

Entwurf und Evaluation einer Transceiver-Applikation für DAB/DAB+

Bachelorarbeit

Moritz Luca Schmid

Hauptreferent : Dr.-Ing. Holger Jäkel
Betreuer : M.Sc. Felix Wunsch

Beginn : 12.06.2017
Abgabe : 11.12.2017

Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig und unter Beachtung der Satzung des Karlsruher Instituts für Technologie (KIT) zur Sicherung guter wissenschaftlicher Praxis in der aktuellen Fassung angefertigt habe. Ich habe keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und wörtlich oder inhaltlich übernommene Stellen als solche kenntlich gemacht.

Karlsruhe, den 11.12.2017

Moritz Luca Schmid

Abstract

Die vorliegende Arbeit beschreibt den Entwurf und die anschließende Evaluation einer DAB/DAB+ Transceiver Applikation im Rahmen einer Bachelorarbeit.

Digital Audio Broadcasting (DAB) ist ein Übertragungsstandard zur Verbreitung von digitalem Hörfunk. Durch die digitale Übertragung und Signalverarbeitung kommen moderne Codierungs- und Modulationsverfahren, wie zum Beispiel OFDM, zum Einsatz.

Mittelfristig wird DAB die analoge UKW Übertragung ersetzen. DAB+ stellt eine Erweiterung zu DAB dar, bei der die weiterentwickelte Audiokompression HE-AAC v2 und eine zusätzliche Fehlerkorrektur eingesetzt wird.

Die theoretischen Grundzüge des DAB/DAB+ Übertragungsstandards werden erläutert, wobei sich die Arbeit an Leser richtet, die mit den Grundlagen der Nachrichtentechnik und Signalverarbeitung vertraut sind.

Die Implementierung des DAB/DAB+ Transceivers erfolgt mit der Open-Source Software GNU Radio. Das entstandene GNU Radio Modul *gr-dab* ist auf GitHub [CEL17] veröffentlicht. Das Modul enthält alle implementierten Signalverarbeitungsblöcke, aus denen ein funktionsfähiger und standardkonformer DAB/DAB+ Transmitter und Receiver aufgebaut werden kann. Dabei sind die meisten Blöcke generisch realisiert, sodass beispielsweise die implementierten Audio-Codecs auch in anderen auf GNU Radio basierenden Projekten verwendet werden können.

Die praktischen Realisierungsschritte des DAB/DAB+ Transceivers werden aufgezeigt, wobei der Fokus der Betrachtung auf der Softwareimplementation liegt.

Zentrale Systemdesignfragen werden ausführlich diskutiert. Die Synchronisation des Empfängers arbeitet mit einer einfachen und sehr effektiven Korrelation des Cyclic Prefix, die sowohl für die Zeit- als auch für die Frequenzsynchronisation eingesetzt wird und die Empfängerstruktur insgesamt sehr einfach hält.

Der implementierte DAB/DAB+ Transceiver ist echtzeitfähig und besitzt eine Empfangsqualität die mit kommerziellen DAB-Geräten vergleichbar ist. Das realisierte Übertragungssystem ist nahezu vollständig auf Softwarebasis implementiert wodurch generalisierte Hardware für den Transceiver verwendet werden kann.

Der Empfang wird mit verschiedenen Hardwarekonfigurationen, von hochwertigen Universal Software Radio Peripherals (USRPs) bis hin zu kostengünstigen RTL-SDR-Dongles, erfolgreich getestet. Durch eine passend gewählte Kanalcodierung ist ein robuster Empfang auch in ungünstigen Empfangssituationen möglich.

Die entwickelte Applikation *DABstep* ergänzt den DAB/DAB+ Transceiver um ein vollwertiges graphisches Benutzerfrontend und erlaubt damit die einfache und benutzerfreundliche Konfiguration und Bedienung des Transceivers.

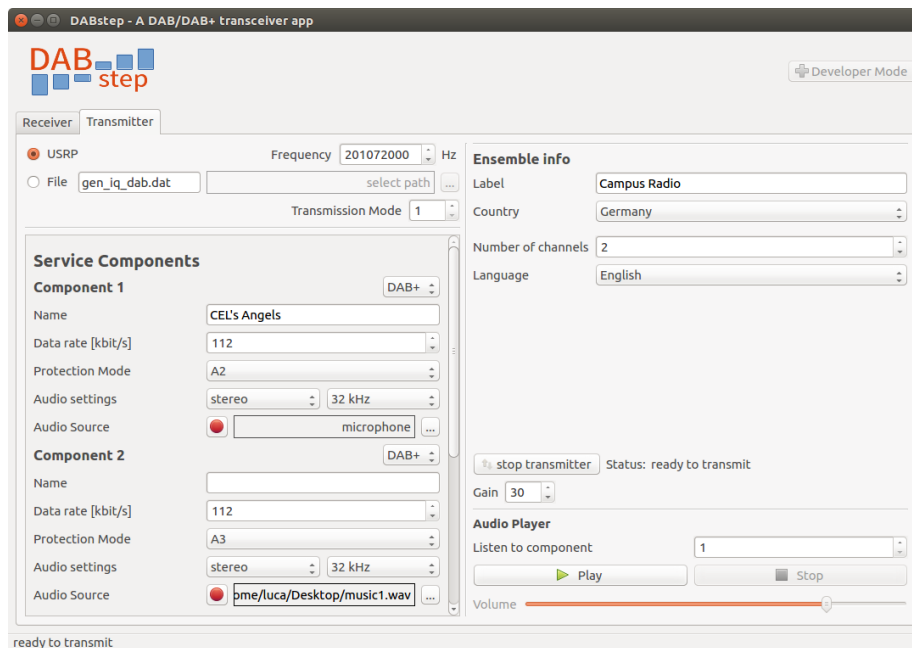


Abbildung 0.1.: DABstep im Sendemodus

Der Transmitter (Abb. 0.1) ermöglicht den Aufbau eines DAB Ensembles, das aus mehreren DAB und DAB+ Audiokanälen bestehen kann. Die Applikation erlaubt eine sehr detaillierte und flexible Konfiguration des Senders, wobei jeder Kanal individuell eingestellt werden kann.

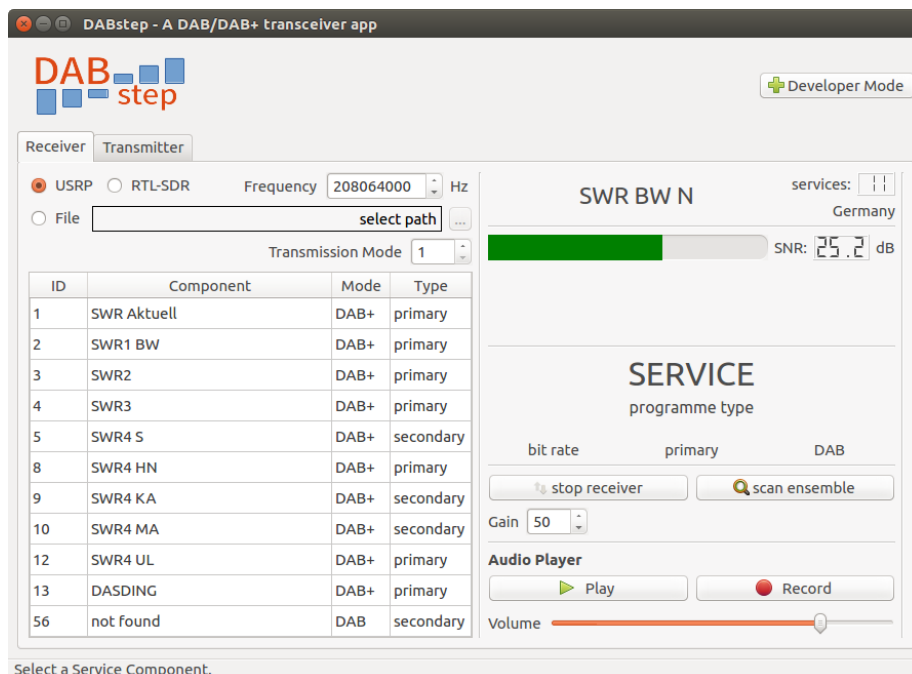


Abbildung 0.2.: DABstep im Empfängermodus

Für den DAB/DAB+ Empfang können aufgenommene IQ Daten oder Hardware wie ein USRP oder ein RTL-SDR-Dongle verwendet werden. Der Empfänger (Abb. 0.2) zeigt automatisch alle empfangenen Radiosender mit den zugehörigen Informationen an. Über den integrierten Audio-Player kann das Abspielen der Audiokanäle gesteuert werden.

Ein weiteres Feature stellt der Entwicklermodus (Abb. 0.3) dar. Er erweitert den Empfänger optional um einige technische Anzeigen, wie zum Beispiel ein Konstellationsdiagramm und mittlere Fehlerraten der verschiedenen Kanäle.

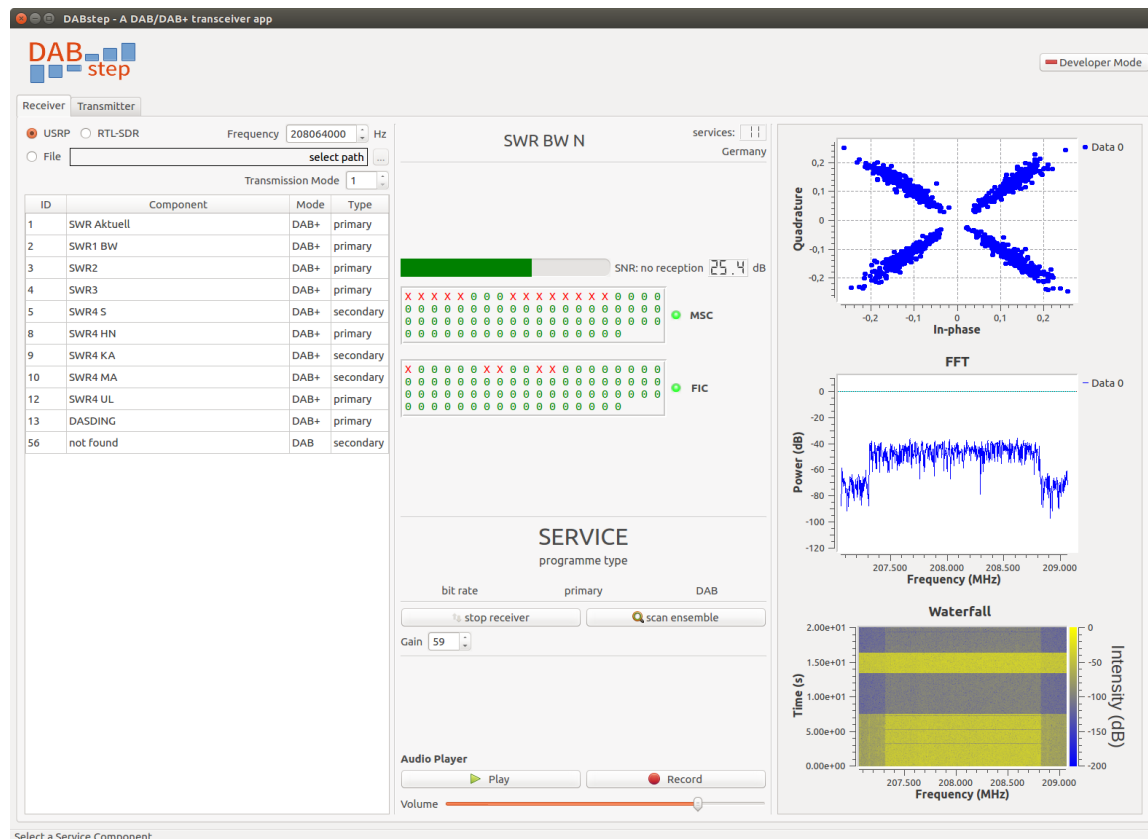


Abbildung 0.3.: DABstep mit aktivem Entwicklermodus

DABstep stellt damit die erste frei zugängliche Applikation für DAB/DAB+ dar, die sowohl einen Sender als auch einen Empfänger enthält.

Das System wird schließlich auf seine Performanz und Robustheit in verschiedenen Übertragungskanälen untersucht. Dazu wird ein AWGN Kanal und verschiedene Fadingkanäle simuliert, die typische Übertragungsszenarien von DAB darstellen. Bei den Fadingmodellen wird zudem der Doppler-Spread variiert um die Auswirkung verschiedener Geschwindigkeiten eines mobilen Empfängers zu untersuchen. Neben Bitfehlerraten werden auch Paketfehler-raten simuliert, die eine subjektiv empfundenen Empfangsqualität gut abbilden können. Dazu wird die gesamte Kanalcodierungskette in die Simulation integrieren. Im AWGN Kanal erreicht der Empfänger schon bei kleinem Signal-zu-Rausch-Verhältnis

sehr geringe Paketfehlerraten von unter 10^{-3} . Abb. 0.4 zeigt zudem den starken Einfluss der Kanalcodierung auf die Empfangsqualität.

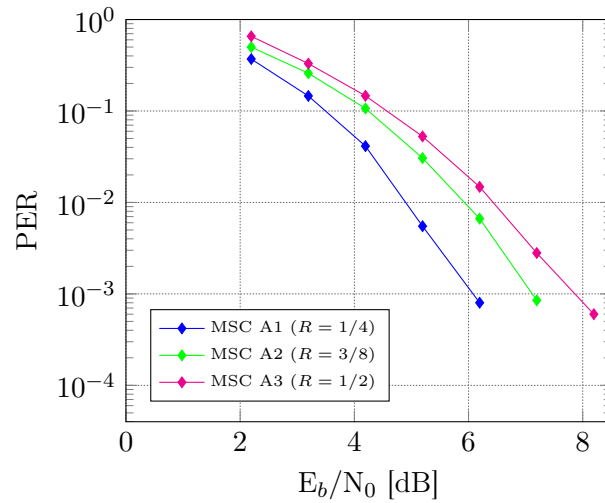


Abbildung 0.4.: PER bei AWGN

Im Fadingkanal erweist sich das System als robust gegen Mehrwegempfang mit hohen Delay-Spreads. Es wurde eine starke Abhängigkeit der Performanz vom Doppler-Spread festgestellt, was zu deutlich erhöhten Bitfehlerraten führt. Die resultierenden Paketfehlerraten erreichen durch die Fehlerkorrektur der Kanalcodierung bis zu einer maximalen Dopplerverschiebung von 20 Hz zufriedenstellende Ergebnisse (Abb. 0.5).

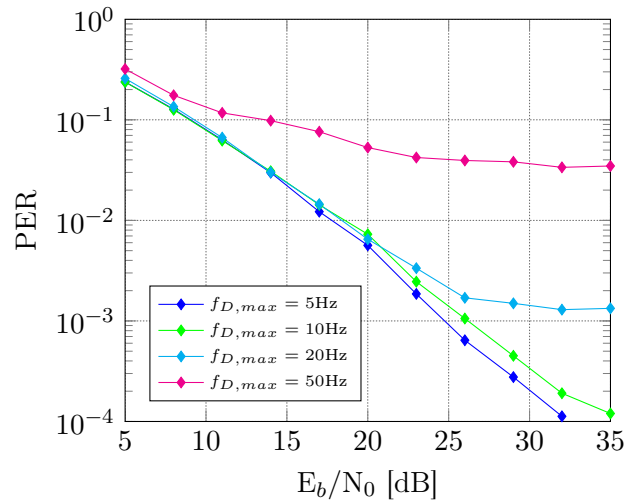


Abbildung 0.5.: PER für Fadingkanal, Coderate $R = 1/3$

Inhaltsverzeichnis

1. Einführung	3
2. DAB Standard	5
2.1. Aufbau eines DAB Ensembles	5
2.2. Fast Information Channel	6
2.2.1. Zyklische Redundanzprüfung (CRC)	7
2.2.2. Scrambling	7
2.2.3. Faltungscodierung	7
2.3. Main Service Channel	8
2.3.1. Audio Kompression	8
2.3.2. Scrambling	9
2.3.3. Faltungscodierung	9
2.3.4. Zeit-Interleaving	10
2.3.5. Multiplexing	10
2.4. Erzeugung des Sendesignals	10
2.4.1. Partitionierung des Sendeframes	11
2.4.2. QPSK Modulation	12
2.4.3. Frequenz-Interleaving	12
2.4.4. Differentielle Modulation	12
2.4.5. Orthogonales Frequenzmultiplexverfahren (OFDM)	13
3. Implementierung eines DAB/DAB+ Transceivers	15
3.1. GNU Radio	15
3.2. Vorgehen bei der Implementierung	16
3.3. Implementierung des Senders	17
3.3.1. FIB Quelle	17
3.3.2. FIC Encoder	18
3.3.3. Audio Quellen und Encoder	18
3.3.4. MSC Encoder	19
3.3.5. Multiplexer	20
3.3.6. OFDM Modulator	20
3.4. Implementierung des Empfängers	22
3.4.1. Zeit Synchronisation	22
3.4.2. Frequenz-Synchronisation	27
3.4.3. FFT	29
3.4.4. Frequenz-Interleaving	29
3.4.5. Demodulation	29
3.4.6. Kanaldecodierung	30
3.4.7. FIC Senke	30
3.4.8. Audio Decoder	31

4. Implementierung einer grafischen Transceiver Applikation	33
4.1. Klassenstruktur der Applikation	33
4.1.1. Python main	33
4.1.2. Grafisches Frontend	33
4.1.3. Umsetzungstabellen	33
4.2. GNU Radio Flowgraphs	35
4.3. Datenfluss	35
4.3.1. JSON	36
4.4. Grafischer Aufbau	36
5. Evaluation des Systems	41
5.1. Verifikation des Empfängers	41
5.2. AWGN Kanal	42
5.2.1. Bitfehlerrate	42
5.2.2. Paketfehlerrate	43
5.3. Realistische Kanalmodelle	46
5.3.1. Kohärenzbandbreite	47
5.3.2. Doppler-Spread	48
5.3.3. Bitfehlerrate	48
5.3.4. Paketfehlerrate	49
6. Fazit	51
A. Abkürzungsverzeichnis	53

1. Einführung

Digital Audio Broadcasting (DAB) ist ein Übertragungsstandard zur Verbreitung von digitalem Radio. Das System wurde in den 90er Jahren im Zuge des Eureka 147 DAB Projekts entwickelt und kommt seit 1997 in vielen Teilen Europas und Asiens zum Einsatz. Mit dem DAB+ System, das eine Weiterentwicklung von DAB darstellt, soll langfristig die analoge Ultrakurzwellen (UKW) Übertragung ersetzt werden.

Die digitale Datenübertragung bietet dabei eine verbesserte Audioqualität, da Übertragungsfehler durch Codierung empfängerseitig korrigiert werden können und somit nicht zu einem verrauschten Audiosignal wie bei FM führen. Eine Audiokompression verringert zusätzlich die Datenrate, was sich in einer gesteigerten spektralen Effizienz pro Audiokanal widerspiegelt. Die digitale Übertragung bietet außerdem viele neue Möglichkeiten der medialen Unterstützung in Form von Service Informationen wie Albumcovers, ausführlichen Stauinformationen oder Wetterkarten.

DAB Sender sind in sog. Ensembles strukturiert. Ein DAB Ensemble enthält ein ganzes Multiplex an Audio- und Datenkanälen. Radiosender dieses Ensembles greifen auf eine Auswahl dieser Kanäle zu. Dieses dynamische Multiplex erlaubt eine flexible und individuelle Programmgestaltung.

DAB stellt vier verschiedene Übertragungsmodi zur Verfügung, die für verschiedene Ausbreitungsszenarien und Frequenzbereiche ausgelegt sind. Der Fokus liegt dabei auf dem mobilen Empfang bei terrestrischer Übertragung, es existiert aber auch jeweils ein Modus für die Satellitenübertragung, sowie die niederfrequente Übertragung per Kabel. Eine Besonderheit bei DAB stellt der Einsatz von sog. Single Frequency Networks (SFNs) dar, bei denen eine Vielzahl von örtlich getrennten Sendestationen auf der gleichen Frequenz ausstrahlen. Dadurch kann ein enormer Gewinn an Frequenzeffizienz erzielt werden. Der Einsatz von SFNs wird bei DAB durch eine hohe Robustheit des Übertragungssystems gegenüber Mehrwegeempfang ermöglicht [HL09].

Ziel dieser Arbeit ist die Implementierung und Evaluation eines Senders und Empfängers für DAB/DAB+. Die Implementierung wird dabei in Form eines Software Defined Radios erfolgen, einem System bei dem ein Großteil der Signalverarbeitung auf Softwareebene durchgeführt wird. Dazu wird die Open-Source Software GNU Radio verwendet. Im Weiteren soll eine grafische Transceiver Applikation realisiert werden, die auf Basis des implementierten Senders und Empfängers eine benutzerfreundliche Oberfläche für das Senden und das Empfangen von DAB/DAB+ Signalen darstellt.

2. DAB Standard

In diesem Kapitel werden die verwendeten Strukturen und Verfahren vorgestellt, die nötig sind um ein funktionierendes DAB¹ Übertragungssystem aufzubauen. Eine vollständige und detailliertere Beschreibung über den kompletten DAB Standard liefert das vom Europäischen Institut für Telekommunikationsnormen (ETSI) veröffentlichte Dokument [ETS17c]. Die in Kapitel 1 angesprochenen Übertragungsmodi basieren alle auf dem selben Prinzip und sind nur durch die Wahl der Parameter auf ihre jeweiligen Einsatzbereiche angepasst. Diese Arbeit bezieht sich bei quantitativen Betrachtungen immer auf den Übertragungsmodus I, der bei der terrestrischen Übertragung verwendet wird. In der praktischen Implementierung wurden alle 4 Übertragungsmodi berücksichtigt.

Abb. 2.1 zeigt als Übersicht ein Blockdiagramm des DAB Transmitters, auf dessen einzelnen Elemente in diesem Kapitel näher eingegangen wird.

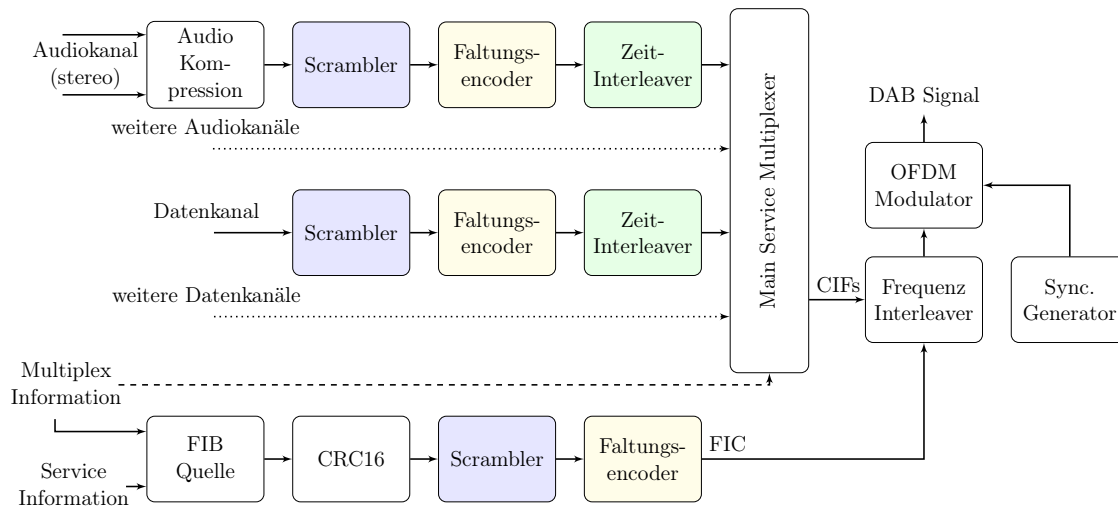


Abbildung 2.1.: Blockdiagramm eines DAB Transmitters

2.1. Aufbau eines DAB Ensembles

Der Aufbau einer DAB Struktur gestaltet sich wesentlich komplexer, aber auch flexibler als die feste Struktur einer UKW Übertragung, bei der jeder Radiosender genau einen Audio-kanal überträgt und ein eigenes Frequenzband belegt. Ein sog. DAB Ensemble beinhaltet in der Regel mehrere Radiosender², die wiederum aus vielen Audio- und Datenkanälen

¹Mit der Bezeichnung DAB ist in den folgenden Kapiteln stets das Übertragungssystem DAB sowie dessen Nachfolger DAB+ gemeint. Bei Unterschieden der beiden Standards wird darauf explizit eingegangen.

²„Radiosender“ bezeichnet hier die Organisationseinheit (engl. „Service“) und nicht den physikalischen Sender.

bestehen können. Ein Beispiel für den Aufbau eines DAB Ensembles zeigt Abb. 2.2. Das Ensemble „Campus Radio“ enthält zwei Radiosender, die jeweils einen separaten Audiokanal beinhalten und beide den Datenkanal „Stauinfo“ sowie den Audiokanal „Nachrichten“ nutzen.

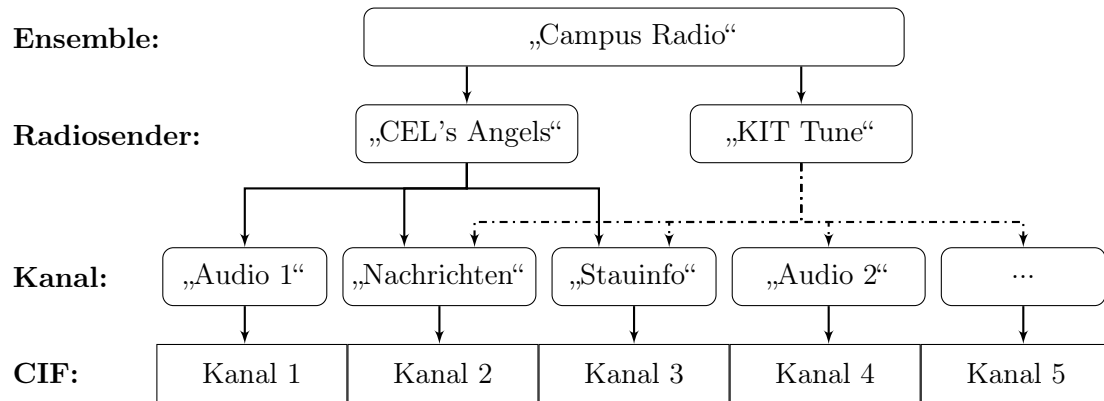


Abbildung 2.2.: Beispielhafte Struktur eines DAB Ensembles

Es erfolgt also eine Entkopplung der Radiosender von den Audio- bzw. Datenkanälen. Diese Trennung erlaubt es, dass mehrere Radiosender den selben Kanal nutzen können, wie es auch im Beispiel 2.2 gezeigt ist. Das spart einerseits redundante Datenübertragung und ermöglicht es den Radiosendern andererseits ihr Programm bzw. Angebot breiter zu fächern und flexibel anzupassen. Alle Kanäle der verschiedenen Radiosender werden im Main Service Multiplexer zu einem Common Interleaved Frame (CIF) gebündelt und in 24ms Frames übertragen.

Ohne Kenntnis über den Aufbau und die Struktur des Multiplex ist ein CIF am Empfänger wertlos, da keine Information über die Lage einzelner Audio Streams im Multiplex bekannt ist. Die Übertragung dieser Multiplex Configuration Information (MCI) ist Aufgabe des Fast Information Channel (FIC).

2.2. Fast Information Channel

Der FIC spielt bei der Übertragung eines DAB Ensembles eine sehr wichtige Rolle. Er überträgt sowohl die MCI, als auch Service Information (SI), wie zum Beispiel den Namen eines Radiosenders. Die Informationen werden paketweise in sogenannten **FIB!s** (**FIB!s**) übertragen.

Die Bedeutung der MCI wurde schon erläutert und ist naheliegend. Auch die SI ist von großer Bedeutung, da ein Nutzer das Radioprogramm zum Beispiel nicht ohne dessen Namen auswählen kann.

Die Übertragung des FIC erfordert deshalb eine hohe Robustheit gegen Übertragungsfehler, was durch eine gute Kanalcodierung sichergestellt wird. Abb. 2.3 zeigt die Kanalcodierungskette des FIC, deren Elemente im Folgenden erläutert werden.

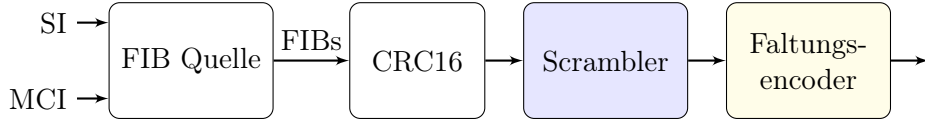


Abbildung 2.3.: Quelle und Kanalcodierung des FIC

2.2.1. Zyklische Redundanzprüfung (CRC)

Jedem **FIB!** der Länge 30 Byte wird ein 16 bit CRC (Cyclic Redundancy Check) Wort angehängt. Das CRC Wort wird über die 30 Nutzbytes mittels des Generatorpolynoms

$$G(x) = x^{16} + x^{12} + x^5 + 1 \quad (2.1)$$

berechnet und im Empfänger mit einer erneuten Berechnung über die Empfangsbits verglichen. Der CRC bietet keine Möglichkeiten zur Fehlerkorrektur, sondern lediglich zur Fehlerdetektion, wofür er nach [ITU97] optimiert ist.

2.2.2. Scrambling

Inhomogene Verteilungen der Datenbits, zum Beispiel durch lange Nullsequenzen, haben schlechte Auswirkungen auf das Peak-to-Average Power Ratio (PAPR). Um die Bitenergie daher gleichmäßiger über der Zeit zu verteilen, wird der Bitstream über eine Modulo 2 Operation mit einer Pseudo-Random Binary Sequence (PRBS) verwürfelt. Die PRBS erfüllt die gewünschten Eigenschaften und kann über das Polynom

$$P(X) = X^9 + X^5 + 1 \quad (2.2)$$

eindeutig generiert werden. Im Empfänger ist die PRBS über das gleiche Polynom exakt reproduzierbar.

2.2.3. Faltungscodierung

Eine Kanalcodierung ermöglicht die Erkennung und Korrektur von Bitfehlern auf Kosten einer geringeren Nutzdatenrate. Das DAB System verwendet eine Faltungscodierung mit der Coderate $R = 1/4$ und einer Einflusslänge von 7 bit. Für jedes Eingangsbit produziert der Encoder also ein Codewort der Länge 4 bit, das über die Polynome

$$\begin{aligned} g_1(x) &= 1 + x^2 + x^3 + x^5 + x^6 \\ g_2(x) &= 1 + x + x^2 + x^3 + x^6 \\ g_3(x) &= 1 + x + x^4 + x^6 \\ g_4(x) &= 1 + x^2 + x^3 + x^5 + x^6 \end{aligned} \quad (2.3)$$

berechnet wird.

Um die Coderate R zu erhöhen wird die punktierte Faltungscodierung verwendet, bei der aus einer Gruppen von Codebits ein Teil der Bits wieder gestrichen (punktiert) wird. Dadurch ergeben sich eine Vielzahl an möglichen Coderaten

$$R_{\text{punktiert}} = \frac{8}{8 + PI} \quad \text{mit} \quad PI \in [1; 24] \quad (2.4)$$

wobei $R = 1/4 \leq R_{\text{punktiert}} \leq 1$ gilt. Der FIC verwendet eine konstante Coderate von $R_{\text{punktiert}}$, FIC = $1/3$.

2.3. Main Service Channel

Der Main Service Channel (MSC) umfasst alle Audio- und Datenkanäle, die im DAB Ensemble enthalten sind. Wie in Abb. 2.4 zu sehen ist, wird jeder Kanal separat codiert. Audiokanäle werden zusätzlich vor ihrer Kanalcodierung komprimiert. Anschließend werden alle Kanäle durch einen Multiplexer zu sogenannten Common Interleaved Frames (CIFs) zusammengefasst. Ein CIF beinhaltet ein sog. Logisches Frame aus jedem Kanal. Das Logische Frame umfasst dabei die Daten jedes Streams in einem Zeitfenster von 24 ms. Die Größe variiert daher in Abhängigkeit der Datenrate eines jeden Kanals. Neben dem Stream Modus, der bei allen Audiokanälen und manchen Datenkanälen verwendet wird, können Daten auch paketweise übertragen werden. Dies wird zum Beispiel bei der Übertragung von Bildern verwendet.

Im Folgenden ist die Kanalcodierungskette und der Multiplex eines Audiokanals beschrieben.

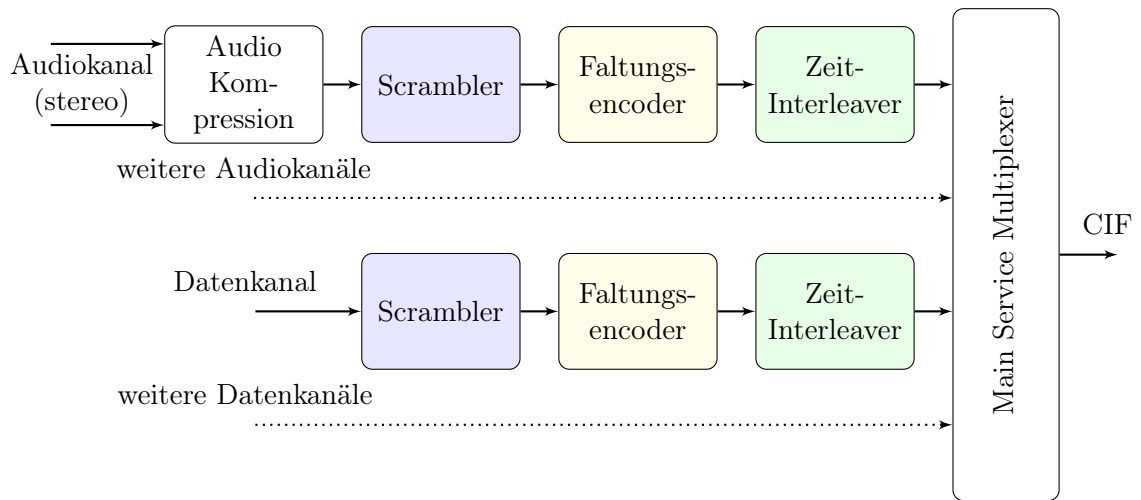


Abbildung 2.4.: Aufbau des Main Service Channels

2.3.1. Audio Kompression

Die subjektiv empfundene Qualität der übertragenen DAB Audio Streams soll der einer Audio CD gleichen. Audio CDs verwenden 16 bit PCM pro Monokanal bei einer Sample-rate von 44,1 kHz. Dies entspricht einer gesamten Bitrate von 1,41 Mbit/s für eine einzelne Audioübertragung mit Stereokanälen. Nimmt man nun noch eine Faltungscodierung mit exemplarischer Coderate von $R = 1/2$ in die Rechnung auf, übersteigt schon diese einzelne Audioübertragung die Gesamtkapazität des Main Service Channel (MSC) von 2,304 Mbit/s (siehe 2.3.5). Um eine Vielzahl von Audiokanälen im MSC unterbringen zu können, ist die Notwendigkeit einer Audiokompression ersichtlich, die eine Datenreduktion von etwa einer Größenordnung erreichen muss. Die verwendete Audiokompression ist MPEG 1/2 Audio Layer II (MPEG 2) [ETS16] für den DAB Standard und der optimierte Audiocodex MPEG

4 HE-AACv2 (MPEG 4) für DAB+ [ETS17a].

MPEG 2 ist ein generischer Audiocodec, der verschiedene Effekte zur verlustbehafteten Kompression verwendet. Zum einen wird die Redundanz des Audiosignals reduziert, indem durch statistische Korrelation eine Vorhersage über das Zeitsignal getroffen werden kann [HL09]. Zum anderen verwendet MPEG 2 ein psychoakustisches Modell des menschlichen Ohres. Damit ist es möglich die spektrale Hörschwelle an einem bestimmten Zeitpunkt zu bestimmen. Diese Information kann genutzt werden, um das in 32 Frequenzbänder zerteilte Signal in Abhängigkeit der temporären Empfindlichkeit des Ohres zu quantisieren und somit das Quantisierungsrauschen in jedem Frequenzband knapp unter der Hörschwelle zu halten. Abb. 2.5 zeigt dieses Prinzip als Blockschaltbild. Das psychoakustische Modell wird dabei zur Laufzeit in Abhängigkeit der Eingangsdaten simuliert und steuert die Quantisierung und Codierung somit dynamisch.

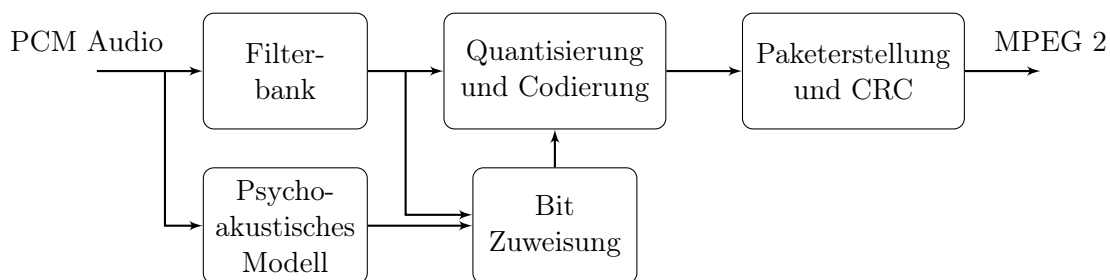


Abbildung 2.5.: Blockdiagramm der MPEG 2 Kompression

Mit MPEG 2 lässt sich die erwähnte subjektive CD Qualität mit einer Bitrate von etwa 192 kbit/s erreichen. Eine noch stärkere Komprimierung bietet MPEG 4, welches nach subjektiven Hörtests dieselbe Qualität bei einer geringeren Daterate von 96 - 128 kbit/s erreicht [EBU03]. MPEG 4 stellt eine Weiterentwicklung von MPEG 2 dar und basiert auf dem selben Prinzip. Um die Datenrate noch weiter zu reduzieren werden Verfahren wie Spektralbandreplikation und Kanalkopplung (Joint-Stereo) genutzt. Der MPEG 4 Codec wurde zusammen mit einer zusätzlichen Reed-Solomon Fehlerkorrektur in den DAB Standard integriert. Dieser wird in Deutschland seit 2011 als DAB+ ausgestrahlt.

2.3.2. Scrambling

Das Scrambling wird wie in Kapitel 2.2.2 auch im MSC durchgeführt.

2.3.3. Faltungscodierung

Die in Abschn. 2.2.3 beschriebene Faltungscodierung mit anschließender Punktierung kann im MSC genutzt werden, um die Coderate jedes einzelnen Kanals individuell einzustellen. Dabei stehen sog. Protection Levels zur Verfügung, die durch eine Kombination verschiedener Punktierungsmuster Coderaten von $R = 1/4$ bis $4/5$ erzeugen.

Neben der Equal Error Protection (EEP) gibt es auch die Möglichkeit einzelne Bit-Blöcke innerhalb eines Frames stärker zu codieren als andere. Diese Unequal Error Protection (UEP) ist zum Beispiel für MPEG Audio Frames sehr sinnvoll, um einzelne, für die subjektive

Audioqualität wichtigen Bits mit einer sehr hohen Redundanz zu versehen, ohne dabei die Gesamtcoderate deutlich zu erhöhen. Ein Bitfehler im Header kann beispielsweise zu einer Fehlskalierung eines ganzen Frequenzbandes führen, was sich beim Hörer als unangenehmes Pfeifen (sog. „birdies“) auswirkt, wogegen ein einzelner Bitfehler im Zeitsignal nahezu unbemerkt bleibt.

2.3.4. Zeit-Interleaving

Die Faltungscodierung kommt schnell an ihre Grenzen, wenn anstelle von einzelnen Bitfehlern ganze Bündelfehler auftreten. Solche zeitlich begrenzten Empfangsausfälle entstehen zum Beispiel bei mobilen Empfängern durch Fadingeffekte beim Mehrwegempfang. Im MSC befindet sich mit dem Zeit-Interleaving deshalb ein zusätzliches Element in der Kanalcodierungskette, das die nicht-korrigierbaren Bündelfehler über die Zeit verteilt, wodurch viele korrigierbare Einzelbitfehler entstehen. Der Zeit-Interleaver verzögert die Bits eines Frames nach den Regeln eines Interleavingvektors um 0 bis 15 Logische Frames, was einer maximalen Verzögerung von $T_{\text{delay, max}} = 15 \cdot 24 \text{ ms} = 360 \text{ ms}$ nach sich zieht. $T_{\text{delay, max}}$ entspricht also der maximalen Länge von Bündelfehlern, die vom Zeit-Interleaver verteilt werden können. Gleichzeitig ist $T_{\text{delay, max}}$ aber auch die Verzögerung der Gesamtübertragung. Ein Frame kann erst decodiert werden nachdem alle Elemente, die maximal eine Verzögerung von $T_{\text{delay, max}}$ haben, empfangen wurden. Diese konstante Verzögerung wird im MSC für die verbesserte Fehlerkorrekturfähigkeit in Kauf genommen. Im FIC ist solch eine Verzögerung nicht akzeptabel, da eventuell zeitkritische Daten, wie zum Beispiel die Umkonfiguration des DAB Multiplexes, übertragen werden. Deshalb wird im FIC kein Zeit-Interleaving angewendet.

2.3.5. Multiplexing

Im Multiplexer werden alle Kanäle des MSC gebündelt. Dabei beinhaltet das resultierende CIF jeweils ein Logisches Frame aus jedem Kanal. Das CIF wird in sog. Capacity Units (CUs) von jeweils 64 bit partitioniert und besitzt eine Gesamtkapazität von 864 CUs, also 55296 bit. Über die CUs erfolgt auch die Adressierung der einzelnen Kanäle, welche die wichtigste Komponente der in 2.2 beschriebenen MCI ausmacht.

Die Kapazität des CIFs ist groß genug um eine Vielzahl von Audio- und Datenkanälen zu transportieren. Ein typischer ³ 16 bit PCM Stereo Audio Stream mit der Abtastrate 32 kHz hat nach der Komprimierung eine Bitrate von $2 \cdot 64 \text{ kbit/s}$. Bei einem Protection Level von A3 ($\hat{=}$ Coderate $R = 1/2$) können also beispielsweise 10 Audiostreams parallel übertragen werden. Wird das CIF nicht komplett mit Kanälen gefüllt, werden die unbesetzten CUs mit einer PRBS (siehe 2.2.2) aufgefüllt, um eine ungleiche Energieverteilung im CIF zu verhindern.

2.4. Erzeugung des Sendesignals

Die generierten und codierten Bitstreams des FIC und MSC sollen nun per Funk vom Sender zum Empfänger übertragen werden. Um die Übertragung dabei möglichst robust und

³Bezogen auf ein in Karlsruhe, Deutschland empfangenes DAB+ Ensembles des Südwestrundfunks.

effizient zu gestalten, werden Verfahren, wie zum Beispiel Orthogonales Frequenzmultiplexverfahren (OFDM) und verschiedene Interleaver, eingesetzt. Die Modulationskette wird nun schrittweise erläutert.

2.4.1. Partitionierung des Sendeframes

Die Daten aus FIC und MSC werden vor der OFDM Modulation jeweils zu einem Sendeframe zusammengefasst. Ein Frame enthält dabei 4 CIFs und deren zugehörigen FIBs, die einer Streamdauer von $4 \cdot 24 \text{ ms} = 96 \text{ ms}$ entsprechen. Die Strukturierung des Sendeframes erfolgt in sogenannten OFDM Symbolen, die Pakete von jeweils 1536 QPSK Symbolen darstellen. $K = 1536$ entspricht dabei der Anzahl der genutzten OFDM Unterträger (siehe Abschn. 2.4.5). Ein OFDM Symbol wird mit $z_{m,l,k}$ bezeichnet, wobei m das Frame, l das OFDM Symbol und k den Unterträger indexiert. Im Folgenden wird bis auf Gl 2.10 die kürzere Bezeichnung $z_{l,k}$ verwendet. Neben den Datensymbolen beinhaltet jedes Frame zusätzlich das Nullsymbol und das Phasenreferenzsymbol für Synchronisationszwecke. Die Struktur des Sendeframes mit den insgesamt 77 OFDM Symbolen ist in Abb. 2.6 dargestellt.

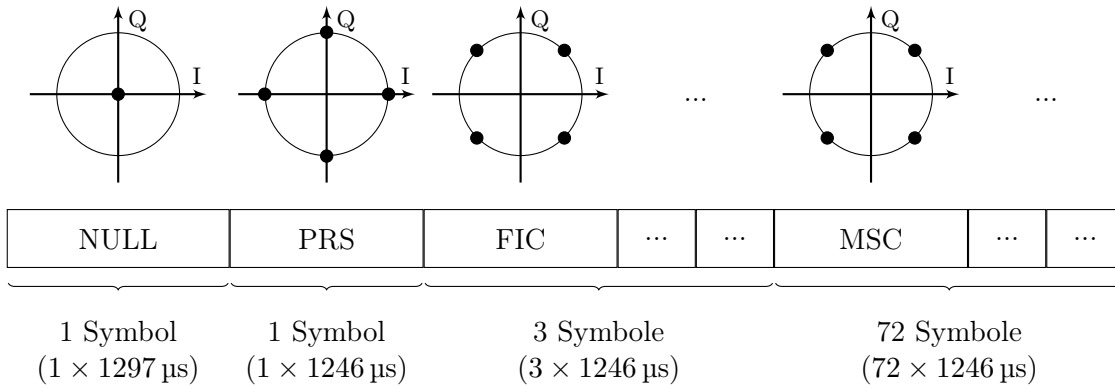


Abbildung 2.6.: Belegung und Modulation der OFDM Symbole

Nullsymbol

Das erste OFDM Symbol jedes Sendeframes ist das Nullsymbol $z_{0,k}$, für das

$$s(t) = 0 \quad \text{für} \quad t \in [0, T_{\text{NULL}}] \quad (2.5)$$

gilt. Diese Energielücke von bekannter Dauer $T_{\text{NULL}} = 1,297 \text{ ms}$ stellt eine robuste Möglichkeit für eine grobe Zeitsynchronisation auf die Sendeframes dar.

Phasenreferenzsymbol

Das Phasenreferenzsymbol $z_{1,k}$ bildet das zweite Symbol jedes Frames. Es enthält eine vordefinierte Sequenz an komplexen Constant Amplitude Zero Auto-Correlation (CAZAC) Symbolen, deren Amplitude stets konstant, und deren Autokorrelation null ist. Das Phasenreferenzsymbol dient zum einen, namensgebend, als Bezugsphase für differentielle Modulation der nachfolgenden D-QPSK Datensymbole. Zum anderen eignet es sich aufgrund

seiner CAZAC Eigenschaften und der Tatsache, dass das Phasenreferenzsymbol auch im Empfänger bekannt ist, ideal für eine feine Zeitsynchronisation, sowie für eine feine und grobe⁴ Frequenzsynchronisation durch Kreuzkorrelation.

2.4.2. QPSK Modulation

Die Bits des FIC und MSC werden mit QPSK moduliert. Dabei werden $2K = 3072$ Bits ($p_{l,n}$ für $n \in [0; 2K)$) zu $K = 1536$ QPSK Symbolen ($q_{l,n}$ für $n \in [0; K)$) moduliert, wobei die erste Hälfte $n \in [0; K)$ der Bits die Realteile und die zweite Hälfte $n \in [K; 2K)$ die Imaginärteile bilden, sodass sich die modulierten Symbole zu

$$q_{l,n} = \frac{1}{\sqrt{2}} [(1 - 2p_{l,n}) + j(1 - 2p_{l,n+K})] \quad (2.6)$$

ergeben. Die möglichen Werte von $q_{l,n}$ sind in Abb. 2.6 im Konstellationsdiagramm schematisch dargestellt.

2.4.3. Frequenz-Interleaving

Die modulierten Symbole werden vor der OFDM Modulation durch eine pseudo-zufällige Folge untereinander vertauscht. An der Stelle von $z_{l,k}$ befindet sich nach der Vertauschung das neue Symbol $y_{l,k}$, das einem $z_{l,\xi}$ mit $\xi \in [0; K)$ entspricht. Bei geringer Geschwindigkeit des Empfängers führt ein Mehrwegeempfang zu langsamem, frequenzselektivem Fading. Daraus resultierende Bündelfehler auf einzelnen Unterträgern übersteigen die Einflussdauer des Zeitinterleavers. Durch das Vertauschen der Unterträger werden die Effekte des frequenzselektiven Fadings minimiert.

2.4.4. Differentielle Modulation

Die QPSK modulierten und frequenzvertauschten Unterträger werden in einem letzten Schritt vor der OFDM Modulation differentiell codiert. Die Phase jedes Sendesymbols ergibt sich aus der Summe von der Phase des Symbols zur Phase des Vorgängers.

$$z_{l,k} = z_{l-1,k} \cdot y_{l,k} = e^{j\varphi_z} \cdot e^{j\varphi_y} \quad (2.7)$$

mit $\varphi_z \in \{n \cdot \frac{\pi}{4} : l \text{ gerade}\} \cup \{\frac{\pi}{4} + n \cdot \frac{\pi}{2} : l \text{ ungerade}\}, n \in \mathbb{N}$

Die zu übertragende Information wandert also von der eigentlichen Phase des Symbols in die relative Änderung der Phase des Vorgängersymbols zum aktuellen Symbol. Der Vorgänger des ersten Datensymbols $z_{1,k}$ ist das Phasenreferenzsymbol $z_{0,k}$ aus Abschn. 2.4.1, welches selbst nicht differentiell moduliert wird. Die differentielle Modulation hat den großen Vorteil, dass der Empfänger den Übertragungskanal nicht schätzen muss. Eine konstante Phasendrehung im Kanal hebt sich beispielsweise durch die Differenzbildung im Empfänger wieder auf.

⁴Nach [HL09] kann durch die CAZAC Sequenz mittels einer Automatic Frequency Control (AFC) ein Frequenzoffset von $\pm 32 \text{ kHz}$ ($\cong \pm 32$ Unterträger) eindeutig erkannt werden.

2.4.5. Orthogonales Frequenzmultiplexverfahren (OFDM)


Das DAB System verwendet ein Frequenzmultiplexverfahren (FDM) für die Übertragung. Dabei werden die D-QPSK Symbole $z_{l,k}$ nicht seriell mit einer Symbolrate von 1,2 MBd übertragen, sondern die zu übertragenden Symbole werden auf $N=2048$ Unterträger verteilt. Die gesamte Bandbreite $B = 2,048$ MHz wird also unter den einzelnen Unterträgern k gleichmäßig aufgeteilt, sodass sich ein Unterträgerabstand von $\Delta f = \frac{B}{N} = 1$ kHz ergibt. Diese schmalbandigen und niederratigen Unterträger haben in der Summe die gleiche Symbolrate wie eine serielle Übertragung, sie bieten aber vorteilhafte Übertragungseigenschaften:

- Der Übertragungskanal kann pro Unterträger in den meisten Fällen als flach angenommen werden.
- Intersymbol Interferenz (ISI) wird wegen der deutlich größeren Symboldauer im Vergleich zur seriellen Übertragung stark reduziert.

Das resultierende Basisbandsignal $S(t)$ ergibt sich als Summe der einzelnen Unterträger, die mit dem Abstand Δf über dem Basisband verteilt sind.

$$s(t) = \frac{1}{K} \sum_{k=-K/2}^{K/2} z_{l,k} g_{k,l}(t - lT_S) e^{j2\pi k \Delta f t} \quad (2.8)$$

Die Symbole werden mit $g_{l,k}(t)$ pulsgeformt. Damit sich die Unterträger nicht gegenseitig beeinflussen und Inter-Träger-Interferenz (ICI) die Performanz des Systems beschneidet, werden die Unterträger orthogonal zueinander gewählt. Orthogonales FDM (OFDM) kann durch eine rechteckige Pulsformung erreicht werden.

$$g_{k,l}(t) = \text{Rect}(t/T_S) = \begin{cases} 1, & \text{für } 0 \leq t < T_S \\ 0, & \text{sonst} \end{cases} \quad (2.9)$$


$$G_{k,l}(f) = T_S \cdot \text{si}(\pi f T_S) e^{-j\pi f T_S}$$

Wie in Abb. 2.7 zu sehen ist, haben die si-Funktionen, die sich nach 2.9 aus den Rechteckfunktionen im Frequenzbereich ergeben, jeweils Nullstellen bei ganzzahligen Vielfachen von Δf (blaue Markierungen), also genau an den Maxima aller anderen Unterträgern (rote Markierung). Diese Orthogonalität der Unterträger vermeidet ICI, da das Gesamtsignal, welches sich aus der Überlagerung aller Unterträger ergibt, an Vielfachen von Δf jeweils genau den Werten des jeweiligen Unterträgers annimmt. Die Orthogonalität gilt nur unter der Annahme, dass die Unterträger an ihrer exakten Mittenfrequenz abgetastet werden.

Von den $N = 2048$ möglichen Unterträgern sind im DAB System nur $K = 1536$ belegt. Der zentrale Unterträger ($k = 0 \Leftrightarrow k \cdot \Delta f = 0$ kHz) wird nicht belegt um einen Gleichanteil im Basisbandsignal zu vermeiden. Die restlichen unbelegten Unterträger liegen am rechten und linken Rand des Spektrums und bilden damit ein Schutzintervall gegen grobe Frequenzverschiebungen.

Wie in [WE71] gezeigt wird, kann das OFDM-Basisbandsignal mit einer Invers Diskreten Fourier Transformation (IDFT) und einer anschließenden parallel-seriell Wandlung aus den

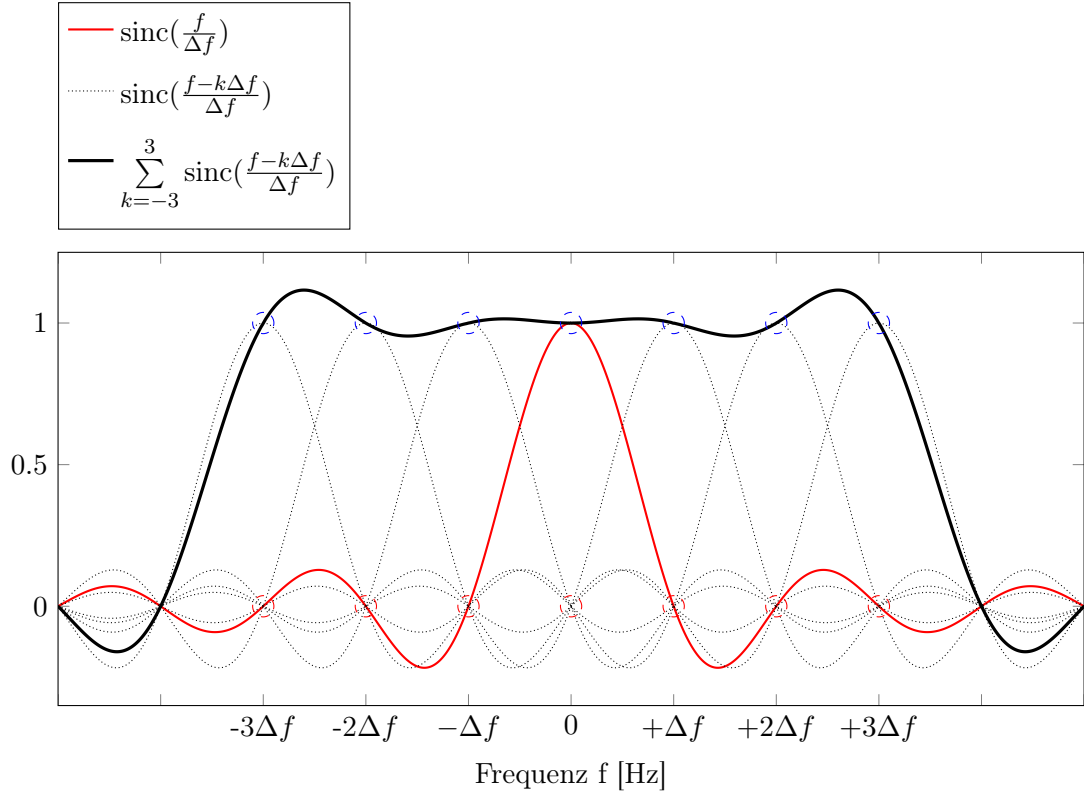


Abbildung 2.7.: Überlagerung frequenzverschobener, orthogonaler si-Funktionen

OFDM Symbolen erzeugt werden.

Um noch robuster gegen Mehrwegempfang zu sein, wird jedem OFDM Symbol ein Guard-Intervall der Länge $T_G = 246 \mu\text{s}$ vorangestellt. T_G entspricht damit der Zeit, in der Echos des Vorgängersymbols abklingen können, ohne das neue Symbol zu stören. Der Inhalt des Guard-Intervalls entspricht dem Ende des darauf folgenden Symbols, was als Cyclic Prefix (CP) bezeichnet wird. Das CP hat den Vorteil, dass durch die periodische Wiederholung des Signals keine Signalanteile wegen Mehrwegempfang verloren gehen [Jon11]. Diese Aussage sowie die komplette Vermeidung von ISI gilt nur unter der Annahme, dass T_G länger als die maximale Verzögerung des Signals ist.

Das gesamte Sendesignal ergibt sich letztendlich nach dem Hochmischen auf die Trägerfrequenz f_c zu dem Bandpasssignal in Gl. 2.10, das über eine Antenne abgestrahlt wird.

$$s_{BP}(t) = \text{Re} \left\{ e^{j2\pi f_c t} \sum_{m=-\infty}^{\infty} \sum_{l=0}^L \sum_{k=-\frac{K}{2}}^{\frac{K}{2}} z_{m,l,k} g_{k,l}(t - mT_F - T_{\text{NULL}} - (l-1)T_S) e^{j2\pi k \Delta f t} \right\} \quad (2.10)$$

3. Implementierung eines DAB/DAB+ Transceivers

3.1. GNU Radio

GNU Radio ist eine kostenlose und weitverbreitete Open-Source Software zur Programmierung und Ansteuerung von Software-Defined Radio (SDR). SDR bezeichnet dabei ein Funksystem, welches die Signalverarbeitung anstatt durch Hardware (integrierte Schaltkreise) auf Softwareebene durchführt. Software hat den Vorteil, dass sie variabel und schnell austauschbar ist. Dadurch ist es möglich mit derselben Hardware viele verschiedene Funksysteme zu realisieren.

GNU Radio verwendet das Konzept von funktionalen Blöcken, die verbunden werden und damit eine Signalverarbeitungskette, einen sog. Flowgraph, bilden. Die einzelnen Blöcke führen jeweils eine spezifische Signalverarbeitungsoperation möglichst recheneffizient aus, weshalb ein GNU Radio Block in den meisten Fällen eine C++ Klasse darstellt. Die Ablaufsteuerung, also die Instanziierung und Anordnung der Blöcke in einem Gesamtsystem, wird in der Skriptsprache Python gesteuert. Dazu werden die verwendeten C++ Klassen mit dem Programmierwerkzeug Simplified Wrapper and Interface Generator (SWIG) nach Python übersetzt.

Neben der Möglichkeit, die GNU Radio Blöcke selbst zu entwerfen, gibt es bereits eine umfangreiche Bibliothek an Signalverarbeitungsblöcken. Diese reichen von grundlegenden Rechenoperationen bis hin zu komplexen Implementierungen wie zum Beispiel ganzen Kommunikationssystemen. Die graphische Oberfläche GNU Radio Companion (GRC) ermöglicht es, einen Flowgraph zu erstellen, ohne Code schreiben zu müssen. Per Drag and Drop werden die GNU Radio Blöcke dazu angeordnet und mit Pfeilen verbunden. Der GRC ist ein nützliches Tool für schnelles Prototyping und die Visualisierung eines Flowgraphs.



Abbildung 3.1.: Logo von GNU Radio

Die in diesem Kapitel vorgestellte Realisierung eines DAB/DAB+ Transceivers wurde mit GNU Radio implementiert. Das dabei entstandene Out-Of-Tree Modul gr-dab enthält alle geschriebenen GNU Radio Blöcke und Flowgraphs. Es ist unter der GNU General Public License Version 3 (GPL3) lizenziert und frei zugänglich auf der Plattform GitHub [CEL17] veröffentlicht. Die dabei verwendete Hardware ist unabhängig von der Implementierung und

kann von kostengünstigen DVB-T Dongles bis hin zu leistungsfähigen Universal Software Radio Peripherals (USRPs) reichen.

In der Evaluation des Systems in Abschn. 5 wurden GNU Radio Flowgraphs zudem als Basis für Simulationen genutzt.

3.2. Vorgehen bei der Implementierung

Die Implementierung des DAB/DAB+ Transceivers in GNU Radio erfolgt nach dem Bottom-up Prinzip. Einzelne Algorithmen bzw. Funktionseinheiten werden jeweils in einem C++ Block realisiert. Jeder implementierte Block wird zuerst durch einen Quality Assurance (QA) Test auf seine korrekte Funktionalität überprüft, bevor er zum Einsatz kommen kann. Oftmals hat ein Block im Sender ein entsprechendes Gegenstück im Empfänger. Es ist sinnvoll diese Blöcke zusammen zu implementieren und gegeneinander in einem Schleifentest (Loopback) zu testen. Aus diesem Grund werden der Sender und der Empfänger synchron implementiert. Aus den einzelnen Blöcken wird anschließend eine Signalverarbeitungskette aufgebaut. Um den GNU Radio Flowgraph übersichtlich zu halten, werden verschieden Gruppen von C++ Blöcken zu sogenannten hierarchischen Blöcken zusammengefasst. Ein hierarchischer Block ist ein GNU Radio Block in Python, der mehrere C++ Blöcke intern verbindet und nach außen hin als einen Gesamtblock darstellt.

Um größere Funktionseinheiten zu testen, werden verifizierte Referenzdaten benötigt, die aus Mangel an Alternativen über eine Antenne von einer aktiven Radiostation in der Umgebung abgegriffen werden. Damit wird zuerst der DAB Empfänger gegen die aufgenommenen Referenzdaten getestet um anschließend den Sender gegen den nun verifizierten Empfänger zu testen.

Im Folgenden ist aus Gründen der Übersichtlichkeit die Implementierung des Senders und Empfängers getrennt erläutert.

3.3. Implementierung des Senders

Abb. 3.2 zeigt den funktionsfähigen Sender im GRC. Der Sender transportiert zwei Audio Kanäle im MSC, wobei der obere Kanal einen DAB+ Audio Stream (MPEG4 und Reed Solomon Encoder) und der untere Kanal einen DAB Audio Stream (MPEG 2 Encoder) transportiert. Jeder Block in der grafischen Ansicht entspricht einer eigenen C++ Klasse, wobei die Kanalcodierer des FIC und MSC, sowie der OFDM Modulator hierarchische Blöcke darstellen, also mehrere C++ Blöcke in sich zusammenfassen. Die Struktur der Blöcke als einzelne Funktionseinheiten ist von der Aufteilung dem DAB Standard nachempfunden. Bei der Implementierung wurde darauf geachtet, möglichst viel Funktionalität in eine Klasse zu integrieren ohne dabei an Flexibilität der einzelnen Blöcke einzusparen.

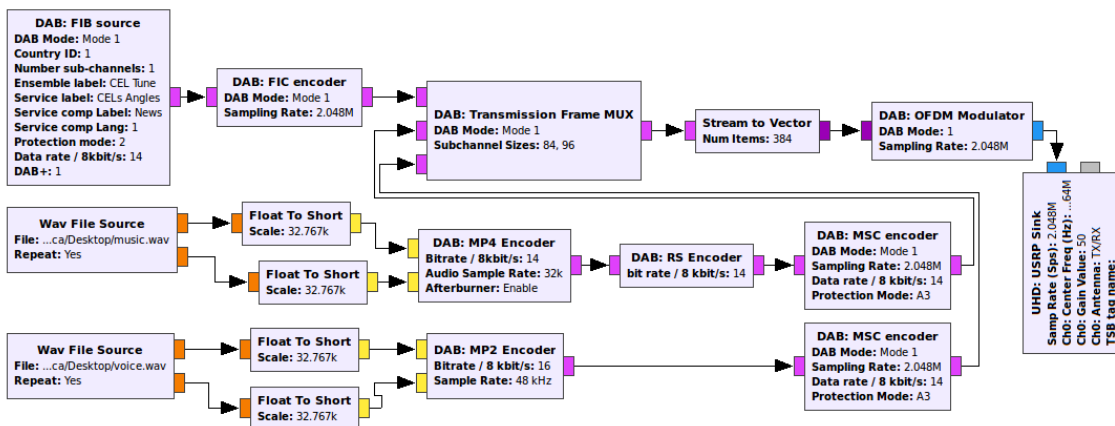


Abbildung 3.2.: DAB/DAB+ Sender im GRC

3.3.1. FIB Quelle

Die FIB Quelle produziert standardkonforme FIBs für den FIC. Die MCI und SI werden der Klasse dabei als Argumente übergeben. Die folgenden Informationen werden gesendet:

- Ensemble Informationen: Länder ID, Ensemble ID, CIF Zähler
- Radiosender-Organisation: Länder ID, Anzahl der Radiosender, Senderbeschreibung
- Radiosender-Komponenten: Kanal ID, Audio Typ, Primär-/Sekundärsender
- Kanal Organisation: Kanal ID, Start und Länge (in CUs), Protection Mode
- Service Information (SI): Namensbeschriftung für das Ensemble und jeden Radiosender und deren Sprache

Diese FIBs umfassen bei weitem nicht das ganze Spektrum der möglichen Metadaten. Sie sind aber ausreichend um dem Empfänger die nötigen MCI für die Decodierung und dem Nutzer die Informationen zur Identifikation der Programme zur Verfügung zu stellen. Weil die MCI eine weit höhere Wichtigkeit als die SI einnimmt, wird sie periodisch mit jedem CIF gesendet. Die Größe der gesamten MCI hängt von der Anzahl der verwendeten Services und

Service Komponenten ab. Bei einer Begrenzung auf maximal 7 Radiosender mit je einem Kanal kann die Information in den ersten zwei FIBs gespeichert werden. Im dritten FIB wird dann die SI geschrieben.

FIB 1			FIB 2	FIB 3
Ensemble Information	Radiosender Information	Komponente Information	Kanal Organisation	Service Information

Abbildung 3.3.: FIC Sendestruktur

Da eine Namensbeschriftung mit 16 Buchstaben ein FIB schon nahezu füllt, werden die einzelnen Labels nacheinander jeweils im dritten FIB verschickt. Dies ist nicht problematisch, da die Namen der Radiosender im Regelfall über die Zeit konstant bleiben und durch eine CIF Rate von $\frac{1}{24ms} > 41 \text{ CIFs/s}$ trotzdem jedes Label mehrmals pro Sekunde gesendet wird.

3.3.2. FIC Encoder

Der FIC Encoder ist in einem hierarchischen Block implementiert, der in Abb. 3.4 dargestellt ist. Er fasst die gesamte Kanalcodierungskette aus Kap. 2.2 zusammen. Während für die CRC Berechnung und die Faltungscodierung eine jeweilige Klasse implementiert wurde, besteht der Scrambler, wie in Gl. 2.2 beschrieben, lediglich aus einer XOR Verknüpfung des Bitstreams mit der PRBS Folge, was mit vorhanden GNU Radio Blöcken realisierbar ist.

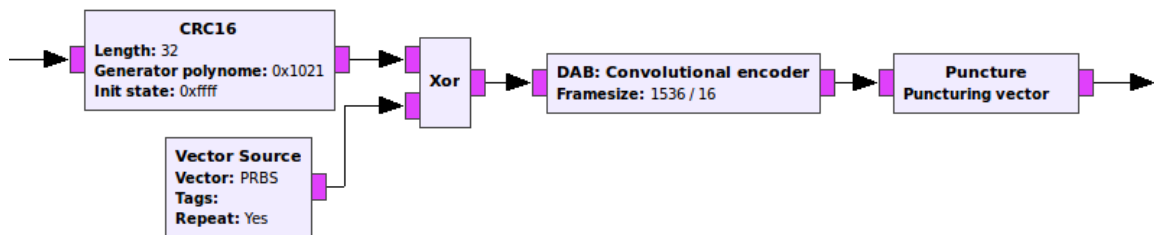


Abbildung 3.4.: Kanalcodierung im hierarchischen Block des FIC Encoders

3.3.3. Audio Quellen und Encoder

Die Eingangsdaten der Audioencoder sind 16 bit Puls-Code-Modulation (PCM) Samples. Um Audio für den Stream zur Verfügung zu stellen gibt es verschiedene Möglichkeiten. Entweder werden Audiosamples über eine Binärdatei als Rohdatenquelle genutzt. Eine andere Möglichkeit bietet der „WAV File Source“ Block von GNU Radio, der aus dem populären „WAVE“ Audioformat die Rohdaten extrahiert und somit auch eine PCM Quelle darstellt. Eine letzte Möglichkeit ist die direkte Generation von PCM Samples über die lokale Soundkarte. Alle drei Möglichkeiten resultieren letztendlich in einem Stream mit 16 bit Gleitkommawerten, dessen Samplingrate der Audiorate entspricht. Weil die implementierten Audiocodex auf Integerebene arbeiten, wird der Stream vor der Audioencodierung auf den neuen Wertebereich abgebildet (siehe Abb. 3.2).

MPEG 1/2 Audio Layer II Encoder (DAB)

Der MPEG 2 Encoder Block verwendet einen Patch der frei verfügbare MPEG2 Encoder Bibliothek *TooLAME*. Der Patch stammt vom *ODR mmbTools* Projekt [ODR17b] und ist speziell auf DAB Audio Encoding angepasst. Der Code wurde in einen GNU Radio Block integriert um ihn in einen Flowgraph einbauen zu können. Die Funktionalität des Encoder Blocks kann mit einem Loopback-Test verifiziert werden, indem der encodierte Stream mit einem Audio-Player erfolgreich und fehlerfrei decodiert und abgespielt wird.

MPEG 4 HE-AACv2 (DAB+)

Der MPEG 4 Encoder basiert auf einem Patch der *FDK-AAC Codec Bibliothek für Android der Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e. V.* [ODR17a]. Die Bibliothek unterstützt eine Reihe von AOTs (Audio Optimization Tools) welche die Audio-kompression nach unterschiedlichen Parametern optimieren. Es stehen die folgenden AOTs zur Verfügung:

- AAC-LC: Geringe Komplexität (1,5 bits/Sample) für minimales Codierungsdelay
- HE-AAC Dualband SBR: Hohe Effizienz (0,625 bits/Sample) durch Spektralbandreplikation.
- HE-AAC v2 PS: Zusätzlich mit parametrisiertem Stereo (0,5 bits/Sample)

Der DAB+ Standard enthält neben der neuen Audiokompression mit HE-AAC v2 eine zusätzliche Fehlererkennung und Fehlerkorrektur. Jeweils 5 Logische DAB Frames bilden ein sog. DAB+ Superframe, das den Inhalt eines MPEG 4 Frames aufnimmt. Der Header dieses Frames wird mit einem CRC Wort, dem sog. Firecode, versehen. Die Fehlerkorrekturfähigkeit wird durch Reed-Solomon Codes RS(120, 110, t=5) erreicht. Ein Superframe wird blockweise um jeweils 10 Codebytes pro 110 Datenbytes ergänzt. Der Reed-Solomon Code ist durch diese eingeführte Redundanz empfängerseitig in der Lage maximal 5 Fehler pro 120 Byte Block zu erkennen und zu korrigieren.

Um den MPEG 4 Encoder Block nicht zu sehr auf DAB+ zu spezialisieren, wurde der Firecode sowie der Reed-Solomon Encoder jeweils in einem separaten Block implementiert.

3.3.4. MSC Encoder

Der MSC Encoder entpricht von Aufbau und Struktur dem FIC Encoder Block aus Kapitel 3.3.2, wobei er keinen CRC Block besitzt und um eine zusätzliche Klasse für das Zeitinterleaving ergänzt ist.

Die durch das Zeit-Interleaving eingeführte Verzögerung um bis zu 15 Logische Frames (siehe Abschn. 2.3.4) stellt die Implementierung des Zeit-Interleaving Blocks vor Herausforderungen. Zum einen benötigt der Block ein Gedächtnis von 15 Frames. Ein dynamisches Speichern dieser Frames in Objekte der Klasse kostet viel Rechenleistung und ist daher unbedingt zu vermeiden um das System echtzeitfähig zu halten. Anstelle dessen werden die jeweils letzten 15 Frames nach deren eigentlichen Verarbeitung nicht aus dem Eingangsbuffer entfernt, sondern der Pointer des Eingangsbuffers wird um 15 Frames verschoben. Der Eingangsbuffer ist damit zu jeder Zeit um 15 Frames größer, es müssen aber keine zusätzlichen Kopieroperationen durchgeführt werden.

Zum anderen stellt der Beginn des Streams ein Problem für den Algorithmus dar. An dieser Stelle liegt dem Zeit-Interleaver noch keine Historie vor, sodass er keine Möglichkeit hat, mit dem Interleaving zu beginnen. Aus diesem Grund wird die Historie des Blocks initial mit 15 Frames gefüllt, die Nullen enthalten.

3.3.5. Multiplexer

Der Multiplexer Block hat prinzipiell die Aufgabe eines Parallel-Seriell Wandlers. Er besitzt einen Eingang für den FIC und eine variable Anzahl an Eingängen für den MSC und setzt alle Kanäle zu einem Sendesignale nach der Struktur aus Kapitel 2.4.1 zusammen. Die Position der jeweiligen Audiostreams im CIF werden dabei über die selbe MCI bestimmt, die in den FIBs des FIC desselben Sendeframes stehen. Die Synchronisationssymbole $z_{0,k}$ und $z_{1,k}$ sind an dieser Stelle noch nicht vorhanden und werden dem Sendeframe im OFDM Modulator hinzugefügt.

3.3.6. OFDM Modulator

Der OFDM Modulator Block ist ein hierarchischer Block mit vielen Unterklassen. Jede Klasse führt dabei einen der in 2.4 beschriebenen Schritte zur Modulation durch.

QPSK Mapping

Der QPSK Mapper bildet 1 Byte auf 4 komplexe QPSK Symbole ab. Da wie in 2.4.2 beschrieben zuerst die Realteile und anschließend die Imaginärteile eines kompletten OFDM Symbols übertragen werden, kann der QPSK Mapper nicht Byte für Byte arbeiten, also jedes Byte direkt auf 4 QPSK Symbole abbilden. Deshalb arbeitet der Mapper auf Vektorbasis der Länge $1536/4$ Bytes am Eingang, bzw. 1536 komplexen Symbolen am Ausgang. Dies umfasst genau die Länge eines OFDM Symbols.

Einfügen des Phasenreferenzsymbols

Das Phasenreferenzsymbol wird schon als ein um $\pi/4$ gedrehtes QPSK Symbol generiert. Daher wird es erst nach der QPSK Modulation der Datensymbole an das Sendeframe angehängt. Das Phasenreferenzsymbol ist bei jedem Sendeframe gleich und kann deshalb im Konstruktor einmalig generiert werden um bei Laufzeit nur noch in den Buffer kopiert zu werden.

Differentielle Modulation

Die Differentielle Modulation stellt eine komplexe Multiplikation jedes Symbols mit dessen Vorgänger dar (siehe 2.4.4). Auch hier wird auf Vektorbasis gearbeitet.

Frequenz-Interleaving

Vor der OFDM Modulation müssen die 1536 D-QPSK Symbole auf die IFFT Länge von 2048 gebracht werden. Dazu werden symmetrisch am Anfang und Ende des Vektors jeweils 256 Nullelemente eingefügt.

Das anschließende Vertauschen der OFDM Unterträger folgt der Regel

$$\begin{aligned} \Pi(i) &= (13 \Pi(i-1) + 511) \bmod 2048 \quad \text{mit } i \in [0, 1536) \\ \text{und } \Pi(0) &= 0 \end{aligned} \tag{3.1}$$

wobei $\Pi(i)$ die Permutation ist, die sich aus deren Vorgänger $\Pi(i-1)$ ergibt. Da diese Vertauschungsregel bei jedem Symbol die selbe ist, wird im Konstruktor des Frequenz-Interleaving Blocks einmalig eine Umsetzungstabelle für die $T_S = 2048$ Permutationen generiert. Dadurch benötigt der Block zur Laufzeit weniger Rechenzeit.

Inverse Schnelle Fourier-Transformation (IFFT)

Die OFDM Operation kann nach 2.4.5 als Inverse Diskrete Fourier Transformation (IDFT) durchgeführt werden. Wegen $2048 = 2^{11}$ kann die IDFT als eine deutlich schnellere IFFT realisiert werden. Eine IFFT Implementierung ist in dem GNU Radio Modul „gr-fft“ vorhanden [GR17a] und wird verwendet.

Einfügen des Cyclic Prefixes

Nach der IFFT liegt das Sendesignal nun im Zeitbereich vor. Das Einfügen des Cyclic Prefixes (CP) entspricht dem Kopieren vom Ende jedes Symbols an den Anfang. Weil das CP eine breite Anwendung in der Nachrichtentechnik hat, existiert in GNU Radio schon ein Block mit dieser Funktionalität, der nur noch mit der passenden Symbollänge $T_S f_s = 2048$ Samples und der Länge des Cyclic Prefixes $T_G f_s = 504$ Samples initialisiert wird. Die Samplingfrequenz ist in dieser Implementierung $f_s = 2,048 \text{ MSamples/s}$

Einfügen des Nullsymbols

Als letzter Schritt in der Sendekette erfolgt das Einfügen des Nullsymbols. Dabei werden $T_{\text{NULL}} f_s = 2656$ Samples mit dem Wert $0 + 0j$ zwischen zwei Sendeframes geschrieben.

Das Basisbandsignal ist nun bereit um hochgemischt und anschließend gesendet zu werden. Beide Schritte sind in dem GNU Radio Block *USRP Sink* vereint, der die Schnittstelle zwischen Software und Hardware bildet. Alternativ lässt sich das Basisbandsignal natürlich auch als I/Q-Daten in einer Binärdatei speichern.

3.4. Implementierung des Empfängers

Bei der Übertragung treten Störungen und Effekte auf, die das Empfangssignal vom Sendesignal abweichen lassen. Vor der Demodulation und Auswertung der Daten ist die erste Stufe des Empfängers deshalb eine Synchronisation. Ziel der Synchronisation ist das Empfangssignal so zu korrigieren, dass es dem Sendesignal wieder möglichst nahe kommt. Die anschließende Kanaldecodierung und Auswertung der Daten im FIC und MSC findet in den entsprechenden hierarchischen GNU Radio Blöcken statt. Abb. 3.5 zeigt den Aufbau eines DAB+ Empfängers im GRC.

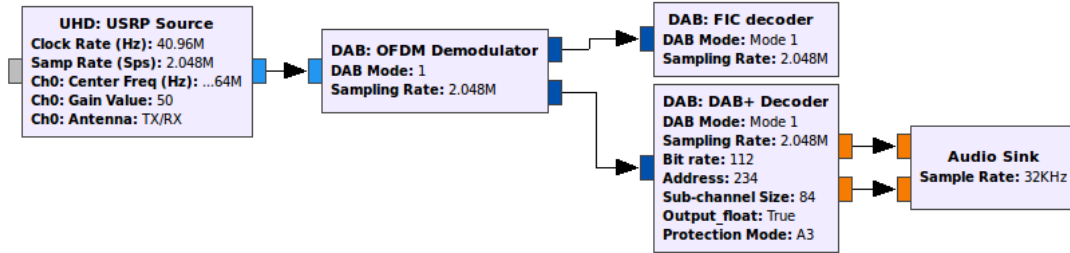


Abbildung 3.5.: DAB+ Empfänger im GRC

Das Empfangssignal $r(t)$ wird in folgender Beziehung zum Sendesignal $s(t)$ angenommen:

$$r(t) = h(t, \tau) * s(t - \tau_{\text{off}}) \cdot e^{j(2\pi f_{\text{off}}t + \varphi_{\text{off}})} + n(t) \quad (3.2)$$

Das Signal wird um einen Zeitoffset τ_{off} verzögert und hat wegen Doppler-Verschiebungen und Ungenauigkeiten der Oszillatoren einen Frequenzoffset f_{off} . Ein konstanter Phasenoffset φ_{off} , der durch Reflektionen der Welle entsteht, wirkt sich in diesem Fall nicht auf das demodulierte QPSK Signal aus, da sich wegen der differentiellen Modulation eine konstante Phase bei der Differenzbildung heraushebt. Es ist also keine Kanalschätzung nötig. Die Amplitude des Signals wird über der Zeit und der Frequenz unterschiedlich stark gedämpft, wobei ein frequenzselektiver Slow-Fading Kanal angenommen wird. Wegen der Schmalbandigkeit der einzelnen Unterträger bei OFDM kann der Kanal trotzdem in jedem Unterträger als flach angenommen werden. Die Dämpfung des Signals ist für die Symbolentscheidung bei PSK zudem nicht ausschlaggebend. Letztendlich wird das Signal durch Additives Weißes Gaußsches Rauschen (AWGN) überlagert.

Um eine möglichst optimale Synchronisation zu erreichen, werden in der folgenden Signalverarbeitungskette relevanten Effekte schrittweise gemessen und korrigiert.

3.4.1. Zeit Synchronisation

Eine geeignete Zeitsynchronisation muss sowohl eine grobe Synchronisation des OFDM Frames, als auch eine feine Synchronisation der einzelnen OFDM Symbole sicherstellen.

Der Beginn jedes Frames k wird durch das Nullsymbol $z_{0,k}$ markiert. Wegen $s(t) = 0$ für $t \in [0, T_{\text{NULL}}]$ stellt es eine zuverlässige Möglichkeit dar, den Anfang von Frames über eine Energiemessung zu detektieren. Die Energiemessung lässt sich nach Gl. 3.3 über eine Autokorrelation des empfangenen und abgetasteten Basisbandsignals $x[i]$ realisieren.

$$E[i] = \sum_{j=i}^{T_G f_s + i - 1} |x[j]|^2 \quad (3.3)$$

Nachdem der Anfang eines Frames detektiert wurde, muss im Folgenden der Beginn jedes OFDM Symbols $z_{l,k}$ ($l \in [1, 76]$) festgelegt werden. Diese feine Zeitsynchronisation erfordert keine sehr hohe Genauigkeit, da jedem Symbol ein CP der Dauer $T_G = 246 \mu\text{s}$ vorgeschoben ist, dessen Inhalt dem Ende des eigentlichen Symbols entspricht. Dadurch ergibt sich ein Zeitbereich von $T_D \in [0, T_G]$, in dem der Symbolanfang grundsätzlich korrekt gesetzt werden kann. Im realistischen Mehrwegempfang treten jedoch zeitverzögerte Echos auf, welche die Signalanteile über der Zeit verwischen und damit zu ISI führen.

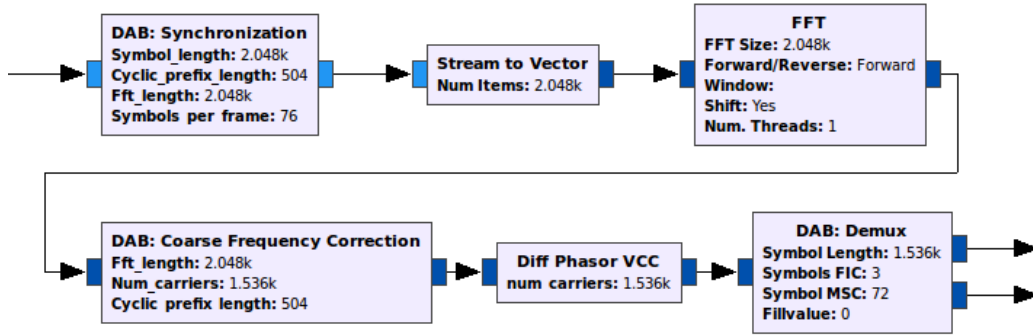


Abbildung 3.6.: Aufbau der kompletten Synchronisationskette im GRC

Um ISI zu minimieren ist der theoretisch optimale Abtastzeitpunkt genau am Ende des CPs zu wählen. Eine komplette ISI Unterdrückung ist wegen $T_G < \tau_{\max}$ mit einer maximalen Echo Verzögerung von $\tau_{\max} = 300 \mu\text{s}$ nicht möglich [HL09]. Jedoch wird die ISI zu einem Minimum reduziert, wenn ein maximal später Startpunkt gewählt wird, da die Interferenzzeit $\tau_{\max} - T_D$, bei $\tau_{\max} < T_G + T_S$, dadurch minimiert wird.

Um den exakten Anfang eines Symbols zu bestimmen, wird die zyklische Wiederholung des Symbolanteils im CP genutzt. Durch eine Korrelation des abgetasteten Empfangssignals $r[i]$ mit einer um T_S verzögerten Version desselben Signals $r[i + T_S f_s]$ über das Intervall $[i, i + T_G f_s]$ kann der Anfang des CP über einen Peak der Korrelation identifiziert werden.

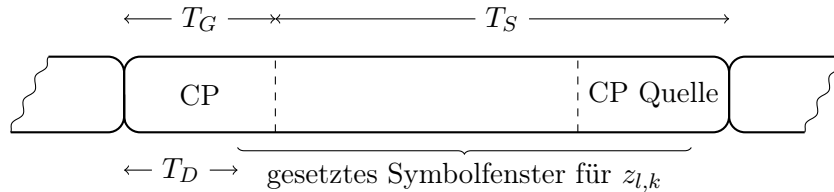


Abbildung 3.7.: OFDM Symbol und Cyclic Prefix

$$y[i] = \sum_{j=i}^{T_G f_s + i - 1} r[j] r^*[j + T_S], \quad y[i] \in \mathbb{C} \quad (3.4)$$

Unter Berücksichtigung von AWGN ist $r[i] = x[i] + n[i]$ mit $n[i] \sim \mathcal{N}(\mu, \sigma^2)$ resultiert für die Korrelation

$$y[i] = \sum_{j=i}^{T_G f_s + i - 1} (r[j] + n[j])(r[j + T_S] + n[j])^* \quad (3.5)$$

$$= \sum_{j=i}^{T_G f_s + i - 1} x[j] r^*[j + T_S] + \sum_{j=i}^{T_G f_s + i - 1} r[j] n^*[j + T_S] \quad (3.6)$$

$$+ \sum_{j=i}^{T_G f_s + i - 1} n[j] x^*[j + T_S] + \sum_{j=i}^{T_G f_s + i - 1} n[j] n^*[j + T_S] \quad (3.7)$$

$$= PT_G + 2(P\sigma^2) + \sigma^4 \approx PT_G + 2(P\sigma^2) \quad (3.8)$$

und damit ein SNR des Korrelationssignals von

$$\text{SNR} = \frac{P_s}{P_n} = \frac{PT_G f_s}{2P\sigma^2} = \frac{T_G f_s}{2\sigma^2} \quad (3.9)$$

Durch eine Mittelung über $T_G f_s = 504$ Samples ist das SNR somit ausreichend groß um eine entsprechend genaue Peakdetektion durchführen zu können.

Die Korrelation ist dabei auf die Energieanteile des Cyclic Prefixes und dessen Quelle nach T_S normiert, sodass das Ergebnis unabhängig von der Empfangsleistung bleibt, die durch Empfangsqualität und verwendeter Hardware stark variieren kann.

$$y_{\text{norm}}[i] = \frac{\sum_{j=i}^{T_G f_s + i - 1} r[j] r^*[j + T_S]}{\sqrt{\sum_{j=i}^{T_G f_s + i - 1} |r[j]|^2 \sum_{j=i}^{T_G f_s + i - 1} |r[j + T_S]|^2}} = \frac{\sum_{j=i}^{T_G f_s + i - 1} r[j] r^*[j + T_S]}{\sqrt{E[i] E[i + T_S]}} \quad (3.10)$$

In Abbildung 3.8 ist die normierte Korrelation aus Gl. 3.10 auf das Empfangssignal berechnet worden. Im Bereich von 1,8 bis 3,0 ms ist im Empfangssignal das Nullsymbol zu erkennen. Ein Peak der Korrelation befindet sich zum Beispiel genau bei 3 ms, was dem Ende des Nullsymbols bzw. dem Anfang des Phasenreferenzsymbols entspricht. Es ist zu erkennen, dass die Flanken der Korrelation linear ansteigen. Die Breite einer Flanke entspricht T_G , also gerade dem Entscheidungsbereich für T_D . Durch die Linearität der Flanke und der Normierung, kann der relative Abtastzeitpunkt innerhalb des CPs über einen Schwellwert eingestellt werden. Der tatsächliche Abtastzeitpunkt wird anschließend mit einer Verzögerung von T_G gesetzt. In der Implementierung von Abbildung 3.8 wurde ein Schwellwert von 0,85 eingestellt, der sich als guter Kompromiss zwischen ISI Unterdrückung und einem Sicherheitsabstand zu $T_D > T_G$ herausgestellt hat. Man beachte, dass bei $T_D f_s > T_G f_s$ Samples vom nachfolgenden Symbol im gesetzten Symbolfenster lägen, was zum Empfang von falscher Information führen würde. Dieser Fall entspricht einer fehlerhaften Zeitsynchronisation.

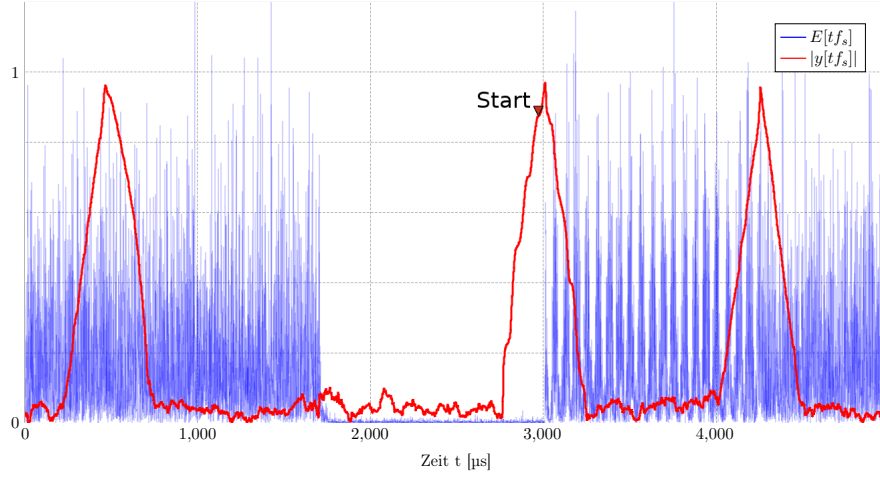


Abbildung 3.8.: Korrelation des Cyclic Prefix

Implementierung

Um eine mehrfache Berechnung von Gleichung 3.3 für die Energiemessung, sowie die normierte Korrelation (Gl. 3.10) zu vermeiden, wurde die feine und grobe Zeitsynchronisation in einer gemeinsamen Klasse implementiert.

Abbildung 3.9 zeigt den Programmablaufplan der Zeitsynchronisation. Er ist aufgeteilt in die beiden Zweige „Suche Frame Start“ und „Kontrolle“. Der Zweig „Suche Frame Start“ kommt bei einem der folgenden Fälle zum Einsatz:

- Die Synchronisation wird initial gestartet.
- Das Programm ist nicht mehr in Synchronisation.
- Das Ende eines Frames wurde erreicht.

Alle diese Situationen haben gemeinsam, dass im Folgenden nach dem Peak des Symbols $z_{1,k}$ (erstes Symbol nach dem Nullsymbol) gesucht wird. Dafür wird iterativ an jedem Sample des Zeitsignals $x[i]$ eine Korrelation $y[i]$ nach Gl. 3.10 durchgeführt. Es können Korrelationspeaks, analog zu Abb. 3.8, detektiert werden, die jeweils dem Start eines Symbols entsprechen. Durch die Energiemessung kann das Symbol $z_{1,k}$ von den restlichen Symbolen $z_{l,k}$, mit $l \in [2,76]$ unterschieden werden.

Die Komplexität der Korrelationsoperation kann drastisch reduziert werden, indem die Summe, die laut Gl. 3.4 in jedem Schritt $T_G f_s = 504$ Additionen durchführt, durch eine gleitende Summe ersetzt wird.

$$y[i + 1] = y[i] - r[i]r^*[i + T_S f_s] + r[i + T_G f_s]r^*[i + (T_G + T_S)f_s] \quad (3.11)$$

Dabei ist zu beachten, dass $y[i = 0]$ komplett berechnet werden muss. Die diskrete Darstellung von Gleitkommawerten im Rechner führt zu Rundungsfehlern. Aus diesem Grund wird die Summe alle $i = 100000$ Samples neu berechnet, um einen Drift von $y[i]$ zu vermeiden.

Mit dem Anfang von $z_{1,k}$ stehen auch die Anfänge aller anderen Symbole des Frames fest, da sie mit der festen und bekannten Verzögerung von $T_G + T_S$ direkt aufeinander folgen.

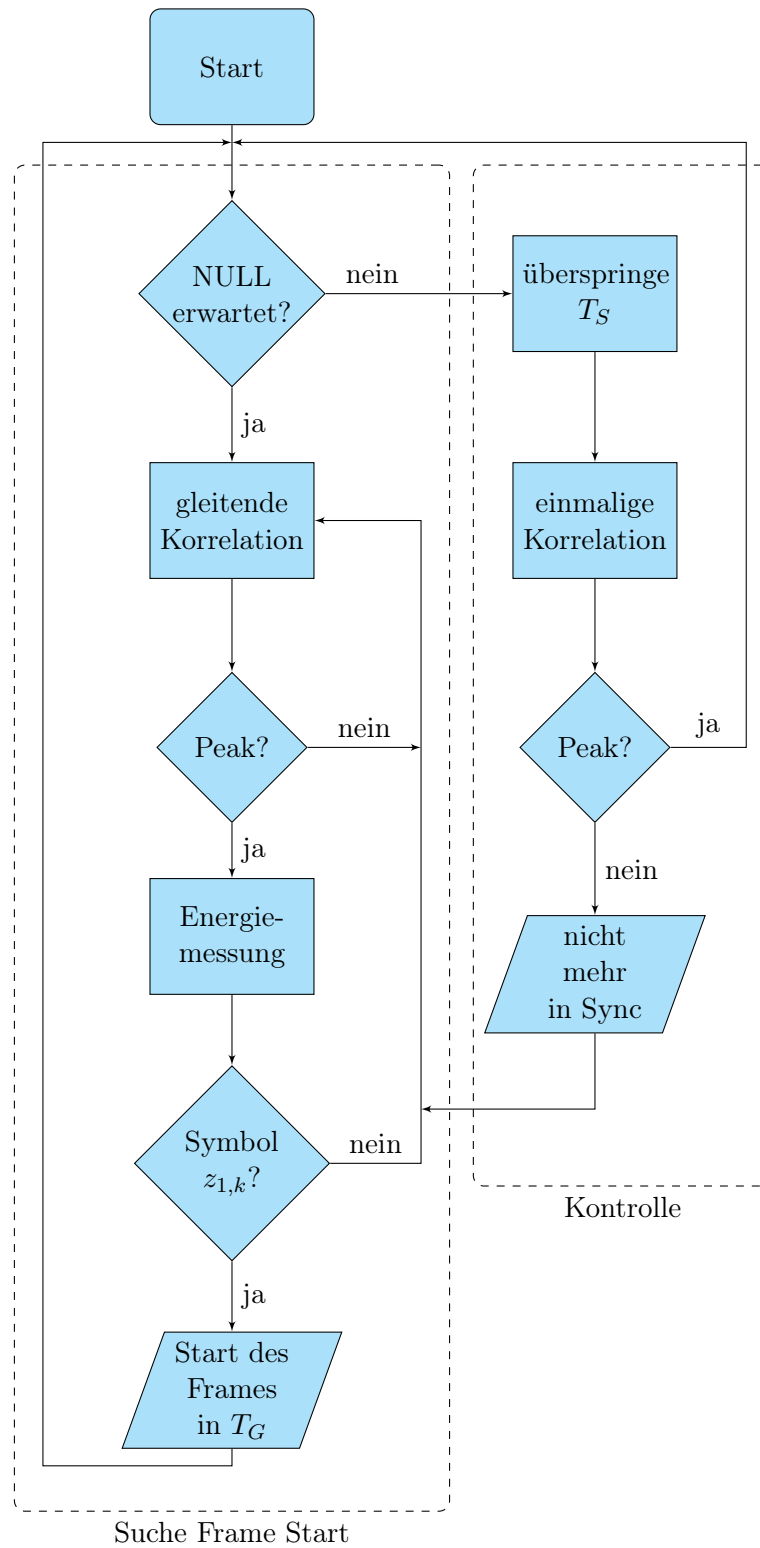


Abbildung 3.9.: Programmablaufplan der Zeitsynchronisation

Der Zweig „Kontrolle“ springt daher nur noch vom Start eines Symbols $x[i]$ zum nächsten $x[i + (T_G + T_S)f_s]$ und berechnet an dieser Stelle einmalig die Korrelation $y[i + (T_G + T_S)f_s]$. Liegt $y[i + (T_G + T_S)f_s]$ über einem Schwellwert, wird das Sample als Anfang des nächsten Symbols bestätigt und die Kontrollschleife iteriert zum nächsten Symbol. Falls an einem erwarteten Symbolanfang der Schwellwert der Korrelation unterschritten wird, wechselt das Programm in den Zweig „Suche Frame Start“.

Weil die Symboldauer genau bekannt ist, mag eine Kontrolle von jedem einzelnen Symbol zunächst überflüssig erscheinen. Der Aufwand ist jedoch gerechtfertigt, um Störeffekte wie einen Clockdrift rechtzeitig erkennen und korrigieren zu können. Ein Clockdrift kann zu einem Verlust der Synchronisation und damit zu Übertragungsfehlern führen. Ein beispielhafter Clockdrift von 50 ppm kann im rauschfreien Fall auf dem äußersten belegten Unterträger zu einem maximalen Phasenfehler von

$$\begin{aligned}\Delta\varphi_{\max} &= 2\pi f_{\max}\Delta t = 2\pi \frac{B}{2} \text{Clockdrift} (T_G + T_S) \\ &= 2\pi \frac{1536 \text{ Hz}}{2} 50 \text{ ppm } 1246 \mu\text{s} \\ &= 0,3 \text{ rad} = 17,2^\circ\end{aligned}\tag{3.12}$$

führen. Nach Rechnung 3.12 führt der Phasenfehler von $17,2^\circ$ zu keiner direkten Fehlentscheidung in der QPSK-Demodulation. Der Phasenooffset erhöht die Bitfehlerrate im ver-rauschten Fall jedoch erheblich und muss daher korrigiert werden. Damit ist der Zweig „Kontrolle“ gerechtfertigt.

3.4.2. Frequenz-Synchronisation

Die Messung und Korrektur eines Frequenzoffsets f_{off} ist der nächste Schritt der Synchronisationskette. Sie spielt in OFDM Systemen eine besonders wichtige Rolle, da ein Frequenzoffset die Orthogonalität zwischen den Unterträgern zerstört und zu ICI führt.

Feine Frequenz-Schätzung

Für die Messung des Frequenzoffsets kann wieder auf die Korrelation aus Gl. 3.4 zurückgegriffen werden. Ein Frequenzoffset lässt sich hier als konstante Änderung der Phase über der Zeit messen, da die Phase von CP und dessen Wiederholung nach T_S gleich sind, wenn kein Frequenzoffset vorliegt.

$$f_{\text{off}}[i] = \frac{\arg(y[i])}{T_S}\tag{3.13}$$

Ein zusätzliches AWGN ändert die Phase jedes einzelnen Samples. Wegen der Mittelwertfreiheit von AWGN kann die Varianz des gemessenen Frequenzoffsets durch eine Mittelung reduziert werden. Durch das Korrelationsintervall von $T_G f_s = 504$ Samples wird eine solche Mittelung durchgeführt, was die relativ kleine Varianz der Frequenzoffsetmessung in Abb. 3.10 bestätigt.

Durch die aktive Mittelung über N Symbole wird eine weitere Senkung der Varianz um Faktor N erreicht. Pro Symbol wird genau eine Korrelation berechnet, also ergibt sich ein Mittelungsfenster der Dauer $N(T_G + T_S)$. Ein zu langes Mittelungsfenster führt zu einer

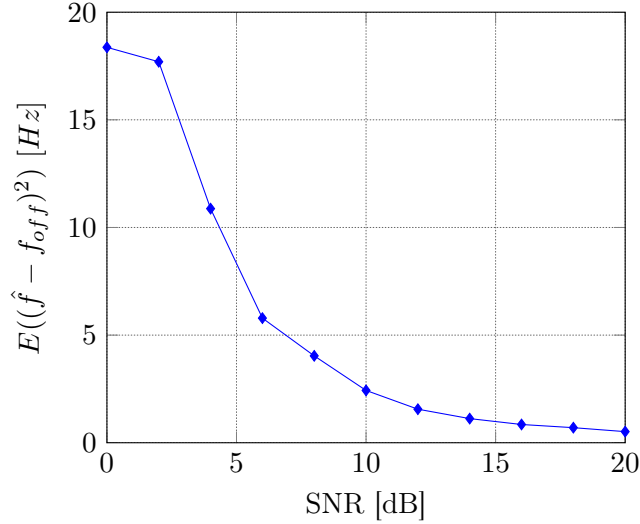


Abbildung 3.10.: Varianz der feinen Frequenzoffsetmessung

Trägheit der Frequenzmessung und damit auch zu einer Zeitverzögerung in der Frequenzkorrektur, was bei schnellen Frequenzänderungen zu Synchronisationsverlusten führen kann. Vor allem bei mobilen DAB Empfängern wird dieser Fall aufgrund der Dopplerverschiebung relevant. Deshalb wird für die obere Grenze der Mittelung die Bedingung gestellt, dass bei einer maximalen, konstanten Beschleunigung a der durch die Mittelung verursachte Messfehler f_M unter der 3σ Grenze der Messabweichung liegt.

$$f_M \stackrel{!}{<} \frac{3\sigma(\text{SNR})}{\sqrt{N}} \quad (3.14)$$

Mit $a = 50 \text{ m/s}^2$ und einer Trägerfrequenz von $f_T = 200 \text{ MHz}$ ist

$$\frac{df}{dt} = f_T \frac{a}{c} = 33,3 \text{ Hz/s} \quad (3.15)$$

und damit

$$\begin{aligned} f_M &= \frac{1}{N} \sum_{i=0}^{N-1} i \frac{df}{dt} (T_G + T_S) \\ &= \frac{1}{N} \frac{df}{dt} (T_G + T_S) \left(\frac{N^2 + N}{2} - N \right) \stackrel{N_{\text{gro\ss}}}{\approx} \frac{df}{dt} (T_G + T_S) N \end{aligned} \quad (3.16)$$

Aus 3.14 und 3.16 ergibt sich eine rauschabhängige Obergrenze für das Mittelungsintervall.

$$N \stackrel{!}{<} \left(\frac{df}{dt} \frac{T_G + T_S}{3\sigma(\text{SNR})} \right)^{-2/3} \stackrel{\text{SNR}=10 \text{ dB}}{\approx} 116 \quad (3.17)$$

Grobe Frequenz-Schätzung

Wegen $\arg(y[i]) \in (-\pi, \pi]$ rad ist der gemessene Frequenzoffset aus Gl. 3.13 nur in $f_{off} \in (-500, 500]$ Hz eindeutig. Dieser Eindeigkeitsbereich entspricht genau der Breite eines

OFDM Unterträgers von 1 kHz. Nach der feinen Frequenzkorrektur liegen also die Unterträger wieder auf dem Frequenzraster und es tritt durch die Orthogonalität keine ICI auf. Jedoch kann das Signal um das ganze Vielfache des Unterträgerabstandes verschoben sein. Das hätte zur Folge, dass Symbole im Empfänger durch eine falsche Zuweisung der FFT Bins fehlinterpretiert würden. Die Information des kompletten Frames wäre in diesem Fall verloren.

Da sowohl die Erkennung, als auch die Korrektur einer Unterträgerverschiebung im Frequenzbereich wesentlich einfacher ist, wird die grobe Frequenz-Korrektur nach der FFT Operation (Abschn. 3.4.3) durchgeführt. Von den 2048 Unterträgern, welche aus der FFT resultieren, sind jeweils die ersten $K = \frac{1536}{2}$ rechts und links des zentralen DC-Trägers belegt. Die restlichen unbelegten Träger enthalten am Empfänger bei vernachlässigter ICI lediglich AWGN. Durch eine Korrelation des FFT Vektors mit dem Vektor der bekannten Trägerbelegung $[1,1,\dots,1,0,1,\dots,1,1]$ der Länge $1536+1$ kann der Trägeroffset n und der erste belegte Unterträger $X[n]$ ermittelt werden.

$$n = \underset{i}{\operatorname{argmax}} \sum_{j=i}^{i+K-1} |X[j]|^2 \quad \text{mit } i \in [0, L_{FFT} - K] \quad (3.18)$$

3.4.3. FFT

Die modulierten Symbole werden durch das OFDM System parallel auf den $K = 1536$ Unterträgern moduliert und sind innerhalb eines Sendesymbols T_s durch ihre Phase und Frequenz vollständig beschrieben [Jon11]. Um aus dem empfangenen Zeitsignal nach der Zeit- und Frequenzsynchronisation nun die D-QPSK Symbole zu erhalten, wechselt man mittels einer Diskrete Fourier-Transformation (DFT) in den Frequenzbereich, was in Gl. 2.8 gezeigt wurde. Wegen $L_{DFT} = \frac{2,048 \text{ M samples/s}}{T_s} = 2048 = 2^{11}$ kann die DFT durch eine Schnelle Fourier-Transformation (FFT) der Länge 2048 effizient implementiert werden [LJK15]. Der FFT Algorithmus ist in GNU Radio [GR17a] bereits implementiert und wird verwendet.

3.4.4. Frequenz-Interleaving

Die im Sender vorgenommene Frequenzspreizung wird rückgängig gemacht, indem die Unterträger wieder in ihre ursprüngliche Reihenfolge gebracht werden. Diese Operation entspricht, genau wie im Sender, einem Vertauschen der Unterträger nach einer festgelegten Vertauschungsregel. Deshalb kann hier die gleiche Klasse wie im Sender genutzt werden, die mit dem invertierten Vertauschungsvektor initialisiert wird.

3.4.5. Demodulation

Die differentielle Demodulation erfolgt durch eine komplexe Multiplikation des D-QPSK Symbols mit dessen komplex konjugierten Vorgängersymbols.

An dieser Stelle würde entsprechend der Umkehrung des Senders nun die QPSK Demodulation erfolgen, indem eine Bitentscheidung der komplexen Eingangswerte zum Beispiel nach dem Maximum Likelihood (ML)-Kriterium erfolgt. In dieser Implementierung wird aber an dieser Stelle noch keine sog. Harddecision durchgeführt. Stattdessen werden die vertauschten QPSK Symbole in ihren Real- und Imaginärteil zerlegt und als Gleitkommawerte, sog. Softbits, weitertransportiert. Eine Bitentscheidung wird in der Faltungsdecodierung über den Viterbialgorithmus mit euklidischem Abstand getroffen.

3.4.6. Kanaldecodierung

Die Kanaldecodierung durchläuft die Encodierungskette in umgekehrter Reihenfolge und stellt so schrittweise den gesendeten Bitstream wieder her.

Zeitinterleaving

Das Zeitinterleaving hat im Sender die Bits nach festen Regeln mit einer Verzögerung versehen. Der Bitstream kann also wiederhergestellt werden, indem dieselbe Verzögerung angewendet wird, um mit dem Schreiben auf das jeweilige Bit zu warten. Damit ein komplettes Symbol geschrieben werden kann, benötigt der Interleaverblock neben dem aktuellen Symbol auch alle 15 nachfolgenden Symbole. Der Block benötigt also ein Gedächtnis von 15 Symbolen. Die in 2.3.4 angesprochene Verzögerung äußert sich in diesem Fall darin, dass die ersten 15 produzierten Symbole des Blocks kein ausreichend großes Gedächtnis haben und die entstandenen Symbole daher fehlerbehaftet sind. Wegen der Fehlerkorrekturfähigkeit des im folgenden Abschnitts beschriebenen Faltungsdecoders ist es aber praktisch sogar möglich, die tatsächliche Verzögerung um einige Symbole zu reduzieren, da die letzten verzögerten Bits vom Decoder schon a priori bestimmt werden können.

Faltungsdecodierung und Viterbi-Algorithmus

Der Faltungsdecoder nutzt die im Encoder eingefügte Redundanz, um aus der empfangenen Bitfolge die wahrscheinlichste Sendefolge zu bestimmen. Das Maximum A-Posteriori (MAP) Kriterium ist wegen der Gleichverteilung der Sendesymbole gleich dem ML Kriterium. Um die Komplexität des Entscheiders von exponentieller Ordnung auf lineare Ordnung zu reduzieren, wird der Viterbi-Algorithmus zur Entscheidungsfindung herangezogen [Jon11]. Eine effiziente Implementierung eines Softbit QPSK Viterbi Decodierers ist im GNU Radio Modul *gr-trellis* vorhanden [GR17c].

Scrambling

Das Scrambling des Signals kann rückgängig gemacht werden, indem die selbe Operation wie im Sender angewendet wird. Die identischen Bits $y[i]$ der PRBS löschen sich zu jedem Sample $i \in \mathbb{N}$ durch die Exklusiv-Oder Verknüpfung gegenseitig aus, sodass wieder das Sendebit $x[i]$ entsteht.

$$x[i] \oplus y[i] \oplus y[i] = x[i], \quad x[i], y[i] \in \{0,1\} \quad (3.19)$$

Die Bits haben nun die Kanaldecodierungskette durchlaufen und entsprechen im besten Fall den FIBs im FIC bzw. den komprimierten Audiostreams im MSC.

3.4.7. FIC Senke

Bevor die Informationen aus den FIBs verarbeitet werden, wird eine Zyklische Redundanzprüfung (CRC) durchgeführt. Dazu wird aus den Datenbits wie in 2.2.1 das CRC Wort berechnet und dieses mit dem gesendeten CRC Wort verglichen. Stimmen die beiden Wörter überein, enthält der FIB mit hoher Wahrscheinlichkeit keine Bitfehler. Korrekt übertragene FIBs können dann im Folgenden gelesen und ausgewertet werden.

3.4.8. Audio Decoder

Die Audiodecoder haben die Aufgabe aus dem Bitstream wieder ein PCM Stream zu generieren, der dem ursprünglichen Audiostream vor der Kompression möglichst nahe¹ kommt.

DAB

Für die Implementierung des MPEG-1/2 Audio Layer II Decoders für DAB wird die Bibliothek *kjmp2* [JK17b] genutzt. Die Decoder werden jeweils in einen GNU Radio Block implementiert.

DAB+

Der Decoder für DAB+ Audio Frames wird in 3 separate Klassen implementiert, die neben der eigentlichen MPEG 4 Decodierung zusätzlich die für den DAB+ Standard eingeführte Fehlerkorrektur realisieren. Abb. 3.11 zeigt den Signalfluss im DAB+ Audio Decoder.

Die erste Klasse beinhaltet den Firecode-Checker, der einen CRC durchführt um Bitfehler im Header des DAB+ Superframes zu identifizieren. In dieser Implementierung hat der Firecode-Checker zusätzlich die Funktion, den Stream auf die DAB+ Superframes zu synchronisieren. Die Synchronisation ist notwendig, weil ein DAB+ Superframe aus 5 Logischen Frames besteht, welche nacheinander über den physikalische DAB/DAB+ Kanal übertragen werden. Vor der MPEG4 Decodierung müssen daher 5 Logische Frames zu einem Superframe gepackt werden, was bei einem fortlaufendem Stream und willkürlichem Betrachtungsfenster nicht eindeutig definiert ist.

Die Synchronisation wird in dieser Implementierung ohne technischen Mehraufwand realisiert, indem bei nicht-erfolgreichem Firecode-Check anstelle eines ganzen Superframes nur ein Logisches Frame verworfen wird. Im nicht-synchronisierten Fall geht der Firecode-Checker somit alle Logischen Frames nacheinander durch, bis er schließlich, im bitfehlerfreien Fall, nach maximal 4 verworfenen Logischen Frames den Anfang des nächsten Superframes durch einen erfolgreichen Firecode-Check identifiziert hat.

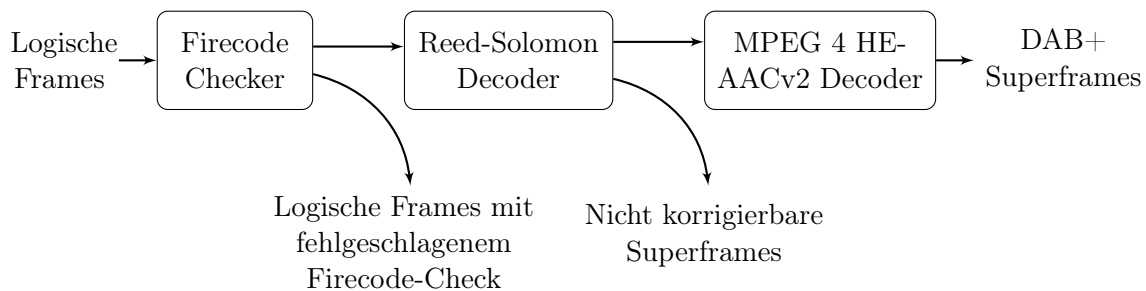


Abbildung 3.11.: DAB+ Audio Decoder

Der nachgeschaltete Reed-Solomon Decoder wurde auf Basis der Bibliothek für Vorwärtsfehlerkorrektur *libfec* implementiert [JK17b]. Ein Superframe wird über die eingefügten

¹Im Sinne einer subjektiven Wahrnehmung des menschlichen Gehörs.

Codewörter auf Blockbasis korrigiert. Nicht korrigierbare Blöcke werden verworfen. Alle Superframes, die den Firecode-Checker sowie den Reed-Solomon Decoder durchlaufen haben, sind mit hoher Sicherheit korrekt und können schließlich vom Audiodecoder verarbeitet werden. Für den eigentlichen MPEG 4 HE-AACv2 wurde eine Minimal-Implementation von [ETS17a] des Open-Source Projekts *qt-dab* genutzt [JK17a].

4. Implementierung einer grafischen Transceiver Applikation

Ein als GNU Radio Flowgraph implementierter DAB/DAB+ Transceiver erfüllt funktions-technisch alle notwendigen Anforderungen. Die Benutzerfreundlichkeit ist allerdings sehr eingeschränkt. Beispielsweise muss die Speicheradresse eines Kanals manuell aus der Konsole abgelesen werden, um sie im MSC Decoder anzugeben. Um die Benutzerfreundlichkeit zu erhöhen und den DAB/DAB+ Transceiver damit zu einer vollwertigen und ansprechenden Applikation werden zu lassen, wird eine Python Applikation mit grafischem Front-End implementiert, die den Namen *DABstep* trägt.

4.1. Klassenstruktur der Applikation

Abb. 4.1 zeigt das Klassendiagramm der Applikation. Python Klassen sind durch abgerundete Ecken von den C++ Klassen unterscheidbar.

4.1.1. Python main

Die Python Klasse *DABstep* stellt das Zentrum der Applikation aus Nutzersicht dar. Sie enthält alle Methoden, die den Nutzer mit der App interagieren lassen. Dazu gehören zum einen sog. Setter-Methoden, die den Transceiver bzw. dessen C++ Blöcke steuern, also beispielsweise den DAB-Empfänger starten oder die Lautstärke regeln. Zum anderen existieren sog. Getter-Methoden, die Informationen aus den verschiedenen Blöcken des Flowgraphs extrahieren und zur Verfügung stellen. Ein wichtiges Beispiel hierfür sind die in der Applikation angezeigten Service Informationen, wie zum Beispiel eine Radiosenderliste. Zusätzlich existieren viele kleine Methoden, die für eine bessere Bedienbarkeit sorgen.

4.1.2. Grafisches Frontend

Die Informationen zum grafischen Design der Applikation liegen in einer separaten Klasse, dem sog. Frontend. Es beinhaltet alle Tabs, Widgets, Beschriftungen, etc. und deren Anordnung im Anzeigefenster der Applikation. Für die Implementierung der grafischen Benutzeroberfläche (GUI) wird die Python Bibliothek PyQt4¹ verwendet. Ein Großteil des Inhaltes dieser Klasse wurde mit dem Programm QtDesigner4 erzeugt.

4.1.3. Umsetzungstabellen

Eine dritte Python Klasse enthält alle Umsetzungstabellen (engl. lookup tables). Diese Umsetzungstabellen übersetzen die in Bits codierten Service Informationen wie zum Beispiel

¹Die aktuelle Nachfolgeversion PyQt5 verwendet Python3, was in GNU Radio noch nicht komplett unterstützt ist.

die Sprache oder das Land des DAB Ensembles in für Menschen lesbare Zeichenketten. Die Umsetzungstabellen sind in einem separaten Standarddokument der ETSI [ETS17b] definiert.

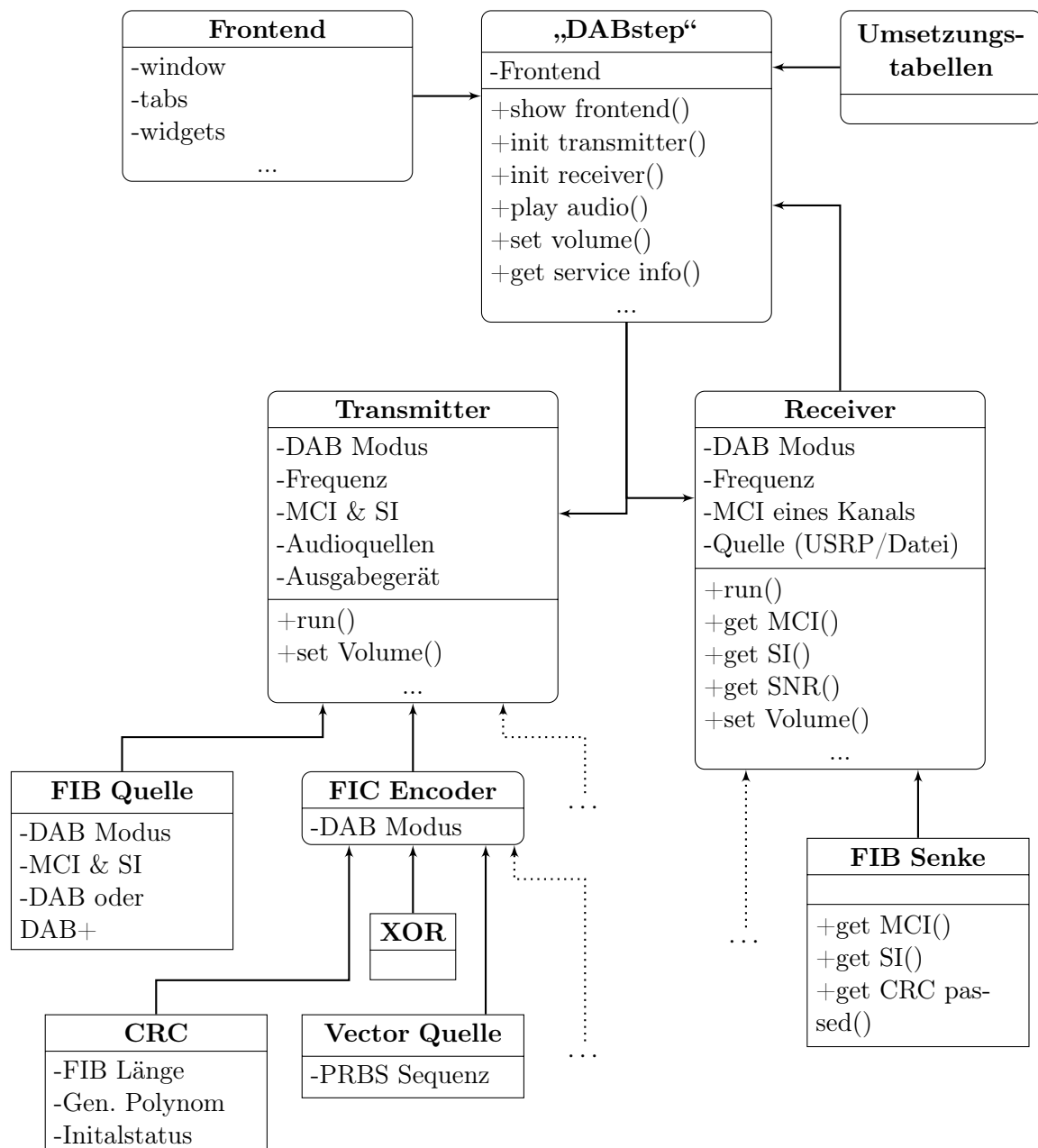


Abbildung 4.1.: Klassendiagramm der DAB/DAB+ Transceiver Applikation

4.2. GNU Radio Flowgraphs

Die funktionellen Kerne der Applikation sind die beiden GNU Radio Flowgraphs für den Transmitter und den Receiver. Sie enthalten die Definition, Instanziierung und Verbindung der einzelnen C++ Funktionalblöcke und bilden einen vollwertigen Transmitter bzw. Receiver. Im groben Aufbau entsprechen sie den bereits in vorherigen Kapiteln vorgestellten GNU Radio Flowgraphs aus Abb. 3.5 bzw. Abb. 3.2. Abhängig von den Argumenten der Klasse werden die Flowgraphs aber individuell initialisiert. Zum Beispiel kann, je nach Konfiguration des Empfängers, eine Binärdatei oder Hardware als IQ-Datenquelle dienen. Im Empfänger können ganze Zweige hinzukommen, wenn mehrere Audiostreams parallel gesendet werden.

Die Flowgraphs laufen in einem separaten Thread. Dies verhindert, dass die komplette Applikation eingefroren wird, wenn der Flowgraph zum Beispiel überlastet ist.

GNU Radio Blöcke

Die grundlegendste Klasse bilden die C++ Funktionalblöcke. Dazu gehören die in Abschn. 3.3 und 3.4 beschriebenen Klassen des in diesem Projekt implementierten Moduls gr-dab, sowie einige grundlegende Blöcke aus anderen GNU Radio Modulen. Manche Gruppen von C++ Blöcken sind aus Gründen der Übersichtlichkeit zu hierarchischen Python-Blöcken zusammengefasst.

Durch die genau definierte Funktion eines jeden Blocks und der hohen Flexibilität der GNU Radio Flowgraphs ist es möglich, nahezu die gesamte Datenverarbeitung und Entscheidungslogik innerhalb der Flowgraphs zu implementieren. Das hat den Vorteil, dass nur ein Minimum an Logik im Benutzerfrontend vorhanden sein muss.

4.3. Datenfluss

Ein Großteil des Datenflusses geschieht zwischen den C++ Blöcken, die große Mengen an komplexen IQ Samples oder Bits mit hohen Raten verarbeiten müssen und diese über Ringbuffer an den jeweils nächsten Block übergeben.

Zusätzlich müssen aber auch Daten, wie zum Beispiel die MCI und SI, von den Datenverarbeitungsblöcken zur Benutzeroberfläche transportiert werden, um sie dort in einer strukturierten und für den Menschen gut lesbaren Form darzustellen. Bei diesem Datenstrom fallen geringe Datenmengen an und der Fokus liegt auf einer guten Strukturierung und Charakterisierung. Eine geeignete Datenstruktur muss daher folgende Anforderungen erfüllen:

- Unterstützung **verschiedener Datentypen** (Zeichenketten für Sendernamen, Integer für Adressierungen, Boolean für Bitschalter)
- **Flexible Struktur** in:
 - Anzahl der Attribute (das Kanalinfo FIB enthält beispielsweise 5 Attribute, während der Sendernamen FIB nur 2 Attribute beinhaltet)
 - Anzahl der Elemente (a priori ist nicht bekannt, wie viele Radiosender im empfangenen Ensemble enthalten sind, bzw. wie viel zusätzliche SI ausgestrahlt wird)

- **Gute Lesbarkeit** für Menschen

Das Datenformat JavaScript Object Notation (JSON) erfüllt diese Eigenschaften hinreichend und wird in dieser Arbeit für den Metadatenfluss zum grafischen Frontend verwendet.

4.3.1. JSON

Die grundlegenden Datenstrukturen des JSON Formats sind:

$$\begin{aligned} object &= \left\{ string : value, string : value, \dots \right\} \\ array &= \left[value, value, \dots \right] \end{aligned} \quad (4.1)$$

mit $value \in \{string, number, object, array, true, false, null\}$

Eine für die beschriebenen Anforderungen vorteilhafte Kombination von JSON Strukturen ist in Gl. 4.2 beispielhaft für ein FIB mit Kanalinformationen dargestellt.

$$\begin{aligned} FIB = \left[\right. & \left\{ "ID" : 4, "address" : 234, "size" : 84, "protection" : 2 \right\} \\ & \left\{ "ID" : 2, "address" : 54, "size" : 84, "protection" : 3 \right\} \\ & \left. \left\{ "ID" : 3, "address" : 138, "size" : 96, "protection" : 1 \right\} \right] \end{aligned} \quad (4.2)$$

Das Array liefert die flexible Anzahl an Elementen und das Objekt bietet eine variable Anzahl an Attributen. Die Daten sind gut lesbar und werden genau in dieser Form als Zeichenkette in den jeweiligen Blöcken, ein Großteil davon in der FIC Senke, generiert. Diese können dann von der Receiver Klasse über eine Getter Funktion abgefragt und an die Klasse *DABstep* weitergeleitet werden wo sie an entsprechender Stelle angezeigt werden. Das Lesen der Daten in *DABstep* gestaltet sich besonders intuitiv, da ein JSON Array als Python Array und ein JSON Objekt als Python Dictionary interpretiert werden kann. Die einzige Datenverarbeitung, die an dieser Stelle vom Benutzerfrontend durchgeführt werden muss, ist das Lesen, Sortieren und Anzeigen der JSON Objekte.

4.4. Grafischer Aufbau

Neben der Funktionalität steht bei der grafischen Applikation *DABstep* vor allem auch die Benutzerfreundlichkeit im Mittelpunkt. Die Applikation soll von Laien intuitiv bedienbar sein und übersichtlich bleiben. Das Programm ist in Empfänger und Sender aufgeteilt, welche über zwei Tabs erreichbar sind. Abb. 4.2 zeigt die Empfängerseite.

Die räumliche Aufteilung ist dem sequentiellen Ablauf einer Empfangssituation nachempfunden:

1. **Initiale Einstellungen:** Wahl der Quelle, DAB Modus und Mittenfrequenz.
2. **Senderliste:** Anzeige und Auswahlfeld der DAB Sender.

3. **Informationen:** Anzeige von zusätzlichen Informationen über das empfangene DAB Ensemble und den ausgewählten Sender.
4. **Audio:** Kontrolle über die abgespielte Musik, zum Beispiel Lautstärkeregelung oder Aufnahmefunktion.

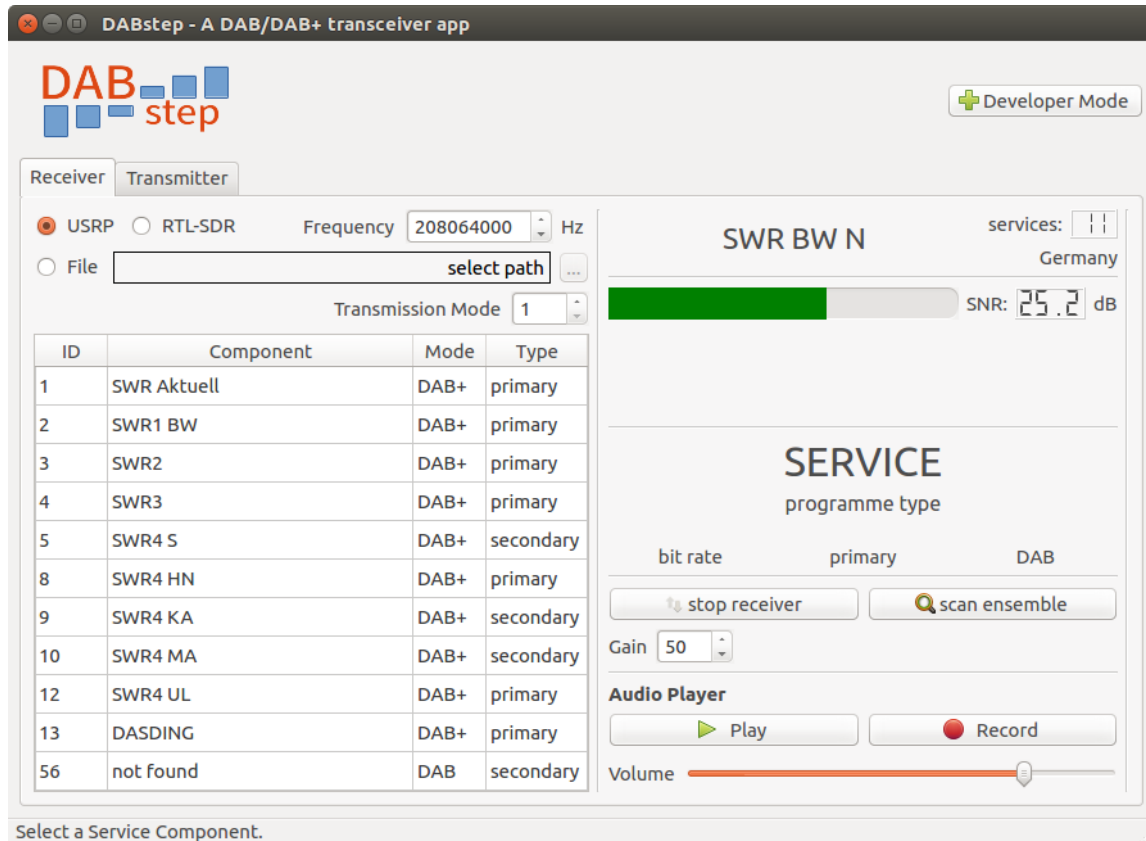


Abbildung 4.2.: DABstep im Empfängermodus

Während die Empfängerseite hauptsächlich Informationen anzeigt, liegt der Schwerpunkt bei der Senderseite auf dem Schreiben von Informationen für das zu sendende DAB Ensemble. Um die Bedienfreundlichkeit zu bewahren, wurde die Aufteilung im Sender von der funktionalen Aufteilung analog zum Empfänger gestaltet. Abb. 4.3 zeigt die GUI im Sendemodus. Die Wahl zwischen Hardware und Binärdatei ist auch im Sendemodus vorhanden und definiert in diesem Fall die Art der Senke. Die Informationen über die einzelnen Radiosender können über die Eingabeformulare individuell eingestellt werden. Es ist dabei auch möglich in einem Ensemble DAB und DAB+ Kanäle zu mischen. Als Audioquelle gibt es neben Audiodateien auch die Möglichkeit, die lokale Soundkarte mit einem Mikrofon zu nutzen. Im rechten oberen Bereich sind, wie im Empfänger, die Ensemble Informationen platziert. Durch das Erhöhen der Anzahl der Kanäle wird die Spalte mit den Informationen zu den einzelnen Kanälen auf der linken Seite automatisch um eine Komponente für den neuen Kanal ergänzt. Die Audioeinstellungen befinden sich auch hier im rechten unteren Bereich der Oberfläche. Der Benutzer muss den Kanal, den er hören will, auswählen, da er logischerweise nicht alle Kanäle des DAB Ensembles gleichzeitig hören kann.

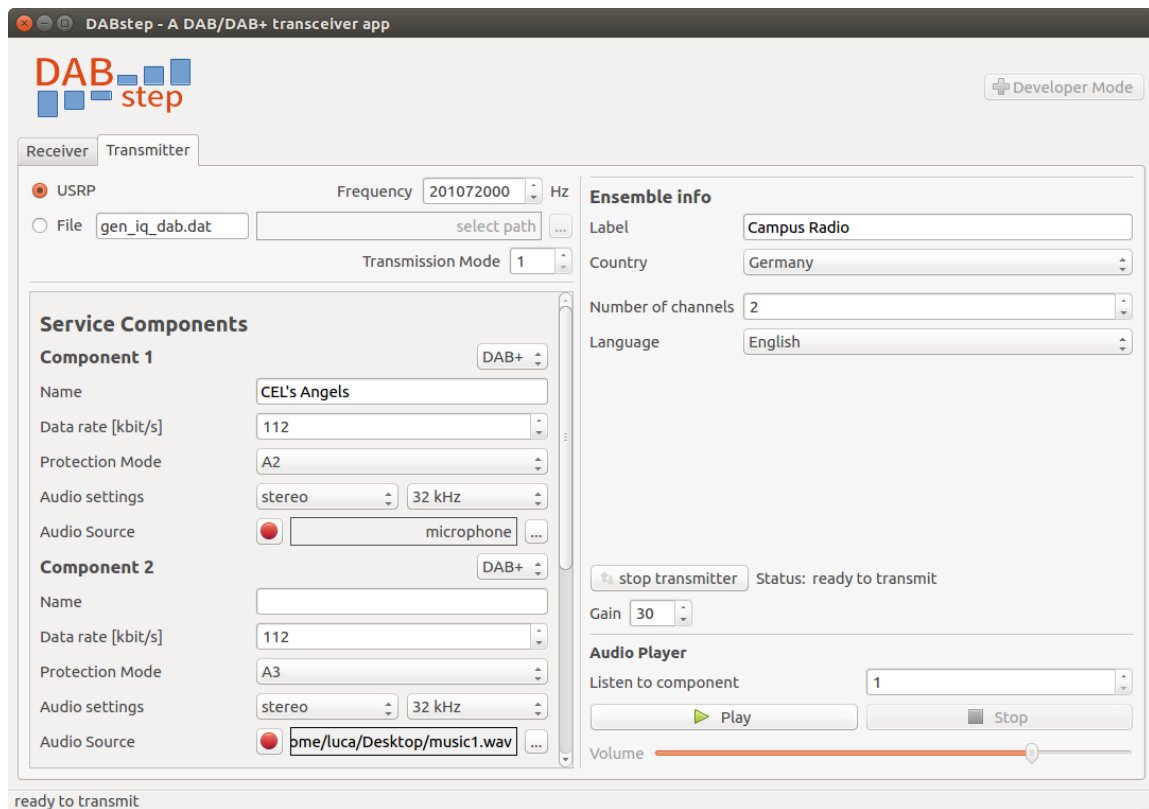


Abbildung 4.3.: DABstep im Sendemodus

Entwicklermodus

Eine Besonderheit stellt der Entwickler-Modus dar, welcher in der Empfängerseite über das „+“ Symbol in der oberen rechten Ecke erreichbar ist. Dieser erweitert die GUI um einige technische Anzeigen (siehe Abb. 4.4).

Eine Anzeige für die Fehlerrate, getrennt für FIC und MSC, stellt das Ergebnis des CRC checks der FIBs bzw. des Firecode Checks für die Audio Frames dar. Die rechte Spalte ergänzt die GUI um ein Konstellationsdiagramm, eine FFT Anzeige und einen Wasserfall Plot. Die drei Widgets wurden dabei aus dem GNU Radio Modul *gr-qtgui* [GR17b] übernommen, in den Flowgraph eingebaut und beim Öffnen des Entwicklermodus an die GUI als Objekte übergeben.

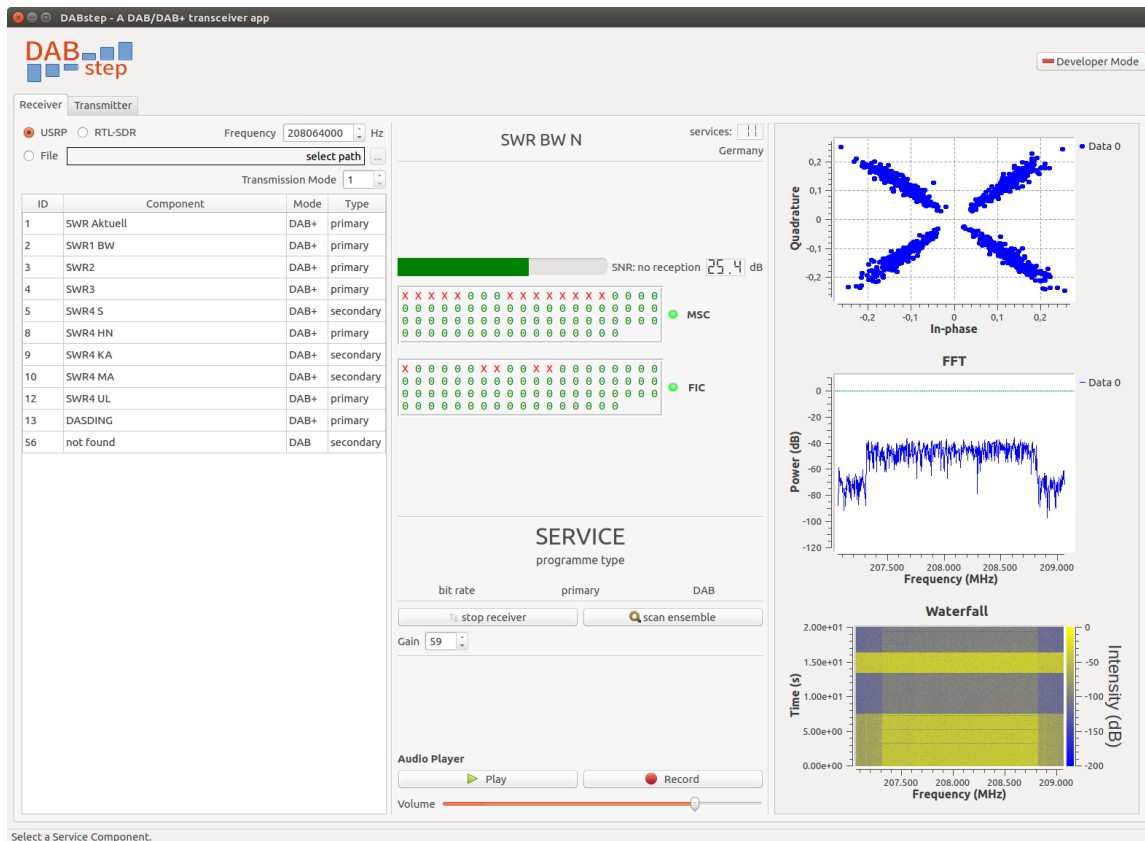


Abbildung 4.4.: DABstep im Entwicklermodus

5. Evaluation des Systems

Nach erfolgreicher Implementierung des DAB/DAB+ Transceivers folgt in diesem Abschnitt eine Evaluation des Systems. Für eine grundsätzliche Verifikation wird der Empfänger gegen das DAB+ Ensemble eines lokalen Radiosenders in einer realen Übertragung getestet. In der Simulation wird der Transceiver anschließend auf sein Verhalten und seine Robustheit bei verschiedenen Übertragungskanälen untersucht.

Für quantitative Aussagen stellt die Bitfehlerrate (BER) ein gängiges Gütekriterium zur Bewertung der Übertragungsqualität des physikalischen Kanals dar. Die BER allein ist jedoch kein ausreichendes Maß für die vom Nutzer subjektiv empfundene Qualität des Empfangssignals. Sowohl im FIC als auch im MSC entscheidet letztlich ein CRC darüber, ob ein FIB beziehungsweise ein MPEG Frame korrekt ist und es dementsprechend weiterverarbeitet oder verworfen wird. Das Audiosignal wird, im Gegensatz zu FM, bei gestörter Übertragung nicht verrauscht, sondern die Ausfallrate, auch Paketfehlerrate (PER) genannt, steigt. Deshalb ist es sinnvoll, die Fehlerrate dieser Pakete über die CRCs als Gütekriterium für eine praktische bzw. subjektive Übertragungsqualität zusätzlich zur Bitfehlerrate heranzuziehen.

Die Performanz des Systems hängt von vielen Faktoren ab. Die Konzeption der Systemarchitektur entscheidet über die grundlegenden Eigenschaften des Systems und ist durch den ETSI DAB Standard [ETSI17c] festgelegt. Ein weiterer wesentlicher Faktor ist das Design der Synchronisation. Diese ist nicht im Standard spezifiziert und hängt deshalb stark von der jeweiligen Implementierung ab.

5.1. Verifikation des Empfängers

Der Empfänger wird gegen ein lokales DAB+ Ensemble des Südwestrundfunks (SWR) verifiziert. Die Sendestation ist in ca. 10 km Entfernung und sendet mit einer effektive Strahlungsleistung (ERP) von etwa 1 kW [UA17]. Der Empfang findet im Gebäude und ohne Bewegung des Empfängers statt. Es wurden insgesamt 3 Empfängerkonfigurationen getestet:

Hardwarekonfiguration	SNR	subjektive Empfangsqualität
USRP B210 mit DAB-Stabantenne	≈ 50 dB	komplett störungsfrei
USRP B210 mit Teleskopantenne	≈ 28 dB	komplett störungsfrei
RTL-SDR mit DVB-T kleiner Stabantenne	≈ 18 dB	teilweise gestört

Abbildung 5.1.: getestete Hardwarekonfigurationen und deren Ergebnisse

Durch den erfolgreichen Empfang des SWR DAB+ Ensembles ist der Empfänger verifiziert. Die SNR Werte spiegeln wie zu erwarten die Eignung der jeweiligen Antenne für den DAB Empfang wider. Der hohe Qualitätsunterschied der SDR-Geräte ist bei der rei-

nen SNR Messung nicht vollständig berücksichtigt, wirkt sich aber im weiteren Verlauf der Synchronisations- und Decodierungskette auf die Empfangsqualität aus.

Zur subjektiven Empfangsqualität seien zwei Anmerkungen gemacht: Die subjektive Audioqualität der DAB+ Übertragung ist im Vergleich zu FM wesentlich besser; dies kommt jedoch nur bei der Verwendung eines hochwertigen Lautsprechers oder Kopfhörers zum Tragen. Bei kostengünstigen Lautsprechern ist ein Qualitätsunterschied kaum hörbar. Bei sehr schlechter Empfangsqualität fallen einzelne DAB+ Audioframes aus und unterbrechen dadurch den Audiostream für die Dauer eines Superframes von 120 ms. Auch wenn die restlichen Audioframes komplett fehlerfrei und rauschfrei sind, ist in diesem Fall das sehr veräuschte aber unterbrechungslose FM Audiosignal angenehmer für den Hörer. Eine mögliche Verbesserung dieser subjektiven Empfindung kann durch weiches Aus- und Einblenden vor bzw. nach einem Frameausfall erreicht werden. Diese Methode wird in einigen kommerziellen DAB+ Radios bereits eingesetzt und erzielt den gewünschten Effekt.

5.2. AWGN Kanal

Der AWGN Kanal ist das einfachste hier verwendete Kanalmodell. Der Kanal addiert bei der Übertragung weißes, gaußsches Rauschen auf das Sendesignal.

$$r(t) = s(t) + n(t) \quad \text{mit} \quad n(t) \sim \mathcal{CN}(0, \sigma^2) \quad (5.1)$$

5.2.1. Bitfehlerrate

Die simulierte Bitfehlerrate des DAB Systems wird über dem SNR gemessen, indem bei bekannter Signalleistung $P_s = 1 \text{ W}$ die Rauschleistung entsprechend dem SNR eingestellt wird. Um aussagekräftige Ergebnisse über die Performanz einer digitalen Übertragung zu erhalten ist nach [Skl01] die Bitenergie im Verhältnis zur Rauschleistungsdichte E_b/N_0 ein sinnvollerer Gütemaß als das SNR, da Energie Bits überträgt, und nicht Leistung. Die Umrechnung von SNR zu E_b/N_0 ergibt:

$$\begin{aligned} \frac{E_b}{N_0} &= \frac{P_s}{N} \cdot \frac{B}{R} \quad \text{mit} \quad P_s = 1, B = 1, N = \sigma^2 \\ &= \frac{1}{\sigma^2} \cdot \frac{1}{R} \\ &= \frac{1}{\sigma^2} \cdot \frac{1}{\frac{K}{N_{Träger}} \cdot \log_2(M) \cdot \frac{T_s}{T_G + T_S}} \end{aligned} \quad (5.2)$$

Die Bitrate R ergibt sich aus den Systemparametern. Symbolratenverluste entstehen durch $K = 1536$ belegten von insgesamt $N_{Träger} = 2048$ verfügbaren OFDM Unterträgern. Die redundante Wiederholung von QPSK Symbolen im Cyclic Prefix verkleinert die effektive Symbolrate weiter. Jedes QPSK Symbol transportiert dabei $\log_2(4) = 2$ Bits. Damit ergibt sich das folgende Verhältnis von SNR zu E_b/N_0 in dB:

$$\frac{E_b}{N_0} = \text{SNR} - 10 \cdot \log_{10} \left(\frac{1536}{2048} \cdot 2 \cdot \frac{2048}{504 + 2048} \right) \approx \text{SNR} - 0,8054 \text{ dB} \quad (5.3)$$

Diese Umrechnung erlaubt es nun die Bitfehlerrate des DAB Systems mit der theoretischen Bitfehlerrate einer reinen D-QPSK Übertragung zu vergleichen. Beide Kurven sind in Abb.

5.2 dargestellt. Für das DAB System wurde in dieser Simulation lediglich der physikalische Kanal, also das OFDM System und die Synchronisierung untersucht. Die anschließende Kanalcodierungskette wurde nicht in die Simulation miteinbezogen.

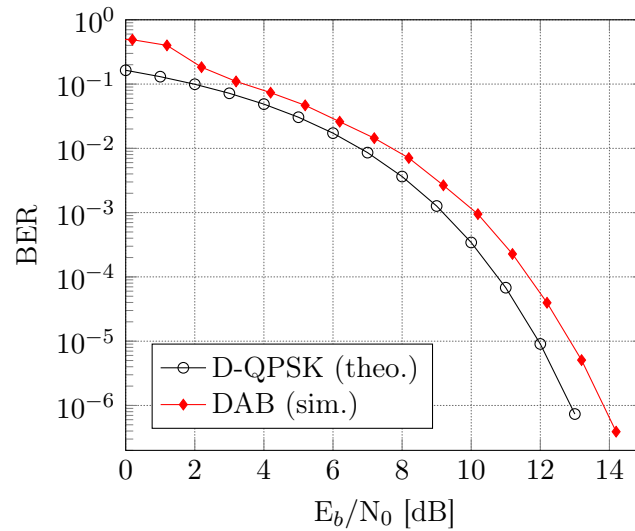


Abbildung 5.2.: BER über E_b/N_0 bei AWGN Kanal

Wie zu erkennen ist, hat die BER Kurve des DAB Systems den gleichen Verlauf wie die D-QPSK BER Kurve und ist um etwa 1 dB nach rechts verschoben. Um also die gleiche BER zu erreichen, benötigt der DAB Empfänger ein um 1 dB besseres E_b/N_0 . Der Performanzverlust ist durch den für die Synchronisation notwendigen Overhead durch Cyclic Prefix und unbelegte Unterträger zu erklären.

Ein Implementationsverlust ist lediglich unter ca. 3 dB zu erkennen, bei dem die BER des DAB Empfänger sehr schnell gegen einen Wert von 0,5 konvergiert. Das Verhalten kann unter Berücksichtigung der Detektionsrate des Synchronisationsblocks erklärt werden. Unter 2 dB nimmt die Rate der detektierten Frames rapide ab und bei etwa 0 dB wird quasi kein Frame von der Synchronisation erkannt. In dieser Simulation wurden die Bits eines nicht erkannten Frames mit $BER = 0,5$ gewichtet, was einem Raten von gleichverteilten Bits entspricht. Unter 2 dB ist die Rauschleistung in der gleichen Größenordnung wie die Signalleistung, wodurch das Nullsymbol nicht mehr mit großer Sicherheit von anderen Symbolen unterschieden werden kann. Aus diesem Grund wurde die Empfindlichkeitsschwelle für die Synchronisation (siehe 3.4.1) bewusst auf 2 dB gesetzt. Ein zu geringer Schwellwert für die Energiedifferenz zwischen Nullframe und Phasenreferenzsymbol sollte unbedingt vermieden werden. Er kann dazu führen, dass Leistungsschwankungen einen Falschalarm bei der Framedetektion auslösen. Zudem liefern E_b/N_0 Werte unter 2 dB so hohe Bitfehlerraten, dass ein sinnvoller Betrieb nicht mehr möglich ist.

5.2.2. Paketfehlerrate

In einem zweiten Schritt wird nun die Paketfehlerrate (PER) untersucht. Die PER Simulation soll dabei so weit wie möglich einer Ende zu Ende Übertragung entsprechen, da dieser Fall der subjektiv empfundenen Übertragungsqualität am nächsten kommt. Deshalb wird

bei der Simulation der PER die komplette Kanalcodierungskette in die Simulationsumgebung integriert.

Fast Information Channel

Im FIC wird nach der Kanalcodierung ein CRC über jedes FIB berechnet, dessen Ergebnis direkt für die PER Berechnung verwendet werden kann. Da der CRC jeweils über das komplette FIB berechnet wird, kann man bei erfolgreichem CRC von einem fehlerfreien FIB ausgehen. Die PER entspricht also direkt der FIB Fehlerrate und ist aussagekräftig für die effektive Übertragungsfehlerrate des FIC. Abb. 5.3 zeigt die PER des FIC über E_b/N_0 .

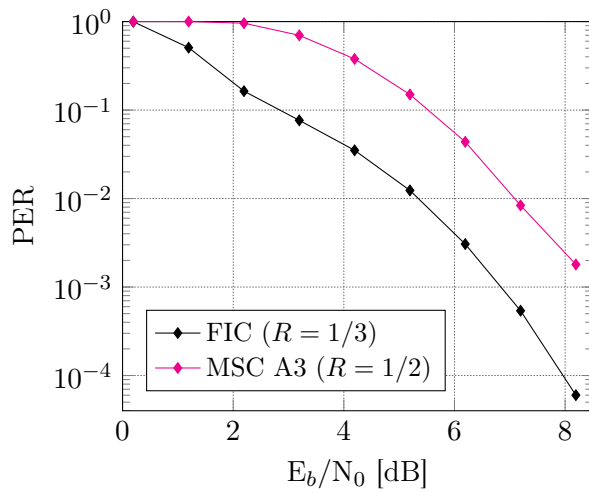


Abbildung 5.3.: PER bei AWGN

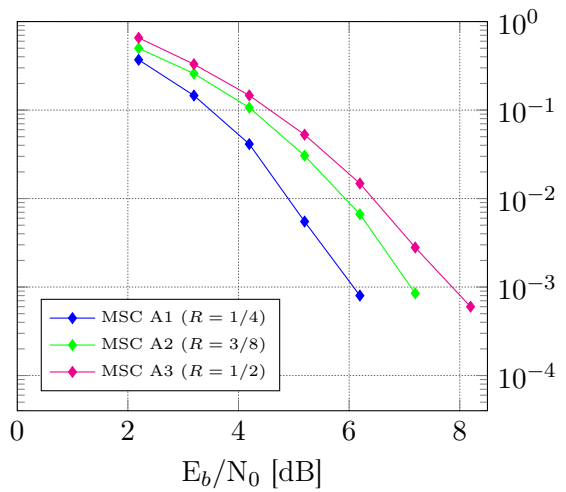


Abbildung 5.4.: PER bei AWGN

Eine PER schneidet in der Regel deutlich schlechter als eine äquivalente BER ab, da alle Bits für ein erfolgreiches Paket korrekt sein müssen. Schon einzelne falsche Bits führen zum Scheitern eines ganzen Paketes, während bei der BER jedes Bit unabhängig zur Fehlerrate beiträgt. Durch die Effekte der Kanalcodierung und der damit verbundenen Fehlerkorrektur des Faltungsdecoders fällt die PER jedoch trotzdem in der gleichen Größenordnung wie die BER aus.

Im Bereich unter 3 dB verlässt die PER des FIC ihren typischen AWGN Verlauf und konvergiert deutlich schneller gegen eine totale Ausfallrate als erwartet. Hier zeigt sich wieder der Einfluss der Detektionsrate der Synchronisation, die schon bei der BER diskutiert wurde. Die PER wurde für den Fall eines nicht erkannten Frames als Fehler berechnet, da ein Paket mit $BER = 0,5$ im Allgemeinen nicht korrigierbar ist.

Main Service Channel (MSC)

Die PER des MSC kann vom Firecode-Check abgeleitet werden, der nach der Kanaldecodierung und vor der Reed-Solomon Korrektur stattfindet. Die angewandte Kanalcodierung unterscheidet sich vom FIC in der Faltungscodierung bzw. deren Coderate. Zusätzlich wird

der MSC um das Zeit-Interleaving ergänzt.

Hier sei angemerkt, dass der Firecode laut DAB+ Standard nicht über das gesamte Superframe berechnet wird, sondern nur über die ersten 9 Byte. Der Firecode hat somit lediglich Einfluss auf die Bits des MPEG Headers. Im Gegensatz dazu wird der CRC des FIC jeweils über das gesamte FIB, also über 30 Byte, berechnet. Ein direkter Vergleich der PER des Firecode mit der PER des FICs ist daher erst nach einer Normierung möglich.

Die PER hängt bei fester BER von der Anzahl der Bits ab, die pro Paket überprüft werden. Diese effektive Bitlänge ist für den CRC im FIC $L_1 = 32$ Byte und für den Firecode $L_2 = 11$ Byte. Dabei kommen jeweils 2 Byte zur eigentlichen Berechnungslänge hinzu, da auch die 16 Bits des CRC- bzw. Firecode-Wortes selbst falsch übertragen werden können. Die Wahrscheinlichkeit, dass ein Paket der Länge L_1 fehlerfrei ist entspricht der Wahrscheinlichkeit, dass genau $\frac{L_1}{L_2}$ unabhängige Pakete der Länge L_2 fehlerfrei sind.

$$\begin{aligned} 1 - P_{e,1} &= (1 - P_{e,2})^{\frac{32}{11}} \\ \Leftrightarrow P_{e,1} &= 1 - (1 - P_{e,2})^{\frac{32}{11}} \end{aligned} \quad (5.4)$$

Mit Gl. 5.4 kann die PER des Firecodes auf die Länge 32 Byte normiert werden. Die resultierende normierte PER ist in Abb. 5.3 aufgetragen und kann nun direkt mit der PER des FIC verglichen werden. Die PER Kurve des MSC hat den selben Verlauf wie die des FIC und ist um etwa 2 dB verschoben. Die schlechtere Performanz kann mit der höheren Coderate des MSC im Protection Mode A3 von $R = 1/2$ im Vergleich zu $R = 1/3$ des FIC erklärt werden. Das zusätzliche Zeit-Interleaving im MSC macht sich im AWGN Fall in der Performanz nicht bemerkbar, da nur unabhängige Einzelfehler auftreten.

Die Normierung der Firecode-PER auf die Länge der FIBs ist sinnvoll, um die beiden PERs miteinander zu vergleichen. Die subjektive Audioqualität, wie sie am Anfang dieses Kapitels motiviert wurde, ist damit aber nicht hinreichend gut beschrieben. Damit ein Superframe fehlerfrei im Audio Decoder ankommt, muss es die Stufen aus Abb. 3.11 durchlaufen.

Nach erfolgreichem Firecode-Check werden im Reed-Solomon Decoder alle restlichen Bitfehler korrigiert, wenn dies möglich ist. Ausschließlich nach erfolgreicher Korrektur wird das Superframe an den Audio Decoder weitergegeben und kann zum Audiostream beitragen. Eine realistische Ausfallwahrscheinlichkeit für die Superframes ergibt sich wie folgt:

$$\begin{aligned} 1 - P_{e,\text{Superframe}} &= (1 - P_{e,\text{Firecode}}) \cdot (1 - P_{e,\text{Reed-Solomon}}) \\ \Leftrightarrow P_{e,\text{Superframe}} &= P_{e,\text{Firecode}} + P_{e,\text{Reed-Solomon}} - P_{e,\text{Firecode}} \cdot P_{e,\text{Reed-Solomon}} \end{aligned} \quad (5.5)$$

Dabei entspricht $P_{e,\text{Firecode}}$ der nicht normierten Ausfallwahrscheinlichkeit $P_{e,2}$ aus Gl. 5.4. $P_{e,\text{Reed-Solomon}}$ wird über die Ausfallrate des Reed-Solomon Decoders bei vorgeschaltetem Firecode-Checker geschätzt. Die resultierende PER des MSC ist für das oft verwendete Protection Level A3 in Abb. 5.3 dargestellt. Die PER Kurve ist um ca. 2 dB schlechter als die Kurve des FIC. Der Verlust kann mit der höheren Coderate von $R = 1/2$ begründet werden.

Ein Vergleich der verschiedenen Protection Levels des MSC ist in Abb. 5.4 zu sehen. Die PERs sind in diesem Diagramm nicht gemäß Gl. 5.4 auf eine andere Länge normiert, weshalb die Kurve für das Protection Level A3 in den Abbildungen 5.3 und 5.4 (jeweils in magenta dargestellt) unterschiedliche Verläufe annimmt. Der Vergleich zeigt, dass die Wahl

des Protection Levels deutlichen Einfluss auf die resultierenden PERs nimmt. Die besten Ergebnisse liefert der Protection Mode A1, der sich geringere PERs durch eine sehr kleine Coderate von $R = 1/4$ erkaufte.

5.3. Realistische Kanalmodelle

Der im vorherigen Kapitel diskutierte AWGN Kanal ist ein stark vereinfachtes Kanalmodell das viele Effekte, die bei einer realen Übertragung auftreten, vernachlässigt. Eine vollständig realistische Übertragungssituation kann in einer Simulation niemals nachgestellt werden, es können aber die wichtigsten Effekte in das Kanalmodell integriert werden um der realistischen Übertragung so nah wie möglich zu kommen. Diese Effekte sind bei einer typischen Rundfunkübertragung:

- **Mehrwegempfang:** Durch Reflexion, Streuung und Beugung erreicht ein gesendetes Signal den Empfänger auf verschiedenen Wegen. Jede Version des Sendesignals durchläuft dabei einen anderen Pfad und erfährt verschiedene Effekte wie Verzögerung, Dämpfung, Phasendrehung und Dopplerverschiebungen. Die Signale überlagern sich am Empfänger additiv und bilden das Empfangssignal.
- **NLOS (Non-Line-of-Sight):** Wegen des sehr großen Einzugsgebiets eines DAB Senders ist es unwahrscheinlich, dass der Empfänger direkte Sichtverbindung zur Sendeantenne hat. In der Menge der Mehrwege ist daher kein direkter Pfad enthalten, der in der Regel die geringste Verzögerung und Dämpfung erfahren würde.
- **Doppler-Spread:** Durch eine Bewegung des Empfängers oder Objekten, an denen das Sendesignal reflektiert wird, entsteht eine Dopplerverschiebung. Durch die Mehrwegeausbreitung variiert die zeitabhängige Dopplerverschiebung über den Mehrwegen, was zu einem ganzen Spektrum an Dopplerverschiebungen führt. Der Doppler-Spread ist vor allem dann von hoher Relevanz, wenn mobile Empfänger betrachtet werden, was im Falle von DAB eine hohe Anwendung in Form von Autoradios findet.

Der beschriebene NLOS Mehrwegeempfang kann unter der Annahme einer gleichverteilten Phasendrehung und normalverteilter Kanalkoeffizienten nach [Pro95] in einem Rayleigh Fading Kanal dargestellt werden. Das Empfangssignal ergibt sich zu

$$r(t) = h(t, \tau) * s(t - \tau_{\text{off}}) \cdot e^{j(2\pi f_{\text{off}}t + \varphi_{\text{off}})} + n(t) \quad (5.6)$$

Zusätzlich addiert der Kanal weißes, gaußsches Rauschen $N(t) \sim \mathcal{CN}(0, \sigma^2)$ auf das resultierende Signal.

Die Anzahl der Mehrwege und deren Verzögerung und mittlere Dämpfung ist stark von der Umgebung abhängig und muss für verschiedene Übertragungssituationen separat festgelegt werden. In dieser Arbeit wurden die folgenden zwei Szenarien betrachtet:

1. DVB-T Kanalmodell MR100: *Motorway Rural channel with 12 paths* [PK13] [ETS09]. Dieses Kanalmodell beschreibt einen Mehrwegempfang mit vielen kurzen, stark gedämpften Echos. Die Ausbreitungsbedingungen von DVB-T sind sehr ähnlich zu DAB, weshalb das Kanalmodell verwendet werden kann. Der Empfang von DAB auf einer Autobahn ist außerdem eine typische Empfangssituation eines Autoradios.

2. DAB Kanalmodell DAB_HT_2: *DAB Hilly Terrain Profile 2* [BHA13]. Das Modell beschreibt eine hügelige Landschaft, bei dem drei wenig gedämpfte Mehrwege mit einer hohen Verzögerung empfangen werden.

Die mittlere Verteilung der Leistung über der Verzögerung kann mit dem Power Delay Profile (PDP) beschrieben werden.

$$P_d(\tau) = E(|h(0, \tau)|^2) \quad (5.7)$$

Es konkretisiert damit die Anzahl der Mehrwege und ihre jeweilige mittlere Dämpfung in Abhängigkeit der Verzögerung. Die PDPs der untersuchten Szenarien sind in Abb. 5.5 und 5.6 dargestellt. Es ist die unterschiedliche Skalierung der Zeit-Achsen zu beachten.

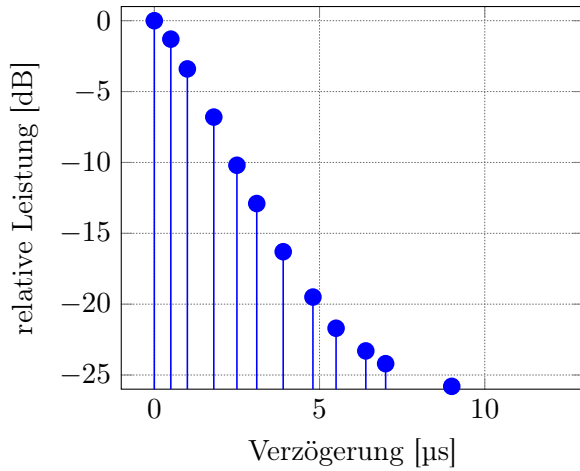


Abbildung 5.5.: PDP von MR100

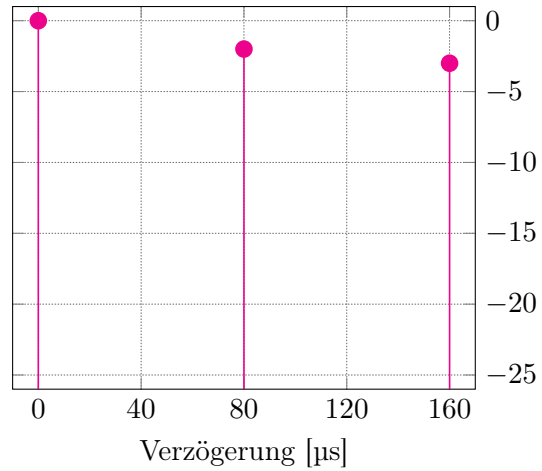


Abbildung 5.6.: PDP von DAB_HT_2

5.3.1. Kohärenzbandbreite

Durch die verschiedenen Kanallaufzeiten der einzelnen Mehrwege kommt es am Empfänger zu konstruktiver und destruktiver Interferenz, die von der Verzögerung der Mehrwege, aber auch von der Frequenz abhängt. In einem Frequenzband der Bandbreite B variiert deshalb die mittlere empfangene Leistung über der Frequenz, was als frequenzselektives Fading bezeichnet wird. Die Bandbreite, in der ein Kanal als flach angenommen werden kann, wird als Kohärenzbandbreite bezeichnet. Für das Kanalmodell DAB_HT_2, das mit $T_d = 160 \mu\text{s}$ die längsten Echos enthält, ist die Kohärenzbandbreite

$$B_c := \frac{1}{2T_d} = 12,5 \text{ kHz} < 1536 \text{ kHz} = B \quad (5.8)$$

Das Gesamtübertragungssystem ist also frequenzselektiv. Durch die parallele Übertragung in OFDM Unterträgern mit der Unterträgerbreite $\Delta f = 1 \text{ kHz} < 12,5 \text{ kHz} = B_c$ kann jedoch jeder Unterträger als frequenzflach angenommen werden.

5.3.2. Doppler-Spread

Für die Wahl der maximalen Dopplerverschiebung wird die Geschwindigkeit eines mobilen Empfängers betrachtet. Bei einer typischen Trägerfrequenz für die terrestrische DAB Übertragung $f_T = 200$ MHz ergibt sich eine Proportionalität zwischen der Geschwindigkeit und der Dopplerverschiebung.

$$|\Delta f_D| \leq f_T \frac{v}{c} = \frac{200 \text{ MHz}}{3 \cdot 10^8 \text{ m/s}} v \quad (5.9)$$

Der Doppler-Spread ist die maximale Differenz der auftretenden Doppler-Verschiebungen [Pro95].

$$D_s = \max_{i,k} |\Delta f_{D,i} - \Delta f_{D,k}| = 2|\Delta f_{D,\max}| \quad (5.10)$$

Die folgenden Dopplerverschiebungen und damit korrespondierenden Geschwindigkeiten werden untersucht:

$ \Delta f_{D,\max} $ [Hz]	v [km/h]	Bewegungsprofil
5	27	Fahrt in Wohngebiet
10	54	Fahrt auf Hauptstraße
20	100	Fahrt auf Autobahn
50	270	Grenzwert für maximale Geschwindigkeit

Abbildung 5.7.: Untersuchte Bewegungsprofile

Kohärenzzeit

Die Kohärenzzeit definiert die Zeitspanne, in welcher sich der Kanal nicht nennenswert ändert. Die Kohärenzzeit berechnet sich aus dem Doppler-Spread.

$$T_c := \frac{1}{4D_s} \geq \frac{1}{4D_{s,\max}} = 2,5 \text{ ms} > 1 \text{ ms} = T_s \quad (5.11)$$

Da selbst für den größten hier betrachteten Doppler-Spread die Kohärenzzeit größer als die Symboldauer ist, liegt in allen betrachteten Szenarien slow fading vor, das heißt der Kanal ändert sich innerhalb der Dauer eines Symbols nicht nennenswert.

5.3.3. Bitfehlerrate

In Abb. 5.8 sind die Bitfehlerraten des DAB OFDM Systems über dem E_b/N_0 im MR100 Kanal aufgetragen. Die simulierten Doppler-Spreads entsprechen den in Tabelle 5.7 gezeigten typischen Geschwindigkeiten. Als Referenz ist die theoretische BER Kurve eines Rayleigh-Fading Kanals bei D-QPSK Modulation mit Antennendiversität 1 abgebildet.

Im Vergleich zur theoretischen BER Kurve im AWGN Kanal (Abb. 5.2), schneidet die BER im Rayleigh-Fading Kanal deutlich schlechter ab. Die Simulationsergebnisse für geringe Dopplerverschiebungen liegen nahe an der theoretischen Kurve. Mit zunehmendem Doppler-Spread laufen die Kurven immer früher in einen Error-Floor. Sehr schlechte Ergebnisse liefert eine Simulation mit der maximalen Dopplerverschiebung von 50Hz. Eine BER von unter 10^{-2} wird selbst bei hohem SNR nicht erreicht.

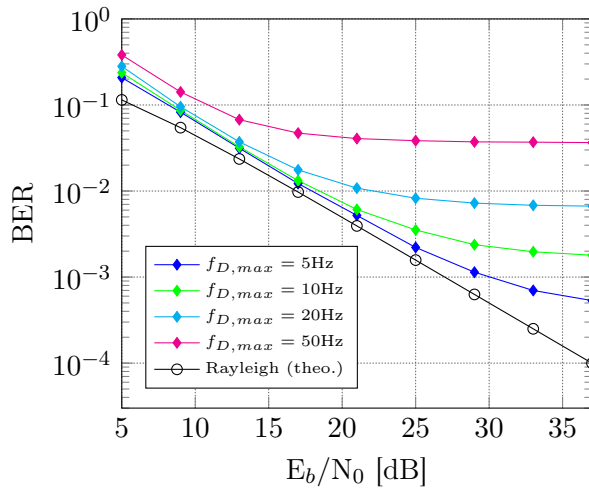


Abbildung 5.8.: BER für MR100

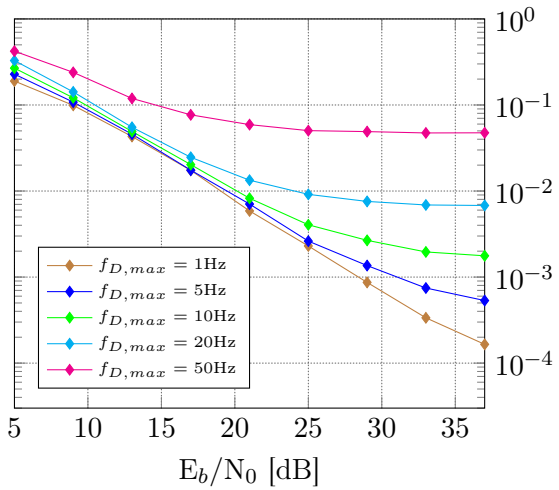


Abbildung 5.9.: BER für DAB_HT_2

In Abb. 5.9 sind die Ergebnisse der BER Simulation für das Kanalmodell DAB_HT_2 dargestellt. Die BER Kurven haben den gleichen Verlauf wie die Kurven in Abb. 5.8 und sind um etwa 3 dB nach rechts verschoben. Die BER für 50 Hz Dopplerverschiebung läuft außerdem in einen schlechteren Error-Floor von etwa $5 \cdot 10^{-2}$. Die geringe Verschlechterung im Kanalmodell DAB_HT_2 im Vergleich zu MR100 demonstriert die Robustheit des DAB Systems gegen stark verzögerte Mehrwege. Die CP Länge von $T_G = 246 \mu\text{s}$ ist dabei noch nicht ausgereizt.

5.3.4. Paketfehlerrate

Die Paketfehlerrate im Rayleigh-Fading Kanal mit realistischem PDP bildet schließlich die Simulation, die der subjektiven Empfangsqualität im Realfall am nächsten kommt. Wegen der Ähnlichkeit von den BER Kurven der beiden Kanalmodelle, wurde die PER nur für das Kanalmodell DAB_HT_2 untersucht. Die Simulationsergebnisse sind in Abb. 5.10 gezeigt.

Die PER Kurven für maximale Dopplerverschiebungen von 5 - 20 Hz weisen deutlich bessere Werte als die entsprechenden BER Kurven auf und sind sehr zufriedenstellend. Die Kanalcodierung trägt hier einen enormen Teil durch die Fehlerkorrektur bei. Lediglich die PER mit einer maximalen Dopplerverschiebung von 50 Hz verbessert sich durch die Kanalcodierung nicht signifikant und läuft in einen Error-Floor mit einer PER von $3 \cdot 10^{-2}$. Es treten zu viele Bitfehler auf, wodurch die Korrekturwahrscheinlichkeit des Faltungscodes sehr gering ist. Die Dopplerverschiebung von 50 Hz entspricht einer Geschwindigkeit von 270 km/h. Der Wert wurde bewusst sehr hoch angesetzt um die Grenzen des Systems zu testen. PERs mit Werten über 10^{-2} stellen eine ungenügende Empfangsqualität dar. Eine tatsächliche Grenze für ausreichenden Empfang ist daher zwischen 20 Hz und 50 Hz anzusetzen.

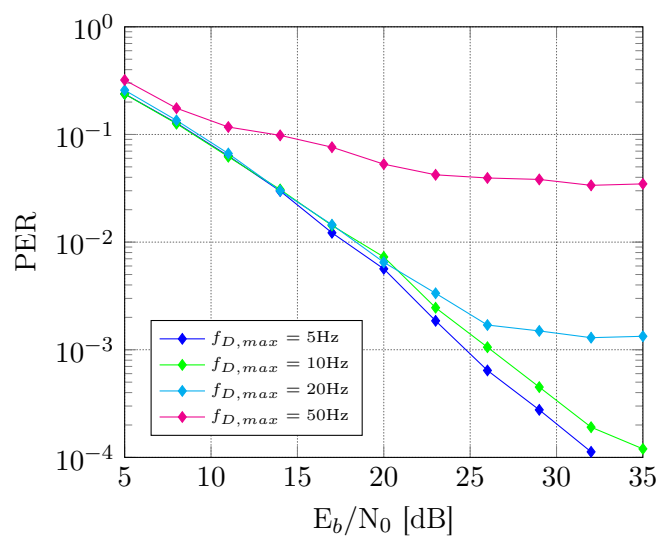


Abbildung 5.10.: PER für DAB_HT_2

6. Fazit

In dieser Arbeit wurde ein standardkonformer, funktionsfähiger DAB/DAB+ Sender und Empfänger implementiert. Das entstandene Transceiver System ist ein Software Defined Radio und kann mit verschiedenen Hardwarekonfigurationen genutzt werden.

Der DAB Sender bietet die Möglichkeit ein individuell anpassbares DAB/DAB+ Ensemble mit bis zu 7 Kanälen und entsprechenden Metainformationen zu erzeugen und abzustrahlen. Der Empfänger ist in der Lage ein DAB/DAB+ Ensemble zu empfangen. Er kann sowohl Metadaten lesen und anzeigen, als auch Audiokanäle decodieren und wiedergeben. Sowohl der Sender als auch der Empfänger wurden auf ihre Funktionalität verifiziert. Sie sind zudem beide in Echtzeit lauffähig.

Zusätzlich wurde die Applikation *DABstep* realisiert, die auf Grundlage des implementierten Transceivers ein grafisches Benutzerfrontend darstellt. Die Applikation ist benutzerfreundlich gestaltet, indem sie Prozesse automatisiert und Metadaten sinnvoll aufbereitet. Durch eine übersichtliche und einfache Strukturierung der grafischen Oberfläche kann die Applikation auch von Laien bedient werden. Der Entwicklermodus erlaubt es, zusätzliche nachrichtentechnisch interessante Details anzeigen zu lassen.

In der Evaluation wurde das System unter verschiedenen Bedingungen auf seine Funktionalität und Performanz getestet. Ein Test mit Referenzdaten sowie ein Schleifentest verifiziert die grundsätzliche Funktionalität des Transceivers.

Im simulierten AWGN Kanal liegt die Bitfehlerrate des OFDM Systems von DAB nahe an der theoretischen Untergrenze. In realistischen Rayleigh-Fading Kanalmodellen mit verschiedenen Mehrwegempfangssituationen wurden deutlich höhere Bitfehlerraten gemessen. Die Paketfehlerraten des Gesamtsystems zeigen jedoch, dass eine Übertragung mit geringen Ausfallwahrscheinlichkeiten dank effektiver Kanalcodierung auch in diesen Kanälen bis zu gewissen Grenzen möglich ist.

Die Ergebnisse zeigen, dass das implementierte DAB/DAB+ System vergleichbare Performanz und Robustheit wie kommerzielle Geräte liefert. Ein großer Vorteil bietet die Realisierung als Software Defined Radio. Sie ermöglicht den Betrieb eines sehr kostengünstigen DAB/DAB+ Transceivers, zum Beispiel durch die Verwendung eines DVB-T Dongles. Der gesamte Transceiver kostet somit, abzüglich des Computers, unter 10€. Im Vergleich kosten entsprechende DAB+ Autoradios in Deutschland das 10 - 100 fache [DRD17].

Für den implementierten Empfänger ist in Zukunft eine weitere Performanzsteigerung anzustreben, die durch eine zusätzliche Fehlerkorrektur im Firecode erreicht werden kann. Zudem kann die Transceiver Applikation um weitere Funktionen, wie zum Beispiel das paketweise Übertragen von Bildern, erweitert werden.

Im Vergleich zum UKW-Rundfunk bietet DAB/DAB+ eine flexiblere Gestaltung der Senderstruktur, mehr Möglichkeiten für Service Informationen und eine bessere, da rauschfreie,

Audioqualität. Außerdem ist die DAB Übertragung deutlich energieeffizienter, weil die digitale Übertragung eine Audiokompression zulässt, die wiederum die Bandbreite pro Kanal reduziert.

Zusammenfassend ist der DAB/DAB+ Standard daher ein sinnvoller und notwendiger Nachfolger der UKW Übertragung. Demgegenüber stehen aktuelle Umfragen über die Nutzung und Verbreitung von DAB Radios, die weit hinter den erwarteten Zahlen zurückbleiben. So verfehlen aktuelle Nutzungszahlen von 35% [BT17] in Großbritannien Prognosen, die diese Abdeckung bereits für 2008 vorhergesagt hatten [Sch99]. Gründe für die schleppende Adaption von DAB ist neben den hohen Gerätepreisen und der bisher unzureichenden Empfangsabdeckung auch die Trägheit der Nutzer sich auf neue Übertragungsstandards einzustellen. Ein Gegenbeispiel bilden Norwegen und die Schweiz, die bereits dieses Jahr mit der Abschaltung der UKW-Rundfunkübertragung beginnen. Hier zeigt sich der Vorteil von DAB+, auch bei sehr schlechten Übertragungsbedingungen eine gute Audioqualität zu liefern, was in diesem Fall die Nutzer direkt betrifft und das System dadurch attraktiv macht.

A. Abkürzungsverzeichnis

AFC	Automatic Frequency Control
AWGN	Additives Weißes Gaußsches Rauschen
BER	Bitfehlerrate
CAZAC	Constant Amplitude Zero Auto-Correlation
CIF	Common Interleaved Frame
CP	Cyclic Prefix
CRC	Zyklische Redundanzprüfung
CU	Capacity Unit
DAB	Digital Audio Broadcasting
DFT	Diskrete Fourier-Transformation
EEP	Equal Error Protection
ERP	effektive Strahlungsleistung
ETSI	Europäischen Institut für Telekommunikationsnormen
FDM	Frequenzmultiplexverfahren
FIC	Fast Information Channel
FFT	Schnelle Fourier-Transformation
GPL3	GNU General Public License Version 3
GRC	GNU Radio Companion
ICI	Inter-Träger-Interferenz
IDFT	Inverse Diskrete Fourier Transformation
IFFT	Inverse Schnelle Fourier-Transformation
ISI	Intersymbol Interferenz
JSON	JavaScript Object Notation
MAP	Maximum A-Posteriori

MCI	Multiplex Configuration Information
ML	Maximum Likelihood
MPEG 2	MPEG 1/2 Audio Layer II
MPEG 4	MPEG 4 HE-AACv2
MSC	Main Service Channel
OFDM	Orthogonales Frequenzmultiplexverfahren
PAPR	Peak-to-Average Power Ratio
PCM	Puls-Code-Modulation
PER	Paketfehlerrate
PRBS	Pseudo-Random Binary Sequency
QA	Quality Assurance
SDR	Software-Defined Radio
SI	Service Information
SWIG	Simplified Wrapper and Interface Generator
UEP	Unequal Error Protection
USRP	Universal Software Radio Peripheral

Literaturverzeichnis

- [BHA13] BAUMGARTNER, S. ; HIRTZ, G. ; APITZSCH, A.: Methods for simulation of the wireless propagation channel for DAB/DAB x002B;/DMB-SDR-receivers. In: *Eurocon 2013*, 2013, S. 138–146
- [BT17] *Digital radio switchover: When will the UK's FM radio signal be turned off?* <http://home.bt.com/tech-gadgets/tech-news/dab-digital-radio-switchover-fm-uk-11364137002537>, 2017. – Zugriff 24.11.2017
- [CEL17] *GitHub, Modul gr-dab*. <https://github.com/kit-cel/gr-dab>, 2017. – Zugriff 07.11.2017
- [DRD17] *DAB+ im Auto*. <http://www.digitalradio.de/index.php/de/dab-im-auto-uebersicht>, 2017. – Zugriff 23.11.2017
- [EBU03] EBU: EBU subjective listening tests on low-bitrate audio codecs. (Juni 2003)
- [ETS09] ETSI: Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television. 2009 (ETSI EN 300 744 V1.6.1). – Forschungsbericht
- [ETS16] ETSI: Digital Audio Broadcasting (DAB); DAB audio coding (MPEG Layer II). 2016 (ETSI TS 103 466 V1.1.1). – Forschungsbericht
- [ETS17a] ETSI: Digital Audio Broadcasting (DAB); DAB+ audio coding (MPEG HE-AACv2). 2017 (ETSI TS 102 563 V2.1.1). – Forschungsbericht
- [ETS17b] ETSI: Digital Audio Broadcasting (DAB); Registered Tables. 2017 (ETSI TS 101 756 V2.2.1). – Forschungsbericht
- [ETS17c] ETSI: Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers. 2017 (ETSI EN 300 401 V2.1.1). – Forschungsbericht
- [GR17a] *GitHub, GNU Radio Modul gr-fft*. <https://github.com/gnuradio/gnuradio/tree/master/gr-fft>, 2017. – Zugriff 09.09.2017
- [GR17b] *GitHub, GNU Radio Modul gr-qgui*. <https://github.com/gnuradio/gnuradio/tree/master/gr-qgui>, 2017. – Zugriff 21.10.2017
- [GR17c] *gr-trellis Repository*. <https://github.com/gnuradio/gnuradio/tree/master/gr-trellis>, 2017. – Zugriff 25.10.2017

- [HL09] HOEG, Wolfgang ; LAUTERBACH, Thomas: *Digital Audio Broadcasting - Principles and Applications of DAB, DAB + and DMB*. 3. New York : John Wiley & Sons, 2009. – ISBN 978-0-470-74619-6
- [ITU97] ITU: Interface between Data Terminal Equipment (DTE) and Data Circuit-terminating Equipment (DCE) for terminals operating in the packet mode and connected to public data networks by dedicated circuit. (1997)
- [JK17a] *GitHub, Modul qt-dab*. <https://github.com/JvanKatwijk/qt-dab>, 2017. – Zugriff 07.10.2017
- [JK17b] *kjmp2 Repository*. https://github.com/JvanKatwijk/ada-dab/blob/master/c_files/kjmp2.c, 2017. – Accessed 04.08.2017
- [Jon11] JONDRALE, F.: *Nachrichtensysteme: Grundlagen, Verfahren, Anwendungen ; 4. Auflage*. Schlembach, 2011. – ISBN 9783935340687
- [LJK15] LEÓN, F.P. ; JÄKEL, H. ; KIENCKE, U.: *Signale und Systeme*. De Gruyter, 2015 (De Gruyter Studium). – ISBN 9783110423839
- [ODR17a] *GitHub, Opendigitalradio Modul fdk-aac*. <https://github.com/Opendigitalradio/fdk-aac>, 2017. – Zugriff 18.11.2017
- [ODR17b] *GitHub, Opendigitalradio Modul ODR-AudioEnc*. <https://github.com/Opendigitalradio/ODR-AudioEnc>, 2017. – Zugriff 09.11.2017
- [PK13] POLAK, L. ; KRATOCHVIL, T.: Exploring of the DVB-T/T2 performance in advanced mobile TV fading channels. In: *2013 36th International Conference on Telecommunications and Signal Processing (TSP)*, 2013, S. 768–772
- [Pro95] PROAKIS, J.G.: *Digital Communications*. McGraw-Hill, 1995 (Electrical and Computer Engineering Series). – ISBN 9780070517264
- [Sch99] SCHRÖDER, Hermann-Dieter: Kurzer Überblick über den Stand des terrestrischen digitalen Hörfunks / Hans-Bredow-Institut für Medienforschung, Universität Hamburg. 1999. – Forschungsbericht
- [Skl01] SKLAR, B.: *Digital communications: fundamentals and applications*. Prentice-Hall PTR, 2001 (Prentice Hall Communications Engineering and Emerging Technologies Series). – ISBN 9780130847881
- [UA17] UKW/TV-ARBEITSKREIS: *Sender-Tabelle Baden-Württemberg (DAB)*. <http://www.ukw.tv.de/cms/deutschland-dab/baden-wuerttemberg-dab.html>, 2017. – Zugriff 15.11.2017
- [WE71] WEINSTEIN, S. ; EBERT, P.: Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform. In: *IEEE Transactions on Communication Technology* 19 (1971), October, Nr. 5, S. 628–634. <http://dx.doi.org/10.1109/TCOM.1971.1090705>. – DOI 10.1109/TCOM.1971.1090705. – ISSN 0018-9332