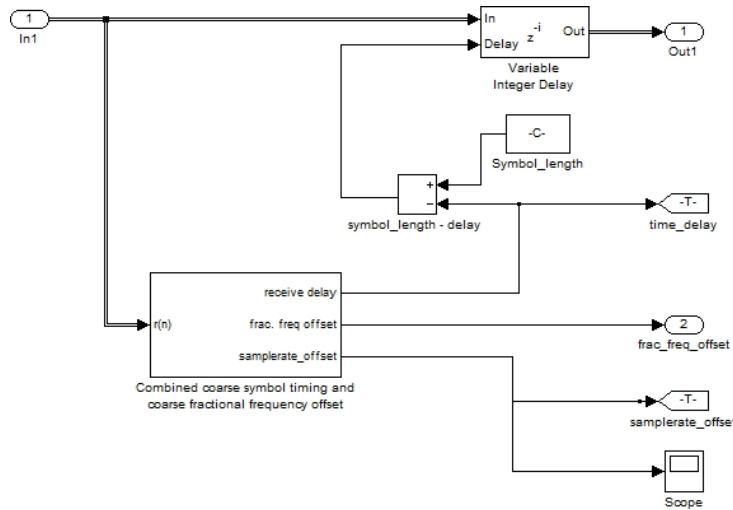


Abbildung 5.42.: Subsystem zur Symbolsynchronisation

Die eigentliche Synchronisation erfolgt, wie in Abbildung 5.42 zu erkennen, über einen *Variable Integer Delay*-Block. Hierbei wird eine Verzögerung von $N_S - \hat{\Theta}$ eingestellt, um der nachfolgenden FFT Symbole exakt zu ihren Beginn zu liefern. Diese Verzögerung um ein Symbol muss jedoch bei der im Abschnitt 5.7.3 beschriebenen Framesynchronisation berücksichtigt werden.

Der ermittelte, partielle Frequenzoffset wird aus dem Subsystem an die anschließende Frequenzsynchro-nisation übergeben.

5.7.2. Frequenzsynchronisation

Zur Ermittlung des ganzzahligen Frequenzoffsets wird ein separates FFT-Subsystem parallel in die Empfängerstruktur eingebracht, wie in Abbildung 5.44 zu erkennen ist, um ein Frequenzabgleich bereits vor der eigentlichen Rücktransformation der OFDM-Symbole zu bewerkstelligen. Hierbei erfolgt jedoch keine Reduktion der Werte der OFDM-Signale auf die genutzten Träger K , da ein vollständiges Symbol der Größe N_U für den folgenden Algorithmus verwendet wird.

Der im Abschnitt 4.2 vorgestellte Algorithmus zur Erkennung der ganzzahligen Trägerverschiebung f_{int} wurde in Simulink, wie in Abbildung 5.43 dargestellt, umgesetzt.

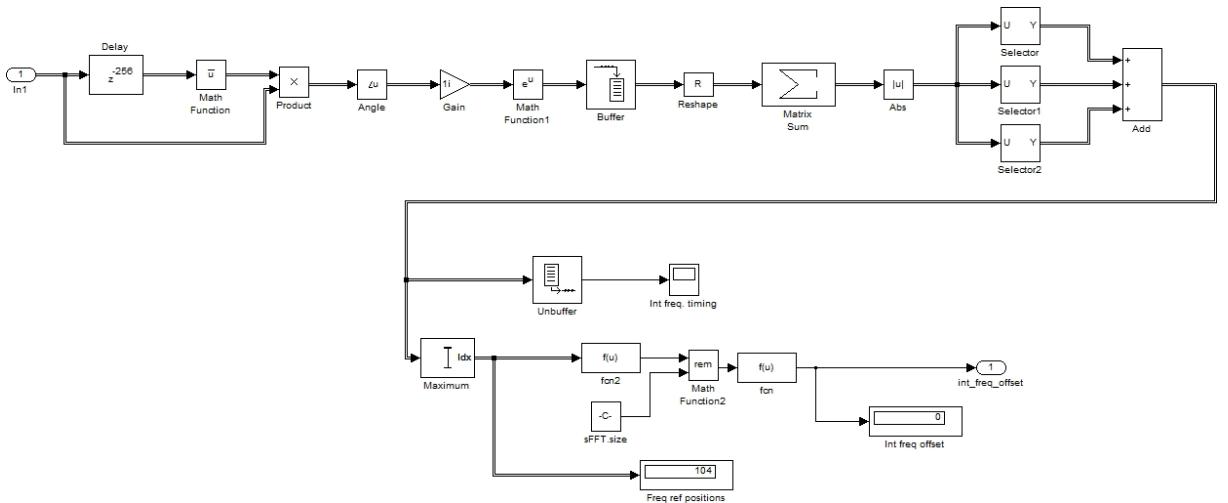


Abbildung 5.43.: Ermittlung der ganzzahligen Trägerverschiebung

Zunächst wird die in Gleichung 4.9 beschriebene Akkumulation der Phasendifferenzen zwischen zwei aufeinander folgenden Symbolen für alle Träger über die Symbole eines Frames durchgeführt. Die folgende Summierung über die verschobenen Indizes der drei Referenzzellen wird mit Hilfe der *Selector*-Blöcke realisiert. Dazu werden bei der Initialisierung in *time_freq_sync.m* die Trägerindizes der drei Piloten *sync.freq_ref_index(p)* selektiert und Vektoren gebildet, die die Indizes dieser Piloten übereinander legen und damit folgendes Aussehen besitzen ($sFFT.size=N_U$):

```
[ sync.freq_ref_index(1)+1:sFFT.size 1:sync.freq_ref_index(1)]
[ sync.freq_ref_index(2)+1:sFFT.size 1:sync.freq_ref_index(2)]
[ sync.freq_ref_index(3)+1:sFFT.size 1:sync.freq_ref_index(3)]
```

Es folgt die Auswahl des Maximums über die N_U -Werte und eine anschließende Korrektur des ermittelten Wertes \hat{m} der ganzzahligen Trägerverschiebung im Bezug auf K_{min} und

5. Umsetzung der Simulation

N_U in folgender Form:

$$m = \begin{cases} \hat{m} - K_{min} - 1, & \text{wenn } \hat{m} \leq \frac{N_U}{2} \\ \hat{m} - K_{min} - 1 - N_U, & \text{wenn } \hat{m} > \frac{N_U}{2} \end{cases}$$

Diese Korrektur wurde ebenfalls aus [SDSGU05] entnommen und zeigt, dass ein Frequenzoffset von maximal $\frac{N_U}{2}$ in positiver oder negativer Richtung erkannt werden kann.

Die folgende Abbildung stellt das Subsystem der Frequenzsynchronisation dar und zeigt die Korrektur der Trägerfrequenz.

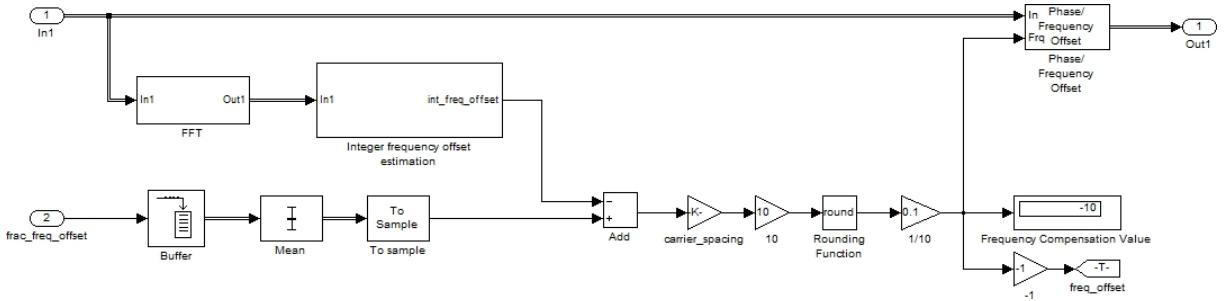


Abbildung 5.44.: Subsystem zur Frequenzsynchronisation

Die beiden ermittelten Anteile der Trägerverschiebung werden zunächst addiert und anschließend über die Multiplikation mit Δf_k zu einem Frequenzoffset f_{off} gewandelt und mit Hilfe des gleichen Blocks, wie bei der manuellen Erstellung der Frequenzverschiebung, *Phase/Frequency Offset* an die Empfangssymbole multipliziert.

Da die Ermittlung des partiellen Frequenzoffsets f_{frac} sehr fein ist und deshalb zu fortlaufenden Variationen der Frequenzverschiebung geführt hat, wurde zur Lösung eine Rundung mit einstellbarer Genauigkeit implementiert. Dies stellt keine optimale Umsetzung dar. Es wäre bei einer Erweiterung des Modells ein Regelalgorithmus für das eigentliche Tracking des geringen Frequenzoffsets angebracht.

Man sieht, dass bei der Simulation keine Differenzierung zwischen Akquisition und Tracking erfolgt, wie es im Theorieteil angedeutet wurde. Dies wäre ein weiterer Punkt zur Optimierung des Empfängermodells, da mit einer nur einmaligen, am Beginn stattfindenden Detektion der groben Frequenzverschiebung Rechenleistung eingespart werden könnte.

5.7.3. Framesynchronisation

Die Schätzung zur Synchronisation der Symbole in einem Rahmen wird exakt nach der im Abschnitt 4.1.2 beschriebenen Methode in Simulink umgesetzt.

Dabei erfolgt zuerst die Bildung der Pilotpaare in der Datei *time_freq_sync.m*, welche mit ihren Indizes im Vektor *sync.pilot_pairs* abgelegt werden. Dieser wird zur Selektion der Referenzzellen zugleich aus den Pilot-Matrizen und den empfangenen Zellen genutzt. Die Abbildung 5.45 stellt die Umsetzung der Gleichung 4.5 mit anschließender Auswahl des Maximums über die in einem Frame befindlichen Symbole dar.

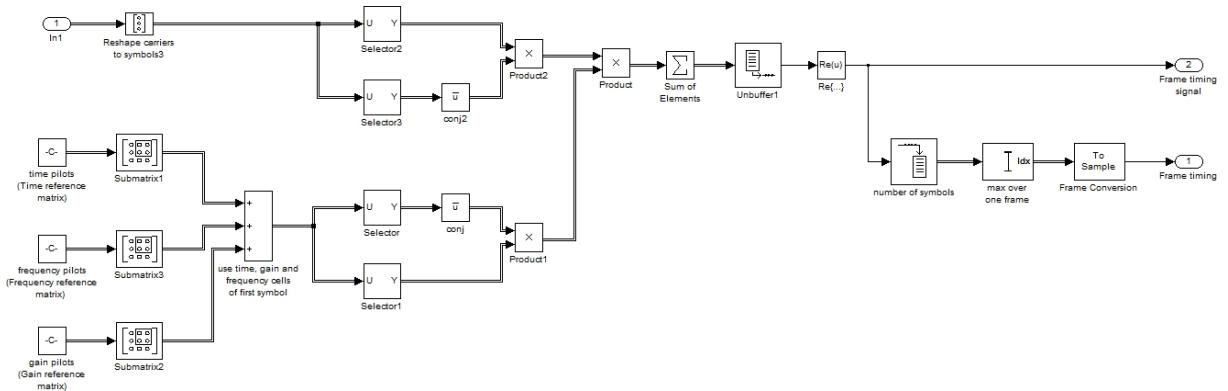


Abbildung 5.45.: Subsystem zur Ermittlung der Lage des ersten Symbols im Rahmen

Das Ergebnis der Korrelation und die ermittelten Maxima über drei Frames werden in der Abbildung 5.46 deutlich. Dabei ist zu erkennen, dass der Maximalwert an der Stelle des zweiten Symbols ($s=1$) liegt. Diese Verzögerung um ein Symbol wird jedoch durch die zuvor beschriebene Symbolsynchronisation im Empfänger selbst erzeugt. Sollte eine zusätzliche Verzögerung um ein ganzes Symbol durch den Nutzer, wie im Abschnitt 5.6 beschrieben, erzeugt werden, so würden sich die Maxima um eine weitere Stelle auf $s=2$ verschieben.

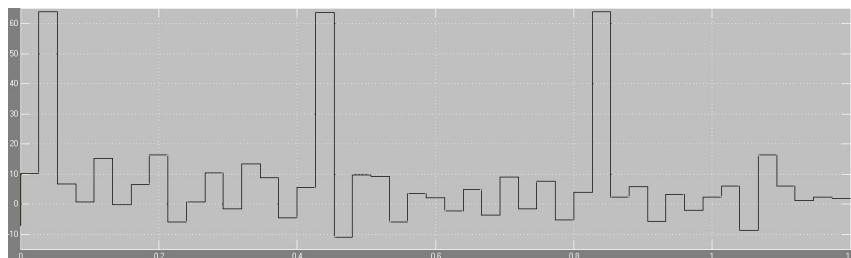


Abbildung 5.46.: Beispiel zur Ermittlung der Lage des ersten Symbols (Mode B)

Nachdem die Lage des ersten Symbols im OFDM-Rahmen ermittelt wurde, wird über eine variable Verzögerung mit Hilfe des Blocks *Variable Integer Delay* dem nachfolgenden

5. Umsetzung der Simulation

OFDM-Demapper stets ein Frame mit der korrekten Symbolordnung übergeben. Hierzu wird wie in Abbildung 5.47 zu erkennen ist, der ermittelte Symbolwert mit der Anzahl der Träger multipliziert und von einem gesamten Superframe subtrahiert, da auch die Lage der SDC-Daten (nur im ersten Frame) für das Demapping berücksichtigt werden muss.

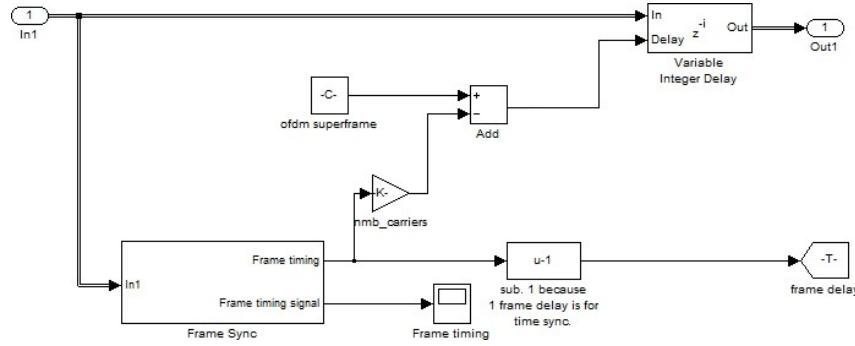


Abbildung 5.47.: Subsystem zur Framesynchronisation

5.7.4. Kanalschätzung und -ausgleich

Die Umsetzung der Kanalschätzung in der Simulation arbeitet nach dem im Abschnitt 4.3 vorgestellten Prinzip. Die Abbildung 5.48 zeigt das entworfene Subsystem zur Kanalschätzung und Kanalkompensation, welches sich in die vier Kategorien Bestimmung der Übertragungsfunktion über die Piloten, Interpolation, Darstellung des geschätzten Kanals und der Kanalkompensation einteilen lässt.

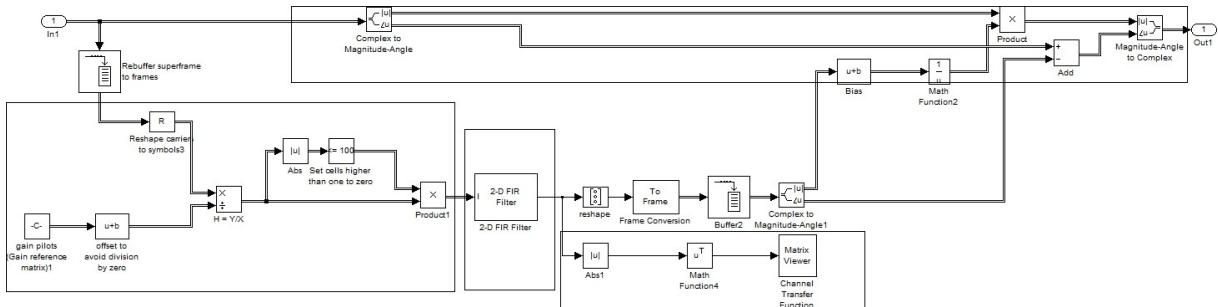


Abbildung 5.48.: Subsystem zur Kanalschätzung und zum Kanalausgleich

Es wird also zunächst die nur auf den Piloten basierende CTF $\hat{H}_{est}(s_p, k_p)$ geschätzt, die über die Division des in Matrixform vorliegenden empfangenen Frames $Y = R(s_p, k_p)$ durch die Matrix der Gain-Piloten $X = S(s_p, k_p)$ ermittelt wird. Hierbei wird die vollständige Matrix der Gain-Piloten `sync.gain_ref_array` eingesetzt, in der im Gegensatz zu der im Abschnitt 5.3.3.1 keine Zellen fehlen, die bereits durch Frequenz- oder Zeitpiloten belegt sind. Zudem wird durch das Hinzufügen eines geringen Offsets durch den

5. Umsetzung der Simulation

Bias-Block eine Division durch Null vermieden. Nach der Division muss $\hat{H}_{est}(s, k)$ in die Form gebracht werden, dass alle nicht berechneten Datenzellen, die nach der Division einen Betrag > 100 haben, den Wert Null erhalten, was über eine Multiplikation dieser Zellen mit Null erreicht wird.

Der zweite Teil des Subsystems ist die Interpolation von $\hat{H}_{est}(s, k)$ auf die vollständige, geschätzte CTF H_{est} . An dieser Stelle sind jetzt diverse Implementierungen von Interpolatoren oder Filtern möglich, wie sie in 4.3 angedeutet wurden. Die Basisumsetzung erfolgt über ein zweidimensionales Filter aus dem *Image Processing Blockset* von Simulink. Es wurde nur eine einfache eindimensionale Mittelwertbildung in Frequenzrichtung zwischen den Pilotzellen realisiert. Das separierbare Filter erhält dazu vertikale Koeffizienten in der Form `ones(1, channel.inter)`, während der horizontale Anteil auf 1 belassen wird. Die Länge des Filters wird je nach Mode über `channel.inter` bestimmt, was den Abständen der Gain-Piloten in Frequenzrichtung D_k entspricht.

Die somit bestimmte Übertragungsfunktion lässt sich mittels *Matrix Viewer* darstellen. Abbildung 5.49 zeigt als Beispiel für den Mode B und einer Spektralbelegung von $B_S=0$ einen Übertragungskanal mit der im Abschnitt 5.6 dargestellten Impulsantwort, wobei zu beachten ist, dass der eingestellte Kanal sich nicht im Basisband befindet.

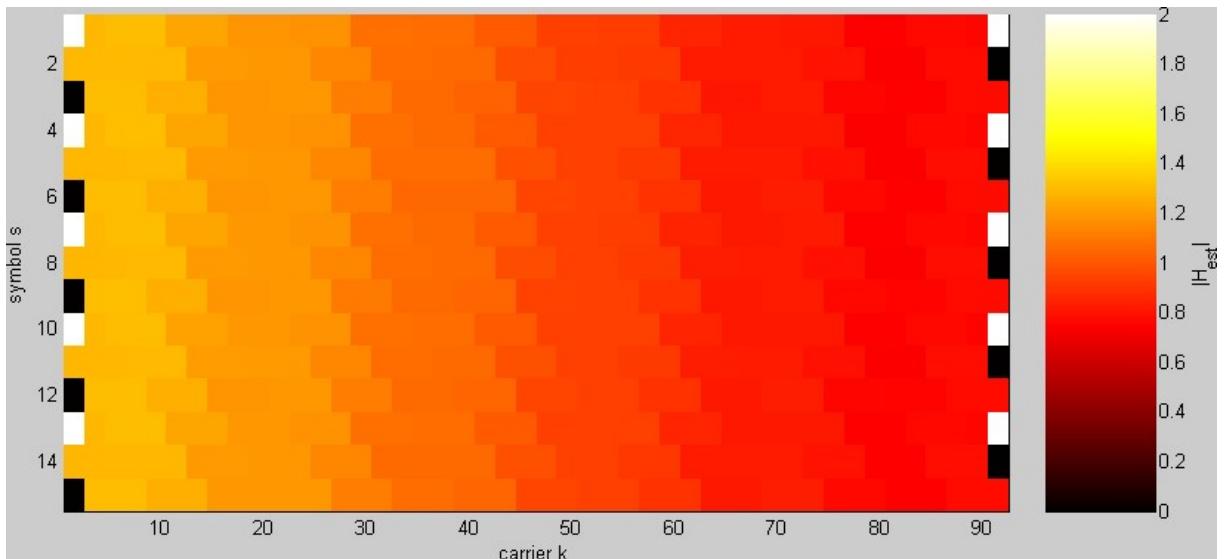


Abbildung 5.49.: Beispiel einer geschätzten Kanalübertragungsfunktion (Mode B, $B_S=0$)

Man erkennt das Tiefpassverhalten des Kanals im Bereich der übertragenen Bandbreite. Zudem sieht man, dass die Schätzmethode noch nicht optimal umgesetzt wurde, da sich zeitliche Änderungen trotz Zeitinvarianz ergeben.

Nachdem die CTF geschätzt wurde wird eine Kanalkompensation durch getrennte Verarbeitung von Betrag und Phase vorgenommen. Der Betrag der empfangene Zellen wird

5. Umsetzung der Simulation

mit dem reziproken Betrag der CTF $\frac{1}{|H_{est}|}$ multipliziert und die Phase der CTF $\angle H_{est}$ wird von der Phase der Zellen subtrahiert.

Wie auch die anderen Algorithmen stellt die Kanalschätzung in diesem Modell nur eine Grundlage für weitere Untersuchungen dar. Eine Optimierung wäre durch den Einsatz eines anderen Filters möglich, indem z.B. auch der Rauschanteil über ein Wiener-Filter bei der Schätzung berücksichtigt werden könnte. [Hen08, S.117]

6. Zusammenfassung und Ausblick

Die in dieser Masterarbeit umgesetzte Simulation stellt wie gefordert ein gesamtes DRM-Übertragungssystem, samt Kanalkodierung, OFDM und ZF-Modulation, dar. Sie beinhaltet alle für eine Nachbildung eines DRM-Systems notwendigen Komponenten und Algorithmen des im Standard beschriebenen Senders und kann über den parallel dazu entwickelten Empfänger diese fehlerfrei rekonstruieren.

Mit der Verifizierung des Sendermodells konnte die Standard gerechte Implementierung der Komponenten mit Hilfe gängiger DRM-Software nachgewiesen werden und es lässt sich aus der fehlerfreien Simulation daraus ableiten, dass auch der Empfänger mit all seinen Komponenten dem Standard entsprechend umgesetzt wurde.

Die Hinzunahme eines Kanalmodells machte es zudem möglich grundlegende Algorithmen zur Synchronisation von Zeit und Frequenz und zur Kanalkompensation im Empfänger hinzuzufügen und zu überprüfen. Die dabei erzielten Ergebnisse zeigten, dass die Algorithmen in ihrer grundsätzlichen Arbeitsweise funktionieren, jedoch nicht die optimale Lösung für sämtliche bei der Übertragung auftretenden Probleme darstellen.

Durch den Einsatz möglichst vieler Blöcke und Funktionen vom Entwicklungswerkzeug MATLAB/Simulink wird eine schrittweise Implementierung des Modells auf eine Zielplattform über den Real-Time-Workshop vereinfacht.

Die vorliegende Masterarbeit bietet eine Basis für folgende Arbeiten, welche problemlos auf der bestehenden Simulation aufsetzen können, um diese zu Erweitern oder zu Optimieren. Dabei sind folgende Möglichkeiten als Ausblick zur Erweiterung für das bestehende System denkbar:

- Fehlende, nicht für die vorliegende Simulation nötige, Teile des DRM-Standards wie HMsym, HMmix oder UEP könnten als Erweiterung des Modells hinzugefügt werden.
- Durch eine Hinzunahme von weiteren Empfangskomponenten wie Robustnessmode-Detektion oder eine Samplerate-Anpassung könnte eine eigenständige Empfangsstruktur entworfen werden, die auch Teststreams vorhandener Software dekodieren und verarbeiten kann.
- Um das gesamte DRM-System zu komplettieren könnten die Algorithmen der Quellenkodierung bzw. -dekodierung des Standards an das Modell angebracht werden. Hierbei ist jedoch zu beachten, dass die vorliegende Simulation bereits sehr hohe Anforderungen an die Hardware stellt und eine Simulation in moderater Zeit schwierig macht.

6. Zusammenfassung und Ausblick

- Die Algorithmen zur Synchronisation und zur Kanalschätzung stellen nur grundlegende Umsetzungen dar, die in der vorliegenden Simulation eingefügten Kanalmodelle nur hinreichend genau behandeln können. An dieser Stelle könnte eine Optimierung der Umsetzung der verschiedenen Algorithmen ein besseres Ergebnis liefern und damit eventuell auch real existierende Störungen im Übertragungskanal (Doppler, Zeitvarianz u.a.) behandeln.
- Zudem wäre eine Implementierung von Teilen des Systems auf ein Embedded System denkbar, indem über den Real-Time-Workshop eine Codegenerierung einzelner Komponenten für die entsprechenden Zielplattformen durchgeführt wird.

Literaturverzeichnis

- [AS06] AHN, In S. ; STEWART, Thomas L.: *Software Defined Radio*. Pedria, Bradley University, Simulink-Projekt, 2006
- [BG07] BRIEM, Harry ; GELBKE, Stefan: *Faltungscodes*. Chemnitz, Technische Universität Chemnitz, Seminar Kodierverfahren SS2007, 2007
- [BSB97] BEEK, Jan-Jaap van d. ; SANDELL, Magnus ; BÖRJESSON, Per O.: *ML Estimation of Time and Frequency Offset in OFDM Systems*. Luleå, Luleå University of Technology, Tech.report, 1997
- [Chi04] CHINCHILLA, Alfonso Luís T.: *Synchronization and Channel Estimation in OFDM: Algorithms for Efficient Implementation of WLAN Systems*. Cottbus, Technische Universität Cottbus, Diss., 2004
- [CPWO06] CHEN, Chen ; PARK, Beomjin ; WEI, Lijun ; OH, Hyun-Seok: *Synchronization Acquisition Methods for DRM Systems*. Beijing, Communication University of China., Tech.report, 2006
- [Eur01] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE (Hrsg.): *Digital Video Broadcasting (DVB); Measurement guidelines for DVB systems ETSI TR 101 290 V1.2.1*. European Telecommunications Standards Institute, 2001
- [Eur09] EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE (Hrsg.): *Digital Radio Mondiale (DRM); System Specification ETSI EN 201980 V3.1.1*. European Telecommunications Standards Institute, 2009
- [FK10] FISCHER, Volker ; KURPIERS, Alexander: *Frequency Synchronization Strategy for a PC-based DRM Receiver*. Darmstadt, TU Darmstadt, Tech.report, o.A. [2010]
- [Hen08] HENKEL, Martin: *Architektur eines DRM-Empfängers und Basisbandalgorithmen für Frequenzakquisition und Kanalschätzung*. Karlsruhe, Universität Karlsruhe, Diss., 2008
- [Int00] INTINI, Anibal L.: *Orthogonal Frequency Division Multiplexing for Wireless Networks*. Santa Barbara, University of California, Tech.report, 2000

- [Kam04] KAMMEYER, Karl-Dirk: *Nachrichtenübertragung*. 3.Auflage. Wiesbaden : B.G. Teubner, 2004
- [Kun07] KUNZE, Joachim: *A robust pilot based frequency tracking method for DRM Receivers*. Merseburg, FH Merseburg, Tech.report, 2007
- [Lan04] LANGTON, Charan: *Orthogonal Frequency Division Multiplex (OFDM)*. Internet-Tutorial. <http://www.complextoreal.com/chapters/ofdm2.pdf>. Version: 2004
- [LLT04] LEE, J. ; LOU, H. ; TOUMPAKARIS, D.: *Maximum likelihood estimation of time and frequency offset for OFDM systems*. Sunnyvale, Marvell Semiconductor, Inc., Tech.report, 2004
- [Loc02] LOCHMANN, Dietmar: *Digitale Nachrichtentechnik - Signale, Codierung, Übertragungssysteme, Netze*. 3.Auflage. Berlin : Verlag Technik, 2002
- [Met06] METZGER, Christoph: *Hauptseminar Drahtlose Netzwerke - Technologien, Anwendungen, Management WLAN IEEE 802.11g/n*. München, Technische Universität München, Seminar Drahtlose Netzwerke, 2006
- [Nus07] NUSZKOWSKI, H.: *Mobile Nachrichtensysteme I*. Dresden, TU Dresden, Vorlesungsscript, 2007
- [o.A10] o.A.: *Dream DRM Receiver*. Internet. http://sourceforge.net/apps/mediawiki/drm/index.php?title>Main_Page. Version: 2010
- [Pas05] PASTOR, Alberto P.: *Simulation and Evaluation of a DVB system using Simulink*. Linköping, Linköpings universitet, Bachelor thesis, 2005
- [Päh98] PÄHLKE, Frank: *Sequentielle Decodierung linearer Blockcodes*. Karlsruhe, Universität Karlsruhe, Diplomarbeit, 1998
- [PM06] PROAKIS, John G. ; MANOLAKIS, Dimitris G.: *Digital Signal Processing*. 4.Auflage. New Jersey : Prentice Hall, 2006
- [Pro01] PROAKIS, John G.: *Digital Communications*. 4.Auflage. New York : McGraw-Hill Higher Education, 2001
- [Pul07] PULIKKOONATTU, Rethnakaran: *Channel estimation in DVB-T and OFDM systems*. India, ST (Genesis Microchip group), Internal technical report, 2007
- [Rei08] REIMERS, Ulrich: *DVB - Digitale Fernsehtechnik: Datenkompression und Übertragung*. 3.Auflage. Berlin, Heidelberg : Springer Verlag, 2008

- [Roc07] ROCA, Amalia: *Implementation of a WiMAX simulator in Simulink*. Wien, Technische Universität Wien, Diplomarbeit, 2007
- [Rud08] RUDOLPH, Dietmar: *Digitalisierung des UKW Rundfunks*. Skript zur Rundfunk-Technik. http://www.diru-beze.de/rundfunktechnik/skripte/UKW_Rundfunk_digital.pdf. Version: 2008
- [SDSGU05] SCHORR, Torsten ; DITTRICH, Andreas ; SAUER-GREFF, Wolfgang ; UR-BANSKY, Ralph: *Diorama - An Open Source Digital Radio Mondiale (DRM) Receiver using MATLAB*. Kaiserslautern, TU Kaiserslautern, Tech.report, 2005
- [SFFM01] SPETH, Michael ; FECHTEL, Stefan ; FOCK, Gunnar ; MEYR, Heinrich: *Optimum Receiver Design for OFDM-Based Broadband Transmission - Part II: A Case Study*. München, Aachen, o.A., IEEE paper, 2001
- [Sho08] SHOOSHTARY, Samaneh: *Development of a MATLAB Simulation Environment for Vehicle-to-Vehicle and Infrastructure Communication Based on IEEE 802.11p*. Wien, Technische Universität Wien, Master's Thesis, 2008
- [Sto01] STOTT, Jonathan: *DRM - key technical features*. EBU technical report. http://www.ebu.ch/en/technical/trev/trev_286-stott.pdf. Version: 2001
- [Thi09] THILLEN, Fränz: *A system overview of Digital Radio Mondiale*. Kaiserslautern, TU Kaiserslautern, Seminar Nachrichtentechnik, 2009
- [VW06] VERCIMAK, Luke ; WEYENETH, Karl: *Software Defined Radio*. Peoria, Bradley University, Tech.report, 2006
- [Wit10] WITTNEBEN, A.: *Signalübertragung mittels Funkempfängern*. Zürich, ETH Zürich, Seminar, 2010
- [YSH99] YUAN, Dong-Feng ; SCHILL, Dietmar W. ; HUBER, Johannes B.: *Robust Hierarchical Broadcasting for AWGN and Flat Rayleigh Fading Channels using Multilevel Codes*. IEEE paper. <http://www.lit.lnt.de/papers/iscta99.pdf>. Version: 1999

Abbildungsverzeichnis

2.1.	Wellenausbreitung von Radiowellen	5
2.2.	Grundelemente eines digitalen Übertragungssystems	6
2.3.	Verschiedene Definitionen des Übertragungskanals [Nus07]	8
2.4.	Mehrwegeszenario bei der Signalübertragung	9
2.5.	Kanalmodell der Mehrwegeübertragung als Verzögerungskette	10
2.6.	Komponenten eines DRM Übertragungssystems	13
2.7.	Blockdiagramm des DRM-Senders [Eur09, fig.1]	14
2.8.	Rahmenstruktur bei DRM im Basisband	17
3.1.	PRNG als Scrambler im DRM System	19
3.2.	Beispiel zum Scrambler/Descrambler-Prozess im DRM Übertragungssystem	20
3.3.	Multilevel-Kodierung in DRM mit 3 Level für SM [Eur09, 115, Abb.29] . .	22
3.4.	Beispiel eines Faltungscodes mit 2 Speichern und das zugehörige Zustandsdiagramm	23
3.5.	Beispiel eines Trellisdiagramms [Loc02, S.523]	23
3.6.	Faltungscoder des DRM Systems [Eur09, Abb.34]	24
3.7.	Beispiel zur Metrikberechnung des Viterbi-Decoders	26
3.8.	Konstellationsdiagramm der 16-QAM [Eur09, 131, Abb.38]	28
3.9.	Prinzip des „long interleaving“	30
3.10.	Einfluss von Burstfehlern auf Daten ohne „long interleaving“	30
3.11.	Einfluss von Burstfehlern auf Daten mit „long interleaving“	30
3.12.	Prinzip eines Faltungsinterleavers	31
3.13.	Spektrum eines OFDM-Signals mit 5 Subträgern	33
3.14.	Prinzip eines OFDM-Übertragungssystems im Basisband	34
3.15.	Rahmenstruktur eines OFDM-Signals	34
3.16.	Vergleich der Bandbreitennutzung zwischen OFDM und FDM [Met06] . . .	35
3.17.	Kanaleinflüsse bei Einzel- und Mehrträgerverfahren [Kam04, S.581] . . .	36
3.18.	Hinzufügen des Schutzintervalls eines OFDM-Symbols	37
3.19.	Verwendete Trägernummern für die Modi bei einer bestimmten Spektralbelegung B_S [Eur09, Tab.84]	40
3.20.	Pilotstruktur für Mode B mit der Spektralbelegung $B_S=0$	43
3.21.	Übersicht zur Zwischenfrequenz-Verarbeitung bei DRM	44
3.22.	Betrachtung der Überabtastung im Frequenzbereich	45
3.23.	Spektren bei der ZF-Modulation	46

3.24. Prinzip eines IQ-Modulators und IQ-Demodulators	47
4.1. Mögliche Empfängerstruktur mit Synchronisationsalgorithmne	48
4.2. Symbole mit Schutzintervall im Betrachtungsfenster	50
4.3. Prinzip der Symbolsynchronisation nach [BSB97]	50
4.4. Bildung von Pilotpaaren im ersten Symbol ($s=0$) eines Frames	52
4.5. Betrachtung der Frequenzpiloten in einem OFDM-Rahmen über N_U Werte	55
4.6. Darstellung der Methode zur Interpolation bei der Kanalschätzung	58
5.1. Sendermodell in Simulink	64
5.2. Subsystem der zyklischen Redundanzprüfung beim FAC	65
5.3. Subsystem für das Energy dispersal	66
5.4. Subsystem zur Kanalkodierung für den SDC mit einer 16-QAM	67
5.5. Subsystem des Faltungscoders	68
5.6. Subsystem zur Punktierung	69
5.7. Subsystem für das QAM-Mapping mit einer 16-QAM	70
5.8. Konstellationsdiagramm 16-QAM in Simulink	71
5.9. Subsystem des MSC Cell-Interleavings	71
5.10. Subsystem des Pilotgenerators	72
5.11. Pilotstruktur in der Simulation (Mode B, $B_S=0$)	73
5.12. Subsystem des OFDM-Cellmappings	74
5.13. Subsystem der IFFT	75
5.14. Subsystem der ZF-Modulation	76
5.15. Subsystem der Überabtastung	76
5.16. Entwurf des Antiimaging-Filters	77
5.17. Spektrum des Ursprungssignals und des überabgetasteten Signals	78
5.18. Subsystem des IQ-Modulators	78
5.19. Spektrum im Basisband (überabgetastet) und nach der ZF-Modulation . .	79
5.20. Modell zur Überprüfung des Sendermodells auf standardgerechte Umsetzung	80
5.21. Diorama: Überprüfung der CRC der Testdatei	81
5.22. Diorama: Empfangsspektrum	82
5.23. Empfängermodell in Simulink	83
5.24. Subsystem der ZF-Demodulation	84
5.25. Subsystem der IQ-Demodulation	84
5.26. Subsystem der Unterabtastung	85
5.27. FFT Subsystem	85
5.28. Subsystem des OFDM-Demappers	86
5.29. Subsystem des MSC Cell-Deinterleavers	87
5.30. Subsystem zur Kanaldekodierung für den SDC mit einer 16-QAM	87
5.31. Subsystem des QAM-Demapping bei einer 16-QAM	88
5.32. Subsystem der Viterbi-Dekodierung	89

5.33. Subsystem zur Depunktierung des Datenstroms	89
5.34. Subsystem zur Auswertung der CRC-Prüfsumme beim FAC	90
5.35. Teile der Subsysteme zur BER-Auswertung	91
5.36. Scatter Plot für den FAC bei Rauschanteil	92
5.37. Subsystem zur Messung des MER (SNR)	93
5.38. Teile des Übertragungskanals im Modell	94
5.39. Subsystem des Übertragungskanals	95
5.40. Kombinierte Ermittlung der Symbollage und der partiellen Trägerverschiebung	97
5.41. Zeitlicher Verlauf der Schätzfunktionen für die Symbolsynchronisation	97
5.42. Subsystem zur Symbolsynchronisation	98
5.43. Ermittlung der ganzzahligen Trägerverschiebung	99
5.44. Subsystem zur Frequenzsynchronisation	100
5.45. Subsystem zur Ermittlung der Lage des ersten Symbols im Rahmen	101
5.46. Beispiel zur Ermittlung der Lage des ersten Symbols (Mode B)	101
5.47. Subsystem zur Framesynchronisation	102
5.48. Subsystem zur Kanalschätzung und zum Kanalausgleich	102
5.49. Beispiel einer geschätzten Kanalübertragungsfunktion (Mode B, $B_S=0$) . .	103
A.1. Multilevel-Kodierung in DRM mit 3 Level für HMsym [Eur09, Abb.30] . .	114
A.2. Multilevel-Kodierung in DRM mit 3 Level für HMMix [Eur09, Abb.31] . .	114
A.3. Multilevel-Kodierung in DRM mit 2 Level SM [Eur09, Abb.32]	115
A.4. Multilevel-Kodierung in DRM mit 1 Level SM [Eur09, Abb.33]	115
A.5. Konstellationsdiagramm der 4-QAM [Eur09, Abb.39]	118
A.6. Konstellationsdiagramm der 64-QAM SM [Eur09, Abb.35]	118

Tabellenverzeichnis

2.1. Kategorien der Wellenausbreitung	4
2.2. Genutzte Frequenzbänder für den Hörrundfunk in Deutschland	5
2.3. Festgelegte Übertragungsmodi bei DRM	16
2.4. Spektralbelegung bzw. Übertragungsbandbreiten in DRM	17
3.1. Beispiel zum Scrambling-Prozess	21
3.2. Zusammensetzung des Codeworts und des Polynoms beim Faltungscoder .	24
3.3. Punktierungsraster für die Coderate 1/4	25
3.4. OFDM Parameter bei DRM	38
3.5. Anzahl der Subträger bei einer Abtastfrequenz von 12 kHz ($B_S = 0, 1, 2, 3$)	39
3.6. Anzahl der Subträger bei einer Abtastfrequenz von 48 kHz ($B_S = 4, 5$) . .	39
3.7. Trägernummer der Frequenzpiloten	41
3.8. Abstände der Gain-Referenzzellen	42
3.9. Für die Simulation gewählte Zwischenfrequenzen	45
A.1. Coderate und Schutzlevel für MSC mit 4-QAM (Mode E)	116
A.2. Coderate und Schutzlevel für MSC mit 16-QAM (Modi A-D)	116
A.3. Coderate und Schutzlevel für MSC mit 16-QAM (Mode E)	116
A.4. Coderate und Schutzlevel für MSC mit 64-QAM (Modi A-D)	117
A.5. Coderate und Schutzlevel für SDC mit 16-QAM (Modi A-D)	117
A.6. Coderate und Schutzlevel für SDC mit 4-QAM (Modi A-D)	117
A.7. Coderate und Schutzlevel für SDC mit 4-QAM (Mode E)	117
A.8. Coderate für FAC (4-QAM)	117
A.9. Übersicht über die verwendeten Initialisierungsdateien und ihre Funktionen 1(2)	119
A.10. Übersicht über die verwendeten Initialisierungsdateien und ihre Funktionen 2(2)	120
A.11. Übersicht über die für die Simulation genutzten Variablen und Strukturen	121

A. Anhang

A.1. Ergänzende Tabellen und Abbildungen des DRM Standards

A.1.1. Arten der Multilevelkodierung

Nachfolgend sind alle weiteren Strukturen der Multilevel-Codierung für die verschiedenen Mappingstrategien aus der DRM-Spezifikation abgebildet.

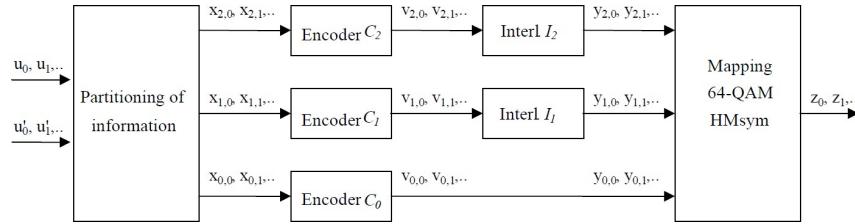


Abbildung A.1.: Multilevel-Kodierung in DRM mit 3 Level für HMsym [Eur09, Abb.30]

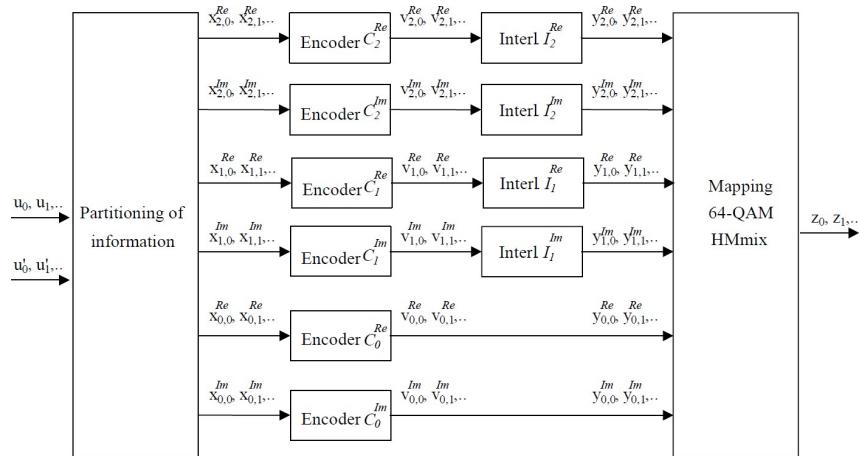


Abbildung A.2.: Multilevel-Kodierung in DRM mit 3 Level für HMmix [Eur09, Abb.31]

A. Anhang

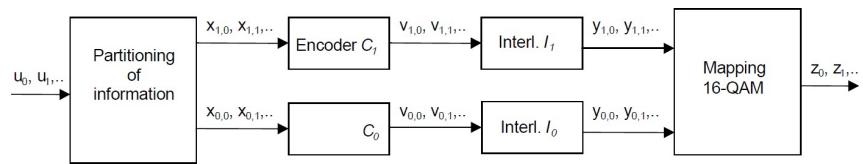


Abbildung A.3.: Multilevel-Kodierung in DRM mit 2 Level SM [Eur09, Abb.32]

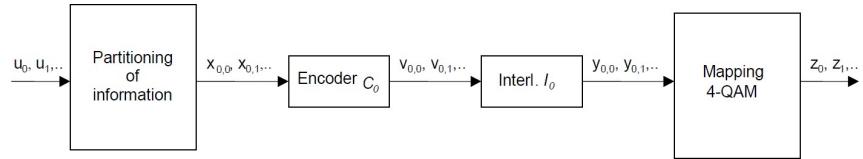


Abbildung A.4.: Multilevel-Kodierung in DRM mit 1 Level SM [Eur09, Abb.33]

A.1.2. Schutzlevel und Coderaten

Die folgenden Tabellen geben eine Übersicht über die in den verschiedenen Schutzlevel definierten Coderaten der einzelnen Kanäle bei unterschiedlicher QAM. Die Tabellen sind aus der Spezifikation [Eur09, S.132 ff.] entnommen. Die Bezeichnungen sind folgende:

R_{all} Gesamte Coderate aller Zweige des Multilevel-Kodierers

R_x Coderate der einzelnen Zweige der Multilevel-Kodierung und somit gleichzeitig die geforderte Punktierung des Faltungscodes (ohne Tail-Bits)

Dargestellt werden nur die Tabellen des Standard Mapping, da nur dieses in der Simulation umgesetzt wurde. Alle weiteren Varianten sind in der Spezifikation [Eur09, S.132 ff.] zu finden.

A.1.2.1. MSC

Schutzlevel	R_{all}	R_0
0	0,25	1/4
1	0,33	1/3
2	0,4	2/5
3	0,5	1/2

Tabelle A.1.: Coderate und Schutzlevel für MSC mit 4-QAM (Mode E)

Schutzlevel	R_{all}	R_0	R_1
0	0,5	1/3	2/3
1	0,62	1/2	3/4

Tabelle A.2.: Coderate und Schutzlevel für MSC mit 16-QAM (Modi A-D)

Schutzlevel	R_{all}	R_0	R_1
0	0,33	1/6	1/2
1	0,41	1/4	4/7
2	0,5	1/3	2/3
3	0,62	1/2	3/4

Tabelle A.3.: Coderate und Schutzlevel für MSC mit 16-QAM (Mode E)

Schutzlevel	R_{all}	R_0	R_1	R_2
0	0,5	1/4	1/2	3/4
1	0,6	1/3	2/3	4/5
2	0,71	1/2	3/4	7/8
3	0,78	2/3	4/5	8/9

Tabelle A.4.: Coderate und Schutzlevel für MSC mit 64-QAM (Modi A-D)

A.1.2.2. SDC

Schutzlevel	R_{all}	R_0	R_1
0	0,5	1/3	2/3

Tabelle A.5.: Coderate und Schutzlevel für SDC mit 16-QAM (Modi A-D)

Schutzlevel	R_{all}	R_0
1	0,5	1/2

Tabelle A.6.: Coderate und Schutzlevel für SDC mit 4-QAM (Modi A-D)

Schutzlevel	R_{all}	R_0
0	0,5	1/2
1	0,25	1/4

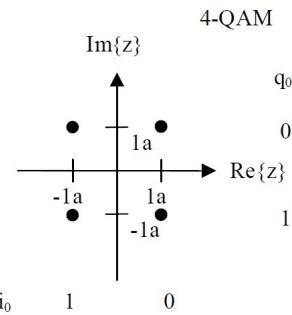
Tabelle A.7.: Coderate und Schutzlevel für SDC mit 4-QAM (Mode E)

A.1.2.3. FAC

Mode	R_{all}	R_0
A-D	0,6	3/5
E	0,25	1/4

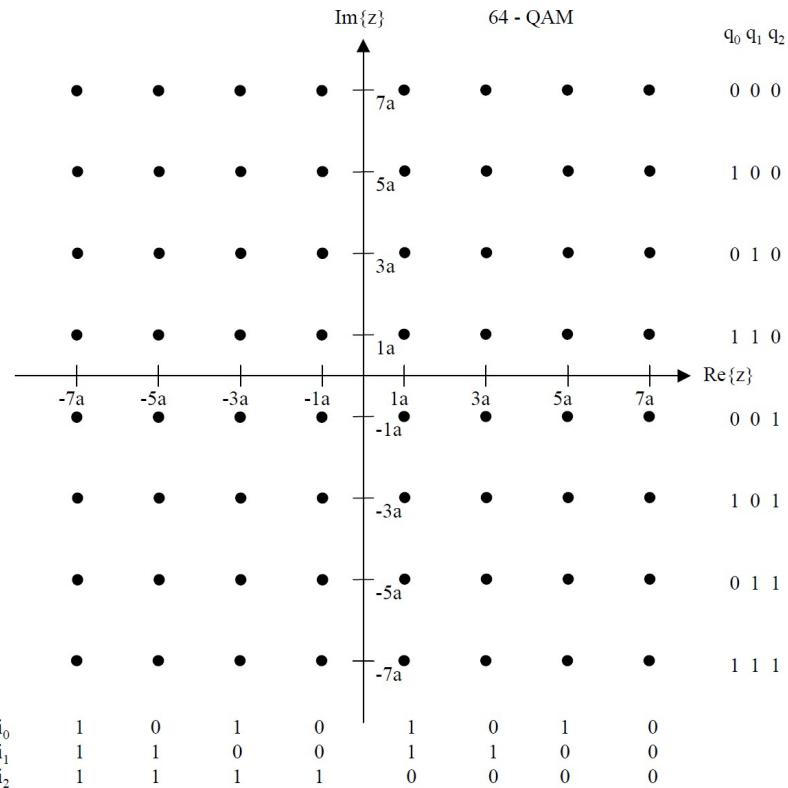
Tabelle A.8.: Coderate für FAC (4-QAM)

A.1.3. QAM Konstellationsdiagramme



Bit ordering: $\{i_0 \ q_0\} = \{y'_0 \ y'_1\}$

Abbildung A.5.: Konstellationsdiagramm der 4-QAM [Eur09, Abb.39]



Bit ordering: $\{i_0 \ i_1 \ i_2 \ q_0 \ q_1 \ q_2\} = \{y'_0 \ y'_1 \ y'_2 \ y'_3 \ y'_4 \ y'_5\}$

Abbildung A.6.: Konstellationsdiagramm der 64-QAM SM [Eur09, Abb.35]

A.2. Initialisierung

A.2.1. Dateien zur Initialisierung

Dateiname	Inhalt und Funktion
analysis.m	Berechnung der Verzögerungen für die BER-Messung
bit_interleaving.m	Umsetzung des beschriebenen Algorithmus für das Bit-Interleaving
cell_interleaving.m	Umsetzung des beschriebenen Algorithmus für das „short interleaving“ beim Cell-Interleaver
channel_est.m	Festlegung der Interpolationslängen für das FIR-Filter der Kanalschätzung
conv_and_puncturing.m	Beinhaltet das Generatorpolynom für den Faltungscoder und die Punktierungsraster für die Nutzdaten
crc_poly.m	Beinhaltet alle in der Spezifikation definierten Generatorpolynome für die CRC
dummy_cells.m	Bestimmt die Anzahl der Dummy-Zellen und weist diesen ihre komplexen Wert zu
energy_dispersal.m	Beinhaltet das Generatorpolynom und das Initialisierungswort für den Scrambler/Descrambler und legt die Samptime des PN-Generators und des Rücksetzvektors fest
fac_cells.m	Beinhaltet alle Indizes der in der Spezifikation definierten Trägernummern der FAC-Zellen für alle Modi
gain_ref_calc.m	Berechnet in Abhängigkeit von Mode und B_S die Lage, den Betrag und die Phase der Gain-Piloten nach der Spezifikation [Eur09, S.147 ff.]
if_modulation.m	Definiert die Zwischenfrequenzen, die Faktoren für Über-/Unterabtastung und die Filterordnung der Antiimaging-/Antialiasing-Filter und berechnet die Samptime und die Framegröße der Sinus-/Cosinus-Generatoren für die IQ-Modulation
init_fac.m	Bestimmt aus Mode und B_S die Größe des Eingangsframes und die Kanalkodierungsparameter für den FAC (Permutationsvektor, Punktierungsraster)
init_fft.m	Definiert die Grundabtastfrequenz des Basisbandsystems und bestimmt aus den Mode die OFDM-Parameter wie T_U , T_G , N_U usw., legt zudem den Verschiebungsvektor der negativen Trägerindizes bei der IFFT/FFT fest

Tabelle A.9.: Übersicht über die verwendeten Initialisierungsdateien und ihre Funktionen 1(2)

A. Anhang

Dateiname	Inhalt und Funktion
init_msc.m	Bestimmt aus Mode und B_S die Größe des Eingangsframes und die Kanalkodierungsparameter für den MSC (Permutationsvektoren, Punktierungsraster), bestimmt zudem das für die MLC genutzte Subsystem in Abhängigkeit des QAM-Typs
init_ofdm.m	Bestimmt die Zuweisungsvektoren der drei Kanäle für das OFDM-Mapping und legt die Matrizen der Pilotzellen an, wie sie für den Pilotgenerator verwendet werden
init_sdc.m	Bestimmt aus Mode und B_S die Größe des Eingangsframes und die Kanalkodierungsparameter für den SDC (Permutationsvektoren, Punktierungsraster), bestimmt zudem das für die MLC genutzte Subsystem in Abhängigkeit des QAM-Typs (Schutzzlevel)
ofdm_parameter.m	Legt die in der Spezifikation definierte Anzahl der Symbole eines Superframes fest, und ermittelt K_{min} und K_{max} aus der Abbildung 3.19 in Abhängigkeit vom gewählten Mode und B_S
qam_cells_select.m	Bestimmt die Anzahl der QAM-Zellen eines Frames der drei Kanäle aus den zugehörigen Tabellen der Spezifikation [Eur09, S.137] und [Eur09, S.113]
qam_mapping.m	Beinhaltet die Matrizen zur Bildung der Konstellationsdiagramme und die Normalisierungsfaktoren für die verschiedenen QAM-Typen
reference_cells.m	Beinhaltet alle in der Spezifikation definierten Trägerindizes und Phasen der Zeit-, Frequenz- und AFS-Referenzzellen
tailbit_select.m	Beinhaltet alle Punktierungsraster der Abschluss-Bits (Tailbits) und selektiert je nach Kodierungsart von SDC und MSC eines dieser
time_freq_sync.m	Detektiert Referenzpaare für die Framesynchronisation und bestimmt den Index der Frequenzpilot für die Frequenzsynchronisation nach der FFT

Tabelle A.10.: Übersicht über die verwendeten Initialisierungsdateien und ihre Funktionen 2(2)

A.2.2. Variablen der Initialisierung

Folgende Tabelle zeigt die bei der Initialisierung erzeugten Variablen und Strukturen, wie sie im *Workspace* der Simulation vorliegen und beschreibt ihre Bedeutung.

Variable	Inhalt
FAC_QAM	QAM-Typ des FAC (immer 4)
MSC_QAM	QAM-Typ des MSC (4, 16 oder 64)
SDC_QAM	QAM-Typ des SDC (4 oder 16)
channel	Struktur mit Parametern zur Kanalschätzung (Interpolationslänge des FIR-Filter)
conv_poly	Polynom des Faltungscoders in Trellis-Struktur
fac	Struktur des FAC (Permutationsvektoren, Punktierungsraster, Anzahl der QAM-Zellen u.a.)
if_mod	Struktur der Zwischenfrequenzverarbeitung (Zwischenfrequenz, Filterordnung, Faktoren für Über/Unterabt. u.a.)
mode	Robustness Mode ('A', 'B', 'C', 'D' oder 'E')
model_type	Name des Modells (zur Umstellung auf andere MATLAB-Versionen)
msc	Struktur des MSC (Permutationsvektoren, Punktierungsraster, Anzahl der QAM-Zellen u.a.)
ofdm	Struktur für OFDM-Parameter (K_{min} , K_{max} , Matrizen der Pilotzellen, Zuweisungsvektoren der Kanäle für das Mapping u.a.)
qam_map	Daten zur Erstellung der Konstellationsdiagramme der QAM
qam_norm	Normalisierungsfaktoren der QAM-Typen
sCRC_poly	Struktur der CRC (Generatorpolynome und Initialisierungswörter)
sFFT	Struktur der IFFT/FFT-Daten (N_U , N_G , T_U , u.a.)
sPunc_pat	Punktierungspattern der Nutzdaten
scrambler	Struktur für den Scrambler / Descrambler (Polynom, Initialisierungsword, Parameter für den PN-Generator u.a.)
sdc	Struktur des SDC (Permutationsvektoren, Punktierungsraster, Anzahl der QAM-Zellen u.a.)
spec_occupancy	Spektralbelegung B_S (0 - 5)
sync	Struktur der Synchronisationsprozesse (Referenzpaare zur Frame-sync., Indizes der Frequenzreferenzen für die Freq.sync. u.a.)

Tabelle A.11.: Übersicht über die für die Simulation genutzten Variablen und Strukturen

B. Inhalt des Datenträgers

Der dieser Masterarbeit beigelegte Datenträger besteht aus nachfolgend aufgelisteten Teilen, welche die angegebene Komponenten beinhalten:

Diorama Das für die Verifizierung verwendete MATLAB-Programm zum Empfang von DRM-Signalen als installierbare Version.
(URL:<http://nt.eit.uni-kl.de/static/diorama/download.html>)

DRM_Simu Vollständiges Modell der Simulation des DRM-Übertragungssystems für die MATLAB-Versionen 2009a und 2008a, mit allen zugehörigen m-Files

DRM_Wav Modell zur Verifizierung des Senders (siehe Abschnitt 5.4)

Referenzen Alle gesammelten Referenzen, die zur Umsetzung und zur Beschreibung der Arbeit Verwendung fanden. Die Dateien sind nach ihren Themen in weiteren Ordner abgelegt.

Samples Ein Teil der mit dem Modell zur Verifizierung erzeugten wav-Dateien.

es_201980v030101p.pdf ETSI ES 201 980 V3.1.1; Digital Radio Mondiale - Systemspezifikation

masters_thesis.pdf Elektronische Version der vorliegenden Ausarbeitung