

Two-dimensional block-based reception for differentially encoded OFDM systems : a study on improved reception techniques for digital audio broadcasting systems

Citation for published version (APA):

Houtum, van, W. J. (2012). *Two-dimensional block-based reception for differentially encoded OFDM systems : a study on improved reception techniques for digital audio broadcasting systems*. [Phd Thesis 1 (Research TU/e / Graduation TU/e), Electrical Engineering]. Technische Universiteit Eindhoven. <https://doi.org/10.6100/IR734135>

DOI:

[10.6100/IR734135](https://doi.org/10.6100/IR734135)

Document status and date:

Published: 01/01/2012

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

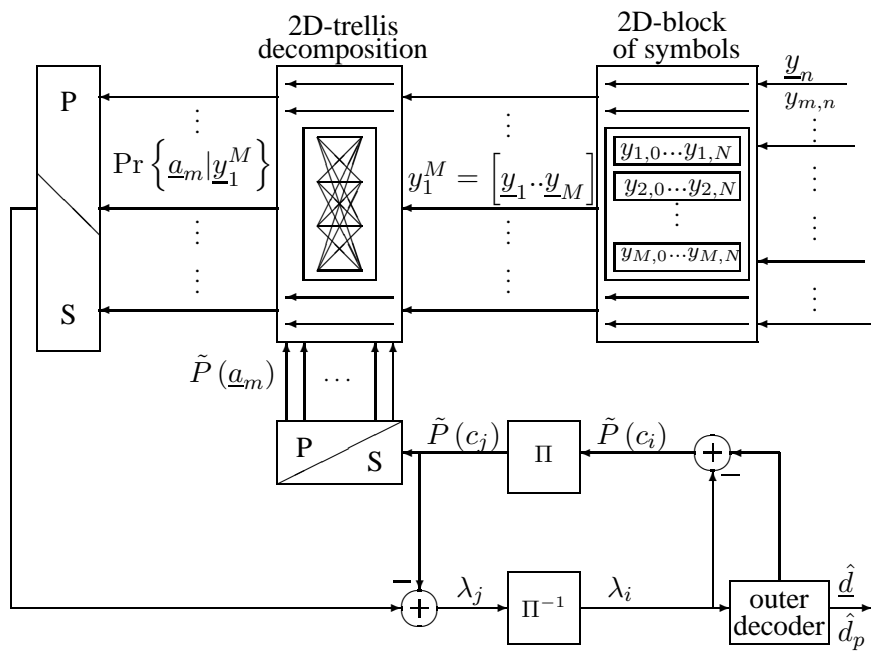
openaccess@tue.nl

providing details and we will investigate your claim.

Two-Dimensional Block-Based Reception for Differentially Encoded OFDM Systems

A study on improved reception techniques for digital audio broadcasting systems

Wim J. van Houtum



Two-Dimensional Block-Based Reception for Differentially Encoded OFDM Systems

*A study on improved reception techniques for digital audio
broadcasting systems*

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de
Technische Universiteit Eindhoven, op gezag van de rector magnificus,
prof.dr.ir. C.J.van Duijn, voor een commissie aangewezen door
het College voor Promoties in het openbaar te verdedigen
op dinsdag 23 oktober 2012 om 16.00 uur

door

Wilhelmus Johannes van Houtum

geboren te Sint-Oedenrode

Dit proefschrift is goedgekeurd door de promotor:

prof.dr.ir. J.W.M. Bergmans

Copromotor:

dr.ir. F.M.J. Willems

The work described in this thesis has been carried out at Catena Radio Design B.V. Eindhoven and at the Technical University Eindhoven as part of the joint development programme between Catena Radio Design B.V. and NXP Car Entertainment Solutions.

A catalogue record is available from the Eindhoven University of Technology Library.

CIP-DATA LIBRARY TECHNISCHE UNIVERSITEIT EINDHOVEN

Two-Dimensional Block-Based Reception for Differentially Encoded OFDM Systems

A study on improved reception techniques for digital audio broadcasting systems

van Houtum, W.J.

-Eindhoven: Technische Universiteit Eindhoven, 2012.

Proefschrift. -ISBN: 978-90-386-3183-7

NUR 959

Subject headings: Iterative decoding / differential detection / OFDM / Turbo-principle / DE-QPSK / BCJR

©Catena Radio Design B.V. 2012

All rights are reserved.

Reproduction in whole or in part is prohibited
without the written consent of the copyright owner.

Samenstelling promotiecommissie:

prof.dr.ir. A.C.P.M. Backx, TU Eindhoven, voorzitter

prof.dr.ir. J.W.M. Bergmans, TU Eindhoven, promotor

dr.ir. F.M.J. Willems, TU Eindhoven, copromotor

prof.dr.ir. L. van der Perre, KU Leuven

prof.dr.ir. A.J.H. Vinck, Univ. Duisburg-Essen

dr.ir. P.F.M. Smulders, TU Eindhoven

*Voor Dorine.
For the pets.*

Summary

Two-Dimensional Block-Based Reception for Differentially Encoded OFDM Systems

A study on improved reception techniques for digital audio broadcasting systems

Digital audio broadcast (DAB), DAB+ and Terrestrial-Digital Multimedia Broadcasting (T-DMB) systems use multi-carrier modulation (MCM). The principle of MCM in the DAB-family is based on orthogonal frequency division multiplexing (OFDM), for which every subcarrier is modulated by $\frac{\pi}{4}$ differentially encoded quaternary phase shift keying (DE-QPSK). In DAB systems convolutional codes and interleaving are used to enable DAB receivers to perform error correction.

The objective of the work, described in the thesis, is to improve reception techniques for DAB, DAB+, and T-DMB systems. In the thesis, two-dimensional (2D) block-based reception for differentially encoded OFDM systems is investigated. The blocks are based on the time and frequency dimension.

Commonly used DAB receivers perform non-coherent two-symbol differential detection (2SDD) with soft-decision Viterbi decoding. It is well-known that 2SDD can be improved if the detection is based on more than two received symbols as, e.g., in non-coherent multi-symbol differential detection (MSDD). For improving the performance of the demodulation procedures of DAB-like streams, demodulation based on 2D blocks of received symbols with a decomposed demodulation trellis is proposed in the thesis.

Peleg and Shamai [58] demonstrated that iterative techniques could increase the performance of the demodulation procedures of DE-QPSK streams even further. In the thesis, their approach is generalized to the 2D setting where again the decomposed demodulation trellis is used. In this way a problem connected to the small lengths of the trellises for each subcarrier is solved.

The application of these iterative decoding techniques in DAB receivers is only feasible if their complexity can be drastically reduced. A significant complexity reduction is obtained by iterating only in a dominant sub-trellis of the decomposed demodulation trellis. In this way, a real-time and bit-true DAB receiver based on iterative decoding techniques is realized.

In Chapter 2, simulation models are introduced. These models are later applied to evaluate the proposed reception methods. The Additive White Gaussian Noise (AWGN) channel model with an input power constraint and the channel model for M -level PSK are first discussed. In addition, the TU-6 (Typical Urban 6 taps) channel model defined in COST-207 [1] is introduced. This channel model is commonly used to assess the performance of DAB, DAB+, or T-DMB transmission. Finally, the basic elements of a DAB transmitter and a standard receiver are described.

In Chapter 3 of the thesis, the state of the art in non-iterative detection and decoding techniques for DE-QPSK streams with convolutional encoding is described. First, as a reference, coherent detection of DE-QPSK with soft-decision Viterbi decoding is studied. Then it is demonstrated that 2SDD of DE-QPSK with soft-decision Viterbi decoding degrades the performance. This non-coherent differential detection scheme can be improved by, for example, MSDD, which is a maximum likelihood procedure for finding a block of information symbols after having observed a block of received symbols. For large numbers of observations, the performance of MSDD approaches the performance of coherent detection of DE-QPSK. Since reference symbols (pilots) are lacking for DAB systems, detection based on observing multiple received symbols is a technique that could lead to reception improvement for DAB receivers. By applying this technique, as will be shown later, a DAB receiver approaches the performance of a receiver that performs coherent detection of $\frac{\pi}{4}$ -DE-QPSK with soft-decision Viterbi decoding.

In Chapter 4, a-posteriori symbol probabilities and log-likelihood ratios (LLRs) for coherently detected $\frac{\pi}{4}$ -DE-QPSK are studied. It is demonstrated, as an extension to the results known in the literature, that an approximation of maximum a-posteriori (MAP) symbol detection, based on selecting dominant exponentials, leads to MAP sequence detection. To improve the performance towards MAP symbol detection, a better approximation is proposed. This approximation relies on piecewise-linear fitting of the logarithm of the hyperbolic cosine and results in a performance quite close to that of MAP symbol detection. For the coded case, where the symbols are produced by convolutional encoding and Gray mapping, the LLRs are investigated. Again a simple approximation based on selecting dominant exponentials and an improved approximation relying on piecewise-linear fits, is proposed. As in the uncoded case, the improved approximation gives a performance quite close to ideal. These improved approximations are also of interest for DAB systems, as will be shown later, if 2D and trellis-based detection is considered as a reception technique.

Peleg et al. [56][57][58] and Chen et al. [18] demonstrated that iterative decoding techniques developed by Benedetto et al. [9] for serially concatenated convolutional codes lead to good results for the concatenation of convolutional and differential encoding, also referred to as Turbo-DPSK. In Chapter 5 the iterative decoding procedures corresponding to these serially concatenated codes are explained. In this chapter also parallel concatenated systems, turbo-codes, first described by Berrou et al. [11] are considered. The iterative decoding procedures for the serially concatenated codes as well as for the turbo-codes are based on modified versions of the BCJR algorithm [4]. The approach taken in Chapter 5 to explain these iterative decoding procedures, is similar to the approach Gallager

[32] followed to investigate iterative procedures for decoding low-density parity-check (LDPC) codes. This way of explaining iterative decoding procedures for the serially concatenated codes as well as for the turbo-codes does not appear in the literature. It is well-known that iterative (turbo) decoding procedures approach channel capacity, e.g., in the AWGN setting. For that reason, in Chapter 6 and Chapter 7, iterative decoding techniques for DAB-like streams are studied.

At the time that the DAB standard was proposed, the results of Berrou et al. [11] on turbo-codes were not available. As a consequence, it is not a common practice to use iterations in DAB receivers. In Chapter 6, motivated by encouraging results on Turbo-DPSK, trellis decoding and iterative techniques for DAB receivers are investigated. Specifically, the usage of 2D-blocks and trellis decomposition in decoding is considered. Each 2D-block consists of a number of adjacent subcarriers of a number of subsequent OFDM symbols. Focussing on 2D-blocks was motivated by the fact that the channel coherence-time is typically limited to a small number of OFDM symbols, and that DAB-transmissions use time-multiplexing of services, which limits the number of OFDM symbols in a codeword. Extension in the subcarrier direction is required then to get reliable phase estimates. The trellis-decomposition method allows for an estimation of the unknown channel phase, since this phase relates to sub-trellises. A-posteriori sub-trellis probabilities are determined, and these probabilities are used for weighting the a-posteriori symbol probabilities resulting from all the sub-trellises. Alternatively, a dominant sub-trellis can be determined from the a-posteriori sub-trellis probabilities and the a-posteriori symbol probabilities corresponding to this dominant sub-trellis can be used. This dominant sub-trellis approach results in a significant complexity reduction, which is the subject of Chapter 7.

In the first part of Chapter 7, complexity reduction of the inner decoder is investigated. This complexity reduction is realized by choosing, based on a-posteriori sub-trellis probabilities, in two different ways a dominant sub-trellis. In the first approach, a method is investigated that is based on finding, at the start of a new iteration, the dominant sub-trellis first and then do the forward-backward processing for demodulation only in this dominant sub-trellis. The second approach involves choosing the dominant sub-trellis only once, before starting with the iterations. In the second part of Chapter 7, an implementation of a MAP channel-phase estimator based on the second dominant sub-trellis approach is described. In addition, an implementation of a channel-gain estimator based on the received symbols within a 2D-block is discussed. Finally, a real-time and bit-true DAB-receiver is sketched. This DAB receiver operates according to the proposed 2D-block-based iterative decoding procedure within a dominant sub-trellis obtained by the second method. The performance improvements of this DAB receiver are evaluated for various numbers of iterations, block-sizes, and Doppler-frequencies.

The main conclusions can be found in Chapter 8. For the non-iterative 2D-case, investigations show that the performance of non-coherent detection based on trellis-decomposition is very close to the performance of coherent detection of DE-QPSK. The gain of 2D trellis-decomposition is modest compared to the standard 2SDD technique. Iterative 2D procedures result in a significantly larger gain. In this context, it needs to be emphasized that part of this gain comes from the 2D-processing. The dominant sub-trellis approach appears to be crucial for achieving an acceptable complexity reduction.

Samenvatting

Two-Dimensional Block-Based Reception for Differentially Encoded OFDM Systems

A study on improved reception techniques for digital audio broadcasting systems

Digital audio broadcast (DAB), DAB+ and Terrestrial-Digital Multimedia Broadcasting (T-DMB) systemen gebruiken multi-carrier modulatie (MCM). Het principe van MCM is gebaseerd op orthogonal frequency division multiplexing (OFDM), waar iedere subcarrier gemoduleerd is met $\frac{\pi}{4}$ differentially encoded quaternary phase shift keying (DE-QPSK). In DAB-systemen zijn convolutional codes en interleaving gebruikt zodat een DAB ontvanger foutcorrectie kan uitvoeren.

Het doel van het werk, beschreven in dit proefschrift, is het verbeteren van ontvangst-technieken voor DAB, DAB+ en T-DMB systemen. In het proefschrift, worden tweedimensionale (2D) blok-gebaseerde ontvang-technieken voor differentially encoded OFDM systemen onderzocht. De blokken zijn gebaseerd op de tijd en frequentie dimensie.

Normaal in gebruik zijnde DAB ontvangers passen niet-coherente two-symbol differential detection (2SDD) met soft-decision Viterbi decoding toe. Het is welbekend dat 2SDD verbeterd kan worden als de detectie gebaseerd is op meer dan twee ontvangen symbolen, zoals bijvoorbeeld in niet-coherente multi-symbol differential detection (MSDD) het geval is. Voor het verbeteren van de prestaties van de demodulatie procedures voor DAB-gebaseerde signalen, wordt in dit proefschrift demodulatie gebaseerd op 2D-blokken van ontvangen symbolen met een decomposed demodulation trellis voorgesteld.

Peleg en Shamai [58] hebben laten zien dat iteratieve technieken de prestaties van de demodulatie procedures van DE-QPSK signalen nog verder kunnen verbeteren. In het proefschrift, is hun aanpak gegeneraliseerd naar de 2D-setting waarbij opnieuw een decomposed demodulation trellis gebruikt is. Op deze manier is een probleem opgelost dat samenhang met korte lengtes van een trellis voor de subcarriers.

Toepassing van deze iteratieve decodeer-technieken in DAB ontvangers is alleen mogelijk als hun complexiteit drastisch verlaagd kan worden. Een aanzienlijke complexiteitsreductie wordt verkregen met alleen itereren in het dominante sub-trellis van de decomposed demodulation trellis. Op deze manier, werd een real-time en bit-true DAB ontvanger gebaseerd op iteratieve decodeer-technieken gerealiseerd.

In Hoofdstuk 2, worden simulatie modellen geïntroduceerd. Deze modellen worden later toegepast om de voorgestelde ontvang-methoedieken te evalueren. Eerst worden het Additive White Gaussian Noise (AWGN) kanaal model met een gelimiteerd vermogen aan de ingang en een M -level PSK kanaal model besproken. Daarbij wordt ook het TU-6 (Typical Urban 6 taps) kanaal model dat gedefinieerd is in COST-207 [1], geïntroduceerd. Dit kanaal model wordt standaard gebruikt voor het beoordelen van DAB, DAB+, of T-DMB transmissies. Tenslotte, worden de basis elementen van een DAB zender en een standaard ontvanger beschreven.

In Hoofdstuk 3 van het proefschrift wordt de huidige stand van de techniek in niet-iteratieve detectie en decodering voor DE-QPSK signalen met convolutional encoding beschreven. Eerst wordt, als referentie, ideale coherente detectie van DE-QPSK met soft-decision Viterbi decoding bestudeerd. Dan wordt aangetoond dat 2SDD van DE-QPSK met soft-decision Viterbi decoding de prestaties vermindert. Dit niet-coherente differential detection schema kan worden verbeterd met, bijvoorbeeld, MSDD. MSDD is een maximum-likelihood procedure voor het vinden van een blok van informatie symbolen nadat een blok van ontvangen symbolen is geobserveerd. Voor een groot aantal observaties benadert MSDD de prestatie van ideale coherente detectie van DE-QPSK. Omdat bij DAB systemen geen referentie symbolen (pilots) voorkomen, zou detectie gebaseerd op meerdere ontvangen symbolen tot ontvangstverbetering kunnen leiden. Door het toepassen van deze techniek, zoals we later zullen zien, kan een DAB ontvanger de prestaties benaderen van een ontvanger met ideale coherente detectie van $\frac{\pi}{4}$ -DE-QPSK met soft-decision Viterbi decoding.

In Hoofdstuk 4 worden a-posteriori symbol probabilities en log-likelihood ratios (LLRs) voor coherent gedetecteerde $\frac{\pi}{4}$ -DE-QPSK bestudeerd. Er wordt aangetoond, als een uitbreiding op de bestaande resultaten in de literatuur, dat een benadering van maximum a-posteriori (MAP) symbool detectie, gebaseerd op het selecteren van dominante exponenten, leidt tot MAP sequence detectie. Voor het verbeteren van de prestatie in vergelijking tot MAP symbool detectie, wordt een verbeterde benadering voorgesteld. Deze verbetering maakt gebruik van het piecewise-linear benaderen van de logaritme van de hyperbolische cosinus en dit resulteert in een prestatie die redelijk dicht bij die van MAP symbool detectie ligt. Voor de gecodeerde situatie, waarbij de symbolen worden gegenereerd door convolutional encoding gevolgd door Gray mapping, zijn ook de LLRs onderzocht. Hierbij worden ook weer een eenvoudige benadering, gebaseerd op het selecteren van dominante exponenten, en een verbeterde benadering, gebruik makend van piecewise-linear fits, voorgesteld. Net als in de ongecodeerde situatie geven de verbeterde benaderingen resultaten die redelijk dicht in de buurt van ideaal komen. Deze verbeterde benaderingen zijn ook interessant voor DAB systemen, zoals later zal worden aangetoond wanneer trellis-gebaseerde detectie wordt toegepast op 2D-blokken.

Peleg et al. [56][57][58] en Chen et al. [18] hebben laten zien dat de iteratieve decodeertechnieken die ontwikkeld zijn door Benedetto et al. [9] voor serieel geconcateneerde convolutional codes, eveneens leiden tot goede resultaten voor de aaneenschakeling van convolutional en differential encoding. Deze decodeer-techniek wordt Turbo-DPSK

genoemd. In Hoofdstuk 5 worden de iteratieve decodeerprocedures, die horen bij deze serieel geconcateneerde codes, besproken en hun werking verklaard. In dit hoofdstuk worden ook de parallel geconcateneerde systemen (turbo-codes), die voor het eerst door Berrou et al. [11] beschreven zijn, besproken. De iteratieve decodeerprocedures voor de serieel geconcateneerde codes en voor de turbo-codes zijn gebaseerd op gemodificeerde versies van het BCJR algoritme [4]. De aanpak in Hoofdstuk 5 voor het uitleggen van deze iteratieve decodeerprocedures, is hetzelfde als de aanpak van Gallager [32] voor het onderzoeken van iteratieve procedures voor het decoderen van low-density parity-check (LDPC) codes. Het op deze manier uitleggen van de iteratieve decodeerprocedures voor de serieel geconcateneerde codes en voor de turbo-codes lijkt niet voor te komen in de literatuur. Het is welbekend dat deze codes met hun iteratieve decodeerprocedures de kanaal-capaciteit benaderen, bijvoorbeeld in de AWGN setting. Daarom worden, in Hoofdstuk 6 en Hoofdstuk 7, iteratieve decodeer technieken voor DAB-verbonden signalen bestudeerd.

Toen de DAB standaard werd voorgesteld, waren de resultaten van Berrou et al. [11] over turbo-codes nog niet beschikbaar. Daarom is het geen normaal gebruik om iteraties toe te passen in DAB ontvangers. In Hoofdstuk 6 worden, gemotiveerd door positieve resultaten voor Turbo-DPSK, trellis decoding en iteratieve technieken voor DAB ontvangers onderzocht. Meer specifiek wordt het gebruik van 2D-blokken en trellis-decompositie bij het decoderen onderzocht. Ieder 2D-blok bestaat uit een aantal aangrenzende subcarriers van een aantal opeenvolgende OFDM symbolen. Het focussen op 2D-blokken werd ingegeven door het feit dat de coherence-time van het kanaal typisch beperkt is tot een klein aantal OFDM symbolen, maar ook omdat het DAB systeem gebruik maakt van time-multiplexing van services. Deze vorm van multiplexing beperkt het aantal OFDM symbolen in een code-woord. Uitbreiding in de subcarrier-richting is dan ook nodig om betrouwbare fase-schattingen te krijgen. De trellis-decompositie methode maakt een schatting van de onbekende kanaalfase mogelijk, omdat deze fase gerelateerd is aan sub-trellises. A-posteriori sub-trellis probabilities worden berekend en deze probabilities worden daarna gebruikt voor het wegen van de a-posteriori symbol probabilities komende van alle sub-trellises. Een andere mogelijkheid is dat er een dominant sub-trellis kan worden bepaald uit de a-posteriori sub-trellis probabilities waarna de a-posteriori symbol probabilities behorende bij dit dominant sub-trellis kunnen worden gebruikt. Deze dominante sub-trellis benadering resulteert in een significante complexiteits-reductie. Dit is het onderwerp van Hoofdstuk 7.

In het eerste gedeelte van Hoofdstuk 7 wordt complexiteits-reductie van de inner decoder onderzocht. Deze complexiteits-reductie wordt gerealiseerd door het op twee verschillende manieren kiezen van een dominant sub-trellis, gebruikmakend van a-posteriori sub-trellis probabilities. In de eerste aanpak wordt een methode onderzocht die gebaseerd is op het vinden van het dominante sub-trellis aan het begin van een nieuwe iteratie en daarna het uitvoeren van forward-backward processing voor demodulatie alleen in dit dominante sub-trellis. Bij de tweede aanpak wordt het dominante sub-trellis maar een keer bepaald en wel voordat begonnen wordt met de iteraties. In het tweede gedeelte van Hoofdstuk 7 wordt een implementatie van een MAP kanaal-fase schatter beschreven die gebaseerd is op de tweede dominante sub-trellis aanpak. Verder wordt een implementatie van een kanaal-versterkingsfactor schatter besproken die gebaseerd is op alle ontvangen

symbolen binnen een 2D-blok. Ten slotte wordt een real-time en bit-true DAB-ontvanger omschreven. Deze DAB ontvanger functioneert volgens de voorgestelde 2D-blok gebaseerde iteratieve decodeerprocedure binnen een dominant sub-trellis, dat verkregen is volgens de tweede methode. De prestatie-verbeteringen van deze DAB ontvanger zijn geëvalueerd voor verschillende aantallen iteraties, blok groottes en Doppler-frequenties.

De hoofd-conclusies zijn beschreven in Hoofdstuk 8. Voor het niet-iteratieve 2D-geval laat het onderzoek zien dat de prestaties van niet-coherente detectie gebaseerd op trellis-decompositie de prestaties van coherente detectie van DE-QPSK goed benadert. De 2D trellis-decompositie geeft hier een bescheiden verbetering op de standaard 2SDD techniek. Iteratieve 2D-procedures resulteren in een duidelijk grotere verbetering. In deze context moet worden benadrukt dat een gedeelte van deze verbetering komt van de 2D-processing. De dominante sub-trellis aanpak is cruciaal voor het bereiken van een acceptabele complexiteit-reductie.

Contents

Summary	ix
Samenvatting	xiii
List of abbreviations and symbols	xxiii
List of Tables	xxix
List of Figures	xxxii
1 Introduction	1
1.1 Outline	1
1.2 Modern communication systems	1
1.3 Thesis outline	2
1.3.1 Objective	2
1.3.2 Concise summary	2
1.3.3 Chapter contents	3
1.4 An overview of some terrestrial COFDM digital radio broadcast systems .	5
1.4.1 Digital Radio Mondiale 30 (DRM30)	5
1.4.2 Digital Radio Mondiale plus (DRM+)	7
1.4.3 HD-radio	7
1.4.4 Digital Audio Broadcasting (DAB)	7
1.4.5 Digital Audio Broadcasting plus (DAB+)	7
1.4.6 Terrestrial-Digital Multimedia Broadcasting (T-DMB)	8
1.5 DAB-family	8
1.5.1 A block-diagram for generating a DAB transmission signal	8
1.6 List of publications and patents by the Author	10
1.6.1 Journals	10
1.6.2 Conference Proceedings	10
1.6.3 Patents	11

2	Channel and DAB System Model	13
2.1	Outline	13
2.2	Channel models	13
2.2.1	Wireless transmission	14
2.2.2	Wireless OFDM transmission	14
2.2.3	AWGN model	18
2.2.4	M -level PSK model	21
2.2.5	TU-6 COST-207 model	24
2.3	DAB system model	26
2.3.1	Block-diagram	26
2.3.2	Rate-Compatible Punctured Convolutional (RCPC) codes	29
2.3.3	Time-interleaving	31
2.3.4	M -level Phase Shift keying (M -PSK)	31
2.3.5	Differentially encoded QPSK (DE-QPSK)	34
2.3.6	$\frac{\pi}{4}$ -DE-QPSK and RCPC codes with OFDM	37
2.3.7	Time-multiplexing	39
3	Detection and Decoding of DE-QPSK	41
3.1	Outline	41
3.2	Introduction	41
3.2.1	A-posteriori symbol probabilities	41
3.3	Coherent detection	44
3.3.1	Coherent symbol-by-symbol detection	44
3.3.2	Soft-decision bit metrics	47
3.3.3	Concluding remarks on coherent detection	50
3.4	Non-coherent detection	50
3.4.1	Two-symbol differential detection (2SDD)	50
3.4.2	Soft-decision bit metrics	51
3.4.3	Multi-symbol differential detection (MSDD)	51
3.4.4	Soft-decision bit metrics	53
3.4.5	Concluding remarks on non-coherent differential detection	53
4	A-Posteriori Symbol Probabilities and Log-Likelihood Ratios for Coherently Detected $\frac{\pi}{4}$-DE-QPSK	55
4.1	Outline	55
4.2	A-Posteriori probabilities	56
4.2.1	First approximation	57
4.2.2	Second approximation	58
4.2.3	Simulations	58
4.3	Log-likelihood ratios	60
4.3.1	Third approximation	60
4.3.2	Fourth approximation	60
4.3.3	Simulations	61
4.4	Conclusions	62

5	The Shannon Limit and Some Codes Approaching It	63
5.1	Outline	63
5.2	Some information theory first	64
5.2.1	Capacity of the AWGN Channel	64
5.2.2	Binary channel inputs	64
5.2.3	From sequences to waveforms	65
5.3	Uncoded binary transmission	65
5.4	Linear codes incur no loss	67
5.5	Convolutional codes	67
5.6	Channel, likelihoods	69
5.7	The BCJR algorithm	70
5.7.1	Decoding algorithms	70
5.7.2	Probabilistic structure	70
5.7.3	Numerator and denominator	71
5.7.4	Splitting up the numerator	72
5.7.5	Alphas and betas and their recursions	72
5.8	Performance convolutional code	73
5.9	Turbo code, encoder structure	73
5.9.1	Introduction	73
5.9.2	Making our code systematic	75
5.9.3	Parallel concatenation of two systematic codes, Interleaving	75
5.10	Turbo decoding procedure	76
5.10.1	BCJR algorithm for a Systematic Code	76
5.10.2	One systematic symbol in two codes	78
5.10.3	All systematic symbols of a vertical code are in different horizontal codes	79
5.10.4	An additional layer of vertical codes	80
5.10.5	Decoding a turbo code	80
5.11	Terminating two trellises	83
5.12	Performance turbo code	83
5.13	Serial concatenation, encoder structure	83
5.14	Serial case, decoding procedure	85
5.14.1	Two connected codes again	86
5.14.2	Two perspectives	87
5.14.3	Decoding procedure	87
5.15	Performance serially concatenated code	88
5.16	Conclusion	89
6	Two-Dimensional Iterative Processing for DAB Receivers Based on Trellis Decomposition	91
6.1	Outline	91
6.1.1	Problem description	92
6.2	Description of a digital audio broadcasting (DAB) system	93
6.2.1	Overview	94
6.2.2	Convolutional coding and interleaving	94

6.2.3	Differential modulation in each subcarrier	95
6.2.4	OFDM in DAB	95
6.2.5	Incoherent reception, channel gain known to receiver	97
6.2.6	Equivalence between DE-QPSK and $\pi/4$ -DE-QPSK	97
6.3	Detection and decoding, single-carrier case, non-iterative	98
6.3.1	Trellis representation, sub-trellises, decomposition	98
6.3.2	Forward-backward algorithm, sub-trellises	99
6.3.3	Combination	102
6.3.4	Dominant sub-trellis approach	104
6.3.5	Simulations	104
6.3.6	Some conclusions	106
6.4	Detection and decoding, multi-carrier case non-iterative	107
6.4.1	Demodulation procedures	107
6.4.2	Simulations	109
6.4.3	Conclusion non-iterative decoding	109
6.5	Detection and decoding, single-carrier case, iterative	111
6.5.1	Serial concatenation	111
6.5.2	Peleg approach	111
6.5.3	Trellis decomposition	112
6.5.4	Dominant sub-trellis approaches	113
6.5.5	Simulations	114
6.6	Detection and decoding, multi-carrier case, iterative	118
6.6.1	Trellis decomposition	118
6.6.2	Dominant sub-trellis approach	119
6.6.3	Simulations	119
6.7	Performance for TU-6 channel model	119
6.8	Conclusions	122
7	A Practical DAB System with 2D-Block-Based Iterative Decoding	125
7.1	Outline	125
7.2	Complexity reduction for non-coherent iteratively detected DE-QPSK based on trellis-decomposition	126
7.2.1	Abstract	126
7.2.2	Introduction	126
7.2.3	Trellis-decomposition for DAB receivers	127
7.2.4	A-posteriori symbol probabilities	128
7.2.5	Conclusions	130
7.3	MAP channel-phase estimator	133
7.3.1	Introduction	133
7.3.2	Complexity reduction	134
7.3.3	Simulations	135
7.4	Channel-gain estimator	138
7.4.1	Introduction	138

7.4.2	Complexity reduction	139
7.4.3	Simulations	139
7.5	A realization of the proposed DAB receiver	139
7.5.1	Block-diagram	139
7.5.2	Bench-test results of the proposed DAB receiver	143
7.6	Conclusions	146
8	Conclusions and Recommendations	147
8.1	Outline	147
8.2	Conclusions	148
8.2.1	Introductory chapters	148
8.2.2	A-posteriori symbol probabilities and log-likelihood ratios for coherently detected $\frac{\pi}{4}$ -DE-QPSK	148
8.2.3	The Shannon limit and some codes approaching it	148
8.2.4	Two-dimensional iterative processing for DAB receivers based on trellis-decomposition	149
8.2.5	A practical DAB system with 2D-block-based iterative decoding	150
8.3	Recommendations	150
8.3.1	Non-ideal conditions and more severe channel distortions	151
8.3.2	Implementation and (specific) optimization	151
8.3.3	Field-testing	151
8.3.4	Other systems	151
A	A-Posteriori Symbol Probabilities and LLRs for Coherent Detection.	153
A.1	A-posteriori symbol probabilities for coherent detection	153
A.2	LLRs for coherent detection	154
A.3	Max-log-MAP approximation of the LLRs	155
B	A-Posteriori Symbol Probabilities and LLRs for 2SDD.	157
B.1	A-posteriori symbol probabilities for non-coherent 2SDD	157
B.2	LLRs for 2SDD	158
	References	161
	Acknowledgments	167
	Curriculum Vitae	169

List of abbreviations and symbols

Abbreviation	Description
2D	Two-Dimensional
2SDD	Two-Symbol Differential Detection
AAC	Advanced Audio Coding
AGC	Automatic Gain Control
AM	Amplitude-Modulation
AP	A-Posteriori
APP	A-Posteriori Probability
AVC	Advanced Video Coding
AWGN	Additive White Gaussian Noise
BCH	Bose, Chaudhuri, Hocquenghem
BCJR	Bahl, Cocke, Jelinek, Raviv
BER	Bit Error Rate
BIFS	BIrary Format for Scene
BSAC	Bit Sliced Arithmetic Coding
CC	Convolutional Code
CELP	Code Excited Linear Prediction
COFDM	Coded Orthogonal Frequency Division Multiplexing
COST	Cooperation in the field of Scientific and Technical research
DAB	Digital Audio Broadcasting
DE-QPSK	Differentially Encoded Quaternary Phase Shift Keying
DFT	Discrete Fourier Transform
DRM	Digital Radio Mondiale
DQPSK	Differential Quaternary Phase Shift Keying
FFT	Fast Fourier Transform
FM	Frequency-Modulation
HE-AAC	High-Efficiency Advanced Audio Coding
HVXC	Harmonic Vector eXcitation Coding
IC	Integrated Circuit
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
ISI	Inter Symbol Interference

Abbreviation	Description
ISDB-T	Integrated Services Digital Broadcasting–Terrestrial
iid	independent and identically distributed
iud	independent and uniformly distributed
LDPC	Low-Density Parity-Check
LLR	Log-Likelihood Ratio
LPC	Linear Phase Correction
<i>M</i> -PSK	<i>M</i> -level Phase Shift Keying
MAP	Maximum A-Posteriori
MCM	Multi-Carrier Modulation
ML	Maximum Likelihood
MLSE	Maximum Likelihood Sequence Estimator
MPEG	Moving Picture Experts Group
MSDD	Multi-Symbol Differential Detection
NLOS	Non-Line Of Sight
OFDM	Orthogonal Frequency Division Multiplexing
PEP	Pairwise Error Probability
PHY	PHYSical layer
PMF	Probability Mass Function
PPM	Pulse Position Modulation
PS	Pseudo Stereo
PSK	Phase Shift Keying
QPSK	Quaternary Phase Shift Keying
RCPC	Rate-Compatible Punctured Convolutional
RS	Reed-Solomon
SBR	Spectral Band Replication
SCCC	Serial Concatenation of Convolutional Codes
SER	Symbol Error Rate
SNR	Signal-to-Noise Ratio
SOVA	Soft-Output Viterbi Algorithm
T-DMB	Terrestrial-Digital Multimedia Broadcasting
TU	Typical Urban
UB	Union Bound
VHF	Very High Frequency

The symbols printed in the table below in roman (X) are scalars, in bold face (**X**) or (X) are vectors or sequences.

Symbol	Description
\mathcal{A}	QPSK set: $\{e^{jp\pi/2}, p = 0, 1, 2, 3\}$
\mathcal{B}	$\frac{\pi}{4}$ -offset QPSK set: $\{e^{jp\pi/2+\pi/4}, p = 0, 1, 2, 3\}$
B	B-points (IFFT)
B_c	Channel coherence bandwidth
B_d	Doppler spread
BER	Average probability of a bit error
C	Channel capacity in bit per second (passband)
\overline{C}	Channel capacity in bit per complex dimension
C_b	Channel capacity in bit per second (baseband)
\overline{C}_b	Channel capacity in bit per transmission (baseband)
\overline{C}_{M-PSK}	Channel capacity of M -level PSK channel
$E[\cdot]$	Expectation
E_b	Average bit energy
$\frac{E_b}{N_0}$	Ratio between average bit energy and noise spectral density
E_s	Average symbol energy
$\frac{E_s}{N_0}$	Ratio between average symbol energy and noise spectral density
$\mathcal{H}(\cdot)$	Entropy
$\Im\{\cdot\}$	Imaginary part of a complex number
$I(s; r)$	Mutual information between variables r and s
K	Number of OFDM-subcarriers per OFDM-symbol
K	Constraint-length convolutional encoder
$K_s(\cdot)$	State likelihood sub-trellis
L	Number of OFDM-symbols per OFDM-frame
L	Number of iterations
M	Number of OFDM-subcarriers per 2D-block
M	Cardinality of input set (constellation) \mathcal{S}
N	Number of observations (e.g., OFDM-symbols per 2D-block)
N_b	Number of information bits per codeword
N_c	Number of coded bits per codeword
N_0	Noise spectral density (one-sided)
P	Average power
PI	Puncturing-Index RCPC codes
$\Pr\{\cdot\}$	Probability mass function
Q	Discrete (Quantized) levels
$Q(\cdot)$	Gaussian tail function
R	Code rate in bits per second
$\Re\{\cdot\}$	Real part of complex number
R_c	Code-rate convolutional encoder
\mathcal{S}	Input signal set (constellation)

Symbol	Description
$S(\tau; \lambda)$	Scattering function
$S_c(\tau)$	Multi-path intensity profile or delay power spectrum
$S_c(\lambda)$	Doppler power spectrum
SE_R	Average probability of a symbol error
SNR	Signal-to-noise ratio
$T(D, N, J)$	Transfer-function convolutional encoder
T	Sampling (basic-pulse) duration
T_c	Channel coherence time
T_{cp}	Cyclic-prefix duration
T_F	(OFDM-)frame duration
T_s	(OFDM-)symbol duration
T_u	Inverse of the OFDM-subcarrier spacing
\mathcal{T}	Demodulator trellis
\mathcal{T}_s	Sub-trellis
W	Passband transmission bandwidth
W_b	Baseband single-sided bandwidth
\mathcal{X}_e	$\frac{\pi}{4}$ -DE-QPSK set: $\{e^{jl\pi/2}, l = 0, 1, 2, 3\}$
\mathcal{X}_o	$\frac{\pi}{4}$ -DE-QPSK set: $\{e^{jl\pi/2+\pi/4}, l = 0, 1, 2, 3\}$
\mathcal{Z}	States of demodulator trellis $\{e^{jl\pi/16}, l = 0, 1, \dots, 31\}$
\mathcal{Z}_s	States sub-trellis $\{e^{jl\pi/16}, l = s + 8p, p = 0, 1, 2, 3, s = 0, 1, \dots, 7\}$
a_n	QPSK transmitted symbol
\tilde{a}_n	Approximated QPSK symbol
\mathbf{a}	QPSK transmitted sequence
\hat{a}	Estimated QPSK symbol
a_d	Number of paths with Hamming weight d
b	(offset) QPSK transmitted symbol
\mathbf{b}	(offset) QPSK transmitted sequence
\hat{b}	Estimated (offset) QPSK symbol
c_i	Coded bit
\underline{c}	Sequence of coded bits
d_{free}	Minimum free distance of convolutional encoder
d_i	Information bit
\hat{d}_i	Estimated information bit
\underline{d}	Sequence of information bits
f_d	Doppler frequency
f_m	m^{th} frequency bin of OFDM-symbol
g_i	Generator polynomials convolutional encoder
h	Channel coefficient (Rayleigh-fading realization)
$ h $	Channel gain
$\widehat{ h }$	Estimated channel gain
j	$\sqrt{-1}$
k	Number of shift-registers convolutional encoder

Symbol	Description
m	Number of bits per modulation symbol
m	OFDM-subcarrier index
$m(\cdot)$	Symbol-metric
n	Number of polynomials (output bits) convolutional encoder
n_n	Circular symmetric complex white Gaussian noise
$p_x(X)$	Probability density function (pdf) of stochastic variable X
r	Code-rate in bits per complex dimension (Spectral efficiency)
r_n	$\frac{\pi}{4}$ -DE-QPSK received symbol
\mathbf{r}	$\frac{\pi}{4}$ -DE-QPSK received sequence
$\tilde{\mathbf{r}}$	Received OFDM-symbol without cyclic-prefix
\tilde{r}	Element of received OFDM-symbol without cyclic-prefix
$\bar{\mathbf{r}}$	Received OFDM-symbol with cyclic-prefix
s	sub-trellis index $s = 0, 1, \dots, 7$
\hat{s}	Estimated sub-trellis
$s_F(t)$	Transmitted OFDM baseband frame signal
$s(t)$	Transmitted OFDM (DAB) signal
s_n	$\frac{\pi}{4}$ -DE-QPSK transmitted symbol
\mathbf{s}	$\frac{\pi}{4}$ -DE-QPSK transmitted sequence
$\tilde{\mathbf{s}}$	Transmitted OFDM-symbol without cyclic-prefix
\tilde{s}	Element of transmitted OFDM-symbol without cyclic-prefix
$\bar{\mathbf{s}}$	Transmitted OFDM-symbol with cyclic-prefix
w	Circularly symmetric complex white Gaussian noise
$\bar{\mathbf{w}}$	Circularly symmetric complex white Gaussian noise sequence
x_n	Transmitted DE-QPSK symbol
\mathbf{x}	Transmitted DE-QPSK sequence
$\bar{\mathbf{x}}$	Transmitted OFDM-symbol with cyclic-prefix and DE-QPSK
y_n	Received DE-QPSK symbol
\mathbf{y}	Received DE-QPSK sequence
$\bar{\mathbf{y}}$	Received OFDM-symbol with cyclic-prefix and DE-QPSK
$\Phi(f)$	Power spectrum
Δf	OFDM-subcarrier spacing
$\Delta\phi_n$	(Information) phase QPSK symbol
$\alpha_n(\cdot)$	Forward (recursive) probability density function
$\beta_n(\cdot)$	Backward (recursive) probability density function
β_d	Number of information bit errors on all incorrect paths with Hamming weight d
$\gamma_n(\cdot)$	Channel likelihood
λ_j	Bit-metrics based on LLRs
η	Decision statistic
σ^2	Noise-variance per component $\sigma^2 = \frac{N_0}{2}$
ϕ	Channel phase
$\hat{\phi}$	Estimated channel phase
ϕ_n	Phase transmitted $\frac{\pi}{4}$ -DE-QPSK symbol

Symbol	Description
ϕ_s	Phase sub-trellis s
τ_m	Multi-path spread

List of Tables

2.1	Required SNR for channel capacity \bar{C}	19
2.2	COST-207 TU-6 channel model	25
2.3	Required SNR for $\text{BER}_{M\text{-PSK}} = 1 \cdot 10^{-4}$	33

List of Figures

1.1	The three fundamental parts of a communication system.	2
1.2	The relation between the chapters of the thesis.	6
1.3	A block-diagram for generating a DAB transmission signal.	9
2.1	Channel capacity \bar{C} in bit per complex dimension.	20
2.2	Channel capacity $C_{M\text{-PSK}}$ for $M = 4, 16$ and 64	23
2.3	COST-207 TU-6 channel model.	26
2.4	Block-diagram of the DAB system model.	27
2.5	$R_c = 1/2$ convolutional encoder with $g_0 = 133_8$ and $g_1 = 171_8$	30
2.6	The $\text{BER}_{M\text{-PSK}}$ for $M = 4, 16, 64$	33
2.7	BER for AWGN of non-coherent 2SDD of uncoded DE-QPSK and coherent detection of uncoded DE-QPSK and QPSK.	36
2.8	Union-bound for AWGN of non-coherent 2SDD of DE-QPSK given in (2.55) and BER for AWGN of non-coherent 2SDD of DE-QPSK and coded DE-QPSK.	38
2.9	DAB convolutional encoder, interleaver, differential encoders, and multi-carrier modulator.	38
2.10	Three services mapped onto consecutive OFDM symbols.	39
3.1	BER for AWGN of coherent detection and non-coherent 2SDD of coded DE-QPSK.	48
3.2	Coherent detection and the max-log-MAP approximation of coded DE-QPSK.	49
3.3	MSDD for coded DE-QPSK.	54
4.1	SER-performance for ideal and approximated versions of the APs.	59
4.2	BER-performance for ideal and approximated versions of the LLRs.	61
5.1	Additive white Gaussian noise channel.	64
5.2	Bit-error probability for uncoded binary transmission.	66
5.3	Convolutional encoder for a $(7, 5)$ -code.	68
5.4	Trellis representation of our example code for $K = 4$	68
5.5	Probabilistic structure corresponding to a general finite-state encoder.	71

5.6	Bit-error rate of our example (7,5)-code decoded with the BCJR procedure based on soft decisions.	74
5.7	Transforming our example code into a systematic code.	74
5.8	The turbo encoder structure as proposed by Berrou, Glavieux, and Thitimajshima.	75
5.9	Probabilistic structure corresponding to a systematic finite-state encoder.	77
5.10	A symbol b_{ij} contained in two different systematic codes.	78
5.11	All symbols in the vertical code belong to different horizontal codes.	80
5.12	Three layers of codes.	81
5.13	Turbo-decoder architecture. The decoder outputs a-posteriori probabilities of the message digits.	82
5.14	Bit-error rate of our turbo code. Also the corresponding Shannon limit is shown.	84
5.15	A systematic (recursive) rate-2/3 convolutional encoder.	85
5.16	Two serially concatenated convolutional codes and a bit-pair interleaver.	85
5.17	A code-bit pair which is both output of the outer encoder and input for the inner encoder.	86
5.18	Decoder architecture for a serially concatenated code.	88
5.19	Bit-error rate of our serially concatenated code.	89
6.1	DAB convolutional encoder, interleaver, differential encoders, and multi-carrier modulator with an IFFT.	94
6.2	Three services mapped onto consecutive OFDM symbols. Note that there is overlap between the service since differential modulation is used.	95
6.3	A 2D-block of symbols out of an OFDM stream. We are interested in M adjacent sequences of $N + 1$ subsequent symbols, where each such sequence corresponds to one of M adjacent sub-carriers.	96
6.4	Trellis representation of the states z_0, z_1, \dots, z_N and the differentially encoded symbols a_1, a_2, \dots, a_N in the incoherent case. An edge between two subsequent states indicates that a transition between these states is possible. Note that the trellis can be decomposed into eight unconnected sub-trellises.	100
6.5	Bit-error performance for LLRs computed as in (6.25), i.e., ideal LLRs, for different trellis-lengths.	105
6.6	Bit-error performance for LLRs computed as in (6.25), i.e., ideal LLRs, and approximated LLRs computed as in (6.26), for different trellis-lengths.	106
6.7	Bit-error performance with ideal LLRs for a decomposed multi-carrier trellis for different values of M and N but with a fixed block-size of $M(N + 1) = 16$	110
6.8	Structure of the receiver.	111
6.9	Bit-error performance of the Peleg method for trellis length $N \rightarrow \infty$ and up to $L = 5$ iterations.	115
6.10	Bit-error performance for the Peleg method for different values of N and $L = 5$ iterations.	116

6.11	Bit-error performance with dominant sub-trellis approach for different values N and $L = 5$ iterations.	117
6.12	Bit-error performance for the multi-carrier case for different values M , where $N + 1 = 4$ and for $L = 5$ iterations.	120
6.13	Bit-error performance with dominant sub-trellis approach for different values N , $M = 8$ and $L = 5$ iterations.	121
6.14	Bit-error performance for the TU-6 COST-207 channel with the solid-lines for $f_d T_u = 0.01$, the dashed-lines for $f_d T_u = 0.02$, $N + 1 = (4, 18)$, $M = (8, 1)$ and $L = 5$ iterations.	123
7.1	BER-performance for the TU-6 COST-207 channel with a perfectly known channel gain $ h $ for the weighting, the 1 st and, the 2 nd method.	131
7.2	BER-performance for the TU-6 COST-207 channel with an estimated channel gain $ \widehat{h} $ for the weighting, the 1 st and, the 2 nd method.	132
7.3	BER-performance with perfect knowledge and estimates of the channel phase.	136
7.4	Detail of the BER-performance with perfect knowledge and estimates of the channel phase.	137
7.5	BER-performance with perfect knowledge and estimates of the channel gain for different values of $M(N + 1)$	140
7.6	Detail of BER-performance with perfect knowledge and estimates of the channel gain for different values of $M(N + 1)$	141
7.7	Block-diagram of the proposed DAB receiver.	142
7.8	Real-time bench-test results of the performance improvement for DAB-transmissions in Mode-I on the AWGN channel.	144
7.9	Real-time bench-test results of the iterative decoding gain for DAB-transmissions in Mode-I on the TU-6 channel.	145

Chapter 1

Introduction

1.1 Outline

In this first chapter we start with a small introduction to modern communication systems. After this introduction we give, in Section 1.3, an overview of the thesis at hand. Then, in Section 1.4, we briefly describe a number of different terrestrial Coded Orthogonal Frequency Division Multiplexing (COFDM) digital radio broadcast systems. In Section 1.5 we will focus on the physical layer (PHY) of the terrestrial Digital Audio Broadcasting (DAB) system as described in [25]. Finally, in Section 1.6, we list the publications and patents that resulted from the work related to this thesis.

1.2 Modern communication systems

Modern communication systems are based on a transformation of the original message into another message, which is then sent over a medium or stored on a medium, such that the original message can be retrieved reliably or up to some fidelity level. There is a wide variety of communication systems, e.g., terrestrial communication systems, satellite communication systems, cable communication systems, storage communication systems, and so on. All these communication systems can be split up in three fundamental parts: (i) the transmitter, (ii) the transmission medium (physical channel), and (iii) the receiver [10, Sec. 4.1, p. 153]. These three parts are shown in Figure 1.1. The receiver tries to reproduce as well as possible the original message that is sent over the channel. In a practical situation the behavior of the channel can be quite complex due to the random nature of different effects such as noise, mobility, and interference. The behavior of the channel is often modeled as a closed form expression that describes, as accurately as possible, the (statistical) relation between its input and output. But one has to keep in mind that a model only approximates reality, as is nicely stated in [43, Ch. 1, p. 7]: “the real world is far too complex to be embraced in a single theory”.

To be robust against the unavoidable errors that occur during transmission, appropriate signal processing algorithms are tuned to the physical properties of the channel. Some of

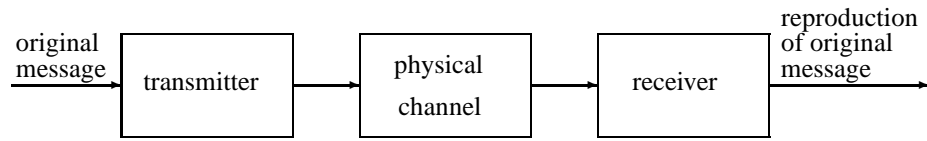


Figure 1.1: *The three fundamental parts of a communication system (i) the transmitter, (ii) the physical channel (the transmission medium), and (iii) the receiver.*

these signal processing algorithms will add redundancy to the original message. The process of adding such redundancy to the original message is commonly referred to as coding. This redundancy is used by the receiver to recover from some errors in the original message. It is now the “communication engineering art” to find a way of not adding too much redundancy, on the one hand, and still being able to recover the original message as accurately as possible, on the other hand. The different ways of accomplishing a trade-off between redundancy and the ability of correcting errors, under the constraints imposed by the physical properties of the channel, is reflected by the development of numerous different transmitter and receiver systems. In this thesis we will focus on improving reception techniques for terrestrial DAB systems as described in [25].

DAB systems use multi-carrier modulation (MCM). MCM for DAB systems is based on Orthogonal Frequency Division Multiplexing (OFDM), for which every subcarrier is modulated by $\frac{\pi}{4}$ -Differentially Encoded Quaternary Phase Shift Keying (DE-QPSK). In DAB systems convolutional codes and interleaving are used to enable DAB receivers to perform error correction.

1.3 Thesis outline

1.3.1 Objective

DAB, DAB+, and Terrestrial-Digital Multimedia Broadcasting (T-DMB) broadcasting systems comprise a combination of convolutional coding, interleaving, and $\frac{\pi}{4}$ -DE-QPSK with OFDM. The $\frac{\pi}{4}$ -DE-QPSK is performed on the QPSK symbols corresponding to the same subcarrier in consecutive OFDM-symbols. Therefore there are as many $\frac{\pi}{4}$ -DE-QPSK streams as there are (active) subcarriers. The objective of the work, described in the thesis, is to improve reception techniques for DAB, DAB+, and T-DMB systems. In particular, two-dimensional (2D) block-based reception for differentially encoded OFDM systems is investigated. The blocks are based on both the time and frequency dimension.

1.3.2 Concise summary

Commonly used DAB receivers perform non-coherent, i.e., no attempt is made to estimate the phase of the carrier, two-symbol differential detection (2SDD) with soft-decision Viterbi decoding. It is well-known that 2SDD can be improved if the detection is based

on more than two received symbols as, e.g., in non-coherent multi-symbol differential detection (MSDD). For improving the performance of the demodulation procedures of DAB-like streams, demodulation based on 2D blocks of received symbols with a decomposed demodulation trellis is proposed in the thesis.

Peleg and Shamai [58] demonstrated that iterative techniques can increase the performance of the demodulation procedures of DE-QPSK streams beyond multi-symbol detection. In the thesis, their approach is generalized to the 2D setting where again the decomposed demodulation trellis is used. In this way a problem connected to the small lengths of the trellises for each subcarrier is solved.

The application of these iterative decoding techniques in DAB receivers is only feasible if their complexity can be drastically reduced. A significant complexity reduction is obtained by iterating only in a dominant sub-trellis of the decomposed demodulation trellis. In this way, a real-time and bit-true DAB receiver based on iterative decoding techniques is realized.

1.3.3 Chapter contents

In Chapter 2, simulation models are introduced. These models are later applied to evaluate the proposed reception methods. The Additive White Gaussian Noise (AWGN) channel model with an input power constraint and the channel model for M -level PSK are first discussed. In addition, the TU-6 (Typical Urban 6 taps) channel model defined in COST-207 [1] is introduced. This channel model is commonly used to assess the performance of DAB, DAB+, or T-DMB systems. Finally, the basic elements of a DAB transmitter and a standard receiver are described.

In Chapter 3, the state-of-the-art in non-iterative detection and decoding techniques for DE-QPSK streams with convolutional encoding is described. First, as a reference, coherent detection of DE-QPSK with soft-decision Viterbi decoding is studied. Then it is demonstrated that non-coherent 2SDD of DE-QPSK with soft-decision Viterbi decoding degrades the performance compared to coherent-detection of DE-QPSK with soft-decision Viterbi decoding. This non-coherent differential detection scheme can be improved by, for example, MSDD, which is a maximum-likelihood procedure for finding a block of information symbols after having observed a block of received symbols. For large numbers of observations, the performance of MSDD approaches the performance of coherent detection of DE-QPSK. Since reference symbols (pilots) are lacking for DAB systems, detection based on observing multiple received symbols is a technique that could lead to reception improvement for DAB receivers. By applying this technique, as will be shown later, a DAB receiver approaches the performance of a receiver that performs coherent detection of $\frac{\pi}{4}$ -DE-QPSK with soft-decision Viterbi decoding.

In Chapter 4, a-posteriori symbol probabilities and log-likelihood ratios (LLRs) for coherently detected $\frac{\pi}{4}$ -DE-QPSK are studied. It is demonstrated, as an extension to the results known in the literature, that an approximation of maximum a-posteriori (MAP) symbol detection, based on selecting dominant exponentials, leads to MAP sequence detection. To improve the performance towards MAP symbol detection, a better approximation is

proposed. This approximation relies on piecewise-linear fitting of the logarithm of the hyperbolic cosine and results in a performance quite close to that of MAP symbol detection. For the coded case, where the symbols are produced by convolutional encoding and Gray mapping, the LLRs are investigated. Again a simple approximation based on selecting dominant exponentials and an improved approximation relying on piecewise-linear fits, is proposed. As in the uncoded case, the improved approximation gives a performance quite close to ideal. These improved approximations are also of interest for DAB systems, as will be shown later, if 2D and trellis-based detection is considered as a reception technique.

Peleg et al. [56][57][58] and Chen et al. [18] demonstrated that iterative decoding techniques developed by Benedetto et al. [9] for serially concatenated convolutional codes lead to good results for the concatenation of convolutional and differential encoding, also referred to as Turbo-DPSK. In Chapter 5 the iterative decoding procedures corresponding to these serially concatenated codes are explained. In this chapter also parallel concatenated systems, i.e., turbo-codes, first described by Berrou et al. [11] are considered. The iterative decoding procedures for the serially concatenated codes as well as for the turbo-codes are based on modified versions of the BCJR algorithm [4]. The approach taken in Chapter 5 to explain these iterative decoding procedures, is similar to the approach Gallager [32] followed to investigate iterative procedures for decoding low-density parity-check (LDPC) codes. This way of explaining iterative decoding procedures for the serially concatenated codes as well as for the turbo-codes does not appear in the literature. It is well-known that iterative (turbo) decoding procedures approach channel capacity, e.g., in the AWGN setting. For that reason, in Chapter 6 and Chapter 7, iterative decoding techniques for DAB-like streams are studied.

At the time that the DAB standard was proposed, the results of Berrou et al. [11] on turbo-codes were not available. As a consequence, it is not a common practice to use iterations in DAB receivers. In Chapter 6, motivated by encouraging results on Turbo-DPSK, trellis decoding and iterative techniques for DAB receivers are investigated. Specifically, the usage of 2D-blocks and trellis decomposition in decoding is considered. Each 2D-block consists of a number of adjacent subcarriers of a number of subsequent OFDM symbols. Focussing on 2D-blocks was motivated by the fact that the channel coherence-time is typically limited to a small number of OFDM symbols, and that DAB-transmissions use time-multiplexing of services, which limits the number of OFDM symbols in a codeword. Extension in the subcarrier direction is required then to get reliable phase estimates. The trellis-decomposition method allows for an estimation of the unknown channel phase, since this phase relates to sub-trellises. A-posteriori sub-trellis probabilities are determined, and these probabilities are used for weighting the a-posteriori symbol probabilities resulting from all the sub-trellises. Alternatively, a dominant sub-trellis can be determined from the a-posteriori sub-trellis probabilities and the a-posteriori symbol probabilities corresponding to this dominant sub-trellis can be used. This dominant sub-trellis approach results in a significant complexity reduction, which is the subject of Chapter 7.

In the first part of Chapter 7, complexity reduction of the inner decoder is investigated.

This complexity reduction is realized by choosing, based on a-posteriori sub-trellis probabilities, in two different ways a dominant sub-trellis. In the first approach, a method is investigated that is based on finding, at the start of a new iteration, the dominant sub-trellis first and then do the forward-backward processing for demodulation only in this dominant sub-trellis. The second approach involves choosing the dominant sub-trellis only once, before starting with the iterations. In the second part of Chapter 7, an implementation of a MAP channel-phase estimator based on the second dominant sub-trellis approach is described. In addition, an implementation of a channel-gain estimator based on the received symbols within a 2D-block is discussed. Finally, a real-time and bit-true DAB-receiver is sketched. This DAB receiver operates according to the proposed 2D-block-based iterative decoding procedure within a dominant sub-trellis obtained by the second method. The performance improvements of this DAB receiver are evaluated for various numbers of iterations, block-sizes, and Doppler-frequencies.

The main conclusions can be found in Chapter 8. For the non-iterative 2D-case, investigations show that the performance of non-coherent detection based on trellis-decomposition is very close to the performance of coherent detection of DE-QPSK. The gain of 2D trellis-decomposition is modest compared to the standard 2SDD technique. Iterative 2D procedures result in a significantly larger gain. In this context, it needs to be emphasized that part of this gain comes from the 2D-processing. The dominant sub-trellis approach appears to be crucial for achieving an acceptable complexity reduction.

For clearness, the relation between the chapters of the thesis is shown by Figure 1.2.

In the next section we will give an overview of a number of different terrestrial COFDM digital radio broadcast systems.

1.4 An overview of some terrestrial COFDM digital radio broadcast systems

1.4.1 Digital Radio Mondiale 30 (DRM30)

Digital Radio Mondiale 30 (DRM30)(actually mode A-D of [26]) is a terrestrial COFDM digital radio broadcast system for short-wave, medium-wave, and long-wave radio for frequencies below 30 MHz. It delivers near-Frequency-Modulation (FM) sound quality and the ease-of-use that comes with digital transmission. The improvement over Amplitude-Modulation (AM) is immediately noticeable. The DRM system uses COFDM where the data produced from the digitally encoded audio and associated signaling are distributed, for transmission, across the large number of closely spaced carriers. Time and frequency interleaving is applied to mitigate fading from multi-path disturbances. Various parameters of the OFDM and the convolutional codes can be adapted to allow DRM to operate successfully in many different propagation environments. The DRM system uses Moving Picture Experts Group (MPEG) 4 High Efficiency Advanced Audio Coding (HE-AAC+ v2) to provide high audio quality at low data rates. HE-AAC+ v2 is a superset of the AAC core audio compression. This superset structure permits three options depending on the required bit rate: (1) plain AAC for high bit rates, (2) AAC and Spectral Band Replication

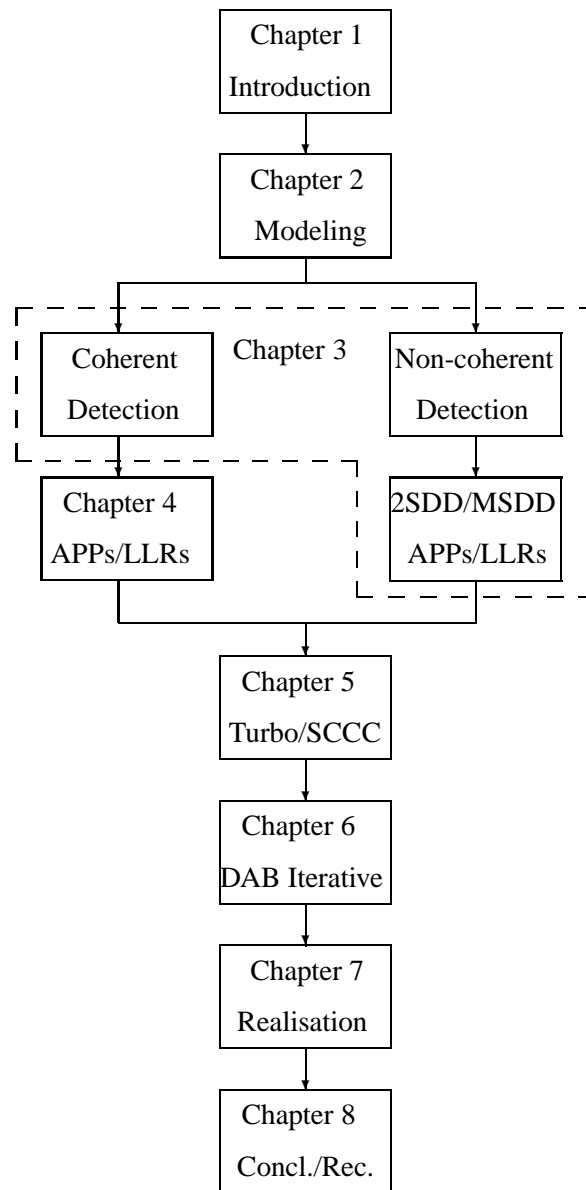


Figure 1.2: *The relation between the chapters of the thesis.*

(SBR), i.e., HE-AAC for medium bit rates, or (3) AAC, SBR, and Pseudo Stereo (PS), i.e., HE-AAC+ v2 for low bit rates. In addition, Code Excited Linear Prediction (CELP) and Harmonic Vector eXcitation Coding (HVXC) speech compression algorithms provide speech-only programming at even lower data rates, see Chapter 5 in [26].

1.4.2 Digital Radio Mondiale plus (DRM+)

Digital Radio Mondiale plus (DRM+) (actually mode E of [26]) is a standard for terrestrial COFDM digital radio broadcast transmissions in Band I and Band II (FM-Band), in principle below Band III, starting at approximately 174 MHz. Also here, OFDM provides a highly efficient usage of spectrum and offers undisturbed mobile reception with no interference. With its bandwidth of 95 kHz, DRM+ fits into the 100 kHz FM pattern used in Europe and can, therefore, be transmitted within the respective gaps in Band II. The maximum effective data rate is up to 186 kbit/s per multiplex. Using the MPEG 4 HE-AAC+ v2 audio compression permits the integration of up to 4 different audio streams including additional data services or even highly compressed video streams on one DRM+ Multiplex.

1.4.3 HD-radio

HD-radio is a method of broadcasting digital radio signals on the same channel and at the same time as the conventional AM or FM signal. Since HD-radio is a closed intellectual-property system, it is only briefly described in this thesis. HD-radio is also a COFDM system that creates a set of digital sidebands on each side of the AM/FM signal. The combined AM/FM and digital radio signal fits in the same spectral mask as is specified for conventional AM/FM. The system supports, for growth towards eventual full utilization of the spectrum by the digital signal, transmissions in three modes: Hybrid, Extended Hybrid, and Full Digital.

1.4.4 Digital Audio Broadcasting (DAB)

DAB was designed in the late 1980s with five original objectives: (1) provide CD-quality radio, (2) provide better in-car reception quality than with FM, (3) use the spectrum more efficiently, (4) allow tuning by the name of the station rather than by frequency, and (5) allow data to be transmitted. The terrestrial DAB broadcasting system comprises a combination of convolutional coding and $\frac{\pi}{4}$ -DE-QPSK with OFDM and time multiplexing of the transmitted services to enable per service symbol processing [25]. The DAB digital radio broadcast system uses MPEG-2 Audio Layer II audio compression.

1.4.5 Digital Audio Broadcasting plus (DAB+)

Digital Audio Broadcasting plus (DAB+) originates from the DAB broadcast system but comprises extra interleaving and Reed-Solomon coding to support the more efficient MPEG-4 HE-AAC+ v2 audio compression algorithms, similar to the DRM broadcast systems. Each audio super frame is carried in five consecutive logical DAB frames which

enable easy synchronization and management of reconfigurations. The Reed-Solomon RS(120, 110, $t = 5$) shortened code, derived from the original systematic RS(255, 245, $t = 5$) code, is applied to 110 byte portions of each audio super frame to generate an error protected packet. The (de-)interleaver can be considered as a row-column block (de-)interleaver.

1.4.6 Terrestrial-Digital Multimedia Broadcasting (T-DMB)

T-DMB is also based on the conventional DAB transmission system according to [25] with extra interleaving and Reed-Solomon coding. Since T-DMB and DAB are delivered on the same system, then by adding a T-DMB video encoder to the existing DAB system, a T-DMB device can receive not only T-DMB multimedia services but also DAB audio services. T-DMB uses Bit Sliced Arithmetic Coding (BSAC) or HE-AAC+ v2 audio coding for audio services, Advanced Video Coding (AVC) for video services, and Binary Format for Scene (BIFS) for interactive data related services.

In the next section, Section 1.5, we introduce a block-diagram for generating a DAB transmission signal.

1.5 DAB-family

In the sequel of this thesis we will use the PHY of the DAB-family as our DE-COFDM system-model for a study on improved reception techniques.

1.5.1 A block-diagram for generating a DAB transmission signal

Figure 1.3 shows a block-diagram for generating a $\frac{\pi}{4}$ -DE-QPSK COFDM transmission signal, which is applicable to the broadcast systems of the DAB-family described in [25]. A convolutional encoder (convolutional coder denoted by CC) encodes N_b information bits $\{d_p\}$ to N_c coded bits $\{c_i\}$. The convolutional encoder has a code-rate $R_c = \frac{1}{4}$, a constraint length of $K = 7$, and generator polynomials, $g_0 = 133, g_1 = 171, g_2 = 145$ and $g_3 = g_0 = 133$. Larger code-rates can be obtained by puncturing the mother code, see for details [25, Sec. 11.1.2, pp. 130-132]. Moreover, these codes are also known as Rate-Compatible Punctured Convolutional (RCPC) codes, see Hagenauer [38]. We use, conform [58], the de-facto industry standard $R_c = \frac{1}{2}$ convolutional code with generator polynomials $g_0 = 133$ and $g_1 = 171$, which is identical to using the convolutional code with puncturing index $PI = 8$ of Table 29 in [25, Sec. 11.1.2, p.131]. The DAB, DAB+, and T-DMB time interleaver and frequency interleaver will be simulated by a bit-wise uniform block interleaver, which is generated for each simulated code block of bits $\{c_i\}$ resulting in interleaved bits $\{c_j\}$. Thus any permutation of the N_c coded bits is a permissible interleaver and is selected with equal probability conform [58]. A QPSK mapper maps, with Gray encoding, a bit pair $(c_{m,n}^1, c_{m,n}^2)$ to a QPSK symbol $b_{m,n} = \exp(j\Delta\phi_{m,n})$ with $\Delta\phi_{m,n} \in \{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\}$, where index-pair (m, n) indicates the m^{th} OFDM-subcarrier of the n^{th} OFDM-symbol. Moreover, the index-pair (m, n) is a function

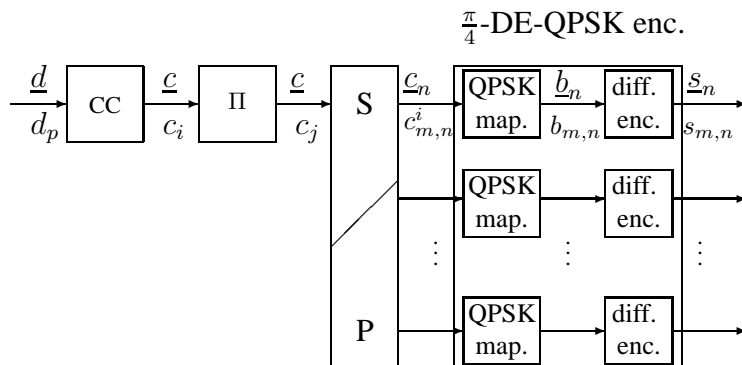


Figure 1.3: A block-diagram for generating a DAB transmission signal.

of bit-index j . A differential encoder performs, for $m = [-\frac{K}{2}, \dots, -1, 1, \dots, \frac{K}{2}]$ and $n = [0, 1, \dots, L - 1]$, differential encoding between symbols $s_{m,n-1} = \exp(j\phi_{m,n-1})$ and $s_{m,n} = \exp(j\phi_{m,n})$ where

$$s_{m,n} = s_{m,n-1} \cdot b_{m,n} \quad (1.1)$$

$$\phi_{m,n} = \phi_{m,n-1} + \Delta\phi_{m,n}, \quad (1.2)$$

with $\phi_{m,n} \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$ for n is even and $\phi_{m,n} \in \{\frac{\pi}{4}, \frac{3\pi}{4}, \frac{5\pi}{4}, \frac{7\pi}{4}\}$ for n is odd. L and K are defined as the frame-length and the frame-width of a DAB transmission frame, respectively. In addition, the K initial $\frac{\pi}{4}$ -DE-QPSK symbols $\{s_{m,0}\}$ are specified in [25]. DE-QPSK symbol $s_{m,n-1}$ modulates an OFDM subcarrier at frequency $f_m = m \cdot \Delta f = m \cdot \frac{1}{T_u}$ with T_u is the inverse of the subcarrier spacing. Moreover, DE-QPSK symbol $s_{m,n}$ modulates a subcarrier at the same frequency f_m of the consecutive OFDM symbol. Without loss of generality we assume that $f_{m=0}$ is not used for data transmission [25]. The transmitted complex baseband signal per frame of L OFDM-symbols each with K modulated subcarriers can now be given by

$$\begin{aligned} s_F(t) &= \sum_{n=0}^{L-1} \sum_{\substack{m=-\frac{K}{2} \\ m \neq 0}}^{\frac{K}{2}} s_{m,n} e^{j2\pi f_m(t-nT_s)} \cdot w(t-nT_s) \\ &\triangleq \sum_{n=0}^{L-1} s_n(t-nT_s), \\ -T_{cp} \leq t &< LT_s - T_{cp} = T_F - T_{cp} \end{aligned} \quad (1.3)$$

where $T_F = LT_s$ is the frame-duration and the rectangular window function $w(\tau)$ for each OFDM symbol is defined as

$$w(\tau) = \begin{cases} 1 & \text{for } -T_{\text{cp}} \leq \tau < T_u \\ 0 & \text{otherwise} \end{cases} \quad (1.4)$$

with $T_s = T_{\text{cp}} + T_u$ is the duration of one OFDM symbol and T_{cp} is the duration of the cyclic prefix, i.e., the guard-interval. Now, our DAB transmitted signal can be written as

$$s(t) = \sum_{F=-\infty}^{+\infty} s_F(t - FT_F), \quad (1.5)$$

where F denotes the frame-index.

In Chapter 2 we will discuss the DAB PHY system model, the channel model, and elaborate on the generation and transmission of terrestrial (COFDM) digital audio broadcast signals.

1.6 List of publications and patents by the Author

The work related to this thesis resulted in the following publications and patents.

1.6.1 Journals

1. W.J. van Houtum and F.M.J. Willems, "A-Posteriori Symbol Probabilities and Log-Likelihood Ratios for Coherently Detected $\frac{\pi}{4}$ -DE-QPSK", *IEEE Communications Letters*, vol. 15, no. 2, pp. 160-162, Feb. 2011.
Chapter 4
2. F.M.J. Willems and W.J. van Houtum, "The Shannon Limit and Some Codes Approaching It", *submitted to IEEE Signal Processing Magazine*.
Chapter 5
3. W.J. van Houtum and F.M.J. Willems, "Two-Dimensional Iterative Processing for DAB Receivers Based on Trellis-Decomposition", *Journal of Electrical and Computer Engineering*, no. 394809, 15 pages, 2012.
Chapter 6
4. W.J. van Houtum and F.M.J. Willems, "Complexity Reduction for Non-Coherent Iteratively Detected DE-QPSK Based on Trellis-Decomposition", *submitted to ETT Eur. Trans. on Telecomm.*
Chapter 7

1.6.2 Conference Proceedings

1. W.J. van Houtum and F.M.J. Willems, "Joint and iterative detection and decoding of differentially encoded COFDM systems", in *Proc. IEEE 17th International Conference on Telecommunications (ICT)*, pp. 36-43, Apr. 2010.

1.6.3 Patents

1. W.J. van Houtum and F.M.J. Willems, "SYSTEM AND METHOD FOR DEMODULATING AND DECODING A DIFFERENTIALLY ENCODED CODED ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING MODULATION CODE USING TWO-DIMENSIONAL CODE BLOCKS", *No. 81383418US01*, Dec. 2009.
2. W.J. van Houtum and F.M.J. Willems, "SYSTEM AND METHOD FOR DEMODULATING AND DECODING A DIFFERENTIALLY ENCODED CODED ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING MODULATION CODE USING TWO-DIMENSIONAL CODE BLOCKS", *No. 81383418EP02*, Nov. 2010.
3. W.J. van Houtum and F.M.J. Willems, "Two-Dimensional Iterative Processing for DAB Receivers Based on Trellis-Decomposition", *Provisional No. 61/499,840 (No. 81426422US01)*, Jun. 2011.

Chapter 2

Channel and DAB System Model

For evaluation purposes we introduce in this chapter the DAB PHY system model, the AWGN channel model, the TU-6 channel model, and elaborate on the generation and transmission of DAB signals.

2.1 Outline

In this chapter, simulation models are introduced. These models are applied later in this thesis to evaluate the proposed reception methods. The AWGN channel model with an input power constraint and the channel model for M -level PSK are first discussed. In addition, the TU-6 channel model defined in COST-207 [1] is introduced. This channel model is typically used to assess the performance of DAB, DAB+, or T-DMB transmission. Finally, the basic elements of a DAB transmitter and a standard receiver are discussed.

After having introduced the channel models, we discuss the channel capacity for serial passband transmission of complex-valued symbols over an ideal time-discrete complex-valued AWGN channel. We demonstrate the minimum required bandwidth and the channel capacity for serial passband transmission with the application of OFDM. We also calculate the channel capacity if uniformly distributed M -PSK with $M = 4, 16$ and 64 symbols were sent over this channel.

Further we calculate the probability of a bit error (BER) for non-coherent and coherent detection of DE-QPSK. Finally, we compare non-coherent detection of coded DE-QPSK with non-coherent detection of DE-QPSK without any coding. For the coded case, a rate $1/2$ convolutional code with soft-decision Viterbi-decoding was applied.

2.2 Channel models

One expects that a suitable channel model describes the physical aspects of a transmission medium that have the largest impact on the original message. These physical aspects will

impose the design rules on the transmitter and also on the receiver because this is the “spitting image” of the transmitter [43, Ch. 1, p. 2].

2.2.1 Wireless transmission

Radio propagation for wireless transmission can be characterized by highly time and frequency dispersive multi-paths [27]. There is also thermal noise present, which originates from the random movement of electrons and is in the literature well studied in for example [59, Sec. 3.1.4]. Some of the statistical properties of this thermal noise will be briefly introduced in Section 2.2.3.

Echoes or replicas of the original message originate from large reflectors and/or scatterers at some distance to the sender and receiver. These replicas have random magnitudes, phases and delays and will sum up destructively or constructively with the original message. Summation results in a small or large amount of signal power at the input of the receiver. Every echo arrives, each via a different path, at the input of the receiver. This multi-path transmission will “smear out” the original signal in time and will cause distortion of symbols, which are transmitted sequentially in time. The distortion of these consecutive symbols is well-known in the literature as “inter symbol interference (ISI)” [59, Sec. 6.2, p. 530]. In Section 2.3 we will discuss COFDM which is a method to combat ISI. Moreover, impulse response measurements in different environments for wireless non-line of sight (NLOS) transmission at different frequencies justify the modeling by a Rayleigh distribution for the magnitude and a uniform distribution for the phase of the echoes [45, Sec 2.1, p. 13] and [62, Sec. 4.6.1, p. 172].

If the channel is changing significantly over the duration of a transmitted symbol or a transmitted frame (multiple symbols), it is said that the channel is a time variant channel. The time variations of the channel are evidenced as a Doppler broadening and, may be, in addition as a Doppler shift of a spectral line [60, Sec. 14.1.1, p. 806].

For modeling a wireless fading channel, we will use the TU-6 COST-207 channel model defined in [1]. The multi-path effects as well as the Doppler effects are captured in this TU-6 channel model, as will be shown in Section 2.2.5. In addition, the TU-6 channel model is typically used to test DAB, DAB+, or T-DMB transmission.

2.2.2 Wireless OFDM transmission

We model wireless OFDM transmission by assuming that we transmit every T seconds a complex-valued pulse, i.e., orthogonal transmission at a rate of $1/T$ complex symbols per second. We will show how this complex orthogonal modulation technique converts a waveform channel into an equivalent time-discrete complex-valued AWGN channel. We will follow the approach of Forney and Ungerboeck in [29], by starting with serial baseband transmission of (perfectly) band-limited real-valued symbols. Then, based on the serial base-band transmission, we will discuss the transmission of (perfectly) band-limited complex-valued symbols by serial passband transmission.

Since OFDM is part of our complex-valued system model it is demonstrated that by the application of OFDM we, again, perform orthogonal transmission at a rate of $1/T$ complex symbols per second over an ideal time-discrete complex-valued AWGN channel.

However, with OFDM each transmitted complex-valued symbol is a sample of a time-discrete periodic function with period $T_u = BT$. This time-discrete periodic function is formed by its frequency-domain representation, i.e., B complex-valued coefficients c_n of the discrete exponential Fourier series. Moreover, these complex-valued frequency-domain coefficients c_n , for $n = 1, 2, \dots, B$, contain the information to be transmitted. Note that the responses of these complex-valued coefficients c_n are (approximately) exponential waveforms. Finally, we will discuss the Shannon limit, i.e., the channel capacity for serial passband transmission over the time-discrete complex-valued AWGN channel. Moreover, the serial passband transmission is discussed without the application of OFDM and with the application of OFDM. Let's first introduce serial baseband transmission with rate $1/T$ of (perfectly) band-limited real-valued symbols.

Serial baseband transmission

Thermal noise $n_w(t)$, is said to be white, i.e., its power spectral density $N_w(f) = N_0/2$ for all f . Hence, it is a stochastic process with a flat (constant) power spectral density over the entire frequency range [78, Sec. 3.5, p. 189], [59, Sec. 3.1.4, p. 156]. For serial baseband transmission of real-valued symbols at a rate of $1/T$ symbols per second we will send out the signal

$$s(t) = \sum_{n=-\infty}^{\infty} a_n p(t - nT), \quad (2.1)$$

where, the coefficients a_n , for integers n , contain the information. Moreover, the shifted baseband pulses $\{p(t - mT), m \text{ integer}\}$ are orthonormal, i.e.,

$$\int_{-\infty}^{\infty} p(t - nT)p(t - mT)dt = \delta_{n-m}, \quad (2.2)$$

where δ_{n-m} is the Kronecker delta function, i.e., $\delta_{n-m} = 1$ if $n = m$ and 0 for $n \neq m$. We also say that these pulses satisfy the Nyquist-criterion for zero intersymbol interference (ISI), see (2.7–2.9) in [29]. The channel output signal $r(t)$ is a noisy version of the input signal $s(t)$, i.e., the channel adds a noise signal $n_w(t)$ to the channel input sequence and the receiver obtains

$$r(t) = s(t) + n_w(t). \quad (2.3)$$

The receiver recovers a sequence $\{r_n\}$ of noisy estimates of the transmitted information symbols a_n , i.e.,

$$\begin{aligned} r_n &= \int_{-\infty}^{\infty} r(t)p(t - nT) dt \\ &= \int_{-\infty}^{\infty} s(t)p(t - nT) dt + \int_{-\infty}^{\infty} n_w(t)p(t - nT) dt \\ &= a_n + n_n. \end{aligned} \quad (2.4)$$

The orthonormality of $\{p(t - nT), n \text{ integer}\}$ guarantees that the sequence $\{n_n, n \text{ integer}\}$ is a set of independent and identically distributed (i.i.d.) Gaussian random

real noise variables with zero mean and variance $\sigma^2 = N_0/2$, i.e.

$$\begin{aligned} E[N_n] &= 0 \\ E[N_n N_m] &= \delta_{n-m} N_0/2, \end{aligned} \quad (2.5)$$

where $E[\cdot]$ denotes expectation. Now, according to optimum detection theory [78], the sequence $\{r_n\}$ in (2.4) contains all information about $\{a_n\}$ that is contained in the time-continuous signal $r(t)$. Thus, the time-continuous received signal $r(t)$ is condensed into the time-discrete sequence $\{r_n\}$ and, the waveform channel is converted to an equivalent time-discrete ideal AWGN channel [29]. Furthermore, if the real-valued baseband transmission pulse is assumed to be a sinc-pulse, i.e.,

$$p(t) = \frac{1}{\sqrt{T}} \frac{\sin\left(\frac{\pi t}{T}\right)}{\frac{\pi t}{T}}, \quad (2.6)$$

this pulse satisfies the Nyquist ISI criterion and has the smallest possible bandwidth $W_b = 1/(2T)$. As a consequence, the real-valued baseband signal contains only frequencies in $[-W_b, +W_b]$ and is perfectly band-limited, i.e., has a ‘‘Brick-wall’’-spectrum. In addition, if the coefficients a_n are i.i.d. we can write for the transmit power of the real-valued baseband symbols

$$P_b \triangleq \frac{E[A_n^2]}{T}. \quad (2.7)$$

The signal-to-noise ratio (SNR) is now equal to

$$\text{SNR} = \frac{E[A_n^2]}{N_0/2} = \frac{P_b T}{N_0/2} = \frac{P_b}{N_0 W_b}. \quad (2.8)$$

Note that $W_b = 1/(2T)$ is the minimum one-sided bandwidth for serial baseband transmission of real-valued symbols at a rate of $1/T$ symbols per second [29].

To investigate orthogonal transmission of complex-valued symbols at a rate of $1/T$ symbols per second we will, in the next section, discuss serial passband transmission for complex-valued symbols.

Serial passband transmission

Since we are interested in transmitting complex-valued symbols we will, in this section, discuss serial passband transmission of complex-valued symbols. We investigate orthogonal transmission of complex-valued symbols at a rate of $1/T$ complex symbols per second. Here we will follow again the approach of Forney and Ungerboeck in [29] and say that the time-continuous passband signal

$$\begin{aligned} s(t) &\triangleq \Re \left\{ \sum_{n=-\infty}^{\infty} s_n p(t - nT) \sqrt{2} \exp(j2\pi f_c t) \right\}, \\ &= \Re \left\{ \sum_{n=-\infty}^{\infty} (a_n + j b_n) p(t - nT) \sqrt{2} \exp(j2\pi f_c t) \right\}, \end{aligned} \quad (2.9)$$

where the coefficients $\{s_n\}$ are now complex symbols containing the information that needs to be transmitted. The signals are in the passband $\pm[f_c - W_b, f_c + W_b]$. By processing (i.e., multiplying with $\sqrt{2} \cos(2\pi f_c t)$, multiplying with $\sqrt{2} \sin(2\pi f_c t)$, low-pass filtering and sampling) the received bandpass signal $r(t) = s(t) + n_w(t)$, we obtain

$$r_n = s_n + n_n, \quad (2.10)$$

where the sequence $\{n_n\}$ is again a sequence of i.i.d. random variables. Each n_n is now circularly symmetric mean zero complex Gaussian noise with variance $N_0/2$ in both the real and imaginary parts. For the transmit power of our complex-valued symbols $\{s_n\}$, we write

$$P = E[|S_n|^2] / T, \quad (2.11)$$

and

$$\text{SNR} = \frac{E_s}{N_0} = \frac{PT}{N_0} = \frac{P}{2W_b N_0}, \quad (2.12)$$

with $E_s \triangleq E[|S_n|^2]$ the average input-symbol energy per complex dimension and $2W_b = 1/T$ is the minimum required bandwidth to transmit $1/T$ complex-valued symbols per second.

Since OFDM is part of our complex-valued communication-system we demonstrate, in the next section, that by the application of OFDM we perform again orthogonal transmission at a rate of $1/T$ complex-valued symbols per second.

Serial passband transmission with OFDM

For serial passband transmission with the application of OFDM, we assume that we transmit every $T_u = BT$ seconds a complex-valued vector $\underline{c} = (c_1, c_2, \dots, c_B)$ of symbols $\{c_n\}$. Moreover, the complex-valued symbols $\{c_n\}$ contain the information that needs to be transmitted. We will demonstrate, in this section, that this is again equivalent to orthogonal transmission at a rate of $1/T = B/T_u$ complex-valued symbols per second. However, with the application of OFDM each transmitted complex-valued symbol $\{s_n\}$ is a sample of a time-discrete periodic function with period $T_u = BT$. This time-discrete periodic function is formed by its frequency-domain representation, i.e., the complex-valued coefficients c_n , for $n = 1, 2, \dots, B$, of the discrete exponential Fourier series. To demonstrate this, we first consider a $B \times B$ complex orthogonal matrix M , where M is the matrix that corresponds to the discrete Fourier transform. The columns m_n , $n = 1, 2, \dots, B$ of this matrix satisfy

$$\langle m_n, m_m^* \rangle = \delta_{n-m}, \quad (2.13)$$

where $\langle \cdot \rangle$ denotes the inner product and δ_{n-m} the Kronecker delta function. If we now want to transmit a complex vector $\underline{c} = (c_1, c_2, \dots, c_B)$, then we first form

$$\underline{s} = M\underline{c}, \quad (2.14)$$

where $\underline{s} = (s_1, s_2, \dots, s_B)$. Subsequently \underline{s} are complex-valued symbols that are transmitted, with serial passband as discussed before. At the receiver side first

$$\underline{r} = \underline{s} + \underline{n}, \quad (2.15)$$

is, similarly as in the previous section, reconstructed, where $\underline{r} = (r_1, r_2, \dots, r_B)$ is the received vector and $\underline{n} = (n_1, n_2, \dots, n_B)$ is the noise vector. Then we form using the inverse matrix M^{-1}

$$\begin{aligned}\hat{\underline{c}} &= M^{-1}\underline{r} = M^{-1}(\underline{s} + \underline{n}) \\ &= M^{-1}M\underline{c} + M^{-1}\underline{n} = \underline{c} + M^{-1}\underline{n}.\end{aligned}\tag{2.16}$$

By the orthonormality of the matrix M^{-1} also the noise vector $M^{-1}\underline{n}$ is i.i.d. with variance of $N_0/2$ per real and imaginary component. Hence, by the application of OFDM we obtain again orthogonal transmission of complex-valued symbols at a rate of $1/T$ symbols per second over the time-discrete complex-valued AWGN channel. As a consequence, for serial passband transmission with the application of OFDM the minimum required bandwidth is $2W_b = 1/T = B/T_u$ for a rate of $1/T = B/T_u$ complex-valued symbols per second. Note, in the previous section we demonstrated that for serial passband transmission of $1/T$ complex-valued symbols per second without the application of OFDM, the minimum required bandwidth was also $2W_b = 1/T = B/T_u$. Thus, in this sense, there are no advantages on more efficient use of the spectrum by applying OFDM. The advantage of using OFDM is that the responses of all c_n , for $n = 1, 2, \dots, B$ are (approximately) exponential waveforms. Such waveforms are eigen-functions of linear time-invariant channels. If now the channel gain is not unity but $H(f)$, i.e., frequency dependent, the attenuation of these exponentials just changes, but can be simply equalized. As we will see later, there is also a cyclic prefix added to an OFDM symbol. This cyclic prefix is useful for combatting multi-path effects. Note, for consistency in notation, we will in the sequel of this thesis denote the complex-valued symbols, which contain the information that needs to be transmitted with $\{s_n\}$ instead of $\{c_n\}$.

2.2.3 AWGN model

As we demonstrated in the previous section, serial passband transmission without and with the application of OFDM, are both orthogonal transmissions at a rate of $1/T$ complex-valued symbols per second within the same bandwidth $2W_b = 1/T = B/T_u$. Now, our channel model will be a time-discrete complex-valued AWGN channel, i.e., an AWGN channel with a single complex-valued input and output and bandwidth $2W_b$. Furthermore, the channel output samples are a noisy version of the channel input samples, i.e., the channel adds a noise sequence to the channel input sequence. The time-discrete complex-valued transmitted signal should satisfy the average power constraint P . Moreover, we assume zero power outside and uniformly distributed power within the transmission bandwidth. Note, that the transmission bandwidth of $2W_b = 1/T = B/T_u$ is the minimum required one-sided bandwidth for orthogonal transmission of $1/T$ complex symbols per second [29].

In the next section we will discuss the Shannon limit, i.e., the channel capacity of the time-discrete complex-valued AWGN channel.

Channel capacity of the AWGN channel

Shannon [66][67] introduced the notion of channel capacity. We will first give the channel capacity per complex dimension for serial passband transmission at a rate of $1/T$ complex-valued symbols per second without and with the application of OFDM. Secondly, we will express the channel capacity, in bit per second. We demonstrated in the previous section that serial passband transmission with the application of OFDM requires the same power and minimum bandwidth of $2W_b = 1/T = B/T_u$ as without the application of OFDM. Therefore, the channel capacity for serial passband transmission without or with the application of OFDM is the same. This is a consequence of using the Fast Fourier Transform (FFT), which is an orthogonal transformation. Moreover, the channel capacity in bit per complex dimension for serial passband transmission over the time-discrete complex-valued AWGN channel can be given by

$$\bar{C} = \log_2 \left(1 + \frac{P}{2W_b N_0} \right) = \log_2(1 + \text{SNR}) \text{ bit/complex dimension, [b/cplx]} \quad (2.17)$$

where $\text{SNR} = P/(2W_b N_0)$, denotes the signal-to-noise ratio. To ease notation, in the sequel of this thesis, we define as transmission bandwidth $W \triangleq 2W_b = 1/T = B/T_u$ for serial passband transmission. Now, the channel capacity in bit per second becomes

$$C = W \log_2(1 + \text{SNR}) \text{ bit/second, [b/s]} \quad (2.18)$$

with $\text{SNR} = P/(W N_0)$. The meaning of the channel capacity is that we can transmit binary digits at the rate R smaller than C bits per second with an arbitrarily small frequency of errors, (6.5-45) in [61, Sec. 6.5-2, p. 367],

$$R < C = W \log_2(1 + \text{SNR}). \quad (2.19)$$

The rate R is the code rate in bits per second of the signalling scheme. However, in the sequel of this thesis, we will use the channel capacity in bit per complex dimension. Since the channel capacity per complex dimension $\bar{C} = C/W$ is in bit/second/Hertz, we will use for the channel capacity in bit per complex dimension the notation [b/s/Hz]. Moreover, Figure 2.1 shows the channel capacity in bit per complex dimension (notation [b/s/Hz]) for serial passband transmission over the time-discrete complex-valued AWGN channel. For channel capacity values $\bar{C} = 2, 4, \text{ and } 6$ b/s/Hz, the required SNR values are given by Table 2.1. From Figure 2.1 can be seen that the channel capacity, \bar{C} , is increasing

Table 2.1: Required SNR for channel capacity \bar{C}

Channel Capacity \bar{C} [b/s/Hz]	SNR [dB]
2	4.8
4	11.8
6	18.0

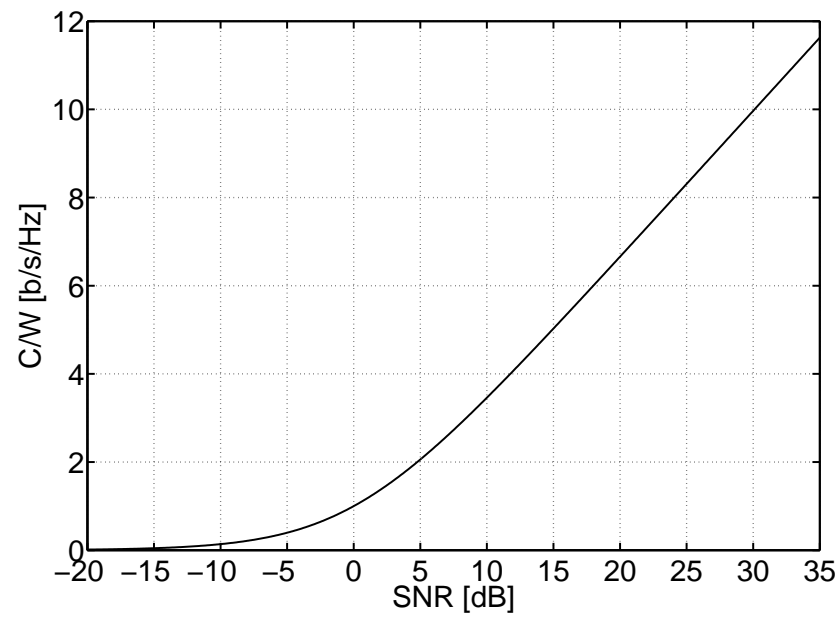


Figure 2.1: Channel capacity \bar{C} in bit per complex dimension.

monotonically if the SNR increases, so for a fixed bandwidth W capacity increases with the increase of the SNR.

Now, let the average energy per information bit be

$$E_b = \frac{E_s}{r} = \frac{PT}{r} = \frac{P}{rW}, \quad (2.20)$$

where $r = R/W$ is the code rate in bits per complex dimension and $W = 1/T$ is the transmission bandwidth. Note that r is also known as the spectral efficiency and can be expressed in [b/s/Hz]. For reliable communication r should be smaller than \bar{C} and by substitution of (2.20) in (2.17), this yields

$$r < \log_2 \left(1 + \frac{P}{WN_0} \right) = \log_2 \left(1 + r \frac{E_b}{N_0} \right). \quad (2.21)$$

Rewriting (2.21), yields a measure for the power efficiency of serial passband transmission of $1/T$ complex symbols per second over the time-discrete complex-valued AWGN channel

$$\frac{E_b}{N_0} > \frac{2^r - 1}{r}. \quad (2.22)$$

For example, from (2.22) can be seen that for reliable communication with a spectral efficiency of 1 b/s/Hz, we require that $\frac{E_b}{N_0} > 1$ or if we express it in [dB] it will become, $\frac{E_b}{N_0} > 10 \cdot \log_{10}(1) = 0$ dB.

The obtained results on serial passband transmission of $1/T$ complex symbols per second over the (ideal) time-discrete complex-valued AWGN channel, will be used as a benchmark for M -level PSK transmission over this channel. We will introduce M -level PSK transmission in the next section.

2.2.4 M -level PSK model

A digital modulator maps digital information, which needs to be transmitted, to an analog (carrier) waveform. In general a digital modulation process involves switching or keying the amplitude, frequency, and/or phase of the carrier according to binary symbols $\{d_n\}$. For complex modulation methods like M -level Phase Shift Keying (M -PSK) it is well-known that $m = \log_2(M)$ bits are mapped (e.g., with Gray-mapping) to form a finite set of complex symbols, i.e., the (complex) signal constellation. From the signal constellation, the complex-valued (M -PSK) symbols are chosen that will be transmitted over the channel. Moreover, we assume that the complex-valued symbols, that need to be transmitted, are chosen from the signal constellation $\mathcal{S} = \{e^{j(m2\pi/M)}, m = 0, 1, \dots, M-1\}$. In addition, we will consider serial passband transmission (without OFDM) of $1/T$ complex symbols per second where the complex-valued coefficients $\{s_n\}$ are chosen from the set \mathcal{S} . In the next section we will evaluate and discuss the channel capacity of this M -level PSK transmission.

Channel capacity of the M -level PSK channel

Massey suggested in his paper [50] that encoding and modulation are both aspects of the signal design problem and that demodulation and decoding are likewise both aspects of the signal detection problem. It might be, therefore, interesting to consider an M -level PSK channel. This M -level PSK channel is the combination of a complex-valued M -level PSK modulator, the time-discrete complex-valued AWGN channel and a complex-valued M -level PSK demodulator. Now, we assume serial passband transmission where the complex-valued coefficients $\{s_n\}$ are chosen from the finite set \mathcal{S} . Moreover, the coefficients $\{s_n\}$ have a uniform distribution, i.e., $\Pr\{s_n = s_m\} = 1/M$ or equivalently all M -PSK symbols have equal probability to be transmitted. Furthermore, the combination of the complex M -level PSK modulator and the time-discrete complex-valued AWGN-channel is characterized by a conditional pdf, see also (2.10),

$$p(r_n|s_n) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|r_n - s_n|^2}{2\sigma^2}\right). \quad (2.23)$$

Now, under the constraint of uniform input probabilities and power constraint P , the channel capacity $C_{M\text{-PSK}}$ of the M -level PSK channel is given by the average mutual information [16]

$$\begin{aligned} C_{M\text{-PSK}} = I(s_n; r_n) &= \mathcal{H}(s_n) - \mathcal{H}(s_n|r_n) \\ &= m - E_{s_n, r_n}[-\log_2 \Pr\{s_n|r_n\}], \end{aligned} \quad (2.24)$$

where $\mathcal{H}(\cdot)$ is the entropy defined in [67, Theorem 2, Sec. 6, p. 393] and $E_{s_n, r_n}[f(s_n, r_n)]$ is the statistical average of f over $\{s_n\}$ and $\{r_n\}$. Now by using Bayes theorem, (2.24) can be rewritten as

$$\begin{aligned} C_{M\text{-PSK}} &= m - E_{s_n, r_n} \left[\log_2 \frac{\sum_{s_m \in \mathcal{S}} p(r_n|s_m)}{p(r_n|s_n)} \right] \\ &= m - E_{s_n, r_n} \left[\log_2 \frac{\sum_{s_m \in \mathcal{S}} p_n(r_n - s_m)}{p_n(r_n - s_n)} \right]. \end{aligned} \quad (2.25)$$

In general the expectations in (2.25) cannot be calculated in closed form so, we performed simulations to obtain numerical results. These simulation results are shown in Figure 2.2.

Figure 2.2 shows the channel capacity of the M -level PSK channel for $M = 4, 16,$ and 64 as function of the SNR. This figure also shows (bold curve), as a reference, the channel capacity of the (ideal) time-discrete complex-valued AWGN channel, given by (2.17).

It is well known from the literature, e.g., [16], [50] that signal space codes, i.e., codes whose words are sequences of symbols $s_n \in \mathcal{S}$, can achieve spectral efficiencies which are less, however, asymptotically equal to the average amount of uncertainty in s_n prior to the observation of r_n . Thus for M -level PSK modulation, the capacity $C_{M\text{-PSK}}$ will achieve asymptotically, i.e., the $\text{SNR} \rightarrow \infty$, the rate $m = \log_2(M)$ b/s/Hz. This is also shown by

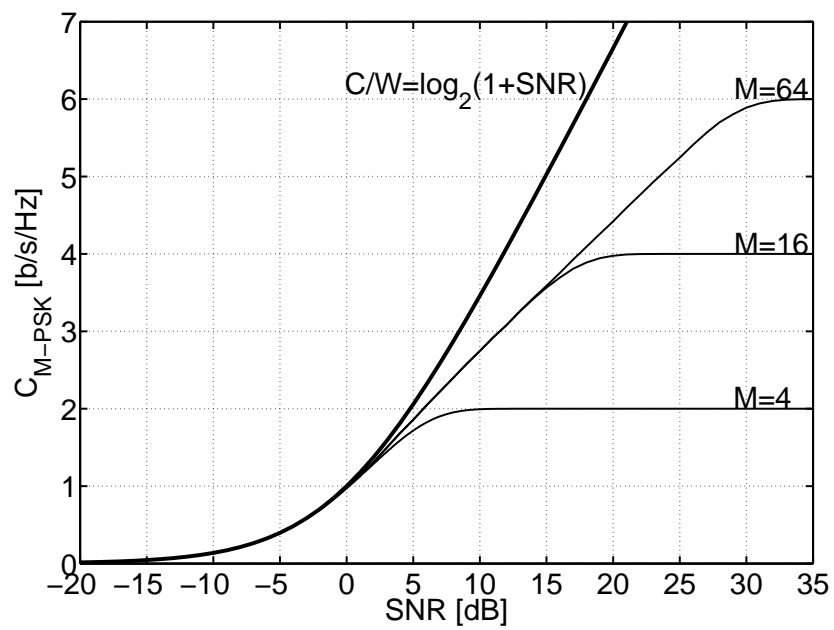


Figure 2.2: Channel capacity C_{M-PSK} for $M = 4, 16,$ and 64 .

Figure 2.2. Since the broadcast systems of the DAB-family, where we are focussing on, use a form of QPSK, i.e., $\frac{\pi}{4}$ -DE-QPSK, see Section 2.3 for more details, we will limit our analysis in this thesis to M -PSK with $M = 4$.

Next we will, in Section 2.2.5, briefly introduce the TU-6 channel model defined in [1], which is commonly used to test DAB, DAB+, or T-DMB transmission.

2.2.5 TU-6 COST-207 model

From (2.10) can be seen that only thermal noise was added to the signal $s(t)$. However, a time varying and multi-path channel will cause time and frequency dispersion or spreading of the signal $s(t)$ and is therefore called “a doubly spread channel”. A doubly spread channel can be characterized by the scattering function $S(\tau; \lambda)$, which represents the power at delay τ and a frequency offset λ (relative to the carrier-frequency). From the scattering function, the multi-path intensity profile or delay power spectrum of the channel can be obtained by [12, Sec. 2, p. 2620]

$$S_c(\tau) = \int_{-\infty}^{\infty} S(\tau; \lambda) d\lambda. \quad (2.26)$$

Also the Doppler power spectrum can be obtained in a similar way

$$S_c(\lambda) = \int_{-\infty}^{\infty} S(\tau; \lambda) d\tau. \quad (2.27)$$

The delay spread τ_m and Doppler spread B_d of the channel are defined as the range of values over which $S_c(\tau)$ respectively $S_c(\lambda)$ are essentially non-zero [59, Sec. 7.1, p. 709]. The Doppler spread is a measure for how fast the channel is changing, which is reflected by the channel coherence time [59, Sec. 7.1, p. 709]

$$T_c \approx \frac{1}{B_d}. \quad (2.28)$$

A measure of the width of the band of frequencies that are similarly affected, i.e., the frequency band for highly correlated fades is the channel coherence bandwidth and is established in a similar way as T_c [59, Sec. 7.1, p. 708]

$$B_c \approx \frac{1}{\tau_m}. \quad (2.29)$$

If the bandwidth of the transmitted signal is larger than B_c the signal will be distorted, in such a case the channel is said to be frequency-selective. An additional distortion is caused by the time variation of the channel. In this case the time-interval of interest, i.e., symbol duration T_s or frame duration T_F is larger than T_c and the received signal strength is fading. The delay power spectrum (delay-spread represented by τ_m) can be used to characterize the frequency selectivity of the channel. The Doppler power spectrum (Doppler-spread represented by B_d) can be used to characterize the rapidity of the fading of the channel.

The COST-207 TU-6 channel model is a tapped delay-line model [61, Sec. 13.5-1, p. 869], where each tap represents a transmission path (6 in total). The six time-variant taps are zero mean complex-valued stationary Gaussian random processes. The magnitudes of these taps are Rayleigh-distributed and their phases are uniformly distributed. For each tap (or path) a stochastic process is available. This stochastic process is characterized by its variance and Doppler power spectrum. The variance of the stochastic process is a measure for the average signal power that can be received via this path. In addition, the corresponding Doppler power spectrum models the time-variant behavior of the channel. The statistical behavior of the 6 different paths for the COST-207 TU-6 channel model are shown in Table 2.2, see also [1].

Table 2.2: *COST-207 TU-6 channel model*

Path No.	Delay [μ s]	Power [dB]	Doppler category
1	0.0	-3	Jakes (Class)
2	0.2	0	Jakes (Class)
3	0.5	-2	Jakes (Class)
4	1.6	-6	BiGaussian (Gauss1)
5	2.3	-8	BiGaussian (Gauss2)
6	5.0	-10	BiGaussian (Gauss2)

Table 2.2 shows that the multi-path spread $\tau_m = 5 \mu$ s, i.e., the delay of path 6, the path with the largest delay. This multi-path spread results in a coherence bandwidth of $B_c \approx 200$ kHz. Moreover, Figure 2.3, shows the defined Doppler power spectra of the different paths for $B_d = 40$ Hz. Figure 2.3 shows that the Doppler-spectra contain their power within the Doppler-frequencies of $[-20, 20]$ Hz, i.e., the maximum Doppler frequency $f_d = B_d/2 = 20$ Hz. Note, a Doppler spread of $B_d = 2f_d = 40$ Hz results in a coherence time of $T_c \approx 25$ ms. Figure 2.3 also shows (the lower right picture) the variations of the amplitudes for the different paths as function of time with $f_d = 20$ Hz, see also [1]. For details on the Jakes and BiGaussian Doppler spectra defined for the COST-207 TU-6 channel-model the reader is referred to [1].

It seems now appropriate to make some remarks on the coherence-time and coherence-bandwidth for DAB transmission in Mode-I (Mode-I is the most commonly used set of PHY-parameters for a DAB transmission, see for details [25]). Note that to prevent ISI in an OFDM-scheme, the delay differences on separate propagation paths need to be less than the cyclic-prefix period [79]. Thus, the channel impulse response length, i.e., the multi-path spread τ_m must satisfy $\tau_m \leq T_{cp}$. Within the DAB-system $T_{cp} \triangleq \frac{63}{256}T_u < \frac{T_u}{4}$ [25] and therefore the coherence-bandwidth $B_c \approx \frac{1}{\tau_m} > 4\frac{1}{T_u}$, which is at least 4 OFDM-subcarriers. This means $B_c > 4$ kHz for DAB transmission in Mode-I, where the subcarrier-spacing $\Delta f = \frac{1}{T_u} = 1$ kHz [25]. A maximum Doppler frequency of, e.g., $f_d = 20$ Hz, represents for DAB transmission (in Band-III, i.e., high-band VHF from

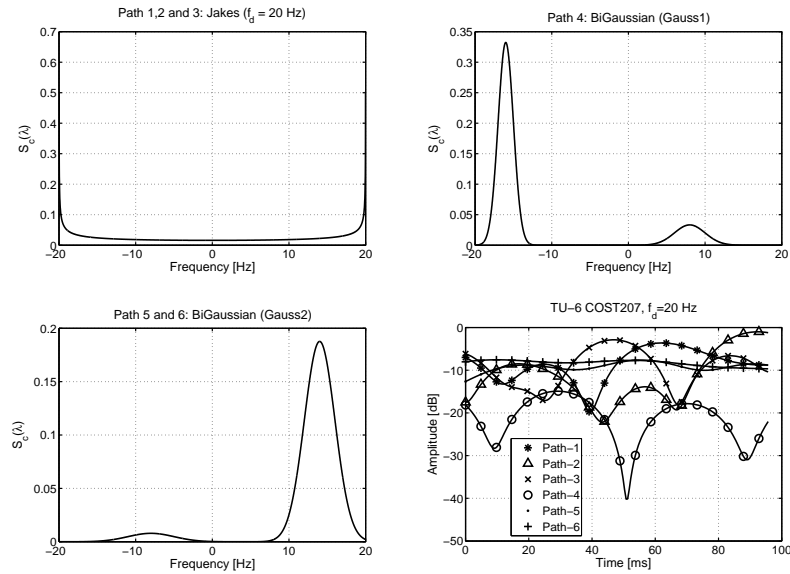


Figure 2.3: COST-207 TU-6 channel model.

174 to 230 MHz) a movement-speed between transmitter and receiver of ≈ 90 km/h. For this Doppler frequency the coherence-time $T_c \approx \frac{1}{2f_d} = 25$ ms, which is ≈ 20 OFDM-symbols (including cyclic prefix) for DAB transmission in Mode-I, where the OFDM symbol period $T_s = T_u + T_{cp} = 1.246$ ms [25].

2.3 DAB system model

2.3.1 Block-diagram

The block-diagram, given by Figure 2.4, shows the basic elements of the DAB system model. The incoming information bits $\{d_p\}$ are encoded by a 1/4-rate convolution encoder with 64-states ($\nu = 6$) using the industry-standard polynomials $g_0 = 133_8$, $g_1 = 171_8$, $g_2 = 145_8$ and $g_3 = g_0 = 133_8$. Larger code-rates can be obtained via puncturing of the mother code, see for details [25, Sec. 11.1.2, pp. 130-132]. Moreover, these codes are also known as RCPC codes, see Hagenauer [38]. To avoid burst errors, the coded bits $\{c_i\}$ are fed into a convolutional interleaver, i.e., time interleaver, see [25, Ch. 12, pp. 137-142]. The time interleaved coded bits $\{c_l\}$ are divided into bit-pairs. Each such bit-pair is converted, via a Gray labeling map, into a QPSK-symbol b_k of the finite (offset) alphabet $\mathcal{B} = \{e^{j(\frac{\pi}{2}p + \frac{\pi}{4})}, p = 0, 1, 2, 3\}$ according to [25, Sec. 14.5, p. 157]. Frequency interleaving is performed on QPSK symbols $\{b_k\}$ as described in [25, Sec. 14.6, pp. 157-161]. Thus, the time- and the frequency-interleavers in DAB perform bit

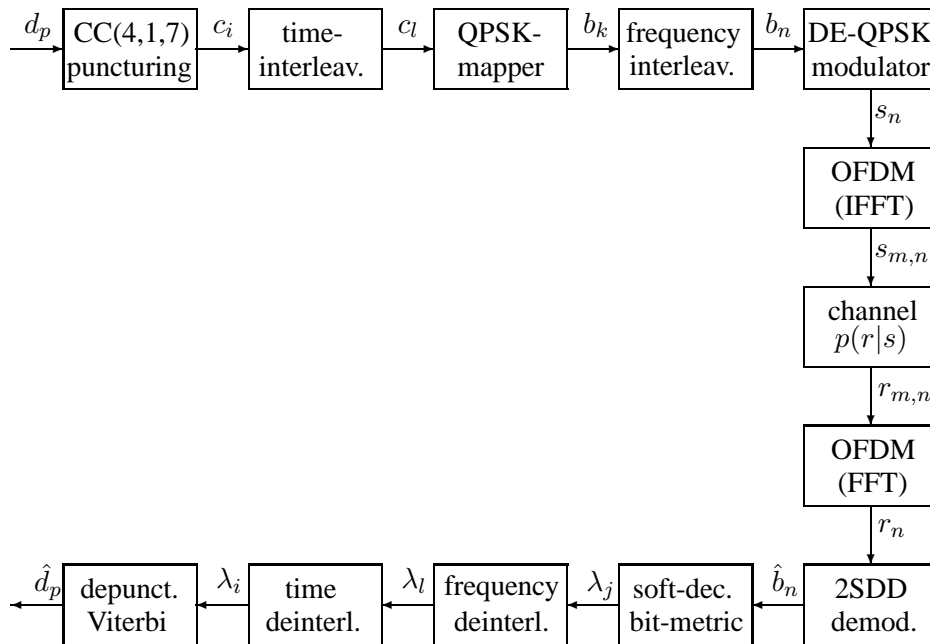


Figure 2.4: Block-diagram of the DAB system model.

and bit-pair interleaving, respectively. As a result the code-bits leaving the convolutional encoder are permuted and partitioned over the sub-carriers of a number of subsequent OFDM-symbols (in subsequent frames). The relationship between indices i , l and n can be determined from the description of the time interleaving procedure described in [25, Ch. 12, pp. 137-142], the mapping-procedure in [25, Sec. 14.5, p. 157], and the frequency interleaving procedure described in [25, Sec. 14.6, pp. 157-161]. Now, for each sub-carrier $\frac{\pi}{4}$ -DE-QPSK modulation is applied. A sequence $\mathbf{b} = (b_1, b_2, \dots, b_N)$ consisting of N symbols (phase differences) b_n for $n = 1, 2, \dots, N$ carries the information that is to be transmitted via this sub-carrier. The symbols $b_n, n = 1, 2, \dots, N$ assume values in the (offset) alphabet $\mathcal{B} = \{e^{j(p\pi/2 + \pi/4)}, p = 0, 1, 2, 3\}$. The transmitted sequence $\mathbf{s} = (s_0, s_1, \dots, s_N)$ of length $N + 1$ follows from \mathbf{b} by applying differential phase modulation, i.e.,

$$s_n = b_n s_{n-1}, \text{ for } n = 1, 2, \dots, N, \quad (2.30)$$

where for the first symbol $s_0 \triangleq 1$. OFDM in DAB is realized using a B -point complex inverse fast Fourier transform (IFFT), where B is 256, 512, 1024, or 2048. To compute the n -th time-domain OFDM-symbol

$\tilde{\mathbf{s}}_n = (\tilde{s}_{1,n}, \tilde{s}_{2,n}, \dots, \tilde{s}_{B,n})$, we determine

$$\tilde{s}_{t,n} = \frac{1}{\sqrt{B}} \sum_{m=1}^B s_{m,n} \exp\left(j2\pi \frac{(t-1)(m-1)}{B}\right), \text{ for } t = 1, 2, \dots, B, \quad (2.31)$$

and $s_{m,n}$ is the n -th differentially encoded symbol corresponding to the m -th sub-carrier, or equivalently the m -th element in the n -th frequency-domain OFDM-symbol¹. Note, that the IFFT is a computationally efficient inverse discrete Fourier transform (IDFT) for values of B that are powers of 2. To prevent ISI resulting from multi-path reception, a cyclic prefix of length L_{cp} is added to the sequence \tilde{s}_n . This leads to the sequence $\bar{s}_n = (\tilde{s}_{B-L_{cp}+1,n}, \dots, \tilde{s}_{B,n}, \tilde{s}_{1,n}, \tilde{s}_{2,n}, \dots, \tilde{s}_{B,n})$ that is finally transmitted. For the AWGN-channel, thermal noise \bar{n}_n is added to \bar{s}_n , resulting in the n -th received OFDM-symbol

$$\bar{r}_n = \bar{s}_n + \bar{n}_n. \quad (2.32)$$

The receiver, in the case of perfect synchronization, removes the (received version of the) cyclic prefix, and then applies a B -points complex FFT on the time-domain received sequence $\tilde{r}_n = (\tilde{r}_{1,n}, \tilde{r}_{2,n}, \dots, \tilde{r}_{B,n})$ which results in the B received symbols

$$r_{m,n} = \frac{1}{\sqrt{B}} \sum_{t=1}^B \tilde{r}_{t,n} \exp\left(-j2\pi \frac{(m-1)(t-1)}{B}\right), \text{ for } m = 1, 2, \dots, B. \quad (2.33)$$

OFDM reception can be regarded as parallel matched-filtering corresponding to B complex orthogonal waveforms, one for each subcarrier, see also Section 2.2.2. Now, for a transmission over, for example, the AWGN-channel, i.e.,

$$r_{m,n} = s_{m,n} + n_{m,n}, \quad (2.34)$$

where $n_{m,n}$ is circularly symmetric complex Gaussian noise with zero mean and variance $\sigma^2 = \frac{N_0}{2}$ per component, a transition pdf representing the AWGN-channel can be given by

$$p(r_{m,n}|s_{m,n}) = \frac{1}{2\pi\sigma^2} \exp\left(\frac{-|r_{m,n} - s_{m,n}|^2}{2\sigma^2}\right). \quad (2.35)$$

For the demodulation procedure in classical DAB-receivers [79], soft-decision bit metrics, $\{\lambda_j\}$, are LLRs computed with a-posteriori symbol probabilities for non-coherent 2SDD of DE-QPSK. The computation of these soft-decision bit metrics will be discussed in more detail in Section 3.4.2. This soft-decision bit-metric calculation is followed by frequency deinterleaving and time deinterleaving, resulting in bit-metrics $\{\lambda_j\}$ and $\{\lambda_i\}$, respectively. As stated before, the relationship between indices j , l and i can be determined from [25]. Note, the time interleaving is over multiple OFDM-symbols in multiple OFDM-frames, i.e., the codeword consists of multiple OFDM-symbols originating from different OFDM-frames. Moreover, for our analysis, the time interleaver and frequency interleaver are modeled by a bit-wise uniform block interleaver generated for each simulated code block of bits. Hence, any permutation of the coded bits is a permissible interleaver and is selected with equal probability, as is done in [58]. Depuncturing is performed by introducing erasures into a Viterbi decoder. Finally, the Viterbi decoder acts as a Maximum Likelihood Sequence Estimator (MLSE) for retrieving estimates $\{\hat{d}_p\}$ of the original information bits $\{d_p\}$.

¹The framing structure of the DAB transmission signal is out of our scope, for framing structure details see [25, Ch. 14, pp. 144-164]

In the sequel of this chapter, i.e., Sections 2.3.2-2.3.7, we will elaborate on RCPC codes, time-interleaving, M -PSK, DE-QPSK, the combination of OFDM with RCPC codes and $\frac{\pi}{4}$ -DE-QPSK and, finally, time-multiplexing of the DAB-services.

2.3.2 Rate-Compatible Punctured Convolutional (RCPC) codes

A code sequence from a convolutional code is generated by passing the information sequence $\{d_i\}$ through a linear finite-state shift register [60, Sec. 8.2, p. 471]. The bit-encoding of the DAB-system according [25] is performed by a convolutional encoder with a code-rate $R_c = k/n = 1/4$, a constraint length of $K = \nu + 1 = 7$, i.e., the shift register consists of $\nu = 6$ bit-delay elements D . This convolutional encoder is based on generator polynomials $g_0 = 1 + D^2 + D^3 + D^5 + D^6 = 133_8$, $g_1 = 1 + D + D^2 + D^3 + D^6 = 171_8$, $g_2 = 1 + D + D^4 + D^6 = 145_8$, and $g_3 = g_0 = 133_8$. To discuss the properties of this 64-state $R_c = 1/4$ -rate convolutional code, we will use the transfer function $T(D, N, J)$ of the code, where the exponent of D indicates the Hamming distance, the exponent of N the number of input bit 1 branch transitions, and the exponent of J indicates the length of the path that merges with the all-zero path for the first time [60, Sec. 8.2.1, p. 479]. Now, for our DAB convolutional-code, a code spectrum can be determined from which the minimum Hamming distance d_{free} , the number a_d of paths with Hamming weight d and the number β_d of information bit errors on all incorrect paths with Hamming weight d can be obtained. This code spectrum will be obtained with the FAST-algorithm introduced by Cederval et al. in [17]. The number a_d of paths with Hamming weight d is the coefficient of D^d in $T(D, 1, 1)$ [60, Sec. 8.2.1], i.e.,

$$\sum_{d=d_{\text{free}}}^{\infty} a_d D^d = T(D, 1, 1) = 2D^{19} + 1D^{20} + 2D^{21} + 2D^{22} + 4D^{23} + 5D^{24} + \dots, \quad (2.36)$$

where the combination of the parameters d_{free} and a_d is referred to as the code distance spectrum [17]. In addition, the number β_d of information bit errors on all incorrect paths with Hamming weight d is the coefficient of D^d in the derivative of $T(D, N, 1)$ with respect to N and setting $N = 1$ [60, Sec. 8.2.3], i.e.,

$$\begin{aligned} \sum_{d=d_{\text{free}}}^{\infty} \beta_d D^d &= \left. \frac{d[T(D, N, 1)]}{dN} \right|_{N=1} \\ &= 4D^{19} + 2D^{20} + 6D^{21} + 6D^{22} + 22D^{23} + 22D^{24} + \dots, \end{aligned} \quad (2.37)$$

where the combination of the parameters d_{free} and β_d is referred to as the code weight spectrum [17]. Note, a large d_{free} and a small β_d characterize codes with good error correcting capabilities with, for example, Viterbi-decoding [60, Sec. 8.2.3, p. 489].

Larger code-rates are obtained via puncturing of the rate $R_c = 1/4$ mother code, these punctured codes are RCPC codes, discussed by Hagenauer et al. in [38]. Puncturing is deleting specific coded bits according to a given perforation pattern. A particular interesting RCPC-code used by the DAB-system is the RCPC code with puncturing-index eight, i.e., $PI = 8$ in Table 29 of [25, Sec. 11.1.2, p. 131]. For $PI = 8$ the perforation matrix is

such that the two streams of coded bits generated by the generator polynomials g_2 and g_3 are entirely removed. However, puncturing the generator polynomials g_2 and g_3 results in the de-facto industry standard $R_c = 1/2$, $K = 7$ convolutional code with generator polynomials $g_0 = 133_8$ and $g_1 = 171_8$. We will use this convolutional code throughout the thesis. This rate $R_c = 1/2$ convolutional encoder generates a code \mathcal{C} , which is optimal in the sense that it has the largest possible minimum free distance d_{free} [60, Sec. 8.2.5, p.492]. The convolutional encoder is shown by Figure 2.5 [25, Sec. 11.1.1, p. 130] in the controller canonical form [44, Sec. 2.1, p. 35]. The code distance spectrum, for this rate

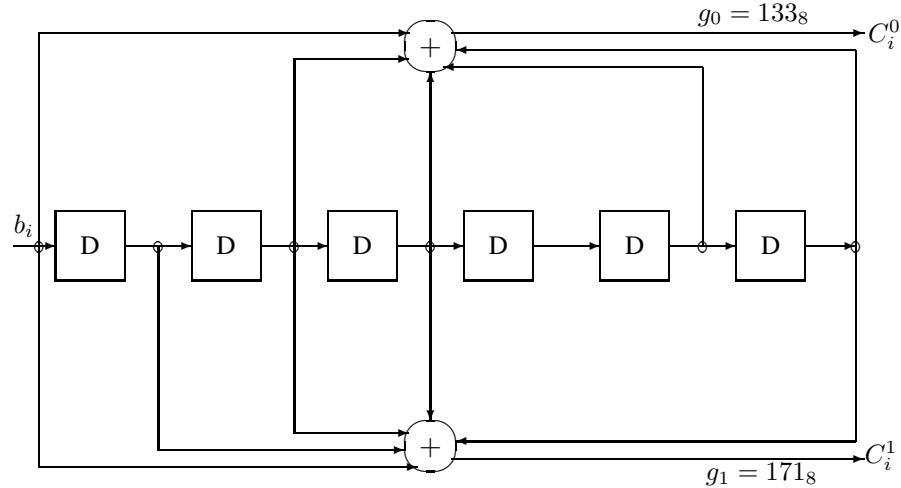


Figure 2.5: $R_c = 1/2$ convolutional encoder with $g_0 = 133_8$ and $g_1 = 171_8$.

$R_c = 1/2$ convolutional-code,

$$\sum_{d=d_{\text{free}}}^{\infty} a_d D^d = T(D, 1, 1) = 11D^{10} + 38D^{12} + 193D^{14} + 1331D^{16} + \dots, \quad (2.38)$$

and the code weight spectrum [17],

$$\sum_{d=d_{\text{free}}}^{\infty} \beta_d D^d = \frac{d[T(D, N, 1)]}{dN} \Big|_{N=1} = 36D^{10} + 211D^{12} + 1404D^{14} + \dots, \quad (2.39)$$

are both obtained with the FAST-algorithm [17]. Now we would like to show, with another small example of the DAB-system, the ability of the RCPC coding approach to make a trade-off between throughput and required SNR. We have chosen for the RCPC code conform $PI = 4$ in Table 29 of [25, Sec. 11.1.2, p. 131]. The perforation matrix for this RCPC code is, see for perforation matrix notation also [60, Sec. 8.2.6, p. 497],

$$[P_{\frac{2}{3}}] = \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_0 \\ \mathbf{C}_1 & \mathbf{C}_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}. \quad (2.40)$$

Note, that also for $PI = 4$ the coded bit streams C_2 and C_3 generated by the polynomials g_2 respectively g_3 are entirely removed by puncturing, see perforation matrix $PI = 4$ in Table 29 in [25, Sec. 11.1.2, p. 131]². From Table I in [5] (with $M = 6$), the weight spectrum of the RCPC code with $PI = 4$, can be found as,

$$\left. \frac{d[T(D, N, 1)]}{dN} \right|_{N=1} = \sum_{d=d_{\text{free}}}^{\infty} \beta_d D^d = D^6 + 16D^7 + 48D^8 + 158D^9 + \dots \quad (2.41)$$

From (2.41) can be seen that the RCPC code with $PI = 4$ reduces the minimum free distance from $d_{\text{free}} = 19$ for the 1/4-rate parent code to $d_{\text{free}} = 6$ for the RCPC code with $PI = 4$. On the other hand, with this RCPC code the code-rate increases from 1/4 to 2/3 and, consequently, a throughput increase by a factor of 8/3 can be obtained by the same rate of transmission. Hence, we are able with RCPC codes to make a trade-off between required SNR and throughput within the allocated transmission bandwidth of, for example, the DAB-system.

In the next section we will briefly discuss time interleaving of our DAB system and the model we are going to use.

2.3.3 Time-interleaving

Convolutional codes are effective if the errors caused by the channel are statistically independent [60, Sec. 8.1.9, p. 468]. Coded systems like the DAB system should operate reliably in multipath and fading channels, which exhibit bursty error characteristics. An effective way to deal with burst error channels is to interleave the coded bits in such a way that the bursty channel is transformed into a channel having independent errors, see also [60, Sec. 8.1.9, p. 468]. In the DAB system a convolutional-interleaver is applied for time interleaving [25, Sec. 12, pp. 137-142]. As a result the code-bits leaving the convolutional encoder are permuted and partitioned over the sub-carriers of a number of subsequent OFDM-symbols in subsequent OFDM-frames, see for details about time interleaving Chapter 12 of [25]. However, as stated earlier, we will for our analysis replace the time (and frequency) interleaving by a bit-wise uniform block interleaving scheme for each codeword.

In the next section we will discuss M -level phase shift keying, as the DAB system also uses a form of M -PSK, i.e., differentially encoded quaternary $M = 4$ PSK.

2.3.4 M -level Phase Shift keying (M -PSK)

In this section, we will discuss the BER of M -PSK as a function of the SNR for $M = 2^m = 4, 16$, and 64. Furthermore, the complex-valued M -PSK symbols that will be transmitted over the time-discrete complex-valued AWGN channel are chosen from the signal constellation $\mathcal{S} = \{e^{j(m2\pi/M)}, m = 0, 1, \dots, M - 1\}$, see Section 2.2.4.

²Perforation matrices in Table 29 are in a sequential order of the coded bit-streams for blocks of 32 coded bits, i.e., $[P_{\text{DAB}}] = [C_0 C_1 C_2 C_3 C_0 C_1 C_2 C_3 \dots C_0 C_1 C_2 C_3 C_0 C_1 C_2 C_3]$.

BER of M -PSK over the AWGN channel

The probability of a symbol error (SER) for M -PSK is shown in the literature, see for example (5.2-56) in Proakis [60, Sec. 5.2.7, p. 268], as an integral that does not reduce to a simple form and must be evaluated numerically except for $M = 2$ and $M = 4$. However, for large values of M and high SNR values, the SER can be approximated by, see (5.2-61) in [60, Sec. 5.2.7, p. 270],

$$\text{SER}_{M\text{-PSK}} \approx 2Q \left(\sqrt{2 \frac{E_s}{N_0}} \sin \left(\frac{\pi}{M} \right) \right), \quad (2.42)$$

where $Q(z) \triangleq \int_z^\infty \frac{1}{\sqrt{2\pi}} \exp(-a^2/2) da$. Note, it turns out that this approximation of the SER is good for all values of M . In addition, with the approximation that the SER is m -times the BER, given by (5.2-62) in [60, Sec. 5.2.7, p. 271], the BER for M -PSK becomes,

$$\text{BER}_{M\text{-PSK}} \approx \frac{2}{m} Q \left(\sqrt{2 \sin^2 \left(\frac{\pi}{M} \right) \cdot \frac{m \cdot E_b}{N_0}} \right), \quad (2.43)$$

where $m = \log_2(M)$ and $E_s = m \cdot E_b$, see also (4.3-17) in [61, Sec. 4.3-2, p. 194]. Note, for the last approximation Gray mapping is assumed, i.e., labels that correspond to adjacent phases differ only in a single bit-position. Now for QPSK the probability of a bit-error can be calculated by substitution of $M = 4$ in (2.43), this yields,

$$\text{BER}_{\text{QPSK}} \approx Q \left(\sqrt{2 \cdot \left(\frac{1}{2} \right) \cdot \frac{2E_b}{N_0}} \right) = Q \left(\sqrt{\frac{2E_b}{N_0}} \right), \quad (2.44)$$

which is, actually, the exact BER for QPSK given by (5.2-57) in [60, Sec. 5.2.7, p. 268]. Moreover, this result is also shown by (8.30) in [2, Sec. 8.1.1.3, p. 201], by (5) in [48] and by (2) in [47].

Figure 2.6 shows the $\text{BER}_{M\text{-PSK}}$ as function of the SNR = $\log_2(M) \frac{E_b}{N_0} = m \frac{E_b}{N_0}$ expressed in [dB], i.e., $10 \cdot \log_{10}(\text{SNR})$, for $m = 2, 4$, and 6 b/s/Hz. Figure 2.6 also shows, as a reference, the required SNR values for the channel capacity \bar{C} , i.e., the "Shannon bound", given by (2.17), for $2, 4$, and 6 b/s/Hz. The circles on the curves represent simulated values of the $\text{BER}_{M\text{-PSK}}$. In the simulation model a combination of m binary symbols are assigned (labeled) via Gray-labeling to a particular M -PSK symbol and send over the AWGN channel. From Figure 2.6 can be seen that the approximation (2.43) fits quite well to the obtained simulation results. For the probability of a bit error $\text{BER}_{M\text{-PSK}} = 10^{-4}$, the required SNR values are given by Table 2.3.

If we compare the second column of Table 2.1 with the second column of Table 2.3, there is a difference of ≈ 6.6 dB, ≈ 10.4 dB, and ≈ 15.9 dB with the "Shannon bound" at a $\text{BER}_{M\text{-PSK}} = 10^{-4}$ for 4-PSK, 16-PSK, and 64-PSK, respectively. Moreover, since increase of the SNR is required to decrease the BER, the gap with the "Shannon-bound" will increase for smaller values of the BER. Figure 2.6 indeed shows that for a smaller $\text{BER}_{M\text{-PSK}}$ the difference increases, e.g., ≈ 12 dB, ≈ 16 dB, and ≈ 22 dB at a $\text{BER}_{M\text{-PSK}} = 10^{-12}$ for $M = 4, 16$, and 64 , respectively.

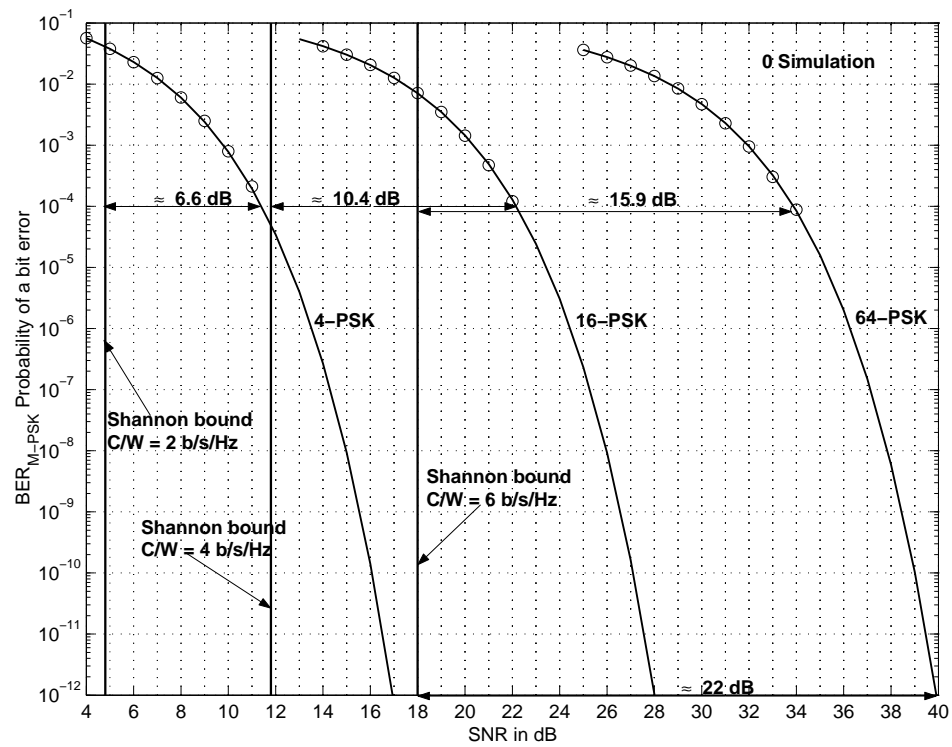


Figure 2.6: The BER_{M-PSK} for $M = 4, 16, 64$.

Table 2.3: Required SNR for $BER_{M-PSK} = 1 \cdot 10^{-4}$

M-PSK	SNR [dB]
4	11.4
16	22.2
64	33.9

After this brief introduction on M -level PSK, we will from now on focus on $M = 4$, i.e., QPSK, as the DAB system uses a form of QPSK, i.e., $\frac{\pi}{4}$ -DE-QPSK.

2.3.5 Differentially encoded QPSK (DE-QPSK)

For DE-QPSK streams, a type of demodulation can be performed where the previous symbol is used as the (noisy) reference for detection, i.e., differential detection. In this case, there is no need to estimate the actual phase of the carrier [60, Sec. 5.2.8, p. 272]. This differential detection technique is known in the literature as differential QPSK (DQPSK) [60, Sec. 5.2.8, p. 272]. Since no phase estimation is required, DQPSK is often considered as a non-coherent communication technique [60, Sec. 5.2.8, p. 272]. We will also take this view throughout the thesis³. Note that for being able to perform non-coherent differential detection of DE-QPSK, it is desirable that the received carrier phase should be constant over at least two received DE-QPSK symbols.

In the case of coherent detection of QPSK, it is assumed that the detector should have perfect knowledge of the carrier frequency plus phase. However, introduced by the carrier-recovery, the carrier phase is known or estimated up to an ambiguity of a multiple of $\frac{\pi}{2}$ [2, Sec. 3.1.4, p. 39]. Thus, coherent detection of QPSK can be successful if these phase ambiguity of a multiple of $\frac{\pi}{2}$ is solved. A manner of resolving this $\frac{\pi}{2}$ -phase ambiguity, is by differentially encoding the phase. It might be expected, as the decision for the information phase is based on two transmitted phases, that there is some loss in BER-performance compared to non-differential encoded QPSK given by (2.44), see also [2, Sec. 3.1.4, p. 39]. Next, to evaluate the performances of DE-QPSK, we compute the BER of DE-QPSK with coherent detection and non-coherent differential detection. We will demonstrate for both detection methods the penalty in required SNR compared to coherent detection of QPSK to obtain a BER = 10^{-4} .

BER of $\frac{\pi}{4}$ -DE-QPSK

In this section we will briefly introduce $\frac{\pi}{4}$ -DE-QPSK and compute the probability of a bit error. The BER is computed for, both, coherent detection and non-coherent differential detection of $\frac{\pi}{4}$ -DE-QPSK. For $\frac{\pi}{4}$ -DE-QPSK the n -th transmitted information symbol b_n assumes a value in the (offset) alphabet \mathcal{B} . Moreover, the first transmitted symbol for $n = 0$ is initialized with $s_0 = 1$. Now, for odd n , the differentially encoded transmitted symbols $\{s_n\}$ will also assume values in \mathcal{B} . However, for even n , the transmitted symbols $\{s_n\}$ assume values in the alphabet $\mathcal{A} = \{e^{j(p\pi/2)}, p = 0, 1, 2, 3\}$. This modulation method is called; $\frac{\pi}{4}$ -DE-QPSK. Moreover, alternating between the sets \mathcal{A} and \mathcal{B} results in a maximum phase change between two consecutive transmitted symbols of $\frac{3\pi}{4}$. Note that the maximum phase change for conventional DE-QPSK (or QPSK) is equal to π . The maximum phase change of $\frac{3\pi}{4}$ results in a smaller instantaneous amplitude change than reversing the sign, i.e., a change of the phase by π . Reversing the sign is the maximum

³Since the previous symbol is a reference for the phase of the carrier, DQPSK is also seen as a differentially coherent detection technique for DE-QPSK, see for example [2, Sec. 3.5, pp. 59-65; Sec. 8.1.5.1, pp. 214-217] and references therein.

instantaneous change of the amplitude. For non-linear transmission, large instantaneous amplitude changes result in large spectral side-lobes of the modulation [2, Sec. 3.5.2, p. 65]. Hence, $\frac{\pi}{4}$ -DE-QPSK modulation with non-linear transmission results in smaller spectral side-lobes compared to DE-QPSK. On a linear AWGN channel with coherent detection no difference in performance is to be expected between $\frac{\pi}{4}$ -DE-QPSK and conventional DE-QPSK. Furthermore, it is well-known and straightforward to show, as we will see in Section 6.2.6 of Chapter 6, that $\pi/4$ -DE-QPSK is equivalent to DE-QPSK. Therefore, we will in the sequel of this thesis treat $\frac{\pi}{4}$ -DE-QPSK similar as DE-QPSK. Next, we will demonstrate the BER of coherent detection of DE-QPSK. For coherent detection it is assumed that the carrier frequency is perfectly known. The BER for coherent detection of DE-QPSK (or $\frac{\pi}{4}$ -DE-QPSK) was found by Simon in [68] and given by (5) in [68], which yields,

$$\begin{aligned} \text{BER}_{\text{DE-QPSK}} &= 2Q\left(\sqrt{\frac{2E_b}{N_0}}\right) - 2Q^2\left(\sqrt{\frac{2E_b}{N_0}}\right), \\ &= 2Q\left(\sqrt{\frac{2E_b}{N_0}}\right) \left(1 - Q\left(\sqrt{\frac{2E_b}{N_0}}\right)\right), \end{aligned} \quad (2.45)$$

where we have used $E_s = 2E_b$ and $\text{erfc}(x) = 2Q(\sqrt{2}x)$. For large values of the signal to noise ratio, (2.45) can be approximated by,

$$\text{BER}_{\text{DE-QPSK}} \approx 2Q\left(\sqrt{\frac{2E_b}{N_0}}\right). \quad (2.46)$$

Coherent detection of DE-QPSK exhibits a two times higher BER compared to the BER of QPSK, given by (2.44), which is not unnatural as the bit errors tend to occur in pairs for DE-QPSK. Figure 2.7 shows that for large signal to noise ratio or equivalently for low BER-values this results in a loss ≈ 0.35 dB in required SNR at a BER = 10^{-4} . Next, we compute the probability of a bit error for non-coherent differential detection of DE-QPSK.

For non-coherent differential detection of DE-QPSK, the phase detections are made by using a demodulated reference signal based on the previous received symbol, i.e., 2SDD, see also [3, Sec. 3.4.2, p. 71]. Moreover, Pawula et al. in [54] found that the SER for uncoded DE-QPSK with non-coherent 2SDD of the DE-QPSK symbols can be given by,

$$\text{SER}_{\text{DE-QPSK}} = \frac{1}{2\sqrt{2}\pi} \int_{\theta=-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{\exp\left(-\frac{E_s}{N_0} \left[1 - \frac{\cos\theta}{\sqrt{2}}\right]\right)}{1 - \frac{\cos\theta}{\sqrt{2}}} d\theta, \quad (2.47)$$

see also (8.84) in [3, Sec. 8.1.5.1, p. 246]. If Gray mapping is applied, then with the results of Lassing et al. in [47] and Pawula in [54], [55] the BER for DE-QPSK with non-coherent 2SDD of the DE-QPSK symbols becomes,

$$\text{BER}_{\text{DE-QPSK}} = F\left(\frac{5\pi}{4}\right) - F\left(\frac{\pi}{4}\right), \quad (2.48)$$

where

$$F(\psi) = -\frac{\sin \psi}{4\pi} \int_{t=-\frac{\pi}{2}}^{\frac{\pi}{2}} \frac{\exp\left\{-\frac{2E_b}{N_0}(1 - \cos \psi \cos t)\right\}}{1 - \cos \psi \cos t} dt, \quad (2.49)$$

see also [48] and (8.86–8.87) in [3, Sec. 8.1.5.1, p. 247]. Figure 2.7 shows, as function of the E_b/N_0 , the BER given by (2.48). Figure 2.7 shows that 2SDD of DE-QPSK exhibits

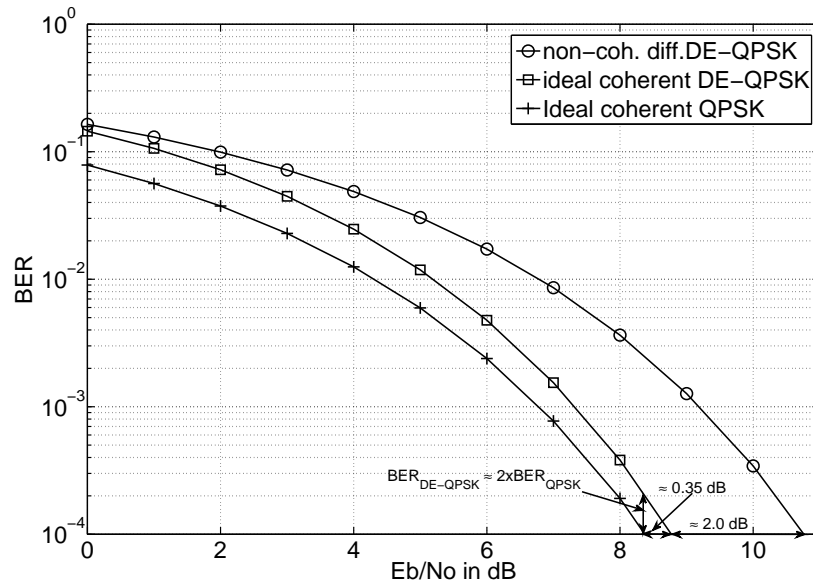


Figure 2.7: BER for AWGN of non-coherent 2SDD of uncoded DE-QPSK and coherent detection of uncoded DE-QPSK and QPSK.

a loss of ≈ 2.0 dB in required SNR at a BER = 10^{-4} , compared to coherent detection of DE-QPSK.

Finally, we will briefly introduce another method to obtain the BER for DE-QPSK with 2SDD. This method is discussed by Simon et al. in [3, Sec. 4.2.1, p. 94] and is useful for computing a bound on the BER, as we will show in the next section. The method uses a first-order Marcum Q-function, given by (4.34) in [3, Sec. 4.2.1, p. 94],

$$Q_1(\alpha, \beta) = \int_{x=\beta}^{\infty} x \exp\left[-\left(\frac{x^2 + \alpha^2}{2}\right)\right] I_0(\alpha x) dx, \quad (2.50)$$

and applies the zero-th order modified Bessel function of the first kind, (4.36) in [3, Sec. 4.2.1, p. 94],

$$I_0(z) = \frac{1}{2\pi} \int_{\theta=-\pi}^{\pi} \exp(-z \sin \theta) d\theta. \quad (2.51)$$

Now, the BER for DE-QPSK with 2SDD computed by Marcum Q-functions becomes,

$$\text{BER}_{\text{DE-QPSK}} = \frac{1}{2} \left[1 - Q_1(\sqrt{b}, \sqrt{a}) + Q_1(\sqrt{a}, \sqrt{b}) \right], \quad (2.52)$$

where

$$a = (2 - \sqrt{2}) \frac{E_b}{N_0}, \quad b = (2 + \sqrt{2}) \frac{E_b}{N_0}, \quad (2.53)$$

see also (8.88), in combination with (8.61), in [3, Sec. 8.1.5.1, p. 247] or (5.2-70–5.2-71) in [60, Sec. 5.2.8, p. 275].

Union bound on the BER of DE-QPSK

Divsalar et al. showed by (22) in [22] for $N = 2$, an upper-bound on the BER for differentially encoded M -PSK with 2SDD. This BER upper-bound, is the union-bound (UB) for DE-MPSK with 2SDD, and yields,

$$\text{BER}_{\text{DE-MPSK}} \leq \frac{1}{\log_2(M)} \sum_{\Delta \hat{\phi}_k \neq \Delta \phi_k} w \{ (c_k^1 c_k^2), (\hat{c}_k^1 \hat{c}_k^2) \} P_2(\hat{\eta} > \eta | \Delta \phi_k), \quad (2.54)$$

with $w \{ (c_k^1 c_k^2), (\hat{c}_k^1 \hat{c}_k^2) \}$ is the Hamming distance between the bit-pairs $(c_k^1 c_k^2)$, respectively, $(\hat{c}_k^1 \hat{c}_k^2)$. Moreover, $P_2(\hat{\eta} > \eta | \Delta \phi_k)$ is the probability that the phase of the information symbol $\Delta \hat{\phi}_k$ is incorrectly chosen when $\Delta \phi_k$ was send, i.e., the Pairwise Error Probability (PEP) with decision statistic η given by (15) in [22]. From the method shown by Divsalar et al. in [22, Sec. III, p. 302] the UB for 2SDD ($N = 2$) of DE-QPSK ($M = 4$), becomes,

$$\text{BER}_{\text{DE-QPSK}} \leq \frac{1}{2} \exp\left(-\frac{2E_b}{N_0}\right) + \frac{1}{2} \left[1 - Q_1(\sqrt{b}, \sqrt{a}) + Q_1(\sqrt{a}, \sqrt{b}) \right], \quad (2.55)$$

with a and b given by (2.53). The UB, (2.55), is computed and shown in Figure 2.8. Note, that the first term in (2.55) for large values of the SNR becomes very small and (2.55) approaches (2.52). Figure 2.8 indeed shows that the UB, (2.55), is quite tight for large values of the SNR.

2.3.6 $\frac{\pi}{4}$ -DE-QPSK and RCPC codes with OFDM

In the previous section, we investigated the BER of DE-QPSK without channel-coding. It is well-known that for transmission over noisy, time-varying and frequency-selective channels, i.e., noisy fading channels, channel-coding can significantly decrease the required signal-to-noise ratio to obtain a specific average BER [61, Sec. 14.3, p. 918]. Therefore, we will study the BER for the (DAB representative) combination of the RCPC codes, interleaving, DE-QPSK and OFDM, see also Fig. 2.9. We assume ideal synchronization, parallel matched-filtering with orthogonal (FFT) waveforms one for each subcarrier, Gray mapping and perfect interleaving. To show the BER performance of DE-QPSK and RCPC codes with OFDM as function of the E_b/N_0 , we choose the $R_c = 1/2$

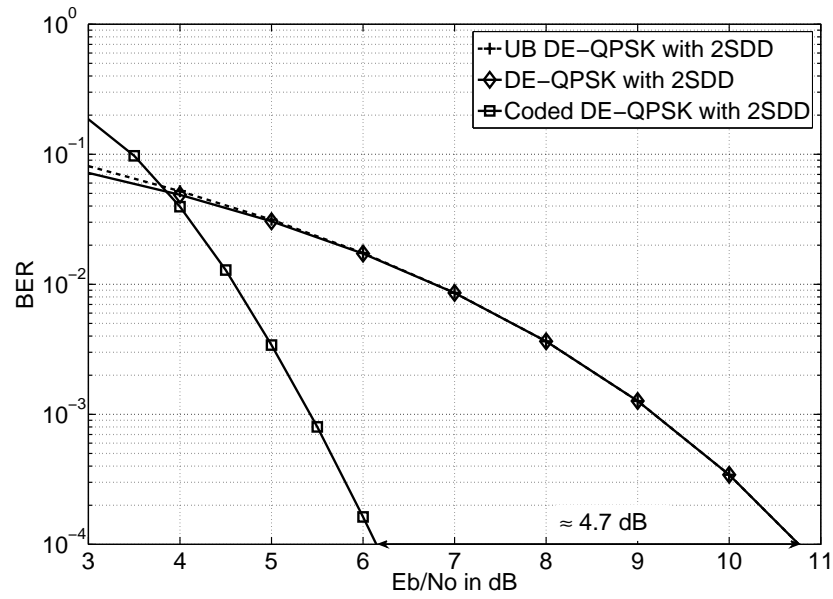


Figure 2.8: Union-bound for AWGN of non-coherent 2SDD of DE-QPSK given in (2.55) and BER for AWGN of non-coherent 2SDD of DE-QPSK and coded DE-QPSK.

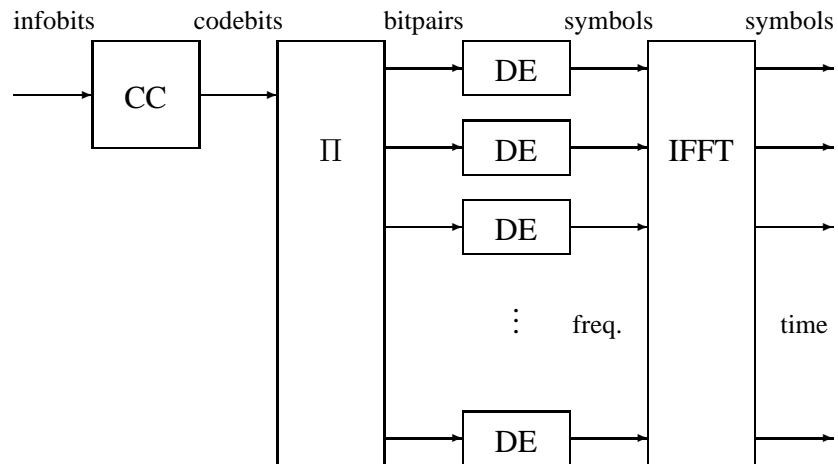


Figure 2.9: DAB convolutional encoder, interleaver, differential encoders, and multi-carrier modulator.

RCPC code with code weight spectrum given by (2.39). Figure 2.8 shows the results of the simulations for the coded case as well as for the uncoded case. Note that the BER performance of 2SDD for DE-QPSK and RCPC codes with OFDM is obtained with soft-decision bit-metrics based on LLRs and Viterbi decoding. The computation of the soft-decision bit-metrics for 2SDD of DE-QPSK on the AWGN channel will be discussed in the next chapter, Chapter 3, in Section 3.4.1. For now we will only consider the coded BER performance for comparison with the uncoded case. Figure 2.8 shows that by using channel coding there is a decrease, compared to the uncoded case, in the required SNR ratio of ≈ 4.7 dB at a BER of 10^{-4} .

2.3.7 Time-multiplexing

Time multiplexing of the transmitted services allows the receiver to perform per service symbol processing [25], see Fig. 2.10. This means that the receiver can decode a cer-

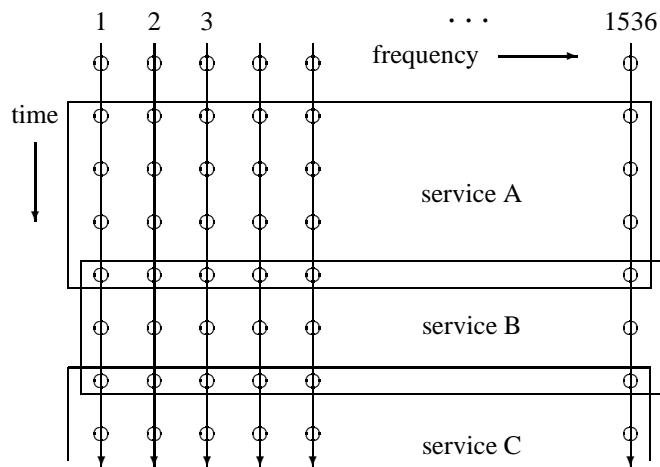


Figure 2.10: *Three services mapped onto consecutive OFDM symbols. Note that there is overlap between the service since differential modulation is used.*

tain service without having to process the OFDM symbols that do not correspond to this service. Consequently, only at particular time instants within a DAB transmission-frame a small number (usually up to four) of OFDM-symbols need to be processed. This results in “idle-time” for the demodulation and decoding processes. Due to this “idle-time”, as we will discuss in Chapter 6, the mix-metric techniques of [31] cannot be applied to DAB-receivers.

In the next chapter, Chapter 3, we elaborate on the detection and decoding of coded DE-QPSK streams. Coherent detection and non-coherent differential detection with a two-symbol observation interval and a multi-symbol (larger than two) observation interval are

investigated. Moreover, a-posteriori symbol probabilities and LLRs for Viterbi decoding are discussed. Some approximations on the a-posteriori symbol probabilities and the LLRs are also introduced.

Chapter 3

Detection and Decoding of DE-QPSK

We are interested in reception improvement techniques for communication systems that apply coded DE-QPSK as in DAB systems. Therefore we evaluate in this chapter coherent detection and non-coherent differential detection procedures for coded DE-QPSK streams. For these detection techniques the a-posteriori symbol probabilities and the LLRs needed for Viterbi decoding are computed.

3.1 Outline

In this chapter, the state-of-the-art in non-iterative detection and decoding techniques for DE-QPSK streams with convolutional encoding is described. First, as a reference, coherent detection of DE-QPSK with soft-decision Viterbi decoding is studied. Then it is demonstrated that non-coherent 2SDD of DE-QPSK with soft-decision Viterbi decoding degrades the performance¹. This non-coherent differential detection scheme can be improved by, for example MSDD, which is a maximum-likelihood procedure for finding a block of information symbols after having observed a block of received symbols. Non-coherent MSDD is evaluated in the last section of this chapter.

3.2 Introduction

3.2.1 A-posteriori symbol probabilities

To discuss detection and decoding of $\frac{\pi}{4}$ -DE-QPSK signals, we will model the received signal, see also (2.34), by

$$r_{m,n} = |h|e^{j\phi}s_{m,n} + n_{m,n}, \quad (3.1)$$

¹Since no phase estimation is required, 2SDD is often considered as a non-coherent detection technique. We will also take this view throughout the thesis.

for subsequent values of n and m , where n is the OFDM-symbol index and m the subcarrier index. The channel gain $|h|$ and phase ϕ are supposed to be unknown to the receiver. Note, that the phase ϕ represents the combination of the transmitter carrier phase and the phase rotation introduced by the propagation through the medium. Later, in Chapter 6, we will show that this channel model is of interest for differentially encoded COFDM systems. However, in the sequel of the present chapter we focus on a single subcarrier. The sequence $\mathbf{s} = (s_0, s_1, \dots, s_N)$ that is transmitted via a certain subcarrier is now observed by the receiver as sequence $\mathbf{r} = (r_0, r_1, \dots, r_N)$. Note that compared to (3.1) we have dropped the subscript m here. Since in practise it is relatively easy to estimate the channel gain, we assume here that it is perfectly known to the receiver and to ease our analysis we take it to be one. The received sequence now relates to the transmitted sequence \mathbf{s} as follows

$$r_n = e^{j\phi} s_n + n_n, \text{ for } n = 0, 1, \dots, N, \quad (3.2)$$

where we assume that n_n is circularly symmetric complex Gaussian with variance σ^2 per component.

Basically we assume that the random channel-phase ϕ is real-valued, uniform over $[0, 2\pi)$, and fixed over all $N + 1$ transmissions. Moreover, in the coherent case it is assumed that the receiver knows this channel phase up to modulo $\frac{\pi}{2}$.

Assuming that the noise samples are independent circularly symmetric complex Gaussian variables with variance $\sigma^2 \triangleq \frac{N_0}{2}$ per component, we get

$$\begin{aligned} p(\mathbf{r}|\mathbf{s}, \phi) &= \frac{1}{(2\pi\sigma^2)^{N+1}} \exp\left(-\frac{\|\mathbf{r} - e^{j\phi}\mathbf{s}\|^2}{2\sigma^2}\right) \\ &= \frac{1}{(\pi N_0)^{N+1}} \exp\left(-\frac{\|\mathbf{r} - e^{j\phi}\mathbf{s}\|^2}{N_0}\right), \end{aligned} \quad (3.3)$$

where $N + 1$ is the length of the observation interval and

$$\|\mathbf{r} - e^{j\phi}\mathbf{s}\|^2 = \sum_{n=0}^N |r_n - e^{j\phi} s_n|^2. \quad (3.4)$$

At this point, it seems appropriate to show that $\pi/4$ -DE-QPSK is equivalent to DE-QPSK. This will make our analysis and notation in the sequel of this chapter simpler. The equivalence follows from

$$\begin{aligned} x_n &= b_n s_{n-1} e^{-jn\pi/4} \\ &= b_n e^{-j\pi/4} s_{n-1} e^{-j(n-1)\pi/4} \\ &= a_n x_{n-1} \\ y_n &= (e^{j\phi} s_n + n_n) e^{-jn\pi/4} \\ &= e^{j\phi} x_n + w_n, \end{aligned} \quad (3.5)$$

where we defined $a_n = b_n e^{-j\pi/4}$, $x_n = s_n e^{-jn\pi/4}$, $y_n = r_n e^{-jn\pi/4}$, and $w_n = n_n e^{-jn\pi/4}$ for $n = 1, 2, \dots, N$, see also (2.30)-(2.35) in Chapter 2. It now follows

that $\frac{\pi}{4}$ -DE-QPSK is equivalent to DE-QPSK, i.e., both a_n and $x_n \in \mathcal{A} = \{e^{jp\pi/2}, p = 0, 1, 2, 3\}$ for $n = 0, 1, \dots, N$, $x_0 = 1$ and that w_n , just like n_n , is circularly symmetric complex Gaussian with variance σ^2 per component. Moreover since \mathbf{b} is Gray-coded with respect to the interleaved code-bits, so is $\mathbf{a} = (a_1, a_2, \dots, a_N)$. In the sequel of this chapter, we will focus on DE-QPSK.

Now for DE-QPSK, $\mathbf{x} = (x_0, x_1, \dots, x_N)$ is the transmitted DE-QPSK symbol sequence, $\mathbf{y} = (y_0, y_1, \dots, y_N)$ is the received DE-QPSK symbol sequence, and $\mathbf{a} = (a_1, a_2, \dots, a_N)$ is the transmitted QPSK information symbol sequence. Moreover, $\{w_n\}$ is independent circularly symmetric complex Gaussian noise and the relation between y_n , x_n , and w_n is now given by (3.5). Since the transmitted DE-QPSK symbol sequence $\mathbf{x} = (x_0, x_1, \dots, x_N)$ is a one-to-one function of the initially transmitted DE-QPSK symbol x_0 and the transmitted QPSK-symbol sequence $\mathbf{a} = (a_1, a_2, \dots, a_N)$ [58], we can write

$$\Pr\{x_0, \mathbf{a}|\mathbf{y}\} = \Pr\{\mathbf{x}|\mathbf{y}\} = \frac{\Pr\{\mathbf{x}\}p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})} = \Pr\{x_0\} \frac{\Pr\{\mathbf{a}\}}{p(\mathbf{y})} p(\mathbf{y}|\mathbf{x}), \quad (3.6)$$

and the a-posteriori probability for $\mathbf{a} = (a_1, a_2, \dots, a_N)$ becomes

$$\Pr\{\mathbf{a}|\mathbf{y}\} = \sum_{x_0} \Pr\{x_0, \mathbf{a}|\mathbf{y}\} = \sum_{x_0} \Pr\{x_0\} \frac{\Pr\{\mathbf{a}\}}{p(\mathbf{y})} p(\mathbf{y}|\mathbf{x}). \quad (3.7)$$

First, by using (3.7) and the relationship given by (3.5) with the unknown channel phase ϕ , we get

$$\begin{aligned} \Pr\{\mathbf{a}|\mathbf{y}\} &= \sum_{x_0} \frac{\Pr\{x_0\} \Pr\{\mathbf{a}\}}{p(\mathbf{y})} \int_{\phi} p(\phi, \mathbf{y}|\mathbf{x}) d\phi \\ &= \sum_{x_0} \frac{\Pr\{x_0\} \Pr\{\mathbf{a}\}}{p(\mathbf{y})} \int_{\phi} p(\mathbf{y}|\mathbf{x}, \phi) p(\phi|\mathbf{x}) d\phi \\ &= \sum_{x_0} \frac{\Pr\{x_0\} \Pr\{\mathbf{a}\}}{p(\mathbf{y})} \int_{\phi} p(\mathbf{y}|\mathbf{x}, \phi) p(\phi) d\phi, \end{aligned} \quad (3.8)$$

where the last equality comes from the fact that the transmitted DE-QPSK sequence \mathbf{x} is independent from the channel phase ϕ . Then, rewriting (3.3) and (3.4) for DE-QPSK yields the likelihood

$$p(\mathbf{y}|\mathbf{x}, \phi) = \frac{1}{(2\pi\sigma^2)^{N+1}} \exp\left(-\frac{\|\mathbf{y} - e^{j\phi}\mathbf{x}\|^2}{2\sigma^2}\right), \quad (3.9)$$

and substitution of this likelihood into the a-posteriori probability given by (3.8), yields an expression for the a-posteriori probability of information sequence $\mathbf{a} = (a_1, a_2, \dots, a_N)$

$$\Pr\{\mathbf{a}|\mathbf{y}\} = \sum_{x_0} \frac{\Pr\{x_0\} \Pr\{\mathbf{a}\}}{p(\mathbf{y}) (2\pi\sigma^2)^{N+1}} \int_{\phi} \exp\left(-\frac{\sum_{n=0}^N |y_n - e^{j\phi}x_n|^2}{2\sigma^2}\right) p(\phi) d\phi, \quad (3.10)$$

since

$$\|\mathbf{y} - e^{j\phi} \mathbf{x}\|^2 = \sum_{n=0}^N |y_n - e^{j\phi} x_n|^2. \quad (3.11)$$

3.3 Coherent detection

Here, we will discuss coherent detection of DE-QPSK signals. For coherent detection both the carrier frequency and the carrier phase are known prior to detection. However, for DE-QPSK streams ($M = 4$) the carrier phase has a $\frac{2\pi}{M} = \frac{\pi}{2}$ phase ambiguity, which is introduced by the carrier recovery methods for QPSK signals. In this section, we also introduce the commonly used max-log-MAP approximation for the a-posteriori symbol probabilities and the LLRs for coherent detection of DE-QPSK signals. Since with max-log-MAP approximations logarithmic or exponential operations are avoided, the computation of a-posteriori symbol probabilities and LLRs becomes simpler. This complexity reduction is especially interesting to enable iterative processing in DAB-receivers, as we will see in Chapter 7.

3.3.1 Coherent symbol-by-symbol detection

For the moment it is assumed that the QPSK information symbols $\{a_n\}$ are independent and uniformly distributed (iud) over the QPSK alphabet $\mathcal{A} = \{e^{jp\pi/2}, p = 0, 1, 2, 3\}$. In Chapter 6 when we discuss iterative processing we will consider the non-iud setting. Colavolpe showed in [19] that if the information symbols are iud, coherent MAP symbol detection reduces to a coherent symbol-by-symbol detection strategy. Since this is an essential result for coherent detection of DE-QPSK, we will derive and analyse this result also here. We will follow the method used by Colavolpe, where he made use of the forward-backward algorithm, i.e., the BCJR algorithm introduced in 1974 by Bahl et al. in [4]. Later, in Chapter 6, we will again discuss the forward-backward algorithm when the differential encoder is used in a code concatenation. The decoding of this code concatenation will be accomplished by two BCJR algorithms and iterative decoding procedures. For coherent detection of DE-QPSK symbols, it is assumed that the carrier frequency is perfectly known and that the channel phase is known with a phase ambiguity of $\frac{\pi}{2}$. This phase ambiguity is introduced by, for example, a suppressed carrier tracking loop. Such a loop locks with equal probability to the channel phase, i.e., the transmitted carrier phase including the phase rotation of the channel plus any of the $M = 4$ transmitted phase values, [2, Sec. 3.1.4, p. 39]. Thus, for coherent detection of DE-QPSK, we assume that the channel phase ϕ in (3.5) is constant over the entire transmission and takes any value in QPSK alphabet \mathcal{A} with equal probability $1/4$. However to show that the MAP symbol strategy reduces to a symbol-by symbol decision strategy when the information symbols are iud we will assume, without loss of generality, that $\phi = 0$ but x_0 is unknown to the receiver, i.e., x_0 takes any value in \mathcal{A} with equal probability $1/4$. With these assumptions we will follow a similar approach as Colavolpe.

The MAP symbol detection strategy for DE-QPSK that minimizes the symbol error prob-

ability is based on the decision rule

$$\hat{a}_n = \arg \max_{a_n} \Pr\{a_n | \mathbf{y}\}. \quad (3.12)$$

We apply the BCJR algorithm to compute the a-posteriori probability $\Pr\{a_n | \mathbf{y}\}$ in (3.12). This yields

$$\begin{aligned} \Pr\{a_n | \mathbf{y}\} &\propto \Pr\{a_n, \mathbf{y}\} \\ &= \sum_{x_{n-1}} \Pr\{y_0, \dots, y_{n-2}, x_{n-1}, y_{n-1}, a_n, y_n, \dots, y_N\} \\ &= \sum_{x_{n-1}} p(y_0, \dots, y_{n-2}, x_{n-1}) p(y_{n-1} | x_{n-1}) \Pr\{a_n\} p(y_n, \dots, y_N | x_{n-1}, a_n) \\ &\triangleq \sum_{x_{n-1}} \alpha_{n-1}(x_{n-1}) \gamma_{n-1}(x_{n-1}) \Pr\{a_n\} \beta_n(x_{n-1} a_n), \end{aligned} \quad (3.13)$$

where we have defined the likelihood

$$\gamma_n(x_n) = p(y_n | x_n) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|y_n - x_n|^2}{2\sigma^2}\right). \quad (3.14)$$

The $\alpha_n(x_n)$ may be calculated by the forward recursion

$$\begin{aligned} \alpha_n(x_n) &= p(y_0, \dots, y_{n-1}, x_n) \\ &= \sum_{(x_{n-1}, a_n) \rightarrow x_n} p(y_0, \dots, y_{n-2}, x_{n-1}, y_{n-1}, a_n) \\ &= \sum_{(x_{n-1}, a_n) \rightarrow x_n} p(y_0, \dots, y_{n-2}, x_{n-1}) p(y_{n-1} | x_{n-1}) \Pr\{a_n\} \\ &= \sum_{(x_{n-1}, a_n) \rightarrow x_n} \alpha_{n-1}(x_{n-1}) \gamma_{n-1}(x_{n-1}) \Pr\{a_n\} \\ &= \sum_{a_n} \alpha_{n-1}(x_n a_n^*) \gamma_{n-1}(x_n a_n^*) \Pr\{a_n\}, \end{aligned} \quad (3.15)$$

where we used for the last equality $x_{n-1} = x_n a_n^*$, see also (3.5). The notation $(x, a) \rightarrow x'$ stands for all states x and symbols a that lead to next state x' . In addition, $\beta_{n-1}(x_{n-1})$ may be calculated by the backward recursion

$$\begin{aligned} \beta_{n-1}(x_{n-1}) &= p(y_{n-1}, y_n, \dots, y_N | x_{n-1}) \\ &= \sum_{a_n} p(a_n, y_{n-1}, y_n, \dots, y_N | x_{n-1}) \\ &= \sum_{a_n} \Pr\{a_n\} p(y_{n-1} | x_{n-1}) p(y_n, \dots, y_N | x_{n-1} a_n) \\ &= \sum_{a_n} \Pr\{a_n\} \gamma_{n-1}(x_{n-1}) \beta_n(x_{n-1} a_n) \\ &= \gamma_{n-1}(x_{n-1}) \sum_{a_n} \beta_n(x_{n-1} a_n) \Pr\{a_n\}, \end{aligned} \quad (3.16)$$

with initializations

$$\begin{aligned}\alpha_0(x_0) &= \frac{1}{4}, & \forall x_0 \in \mathcal{A}, \\ \beta_N(x_N) &= \gamma_N(x_N), & \forall x_N \in \mathcal{A}.\end{aligned}\quad (3.17)$$

Now with iud QPSK information symbols $\{a_n\}$, i.e., $\Pr\{a_n\} = 1/4$, the forward recursion (3.15) for computing $\alpha_n(x_n)$ becomes

$$\begin{aligned}\alpha_n(x_n) &= \frac{1}{4} \left[\sum_{a_n} \alpha_{n-1}(x_n a_n^*) \gamma_{n-1}(x_n a_n^*) \right] \\ &\propto \frac{1}{4},\end{aligned}\quad (3.18)$$

since we add always the same terms, within the brackets, for each x_n , see also (11) in [19]. In Chapter 6, Section 6.3, we will make use of this property to calculate, in an efficient way, a-posteriori symbol probabilities for non-coherent detection of DE-QPSK based on trellis-decomposition. However, for now it is sufficient to note that the forward recursion with iud information symbols is proportional to $1/4$. In addition, the backward recursion (3.16) for computing $\beta_{n-1}(x_{n-1})$ becomes with iud information symbols

$$\begin{aligned}\beta_{n-1}(x_{n-1}) &= \gamma_{n-1}(x_{n-1}) \left[\frac{1}{4} \sum_{a_n} \beta_n(x_{n-1} a_n) \right] \\ &\propto \gamma_{n-1}(x_{n-1}),\end{aligned}\quad (3.19)$$

since also here we add always the same terms, within the brackets, for each x_n , see also (12) in [19]. Again for the backward recursion we will, in Chapter 6 in Section 6.3, make use of (3.19) to calculate efficiently a-posteriori symbol probabilities for non-coherent detection of DE-QPSK based on trellis-decomposition. If we now combine the results shown in (3.18) and (3.19) for the forward and backward pass, respectively, with (3.13), we obtain for the a-posteriori symbol probabilities

$$\Pr\{a_n|\mathbf{y}\} \propto \sum_{x_{n-1}} \gamma_{n-1}(x_{n-1}) \gamma_n(x_{n-1} a_n), \quad (3.20)$$

which depends only on the likelihoods of the received symbols y_{n-1} and y_n . Hence, for coherent detection of DE-QPSK with iud information symbols, MAP symbol detection reduces to a symbol-by-symbol detection strategy with decision rule

$$\hat{a}_n = \arg \max_{a_n} \Pr\{a_n|y_n, y_{n-1}\}. \quad (3.21)$$

Now as can be seen from (3.21), two received symbols are required for coherent detection of DE-QPSK with iud information symbols. Adapting the a-posteriori symbol probability given by (3.10) appropriately with $\Pr\{a_n\} = 1/4$, leads to

$$\begin{aligned}\Pr\{a_n|y_n, y_{n-1}\} &= \sum_{x_{n-1}} \frac{\Pr\{x_{n-1}\}}{4p(y_n, y_{n-1}) (2\pi\sigma^2)^2} \\ &\cdot \int_{\phi} \exp\left(-\frac{\sum_{k=0}^1 |y_{n-k} - e^{j\phi} x_{n-k}|^2}{2\sigma^2}\right) p(\phi) d\phi.\end{aligned}\quad (3.22)$$

Section A.1 of Appendix A gives an expression of the a-posteriori symbol probability for coherent detection of DE-QPSK with a phase ambiguity of $\frac{\pi}{2}$ on the AWGN channel. This yields

$$\begin{aligned} \Pr\{a_n|y_n, y_{n-1}\} &= \frac{K_A}{2} \left[\cosh\left(\frac{v_n}{\sigma^2}\right) + \cosh\left(\frac{w_n}{\sigma^2}\right) \right] \\ &\propto \cosh\left(\frac{v_n}{\sigma^2}\right) + \cosh\left(\frac{w_n}{\sigma^2}\right), \end{aligned} \quad (3.23)$$

$$\text{with } \begin{cases} K_A = \frac{\exp\left(-\frac{\sum_{k=0}^1 |y_{n-k}|^2 + 2}{2\sigma^2}\right)}{4p(y_n, y_{n-1})(2\pi\sigma^2)^2} \\ v(a_n) \triangleq v_n = \Re\{y_n a_n^* + y_{n-1}\} \\ w(a_n) \triangleq w_n = \Im\{y_n a_n^* + y_{n-1}\}. \end{cases}$$

From (3.23) can be seen that with iud information symbols the a-posteriori information symbol probability is proportional to the sum of two cosh-functions. Simon and Divsalar in [69] obtained comparable results, see (9) for $M = 4$, by computing the conditional-likelihood function $p(y_n, y_{n-1}|x_n, x_{n-1})$.

3.3.2 Soft-decision bit metrics

The desired soft-decision bit metrics related to transmission n , i.e., the LLRs [37]

$$\lambda_n^i \triangleq \ln \left(\frac{\Pr\{b_i = 1\}}{\Pr\{b_i = 0\}} \right), \quad (3.24)$$

can be expressed as

$$\lambda_n^1 = \ln \left(\frac{e^{m(\pi)} + e^{m(3\pi/2)}}{e^{m(0)} + e^{m(\pi/2)}} \right), \lambda_n^2 = \ln \left(\frac{e^{m(\pi/2)} + e^{m(\pi)}}{e^{m(0)} + e^{m(3\pi/2)}} \right), \quad (3.25)$$

with symbol metric

$$m(\phi) = \ln \left(\Pr\{a_n = e^{j\phi} | \mathbf{y}\} \right), \quad (3.26)$$

where λ_n^1 corresponds to bit b_1 , λ_n^2 to bit b_2 and with Gray mapping conform

$$\begin{array}{c|cccc} b_1 b_2 & 00 & 01 & 11 & 10 \\ \hline a(b_1 b_2) & 1 & e^{j\pi/2} & e^{j\pi} & e^{j3\pi/2} \end{array}.$$

Section A.2 of Appendix A shows details of computing the LLRs from corresponding a-posteriori symbol probabilities given by (3.23). This yields

$$\lambda_n^1 = \ln \left(\frac{\cosh\left(\frac{v(\pi)}{\sigma^2}\right) + \cosh\left(\frac{w(\pi)}{\sigma^2}\right) + \cosh\left(\frac{v(\frac{3\pi}{2})}{\sigma^2}\right) + \cosh\left(\frac{w(\frac{3\pi}{2})}{\sigma^2}\right)}{\cosh\left(\frac{v(0)}{\sigma^2}\right) + \cosh\left(\frac{w(0)}{\sigma^2}\right) + \cosh\left(\frac{v(\frac{\pi}{2})}{\sigma^2}\right) + \cosh\left(\frac{w(\frac{\pi}{2})}{\sigma^2}\right)} \right). \quad (3.27)$$

In addition, with the used Gray-mapping shown above, the soft-decision bit metric λ_n^2 is obtained by interchanging $\frac{\pi}{2}$ with $\frac{3\pi}{2}$ in (3.27). The a-posteriori probability given by (3.23) and the correspondingly LLRs given by (3.25) are functions of the currently received DE-QPSK symbol y_n and the previously received DE-QPSK symbol y_{n-1} . Consequently, the LLRs for bits b_1 and b_2 require only the knowledge of the received DE-QPSK symbol pair (y_{n-1}, y_n) . Figure 3.1 shows the simulation results of coded DE-QPSK with coherent detection, bit-interleaving, and Viterbi-decoding with LLRs computed by (3.27). The de facto-standard $R_c = 1/2$ RCPC code, as described in Section 2.3.2, is used for the encoding. This figure also shows the BER performance of 2SDD for coded DE-QPSK

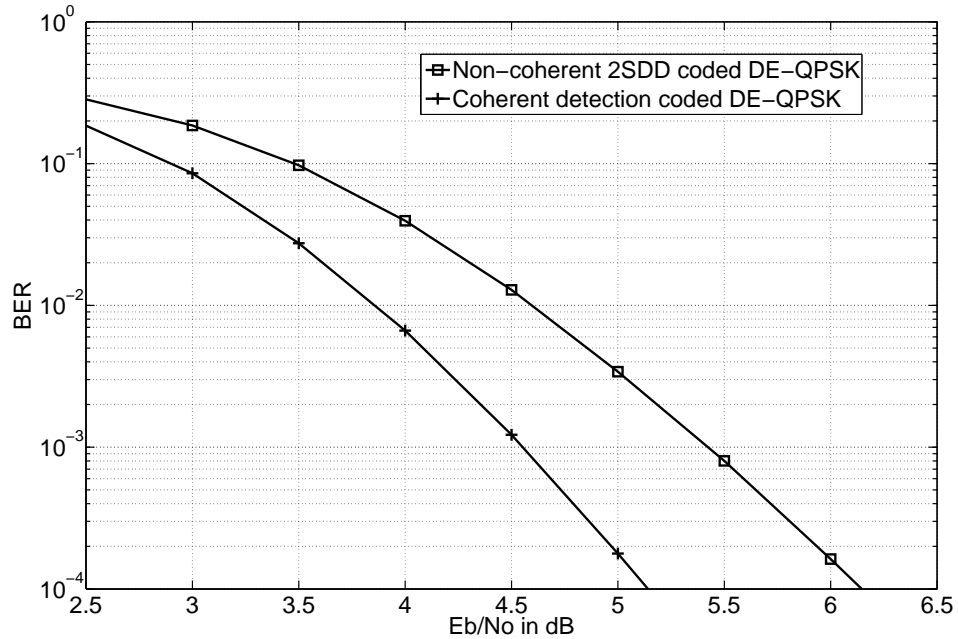


Figure 3.1: BER for AWGN of coherent detection and non-coherent 2SDD of coded DE-QPSK.

with soft-decision bit-metrics based on LLRs and Viterbi decoding. The computation of the soft-decision bit-metrics for 2SDD of DE-QPSK will be discussed in Section 3.4.1. It is shown in Figure 3.1 that coherent detection of coded DE-QPSK improves the required signal to noise with ≈ 1 dB compared to coded DE-QPSK with non-coherent 2SDD at a BER = 10^{-4} .

Max-log-MAP approximation

To avoid logarithmic or exponential functions for reducing the complexity in calculating the a-posteriori symbol probabilities and LLRs, we will discuss the max-log-MAP approximation. The max-log-MAP approximation is selecting the dominant exponential

[30]. A similar method is used by Bottomley et al. in [15]. Section A.3 of Appendix A shows details of applying the max-log-MAP approximation to the LLRs given by (3.27), which yields

$$\lambda_n^1 \approx \frac{1}{\sigma^2} \left\{ \max(|v(\pi)|, |w(\pi)|, |v(\frac{3\pi}{2})|, |w(\frac{3\pi}{2})|) - \max(|v(0)|, |w(0)|, |v(\frac{\pi}{2})|, |w(\frac{\pi}{2})|) \right\}. \quad (3.28)$$

In addition, with the previously used Gray-mapping, the soft-decision bit metric λ_n^2 is obtained by interchanging $\frac{\pi}{2}$ with $\frac{3\pi}{2}$ in (3.28). From (3.28) can be concluded that if the noise variance σ^2 is constant, over all transmissions, it is not actually needed for computing the LLRs. Furthermore, to compute the LLRs, the max-log-MAP approximation requires only subtractions after the maximum is found. Hence, no logarithmic or exponential functions are required for the max-log-MAP approximation. Figure 3.2 shows simulation results of coded DE-QPSK with coherent detection, bit-interleaving, and Viterbi-decoding with a max-log-MAP approximation of the soft-decision bit metric pair $(\lambda_n^1, \lambda_n^2)$ given by (3.28). The de facto-standard $R_c = 1/2$ RCPC code, as described in Section 2.3.2, is used for the encoding. Figure 3.2 demonstrates already good per-

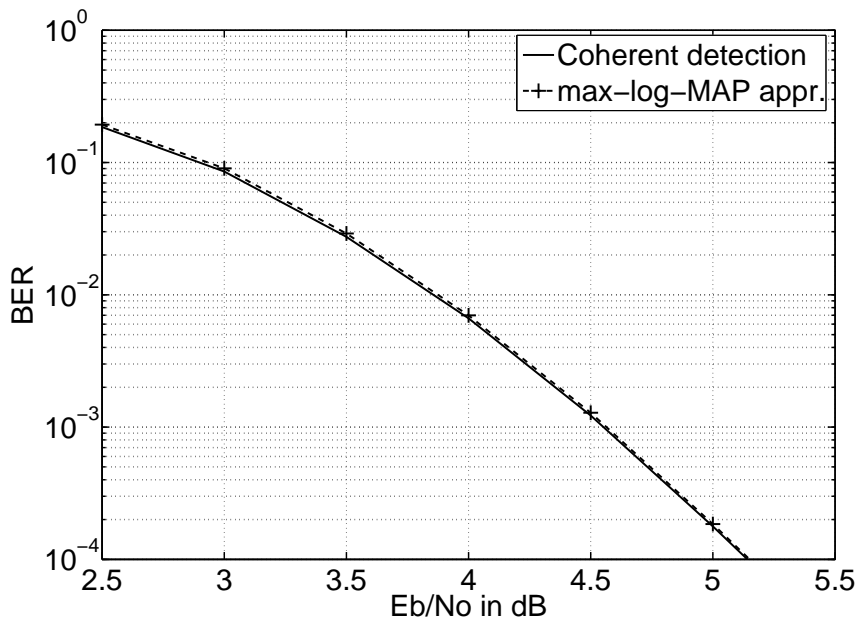


Figure 3.2: *Coherent detection and the max-log-MAP approximation of coded DE-QPSK.*

formances for the max-log-MAP approximation. However we will provide in Chapter 4 even better approximations based on a piecewise linear approximation of the $\ln(\cosh)$ function.

3.3.3 Concluding remarks on coherent detection

In the particular case that the information QPSK-symbols $\{a_n\}$ are iud, coherent MAP symbol detection reduces to coherent symbol-by-symbol detection [19]. Then the soft-decision bit metrics are a function of the currently received DE-QPSK symbol y_n and the previously received DE-QPSK symbol y_{n-1} , and are the sum of cosh-functions. Moreover, coherent symbol-by-symbol detection is a particularly interesting detection strategy if efficient calculation of the soft-decision bit metrics λ_n^i with $i \in \{1, 2\}$ is concerned. Complexity reduction by using a symbol-by-symbol detection strategy will be further discussed in Section 6.3.3 of Chapter 6.

The max-log-MAP approximation results in good performances for coherent detection. However, we will suggest in Chapter 4 some improved approximations. The max-log-MAP approximation requires only subtractions after the maximum is found hence no logarithmic or exponential functions are required, see also (3.28).

Since reference symbols (pilots) are lacking in DAB systems the commonly used detection techniques for DAB-systems are non-coherent detection techniques [79], which is the subject of the next section. We will use the previously obtained results for coherent detection as a reference for the performance of non-coherent detection.

3.4 Non-coherent detection

In this section we will discuss non-coherent detection of DE-QPSK signals. Now, no attempt is made to get any information on the channel phase, i.e., the phase of the carrier plus the phase rotation introduced by the channel. Thus, for non-coherent detection, the carrier frequency is known but the channel phase is assumed to be unknown. However, the channel phase is assumed to be fixed over a block of consecutive DE-QPSK symbols.

3.4.1 Two-symbol differential detection (2SDD)

In this section, differential detection is considered under the condition that the channel phase is uniform. This differential detection technique is known in the literature as non-coherent DQPSK [60, Sec. 5.2.8, p. 272]. For DQPSK, the observation interval has a length of two DE-QPSK symbols, i.e., 2SDD and is widely used. Moreover, 2SDD is also a commonly used non-coherent detection method for DAB receivers [79]. Since we are interested in finding the soft-decision metrics for 2SDD, we will first compute the a-posteriori symbol probabilities. For 2SDD we will use (3.22), hence

$$\Pr\{a_n|y_n, y_{n-1}\} = \sum_{x_{n-1}} \frac{\Pr\{x_{n-1}\}}{4p(y_n, y_{n-1}) (2\pi\sigma^2)^2} \cdot \int_{\phi} \exp\left(-\frac{\sum_{k=0}^1 |y_{n-k} - e^{j\phi} x_{n-k}|^2}{2\sigma^2}\right) p(\phi) d\phi, \quad (3.29)$$

where we now model the channel phase ϕ uniformly distributed over $[0, 2\pi)$ with $p(\phi) = \frac{1}{2\pi}$. Section B.1 of Appendix B shows an expression for the a-posteriori symbol probabilities for non-coherent 2SDD of DE-QPSK on the AWGN channel. This yields

$$\begin{aligned} \Pr\{a_n|y_n, y_{n-1}\} &= K_A I_0\left(\frac{1}{\sigma^2} |y_n a_n^* + y_{n-1}|\right) \\ &\propto I_0\left(\frac{1}{\sigma^2} |y_n a_n^* + y_{n-1}|\right), \end{aligned} \quad (3.30)$$

where $I_0(\cdot)$ is a zero-th order modified Bessel function of the first kind. From (3.30) it can be seen that with iud information symbols the a-posteriori information symbol probability for 2SDD can be expressed in terms of Bessel functions. Divsalar and Simon in [22] obtained comparable results, see (9) for $N = 2$, by computing the conditional-likelihood function $p(y_n, y_{n-1}|x_n, x_{n-1})$.

3.4.2 Soft-decision bit metrics

Section B.2 of Appendix B shows details of computing the LLRs from the corresponding a-posteriori symbol probabilities given by (3.30). This yields

$$\lambda_n^1 = \ln \left(\frac{I_0\left(\frac{1}{\sigma^2} |y_n e^{-j\pi} + y_{n-1}|\right) + I_0\left(\frac{1}{\sigma^2} |y_n e^{-j\frac{3\pi}{2}} + y_{n-1}|\right)}{I_0\left(\frac{1}{\sigma^2} |y_n + y_{n-1}|\right) + I_0\left(\frac{1}{\sigma^2} |y_n e^{-j\frac{\pi}{2}} + y_{n-1}|\right)} \right). \quad (3.31)$$

Here again, with the previously used Gray-mapping, the soft-decision bit metric λ_n^2 is obtained by interchanging $\frac{\pi}{2}$ with $\frac{3\pi}{2}$ in (3.31). The a-posteriori probability given by (3.30) and the corresponding LLRs given by (3.31) are functions, as can be expected with 2SDD, of the currently received DE-QPSK symbol y_n and the previously received DE-QPSK symbol y_{n-1} . Consequently, the LLRs for bits b_1 and b_2 require only the knowledge of the received DE-QPSK symbol pair (y_{n-1}, y_n) . Figure 3.1 in Section 3.3.1 contains the simulation results of coded DE-QPSK with non-coherent 2SDD, bit-interleaving, and Viterbi-decoding with LLRs computed by (3.31). The de facto-standard $R_c = 1/2$ RCPC code, as described in Section 2.3.2, is used for the encoding. It can be observed from this figure that coded DE-QPSK with non-coherent 2SDD requires a ≈ 1 dB higher signal to noise ratio than coherent detection of coded DE-QPSK to obtain a BER = 10^{-4} . Divsalar and Simon [22] showed that by increasing the number of observations on the received DE-QPSK symbols, the performance can be improved compared to 2SDD. This MSDD technique is the subject of the next section.

3.4.3 Multi-symbol differential detection (MSDD)

In this section, we discuss a type of demodulation for DE-QPSK where a block of DE-QPSK symbols is used for detection. We will assume that multiple concatenated DE-QPSK symbols have a similar carrier-phase. Now the observation interval containing the received DE-QPSK symbols might be extended compared to 2SDD. This method is called MSDD, and was introduced by Divsalar and Simon in 1990 [22]. For the AWGN

channel with a time-invariant (unknown) phase, Divsalar and Simon [22] showed that by applying MSDD the performance can be improved relative to 2SDD. They propose to make use of MLSE of the transmitted phases rather than symbol-by-symbol detection as in conventional differential detection [22]. Actually, non-coherent MSDD is a maximum likelihood procedure for finding a block of information symbols after having observed a block of received symbols. For large numbers of observations, the performance of MSDD approaches the performance of coherent detection of DE-QPSK. In this section we will demonstrate, by simulations, that MSDD improves the performances compared to 2SDD also for coded DE-QPSK. Improving performance by increasing the observation interval might be also of interest for DAB receivers. Our simulations are performed with soft-decision bit metrics based on LLRs and Viterbi-decoding. The LLRs are computed in the sequel of this section.

We are interested in finding the soft-decision metric of MSDD for $N + 1$ -length sequences. Adapting the a-posteriori symbol probability given by (3.10) appropriately with $\Pr\{a_n\} = 1/4$, leads to

$$\Pr\{\mathbf{a}|\mathbf{y}\} = \sum_{x_{n-N}} \frac{\Pr\{x_{n-N}\}}{4^N p(y_n, y_{n-1}, \dots, y_{n-N}) (2\pi\sigma^2)^{N+1}} \cdot \int_{\phi} \exp\left(-\frac{\sum_{k=0}^N |y_{n-k} - e^{j\phi} x_{n-k}|^2}{2\sigma^2}\right) p(\phi) d\phi. \quad (3.32)$$

Now, after some manipulations, similar to 2SDD in (3.32) we end up with

$$\begin{aligned} \Pr\{\mathbf{a}|\mathbf{y}\} &= K_{A_N} I_0 \left(\frac{1}{\sigma^2} \left| \sum_{k=0}^N y_{n-k} x_{n-k}^* \right| \right) \\ &= K_{A_N} I_0 \left(\frac{1}{\sigma^2} \left| \sum_{k=0}^{N-1} y_{n-k} \left(\prod_{m=0}^{N-1-k} a_{n-k-m}^* \right) + y_{n-N} \right| \right) \\ &\propto I_0 \left(\frac{1}{\sigma^2} \left| \sum_{k=0}^{N-1} y_{n-k} \left(\prod_{m=0}^{N-1-k} a_{n-k-m}^* \right) + y_{n-N} \right| \right), \end{aligned} \quad (3.33)$$

where

$$K_{A_N} = \frac{\exp\left(-\frac{\sum_{k=0}^N |y_{n-k}|^2 + N + 1}{2\sigma^2}\right)}{4^N p(y_n, \dots, y_{n-N}) (2\pi\sigma^2)^{N+1}}.$$

From (3.33) can be seen that with iud information symbols the a-posteriori information symbol probabilities for MSDD can be expressed in terms of Bessel functions. As one might be expecting, (3.33) for $N + 1 = 2$ gives the same results as (3.30). Divsalar and Simon in [22] obtained comparable results, see (9) and (15), by computing the conditional-likelihood function $p(\mathbf{y}|\mathbf{x})$.

3.4.4 Soft-decision bit metrics

As soft-decision bit metrics we will use LLRs [37]. Assuming iud information symbols $\{a_n\}$ with appropriate bit-interleaving, such that

$$\Pr\{a_n\} = \prod_{i=1}^2 \Pr\{b_n^i\} = \frac{1}{4}, \quad (3.34)$$

the soft-decision bit metrics, $\{\lambda_{n-k}^i\}$ with $i \in \{1, 2\}$ and $k \in \{0, 1, \dots, N-1\}$ each be computed by

$$\lambda_p = \ln \left(\frac{\sum_{C \in \mathbf{C}_p} I_0 \left(\frac{1}{\sigma^2} \left| \sum_{k=0}^{N-1} y_{n-k} \left(\prod_{m=0}^{N-1-k} a_{n-k-m}^* \right) + y_{n-N} \right| \right)}{\sum_{C \in \bar{\mathbf{C}}_p} I_0 \left(\frac{1}{\sigma^2} \left| \sum_{k=0}^{N-1} y_{n-k} \left(\prod_{m=0}^{N-1-k} a_{n-k-m}^* \right) + y_{n-N} \right| \right)} \right), \quad (3.35)$$

where \mathbf{C}_p is the set of all the modulation codewords corresponding to sequences of mapped bits $[b_1, \dots, b_{2N}]$ having $b_p = 1$ and $\bar{\mathbf{C}}_p$ is the set of codewords for which $b_p = 0$, with p is a function of n, k and i . As one might be expecting, (3.35) for $N+1=2$ gives the same results as (3.31). As stated before, the assumption of equal a-priori probabilities will not hold anymore in the iterative case, as will be discussed in Chapter 5. Figure 3.3 shows simulation results of coded DE-QPSK with non-coherent MSDD, bit-interleaving, and Viterbi-decoding for different values of $N+1$. The soft-decision bit metrics are conform (3.35). The de facto-standard $R_c = 1/2$ RCPC code, as described in Section 2.3.2, is used for the encoding. Figure 3.3 shows indeed that the performance improves when $N+1$ increases. As we will show in Chapter 6, increasing the number of observations on the received DE-QPSK symbols is an interesting technique to improve the performance of coded DE-QPSK systems like the DAB system.

3.4.5 Concluding remarks on non-coherent differential detection

The soft-decision bit metrics for non-coherent differential detection of coded DE-QPSK with Gray-mapping can be computed using zero-th order modified Bessel functions of the first kind.

The well-known DQPSK detection technique is the particular case that the observation interval is only two symbols long, i.e., 2SDD. With 2SDD the previously received DE-QPSK symbol is used as the (noisy) reference for differential detection. In this case the soft-decision bit metrics are a function of the currently received DE-QPSK symbol y_n and the previously received DE-QPSK symbol y_{n-1} .

It is further shown, by simulations, that the required signal-to-noise ratio for 2SDD is ≈ 1 dB higher than the required signal-to-noise ratio of coherent detection of coded DE-QPSK at a BER = 10^{-4} . However, as is demonstrated in Section 3.4.3 by simulations, an expansion of the observation interval from two received symbols to multiple received

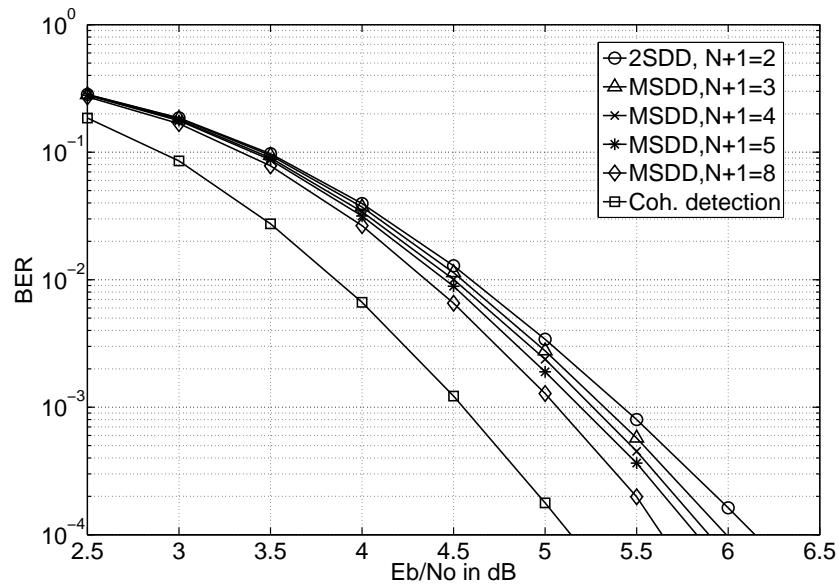


Figure 3.3: MSDD for coded DE-QPSK.

symbols, i.e., moving from 2SDD to MSDD improves the performance of coded DE-QPSK. To improve the performance of coded DE-QPSK even further we will introduce, in Chapter 5, iterative techniques for decoding and demodulation. It needs to be stated here, that the performance improvement of coded DE-QPSK using iterative techniques is partly obtained from increasing the number of received DE-QPSK symbols.

In the next chapter, Chapter 4, it is demonstrated, as an extension to the results known in the literature, that an approximation of MAP symbol detection for $\frac{\pi}{4}$ -DE-QPSK, based on selecting dominant exponentials, leads to MAP sequence detection. Moreover, to improve the performance relative to MAP symbol detection, better approximations for the a-posteriori symbol probabilities and LLRs for coherently detected $\frac{\pi}{4}$ -DE-QPSK are proposed.

Chapter 4

A-Posteriori Symbol Probabilities and Log-Likelihood Ratios for Coherently Detected $\frac{\pi}{4}$ -DE-QPSK

In the previous chapter, Chapter 3, we observed that by applying MSDD techniques, a DAB receiver approaches the performance of a receiver that performs coherent detection of $\frac{\pi}{4}$ -DE-QPSK with soft-decision Viterbi decoding. For that reason a-posteriori symbol probabilities and LLRs for coherently detected $\frac{\pi}{4}$ -DE-QPSK are studied in the current chapter¹.

4.1 Outline

In this chapter, we discuss *a-posteriori* (AP) symbol-probabilities and optimal symbol-detection, but also LLRs for $\frac{\pi}{4}$ -DE-QPSK systems when coherent detection is performed. Our analysis carries over to common DE-QPSK. We assume that the detector has perfect knowledge of the carrier frequency. The carrier phase is known up to an ambiguity of a multiple of $\frac{\pi}{2}$.

Colavolpe [19] mentioned that for differentially encoded phase-shift keying, MAP sequence detection has a symbol-error performance *practically identical* to MAP symbol detection. We will demonstrate here that MAP sequence detection can be regarded as a straightforward approximation of MAP symbol detection. We will also discuss a better approximation of MAP symbol detection, which outperforms MAP sequence detection.

¹This chapter is based on the paper published as *W.J. van Houtum and F.M.J. Willems, "A-Posteriori Symbol Probabilities and Log-Likelihood Ratios for Coherently Detected $\frac{\pi}{4}$ -DE-QPSK", IEEE Communications Letters, vol. 15, no. 2, pp. 160-162, Feb. 2011.*

This improvement is based on a piecewise-linear approximation of the logarithm of the hyperbolic cosine.

In the second part of this chapter we will focus on coded systems and investigate a-posteriori LLRs, and some approximations for these ratios. Although the performance gain is small we show that practically optimal performance can be achieved again using the piecewise-linear approximation mentioned above.

4.2 A-Posteriori probabilities

In the sequence $\mathbf{s} = s_0, s_1, \dots, s_N$ of transmitted $\frac{\pi}{4}$ -DE-QPSK code symbols, the symbols $s_n \in \mathcal{X}_e = \{e^{j\pi l/2}, l = 0, 1, 2, 3\}$ for n even and the symbols $s_n \in \mathcal{X}_o = \{e^{j\pi l/2 + \pi/4}, l = 0, 1, 2, 3\}$ for n odd. Furthermore, these symbols are determined by the differential encoding rule

$$s_n = b_n s_{n-1}, \text{ for } n = 1, 2, \dots, N, \quad (4.1)$$

where the first symbol $s_0 \in \mathcal{X}_e$, with $\Pr\{s_0 = e^{j\pi l/2}\} = 1/4$, for $l = 0, 1, 2, 3$.

As in Colavolpe [19] we consider the case where the information symbols $\mathbf{b} = b_1, b_2, \dots, b_N$ are independent of each other and uniformly distributed (*iud*) over $\mathcal{X} = \{e^{j\pi l/2 + \pi/4}, l = 0, 1, 2, 3\}$. Denoting the received sequence by $\mathbf{r} = r_0, r_1, \dots, r_N$, we can write for channel output r_n for $n = 0, 1, \dots, N$

$$r_n = s_n + n_n, \quad (4.2)$$

where n_n is circularly symmetric complex white Gaussian noise with variance σ^2 per component.

Now, as in (13) in [19], the AP of an *iud* $\frac{\pi}{4}$ -DE-QPSK information symbol b_n for $n = 1, 2, \dots, N$ can be expressed as

$$\Pr\{b_n | \mathbf{r}\} \propto \begin{cases} \sum_{s_{n-1} \in \mathcal{X}_e} \exp\left(\frac{1}{\sigma^2} \Re\{s_{n-1}^* [r_n b_n^* + r_{n-1}]\}\right), & n \text{ odd,} \\ \sum_{s_{n-1} \in \mathcal{X}_o} \exp\left(\frac{1}{\sigma^2} \Re\{s_{n-1}^* [r_n b_n^* + r_{n-1}]\}\right), & n \text{ even.} \end{cases} \quad (4.3)$$

which results in the MAP symbol decision rule

$$\hat{b}_n = \arg \max_{b_n} \Pr\{b_n | \mathbf{r}\}. \quad (4.4)$$

For MAP sequence detection in the case of *iud* information symbols (i.e., maximum-likelihood (ML) detection), the decision rule is, see (17) in [19],

$$\hat{b}_n = \hat{s}_n \hat{s}_{n-1}^*, \quad (4.5)$$

with

$$\hat{b}_n = \arg \min_{b_n} |r_n - s_n|^2. \quad (4.6)$$

Now consider (4.3) for $n - 1$ is even and for some fixed $b_n \in \mathcal{X}$. If we define

$$v_n \triangleq r_n b_n^* + r_{n-1}, \quad (4.7)$$

we obtain

$$\begin{aligned} \Pr\{b_n | \mathbf{r}\} &\propto \exp\left(\frac{1}{\sigma^2} \Re(v_n)\right) + \exp\left(-\frac{1}{\sigma^2} \Im(v_n)\right) \\ &\quad + \exp\left(-\frac{1}{\sigma^2} \Re(v_n)\right) + \exp\left(\frac{1}{\sigma^2} \Im(v_n)\right) \\ &= 2 \cosh(R) + 2 \cosh(I), \end{aligned} \quad (4.8)$$

with $R \triangleq \frac{\Re(v_n)}{\sigma^2}$ and $I \triangleq \frac{\Im(v_n)}{\sigma^2}$. Note that for odd $n - 1$ we get an identical expression for $\Pr\{b_n | \mathbf{r}\}$ if we define

$$v_n \triangleq (r_n b_n^* + r_{n-1}) e^{j\frac{\pi}{4}}. \quad (4.9)$$

4.2.1 First approximation

From (4.8) we may conclude that the AP of an iud information symbol is proportional to the sum of two cosh-functions. To avoid underflow, calculations are often carried out in the logarithmic domain, and therefore instead of (4.8) the metric

$$m_0(b_n) \triangleq \ln(2 \cosh(R) + 2 \cosh(I)), \quad (4.10)$$

can be applied. As approximation for $m_0(b_n)$ we can now use

$$m_1(b_n) \triangleq \ln\left(\max\left(e^{|R|}, e^{|I|}\right)\right) = \max(|R|, |I|). \quad (4.11)$$

Note that this approximation avoids calculating exponentials and logarithms. To demonstrate that this approximation leads to a performance identical to that of MAP sequence detection, note that for iud information symbols, the MAP sequence decision rule (4.5) can be written as:

$$\begin{aligned} \hat{b}_n &= \arg \min_{b_n} \min_{s_{n-1}} (|r_n - s_{n-1} b_n|^2 + |r_{n-1} - s_{n-1}|^2) \\ &= \arg \max_{b_n} \max_{s_{n-1}} \frac{1}{\sigma^2} \Re\{s_{n-1}^* (r_n b_n^* + r_{n-1})\}. \end{aligned} \quad (4.12)$$

If we now call

$$m_{\text{seq}}(b_n) \triangleq \max_{s_{n-1}} \frac{1}{\sigma^2} \Re \{ s_{n-1}^* (r_n b_n^* + r_{n-1}) \}, \quad (4.13)$$

then, for $n - 1$ even and all b_n we obtain

$$\begin{aligned} m_{\text{seq}}(b_n) &= \max \left(\frac{\Re \{v_n\}}{\sigma^2}, -\frac{\Im \{v_n\}}{\sigma^2}, -\frac{\Re \{v_n\}}{\sigma^2}, \frac{\Im \{v_n\}}{\sigma^2} \right) \\ &= \max (|R|, |I|), \end{aligned} \quad (4.14)$$

which is identical to our first approximation given by (4.11). As a consequence, our first approximation will result in the same symbol estimates as MAP sequence detection.

In the next subsection we discuss a better approximation for the AP symbol metric $m_0(b_n)$ than $m_1(b_n)$.

4.2.2 Second approximation

To improve approximation $m_1(b_n)$ of $m_0(b_n)$, we propose

$$m_2(b_n) \triangleq f \left(\frac{R+I}{2} \right) + f \left(\frac{R-I}{2} \right) + 2 \ln 2, \quad (4.15)$$

where we used the identity

$$\cosh(R) + \cosh(I) = 2 \cosh \left(\frac{R+I}{2} \right) \cosh \left(\frac{R-I}{2} \right), \quad (4.16)$$

and approximated the $\ln(\cosh(g))$ by the piecewise-linear function:

$$f(g) = \begin{cases} |g| - \ln 2, & |g| > \ln 2 \\ 0, & |g| \leq \ln 2. \end{cases} \quad (4.17)$$

To see that this is reasonable note that for large $|g|$ one of the exponentials in $\cosh(|g|)$ dominates, which results in linearity.

4.2.3 Simulations

We have simulated the first and second approximation in terms of the uncoded SER versus the signal-to-noise ratio $E_s/N_0 = \frac{1}{2\sigma^2}$, where E_s is the received signal energy of a code symbol and $N_0/2$ the two-sided power spectral density of the noise. The results of these simulations can be found in Fig. 4.1, together with the optimum symbol-detection results, i.e., the results based on metrics $m_0(\cdot)^2$.

It can be observed that optimal symbol detection and our second approximation outperform MAP sequence detection (first approximation). In contrast to the other two methods, MAP sequence detection requires no knowledge of the noise variance however.

²To clearly view the results a small range of a low value of the E_s/N_0 is shown.

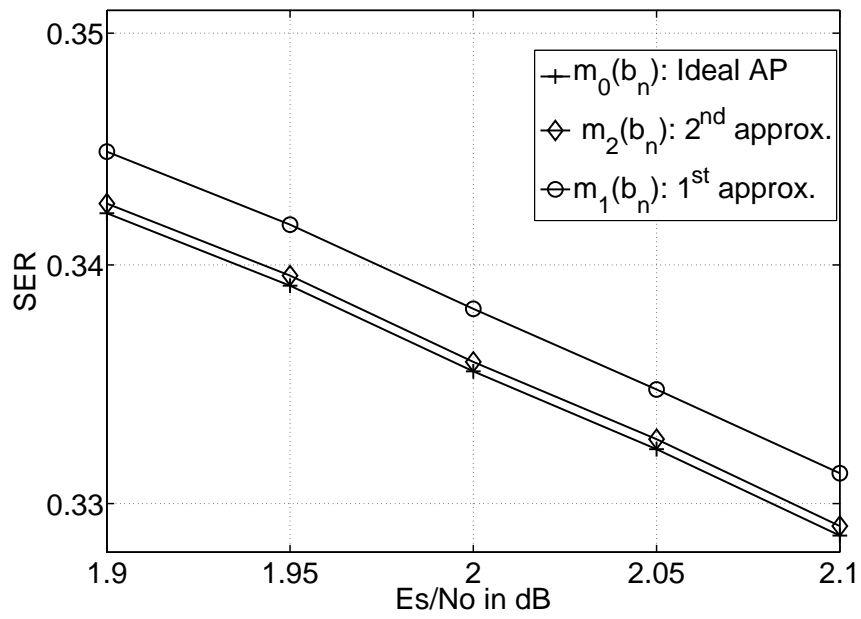


Figure 4.1: SER-performance for ideal and approximated versions of the APs.

4.3 Log-likelihood ratios

In a coded situation the detector is followed by a Viterbi-decoder which needs soft-information about the coded bits. The desired metrics for transmission n , i.e., the LLRs [37], in the case of Gray encoding, see [25], can be expressed as

$$\lambda_n^1 = \ln \left(\frac{e^{m(\frac{3\pi}{4})} + e^{m(\frac{5\pi}{4})}}{e^{m(\frac{\pi}{4})} + e^{m(\frac{7\pi}{4})}} \right), \lambda_n^2 = \ln \left(\frac{e^{m(\frac{5\pi}{4})} + e^{m(\frac{7\pi}{4})}}{e^{m(\frac{\pi}{4})} + e^{m(\frac{3\pi}{4})}} \right), \quad (4.18)$$

where λ_n^1 corresponds to the first bit and λ_n^2 to the second bit. Ideally the symbol-metric $m(\cdot)$ should be $m_0(\cdot)$ but to reduce complexity we could also use an approximated version, i.e., $m_1(\cdot)$ or $m_2(\cdot)$.

4.3.1 Third approximation

To avoid arithmetic based on exponential and logarithmic functions, we introduce a third approximation, which just as our first approximation, selects the dominant exponential. Observe that a LLR is a difference of the logarithm of a numerator and the logarithm of a denominator. Both the numerator and denominator consist of two exponentials. Therefore we make the following approximation

$$\ln \left(e^{m(b)} + e^{m(b')} \right) \approx \max(m(b), m(b')). \quad (4.19)$$

Note, that in this approximation we can use the ideal value $m_0(\cdot)$ for $m(\cdot)$, but also one of its approximations $m_1(\cdot)$ or $m_2(\cdot)$. If we take $m_1(\cdot)$ we get an approximation which is identical to the approximation (10) by Bottomley et al. [15].

4.3.2 Fourth approximation

Motivated by the fact that our first approximation could be improved, we propose a fourth approximation, similar to the second one. Based on the identity

$$e^{m(b)} + e^{m(b')} = 2e^{\frac{m(b)+m(b')}{2}} \cosh \left(\frac{m(b) - m(b')}{2} \right), \quad (4.20)$$

and (4.17), we can approximate

$$\begin{aligned} \ln \left(e^{m(b)} + e^{m(b')} \right) & \\ \approx \frac{m(b) + m(b')}{2} + f \left(\frac{m(b) - m(b')}{2} \right) + \ln 2. & \end{aligned} \quad (4.21)$$

Again we could use the ideal symbol metrics $m_0(\cdot)$ for $m(\cdot)$, but also the approximations $m_1(\cdot)$ or $m_2(\cdot)$.

4.3.3 Simulations

Fig. 4.2 shows simulation results when we use the ideal symbol-metrics $m_0(\cdot)$ with the ideal LLR computations as in (4.18), but also for the case where we use the approximated symbol-metrics $m_1(\cdot)$ in combination with LLR-approximation three (4.19), and for the case in which we use the improved symbol-metrics $m_2(\cdot)$ together with the improved fourth approximation (4.21). The BER versus the signal-to-noise ratio $E_b/N_0 = \frac{1}{2\sigma^2}$ is shown, where E_b is the received signal energy of an information bit.

We used, conform [25], the de-facto industry standard $R_c = \frac{1}{2}$, $K = 7$, convolutional code with generator polynomials $g_0 = 133$ and $g_1 = 171$. Its output is randomly bit-interleaved and differentially encoded after Gray mapping. The BER simulations are performed with the Viterbi-algorithm for decoding the convolutional code. Note that even if we do ideal LLR-computations based on ideal symbol-metrics $m_0(\cdot)$, the concatenated demodulator/decoder can only be close to optimal. As expected the ideal symbol-metrics $m_0(\cdot)$ together with ideal LLR-computation result in the best (ideal) results, and using approximated symbol-metrics $m_1(\cdot)$ together with the LLR-approximation three yields the worst results. Using symbol-metrics $m_2(\cdot)$ together with LLR-approximation four achieves results that are only slightly worse than the best results. Note again that the combination of the second and fourth approximation requires knowledge of the noise variance³.

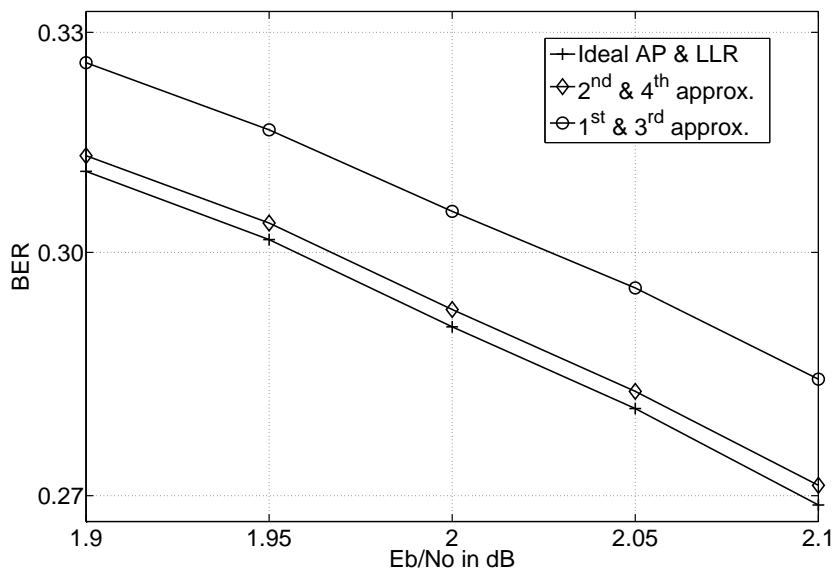


Figure 4.2: BER-performance for ideal and approximated versions of the LLRs.

³Again, to clearly view the results a small range of a low value of the E_b/N_0 is shown.

4.4 Conclusions

We have shown that a straightforward approximation of MAP symbol detection is actually identical to MAP sequence detection, which is suboptimal for coherent symbol detection of $\frac{\pi}{4}$ -DE-QPSK. This approximation is based on selecting dominant exponentials. Simulations showed that MAP symbol detection outperforms MAP sequence detection, but the difference is small. In this way we have made the statement of Colavolpe [19] more precise. Moreover we have proposed an improved approximation of the symbol-metrics that results in a symbol-error performance close to optimal. The resulting arithmetic is based on a piecewise-linear function.

In the coded case LLRs must be computed. Apart from exact computation of these LLRs, we have considered an approximation based again on selecting dominant exponentials, i.e., Bottomley's [15] approximation, but also a better approximation based on the piecewise-linear function mentioned before. Improved LLR-approximation combined with improved symbol-metric approximation gives a performance close to that of exact LLR-computation based on exact symbol metrics.

Selecting dominant exponentials is similar to max-log-MAP detection applied in turbo-decoders, see Robertson [63]. Our improved approximations are based on a piecewise-linear fit for the logarithm of a hyperbolic cosine and can be shown to be similar to approximations of the Jacobian proposed by Talakoub et al. in [70] for turbo decoding. While the particular examples of our considered approximations show modest gains in performance, it provides a way of improving performance when needed.

Chapter 5

The Shannon Limit and Some Codes Approaching It

Iterative decoding techniques for serially concatenated convolutional codes lead to good results for the concatenation of convolutional and differential encoding, also referred to as Turbo-DPSK. Later in the thesis iterative decoding techniques are proposed for DAB-like streams. Therefore we explain in this chapter, in a tutorial setting, the iterative decoding procedures corresponding to these serially concatenated codes¹.

5.1 Outline

Shannon [66][67] showed that reliable communication is possible only if the transmission rate is smaller than channel capacity. For additive white Gaussian noise channels this capacity is determined by the signal-to-noise ratio. Conversely we can say that for every transmission rate there is a minimum signal-to-noise ratio, the so-called Shannon limit, that can lead to reliable communication. Without coding the signal-to-noise ratio needed to achieve reliable communication is roughly eight dB larger than the Shannon limit (at a rate of one bit per transmission). With a very simple rate-1/2 convolutional code this gap can be reduced to roughly six dB. The gap can be bridged, up to roughly one dB, by applying very simple rate-1/3 turbo-codes (Berrou, Glavieux, & Thitimasjima [11]). Turbo codes are based on parallel concatenation of two convolutional codes. Decoding is done iteratively. In 1996, Benedetto & Montorsi [9] proposed an iterative decoding procedure for serially concatenated convolutional codes. These codes perform slightly better than turbo-codes. It is the objective of this chapter to discuss and explain the iterative decoding procedures behind these codes. These procedures are based on modified versions of the BCJR algorithm [4]. Our approach is similar to the approach Gallager [32] followed to investigate iterative procedures for decoding LDPC codes.

¹This chapter is based on a paper that has been submitted as *F.M.J. Willems and W.J. van Houtum, "The Shannon Limit and Some Codes Approaching It"*, to *IEEE Signal Processing Magazine*.

5.2 Some information theory first

5.2.1 Capacity of the AWGN Channel

Shannon [66][67] introduced the notion of channel capacity. For an AWGN channel, see Fig. 5.1, this capacity is given by the expression

$$\bar{C}_b = \frac{1}{2} \log_2(1 + \text{SNR}) \text{ bit/transmission}, \quad (5.1)$$

where SNR denotes the so-called signal-to-noise ratio.

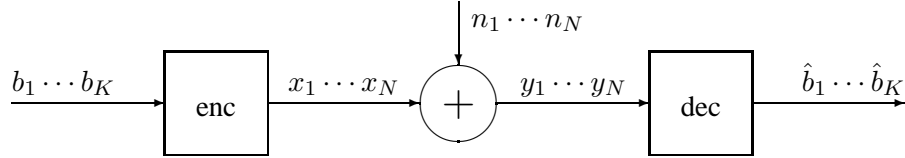


Figure 5.1: Additive white Gaussian noise channel.

To explain what capacity is, we assume that communication over this channel is arranged in blocks. In each block the encoder accepts a message sequence b_1, b_2, \dots, b_K of K independent and uniformly distributed binary digits, and transforms it into a sequence x_1, x_2, \dots, x_N of N real-valued channel input symbols. Transmitting this sequence, also called codeword, requires N channel transmissions, one for each symbol. The rate of the encoder is therefore $r_b = K/N$ bit per transmission. Transmission over the channel is not ideal and the channel output sequence y_1, y_2, \dots, y_N is a noisy version of the input sequence, i.e., the channel adds a noise sequence n_1, n_2, \dots, n_N to the channel input sequence, and the resulting channel output sequence y_1, y_2, \dots, y_N is presented to the decoder. This decoder must produce estimates $\hat{b}_1, \hat{b}_2, \dots, \hat{b}_K$ of the transmitted message digits. The noise variables N_1, N_2, \dots, N_N are assumed to be independent Gaussians, all having mean 0 and unit variance. Therefore the signal-to-noise ratio $\text{SNR} = E[\sum_{i=1,N} X_i^2] / E[\sum_{i=1,N} N_i^2] = E[\sum_{i=1,N} X_i^2] / N$. Shannon showed that, when the channel capacity is \bar{C}_b , there exist encoders with rate $r_b < \bar{C}_b$ such that the message digit error probability can be made arbitrarily small by increasing N , while this is impossible for rates $r_b > \bar{C}_b$.

From capacity expression (5.1) it follows that for reliable transmission over an AWGN channel at rate r_b , the signal-to-noise ratio has to satisfy

$$\text{SNR} \geq 2^{2\bar{C}_b} - 1 \geq 2^{2r_b} - 1. \quad (5.2)$$

The minimum signal-to-noise ratio $2^{2r_b} - 1$ is called the Shannon limit. E.g., for $r_b = 1/3$ bit per transmission the Shannon limit is $2^{2/3} - 1 = 0.587$ or -2.31 dB.

5.2.2 Binary channel inputs

It would be simpler to transmit, instead of sequences of N real-valued channel input-symbols, binary sequences of length N , i.e., sequences consisting of N symbols from

alphabet $\{-\sqrt{\text{SNR}}, +\sqrt{\text{SNR}}\}$. However, as always there is a penalty for making things simpler. Fortunately for rates $r_b \ll 1$ the penalty is negligible, and, e.g., for $r_b = 1/3$ the minimum SNR = 0.595 or -2.26 dB, resulting in a penalty of only 0.05 dB.

5.2.3 From sequences to waveforms

So far we have discussed only transmission of sequences over a discrete-time AWGN channel. The discrete time AWGN channel models transmission of power- and bandlimited waveforms over a waveform channel, i.e., a channel that adds white noise to its input. If the transmitter power is P_b , the channel bandwidth W_b , and the power spectral density of the white noise is $N_0/2$, the capacity in bit per second of this bandlimited waveform channel is given by

$$C_b = W_b \log_2 \left(1 + \frac{P_b}{N_0 W_b} \right) \text{ bit/second.} \quad (5.3)$$

Comparing this waveform-channel capacity with its discrete-time counterpart in (5.1) we can observe that transmitting a second of waveform is equivalent to performing $2W_b$ transmissions, which is not unnatural having the sampling theorem in mind. For the signal-to-noise ratio in the waveform domain we can write

$$\text{SNR} = \frac{P_b}{N_0 W_b}. \quad (5.4)$$

The equivalence between bandlimited waveform channel and discrete-time channel has been discussed in full detail by Wozencraft and Jacobs [78].

5.3 Uncoded binary transmission

Uncoded binary transmission, i.e., transmitting independent and equally likely message symbols from $\{-\sqrt{\text{SNR}}, +\sqrt{\text{SNR}}\}$ results in a bit-error probability

$$P_e^{\text{unc}} = Q(\sqrt{\text{SNR}}), \quad (5.5)$$

where $Q(z) \triangleq \int_z^\infty \frac{1}{\sqrt{2\pi}} \exp(-a^2/2) da$. Fig. 5.2 contains a plot of this error probability as a function of the signal-to-noise ratio SNR.

The figure also contains the Shannon limit which is $2^2 - 1 = 3$ or 4.77 dB for rate $r_b = 1$. It can be checked that an SNR of roughly 12.6 dB is needed to obtain practically error-free behavior, which (here) is an error probability of 10^{-5} . Note that uncoded binary transmission is roughly 7.8 dB above the Shannon limit at a BER of 10^{-5} .

To achieve the Shannon limit we must find the codes whose existence is guaranteed by Shannon's result. It took almost fifty years before codes were discovered with a non-vanishing rate² that approximate the Shannon limit within one dB. In what follows we will focus on these codes and discuss how they are constructed and how they can be decoded.

²Golay [34] showed already in 1949 that Shannon's limit can be achieved using pulse-position modulation (PPM), however only for rates r_b approaching 0.

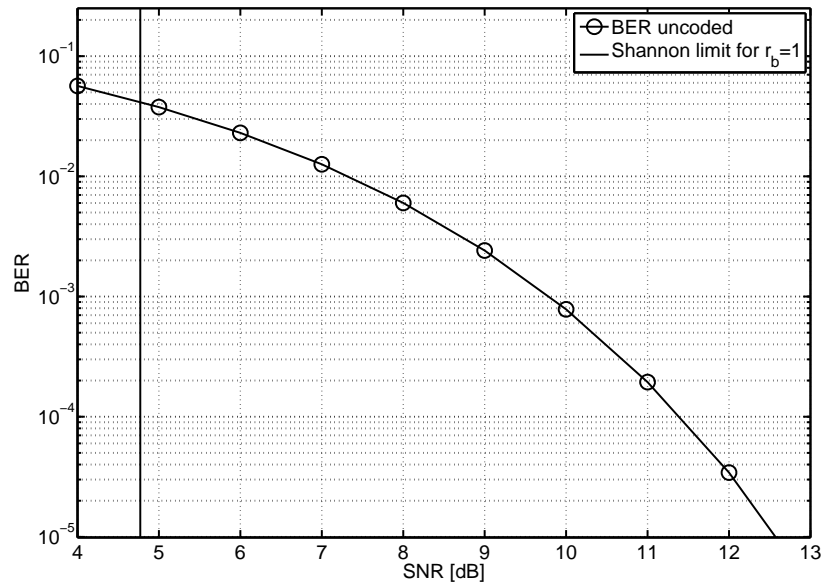


Figure 5.2: Bit-error probability for uncoded binary transmission.

5.4 Linear codes incur no loss

Shannon demonstrated that capacity-achieving codes exist. It would be nice if these codes would had some structure that could lead to simple encoding and decoding procedures. Elias [24] demonstrated that codes whose structure is linear, can achieve capacity for channels with binary inputs being symmetric in some manner. Our AWGN vector channel with binary input alphabet $\{-\sqrt{\text{SNR}}, +\sqrt{\text{SNR}}\}$ has this symmetry. Therefore from now on we can focus on linear codes. An encoder based on a linear code first determines the codeword (c_1, c_2, \dots, c_N) by multiplying the binary input vector (b_1, b_2, \dots, b_K) with an $N \times K$ matrix G , as follows:

$$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{pmatrix} = \begin{pmatrix} g_{11} & g_{12} & \cdots & g_{1K} \\ g_{21} & g_{22} & \cdots & g_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ g_{N1} & g_{N2} & \cdots & g_{NK} \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_K \end{pmatrix}. \quad (5.6)$$

It is assumed that all elements of (b_1, b_2, \dots, b_K) and of matrix G are either 0 or 1, and that operations are based on modulo-2 arithmetic. Matrix G is called the generator matrix. The channel input sequence (x_1, x_2, \dots, x_N) finally results from

$$x_n = (2c_n - 1)\sqrt{\text{SNR}}, \text{ for } n = 1, 2, \dots, N. \quad (5.7)$$

Hamming Codes [39] and BCH codes [40], [14], [13] are two elementary types of error-correcting codes fitting into the linear framework. These codes are referred to as linear block codes. In the next section we will discuss another type of linear codes, i.e., the convolutional codes.

5.5 Convolutional codes

Convolutional codes were proposed by Elias [24]. Although they can be cast into a matrix form, it is easier to describe these codes by describing the corresponding encoder. In Fig. 5.3 we see an encoder that appears in almost every basic text on convolutional codes. It consists of two connected delay elements and two modulo-2 adders. The output of a delay element at time k is equal to the input at time $k - 1$, where the time k is assumed to be integer.

Let K be the number of message digits that is to be encoded. At time $k = 1$ both delay elements are in state zero, i.e., $s_{11} = s_{21} = 0$. For the message digits b_k that enter the encoder we assume that $b_k \in \{0, 1\}$, for $k = 1, 2, \dots, K$. We assume that message digits are equiprobable. Following the message digits there are two so-called tail-bits $b_{K+1} = b_{K+2} = 0$ shifted into the encoder, driving both delay elements into state zero, hence $s_{1,K+3} = s_{2,K+3} = 0$.

For the outputs c_{1k} and c_{2k} of the encoder we then have that

$$\begin{aligned} c_{1k} &= b_k \oplus s_{1k} \oplus s_{2k}, \\ c_{2k} &= b_k \oplus s_{2k}, \text{ for } k = 1, 2, \dots, K + 2, \end{aligned} \quad (5.8)$$

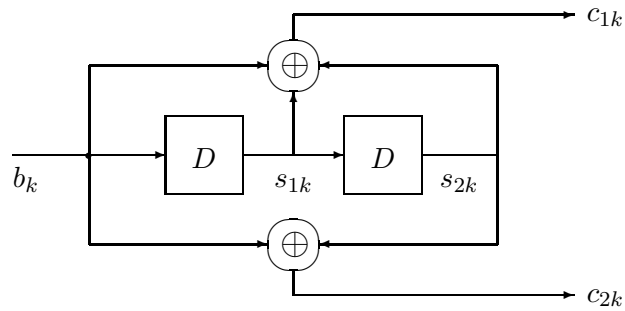


Figure 5.3: Convolutional encoder for a $(7, 5)$ -code.

while the states s_{1k} and s_{2k} satisfy $s_{1k} = b_{k-1}$, and $s_{2k} = s_{1,k-1}$, for $k = 2, 3, \dots, K + 3$. We call this code a $(7, 5)$ -code since the first code output c_{1k} is connected to b_k, b_{k-1} and b_{k-2} while the second output c_{2k} is based on b_k and b_{k-2} , but not on b_{k-1} .

There are now four possible encoder-states, hence $(s_{1k}, s_{2k}) \in \{(0, 0), (0, 1), (1, 0), (1, 1)\}$, and the encoder starts and stops in the all-zero state $(0, 0)$. Codewords have the form $(c_{11}, c_{21}), (c_{12}, c_{22}), \dots, (c_{1,K+2}, c_{2,K+2})$. Fig. 5.4 contains a so-called trellis representation of the code. The squares represent the encoder-states. They are labeled with $s_1 s_2$. Time advances from left to right. The edges between the states represent the transitions that are possible. The label corresponding to a transition is $b|c_1 c_2$. Observe that $K = 4$ here and that there are $2^K = 16$ different codewords, i.e., 16 different paths in the trellis from the start-state to the stop-state.

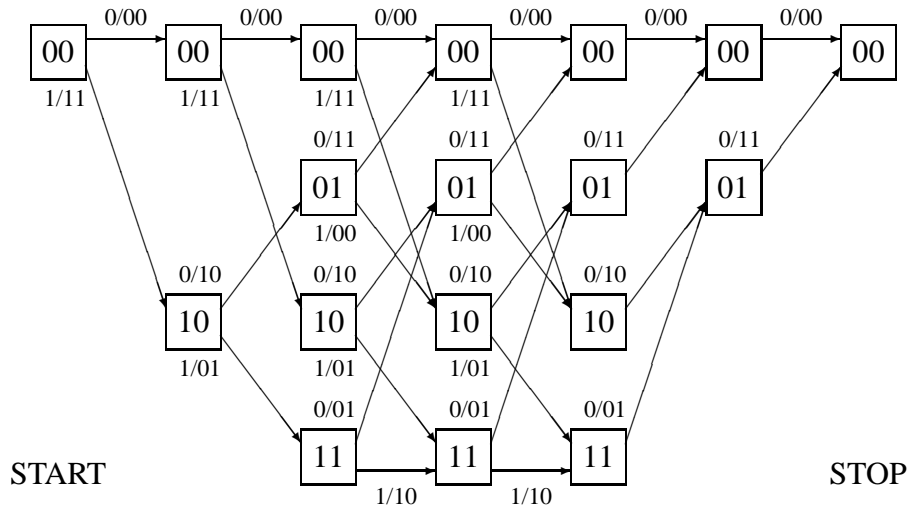


Figure 5.4: Trellis representation of our example code for $K = 4$.

In general there are 2^K equiprobable codewords of length $N = 2(K + 2)$ and hence the rate

$$R_K = \frac{K}{2(K + 2)} = \frac{1}{2} - \frac{1}{K + 2}. \quad (5.9)$$

Since $r_{K \rightarrow \infty} = 1/2$ we call our code a rate-1/2 code. It is easy to see that this code is linear.

Now that we have described the encoding procedure we are ready to discuss the channel and a decoder. We will start with the channel.

5.6 Channel, likelihoods

In the coded case, sequences c_1, c_2, \dots, c_N containing redundancy, i.e., codewords, are transmitted. If we note, see (5.7), that code symbol $c = 0$ results in channel input $x = -\sqrt{\text{SNR}}$ and $c = 1$ in $x = +\sqrt{\text{SNR}}$, and if we assume that $Y = y$ is short for $Y \in [y, y + \Delta)$ for some small enough Δ , then we can determine the conditional probabilities

$$\begin{aligned} P(Y = y|C = 0) &= P(Y \in [y, y + \Delta)|X = -\sqrt{\text{SNR}}) \\ &\approx \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y + \sqrt{\text{SNR}})^2}{2}\right)\Delta, \\ P(Y = y|C = 1) &= P(Y \in [y, y + \Delta)|X = +\sqrt{\text{SNR}}) \\ &\approx \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(y - \sqrt{\text{SNR}})^2}{2}\right)\Delta. \end{aligned} \quad (5.10)$$

These conditional probabilities are called likelihoods. The Δ 's are typically omitted in computations and expressions, and we will do so too.

Our example code is a rate-1/2 convolutional code. For this code the code-symbols leave the encoder in pairs, and since channel transmissions are independent of each other, we can write

$$P(\underline{Y}_k = \underline{y}|\underline{C}_k = \underline{c}) = P(Y_{1k} = y_1|C_{1k} = c_1)P(Y_{2k} = y_2|C_{2k} = c_2), \quad (5.11)$$

where $\underline{Y}_k = (Y_{1k}, Y_{2k})$, $\underline{y} = (y_1, y_2)$, $\underline{C}_k = (C_{1k}, C_{2k})$, and $\underline{c} = (c_1, c_2)$, for $k = 1, 2, \dots, K + 2$.

We have seen before that the penalty from using binary channel inputs can be ignored for small SNR. The penalty for processing, instead of the real-valued channel outputs Y_{1k} and Y_{2k} , their signs, can not be ignored. Making so-called *hard-decisions* results, at low SNR, in a capacity decrease which can be compensated by increasing the SNR by a factor $\pi/2$ which is roughly 1.96 dB, see Viterbi and Omura [77]. Since this loss is quite large, we will focus on processing the likelihoods in (5.10), which can be regarded as the so-called *soft decisions*.

5.7 The BCJR algorithm

5.7.1 Decoding algorithms

An efficient and very popular method for decoding convolutional codes is the Viterbi algorithm [76]. It finds the codeword that has ML³ and it achieves the smallest possible codeword-error probability since all codewords are equally likely. Here however, we will focus on a decoding technique that produces estimates of the message digits instead of an estimate of the codeword. These estimates are optimum in the sense that the bit-error probability is minimized. More important however is that this method provides reliability information on the estimated bits, the so-called soft outputs. This decoding algorithm was proposed by Bahl, Cocke, Jelinek, and Raviv [4]. It turns out to be an essential part of a turbo-decoder as we shall see soon.

As a final remark in this subsection we mention that Hagenauer and Hoehner [37] developed a soft-output Viterbi algorithm (SOVA) which produces soft outputs on message bits estimated by the Viterbi procedure.

5.7.2 Probabilistic structure

Observe that our convolutional encoder is a finite-state machine. It is driven by the message digits, which are independent of each other. Also the channel transmissions are independent of each other. Therefore, when the encoder has reached a certain state at some time instant, all future observations depend on the past ones, only through this state. We can thus write for the joint probability of the variables corresponding to the transmission of message digit b_k , i.e., the variables corresponding to trellis section k , that

$$\begin{aligned}
 & P(\underline{y}_1, \dots, \underline{y}_{k-1}, \underline{s}_k, b_k, \underline{c}_k, \underline{y}_k, \underline{s}_{k+1}, \underline{y}_{k+1}, \dots, \underline{y}_{K+2}) \\
 &= P(\underline{y}_1, \dots, \underline{y}_{k-1}, \underline{s}_k) P(b_k) P(\underline{c}_k, \underline{s}_{k+1} | \underline{s}_k, b_k) \\
 &\quad \cdot P(\underline{y}_k | \underline{c}_k) P(\underline{y}_{k+1}, \dots, \underline{y}_{K+2} | \underline{s}_{k+1}),
 \end{aligned} \tag{5.12}$$

see Fig. 5.5. Observe that the encoder starts trellis section k in a certain state \underline{s}_k . Message digit b_k enters the encoder and an output-pair \underline{c}_k is produced, depending on state \underline{s}_k . The channel generates the output-pair \underline{y}_k depending on input-pair \underline{c}_k . Moreover the input b_k causes the encoder to go to a next state \underline{s}_{k+1} depending on state \underline{s}_k . Then we move to the next trellis section.

Observe that the probabilistic structure and the resulting BCJR algorithm is quite general and based primarily on the finite-state character of the encoder.

³The Viterbi algorithm performs ML decoding, but it can be modified to do MAP codeword decoding.

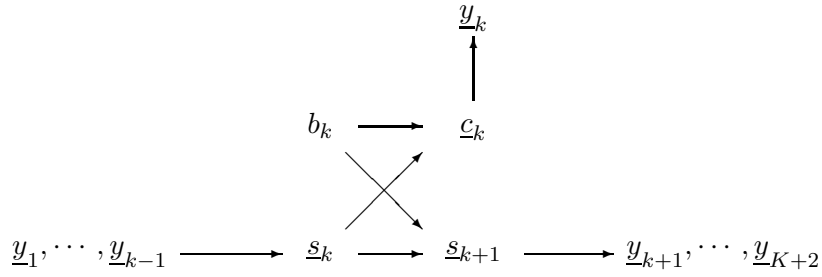


Figure 5.5: Probabilistic structure corresponding to a general finite-state encoder.

5.7.3 Numerator and denominator

It is our objective now to compute, for all $k = 1, 2, \dots, K$ the a-posteriori probabilities (APPs) of the message digit B_k , i.e.,

$$\begin{aligned}
 & P(B_k = b | \underline{y}_1, \dots, \underline{y}_{k-1}, \underline{y}_k, \underline{y}_{k+1}, \dots, \underline{y}_{K+2}) \\
 &= \frac{P(\underline{y}_1, \dots, \underline{y}_{k-1}, B_k = b, \underline{y}_k, \underline{y}_{k+1}, \dots, \underline{y}_{K+2})}{P(\underline{y}_1, \dots, \underline{y}_{k-1}, \underline{y}_k, \underline{y}_{k+1}, \dots, \underline{y}_{K+2})}, \quad (5.13)
 \end{aligned}$$

for $b = 0, 1$. Since the denominator in (5.13) can be obtained from

$$\begin{aligned}
 & P(\underline{y}_1, \dots, \underline{y}_{k-1}, \underline{y}_k, \underline{y}_{k+1}, \dots, \underline{y}_{K+2}) \quad (5.14) \\
 &= \sum_{b=0,1} P(\underline{y}_1, \dots, \underline{y}_{k-1}, B_k = b, \underline{y}_k, \underline{y}_{k+1}, \dots, \underline{y}_{K+2}),
 \end{aligned}$$

we only need to compute the numerator terms.

5.7.4 Splitting up the numerator

Based on the probabilistic structure, we can split up the numerator in (5.13) as follows:

$$\begin{aligned}
 & P(\underline{y}_1, \dots, \underline{y}_{k-1}, B_k = b, \underline{y}_k, \underline{y}_{k+1}, \dots, \underline{y}_{K+2}) \\
 &= \sum_{\underline{s}, \underline{c}, \underline{s}'} P(\underline{y}_1, \dots, \underline{y}_{k-1}, \underline{S}_k = \underline{s}, B_k = b, \underline{C}_k = \underline{c}, \\
 & \quad \underline{Y}_k = \underline{y}_k, \underline{S}_{k+1} = \underline{s}', \underline{y}_{k+1}, \dots, \underline{y}_{K+2}) \\
 &= \sum_{\underline{s}, \underline{c}, \underline{s}'} P(\underline{y}_1, \dots, \underline{y}_{k-1}, \underline{S}_k = \underline{s}) P(B_k = b) \\
 & \quad \cdot P(\underline{C}_k = \underline{c}, \underline{S}_{k+1} = \underline{s}' | \underline{S}_k = \underline{s}, B_k = b) \\
 & \quad \cdot P(\underline{Y}_k = \underline{y}_k | \underline{C}_k = \underline{c}) P(\underline{y}_{k+1}, \dots, \underline{y}_{K+2} | \underline{S}_{k+1} = \underline{s}') \\
 &= \sum_{\underline{s}} P(\underline{y}_1, \dots, \underline{y}_{k-1}, \underline{S}_k = \underline{s}) P(B_k = b) \\
 & \quad \cdot P(\underline{Y}_k = \underline{y}_k | \underline{C}_k = \gamma(\underline{s}, b)) \\
 & \quad \cdot P(\underline{y}_{k+1}, \dots, \underline{y}_{K+2} | \underline{S}_{k+1} = \sigma(\underline{s}, b)). \tag{5.15}
 \end{aligned}$$

where $\gamma(\underline{s}, b)$ is the encoder output and $\sigma(\underline{s}, b)$ the next state if \underline{s} is the current state and b the encoder input.

5.7.5 Alphas and betas and their recursions

If for $k = 1, 2, \dots, K + 2$ we define

$$\begin{aligned}
 \alpha_k(\underline{s}) &\triangleq P(\underline{y}_1, \dots, \underline{y}_{k-1}, \underline{S}_k = \underline{s}), \\
 \beta_{k+1}(\underline{s}') &\triangleq P(\underline{y}_{k+1}, \dots, \underline{y}_{K+2} | \underline{S}_{k+1} = \underline{s}'), \tag{5.16}
 \end{aligned}$$

then we can rewrite (5.15) as

$$\begin{aligned}
 & P(\underline{y}_1, \dots, \underline{y}_{k-1}, B_k = b, \underline{y}_k, \underline{y}_{k+1}, \dots, \underline{y}_{K+2}) \\
 &= \sum_{\underline{s}} \alpha_k(\underline{s}) P(B_k = b) P(\underline{Y}_k = \underline{y}_k | \underline{C}_k = \gamma(\underline{s}, b)) \beta_{k+1}(\sigma(\underline{s}, b)). \tag{5.17}
 \end{aligned}$$

Of crucial importance is now that the α 's and β -s can be computed recursively. First set $\alpha_1(0, 0) = 1$ and $\alpha_1(\underline{s}) = 0$ for all $\underline{s} \neq (0, 0)$. Then (forward recursion) for all trellis sections $k = 1, 2, \dots, K + 2$ and all states \underline{s}'

$$\begin{aligned}
 \alpha_{k+1}(\underline{s}') &= P(\underline{y}_1, \dots, \underline{y}_k, \underline{S}_{k+1} = \underline{s}') \tag{5.18} \\
 &= \sum_{\underline{s}, b \rightarrow \underline{s}'} P(\underline{y}_1, \dots, \underline{y}_{k-1}, \underline{S}_k = \underline{s}) P(B_k = b) P(\underline{Y}_k = \underline{y}_k | \underline{C}_k = \gamma(\underline{s}, b)) \\
 &= \sum_{\underline{s}, b \rightarrow \underline{s}'} \alpha_k(\underline{s}) P(B_k = b) P(\underline{Y}_k = \underline{y}_k | \underline{C}_k = \gamma(\underline{s}, b)),
 \end{aligned}$$

where $\underline{s}, b \rightarrow \underline{s}'$ denote the state-input combinations leading to state \underline{s}' . Similarly set $\beta_{K+3}(0, 0) = 1$ and $\beta_{K+3}(\underline{s}') = 0$ for all $\underline{s}' \neq (0, 0)$. Then (backward recursion) for all trellis sections $k = 1, 2, \dots, K + 2$ and all states \underline{s}

$$\begin{aligned}\beta_k(\underline{s}) &= P(\underline{y}_k, \dots, \underline{y}_{K+2} | \underline{S}_k = \underline{s}) \\ &= \sum_b P(B_k = b) P(\underline{Y}_k = \underline{y}_k | \underline{C}_k = \gamma(\underline{s}, b)) \beta_{k+1}(\sigma(\underline{s}, b)).\end{aligned}\quad (5.19)$$

Note that, just like the Viterbi algorithm, the forward and backward recursions are operations on the trellis corresponding to the code. When both recursions have been carried out, the a-posteriori message digit probabilities for $b = 0, 1$ are computed using (5.17) (and 5.13), which is another trellis operation. The bit-error probability can now be minimized by taking

$$\hat{b}_k = 1 \text{ if } P(B_k = 1 | \underline{y}_1, \dots, \underline{y}_{K+2}) > 1/2, \quad (5.20)$$

and $\hat{b}_k = 0$ otherwise.

The recursions have led to a second name under which the BCJR algorithm became popular, the ‘‘Forward-Backward’’ algorithm.

5.8 Performance convolutional code

In Fig. 5.6 we have plotted the results of a simulation based on the BCJR procedure for our example (7,5)-code. The horizontal axis gives the SNR in dB, along the vertical axis we have the observed bit-error rate. For SNR = -1.0, -0.5, 0, \dots , 6.0 dB we have simulated 2048 codewords with $K = 4096$. Note that the Shannon limit for rate 1/2 is equal to 1, which is 0 dB. Our code needs a signal-to-noise ratio of roughly 5.8 dB to achieve a bit-error rate of 10^{-5} . Relative to the uncoded case we have therefore gained roughly 2 dB using this convolutional code in combination with the BCJR decoding algorithm. We are still 5.8 dB from the Shannon limit however. In the next sections we will show how this gap can be decreased.

5.9 Turbo code, encoder structure

5.9.1 Introduction

One of the most important results in channel-coding theory is probably the invention of Turbo Codes. Berrou, Glavieux, and Thitimajshima [11] proposed this method at the ICC in 1993, and demonstrated its near-Shannon-limit behavior. Both the structure of the code and the proposed decoding technique are quite remarkable. We will first describe the encoder and then discuss the corresponding decoding technique. The encoder is based on (a) systematic recursive convolutional codes, (b) parallel concatenation, and (c) interleaving systematic symbols. The decoder will be discussed in Sec. 5.10.

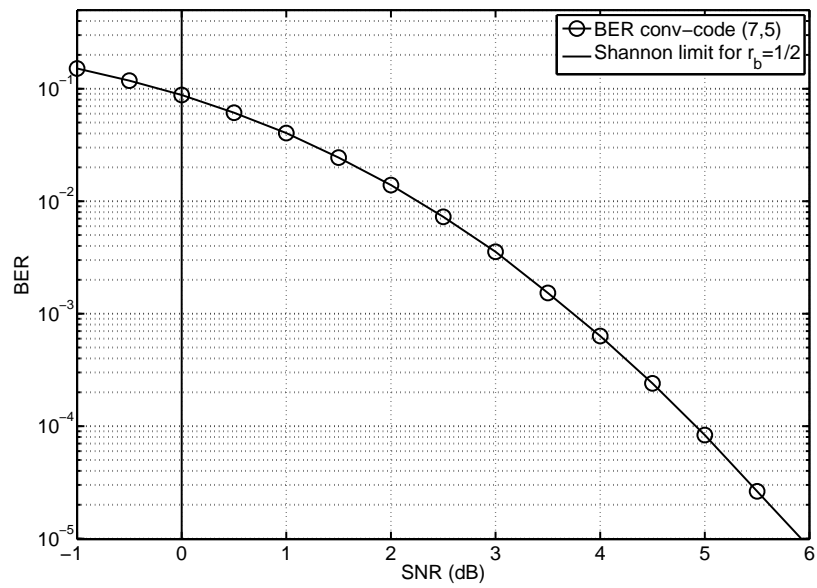


Figure 5.6: Bit-error rate of our example (7,5)-code decoded with the BCJR procedure based on soft decisions.

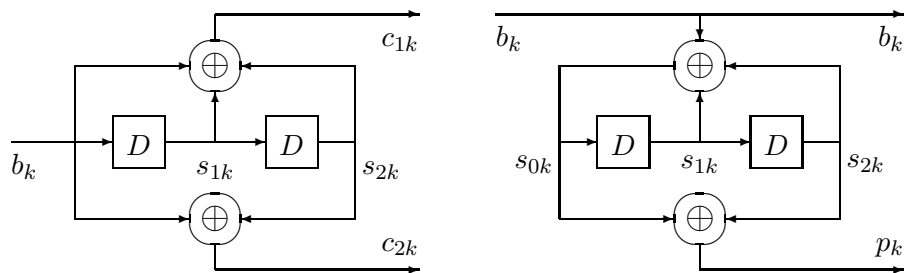


Figure 5.7: Transforming our example code into a systematic code.

5.9.2 Making our code systematic

A turbo code is constructed from two component codes, which are both systematic. For a systematic code, the message-digits are directly recognizable in the stream of code bits. The convolutional encoder that was discussed in Sec. 5.5 is non-systematic. It is not so hard to make our encoder systematic however. To see this, assume that the first code-output stream is systematic, i.e., $c_{1k} \equiv b_k$. This can only be accomplished if we take, see Fig. 5.7,

$$s_{0k} = b_k \oplus s_{1k} \oplus s_{2k} \quad (5.21)$$

for $k = 1, 2, \dots, K + 2$. Now

$$c_{2k} = p_k = s_{0k} \oplus s_{2k}. \quad (5.22)$$

We now call the second outputs $c_{2k} \equiv p_k$, the parity outputs of the encoder. The systematic code that we have obtained in this way again corresponds to a trellis having four states.

It is very important to observe that the new encoder is not only systematic but also recursive. This has the consequence that a difference of one bit in systematic symbols results in many different parity symbols, and therefore can be detected quite well. Our example (7,5)-code in Fig. 5.3, since it is feedforward instead of recursive, does not have this property. If we toggle a single message digit (bit), only five code digits flip.

5.9.3 Parallel concatenation of two systematic codes, Interleaving

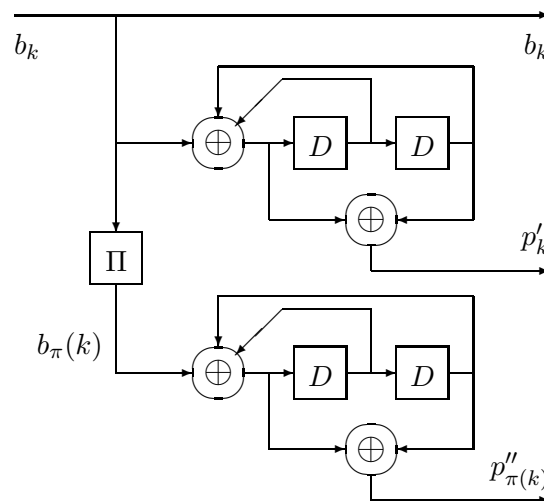


Figure 5.8: *The turbo encoder structure as proposed by Berrou, Glavieux, and Thitimajshima.*

A turbo encoder, see Fig. 5.8, consists of two systematic convolutional codes. There is a systematic stream (b_k) which is encoded by the first encoder into a first parity stream (p'_k), but the systematic stream is also encoded by a second encoder into a second parity stream ($p''_{\Pi(k)}$). The surprising idea behind the turbo-encoder is however that the systematic stream is interleaved before it enters the second encoder. The interleaver based on permutation $\Pi(\cdot)$, permutes the systematic stream, so that systematic symbols which are close to each other before interleaving, are far apart after interleaving, see table below.

k	1	2	3	4	5	6	7	8	9
$\Pi(k)$	1	4	7	2	5	8	3	6	9

The interleaver in the table is a so-called block-interleaver. It is based on a 3×3 block. Interleaving is done by writing the systematic symbols in this block row by row, and then read them out column by column. Apart from block-interleavers there are so-called random interleavers for which the permutation is determined at random, and therefore these interleavers have no structure. Note that always the decoder is aware of the permutation that is used by the encoder. For every message digit entering the encoder there are three code digits leaving. Therefore the rate of our turbo code is $1/3$.

Since we have two encoders working in parallel on the same systematic bits, we say the two codes are parallel concatenated. Later, in Sec. 5.13, we will discuss serial concatenation of two codes.

Our example (feed-forward) convolutional code started in the all-zero state and was, at the end of the codeword, driven back to the all-zero state by making the last two message digits zero, see Sec. 5.5. The two encoders in a turbo code also start and end in the all-zero state. To guarantee this, four message digits cannot be arbitrarily chosen, and therefore the trellis length of both codes is $K + 4$, where K is the number of message digits that can be freely chosen. For details we refer to Sec. 5.11.

5.10 Turbo decoding procedure

5.10.1 BCJR algorithm for a Systematic Code

The BCJR algorithm for decoding a systematic code is a special version of the general procedure described in Sec. 5.7 applying to any finite-state encoder. It is this special version that makes interaction between the two codes possible and consequently turbo-decoding work. Therefore we will first discuss the probabilistic structure corresponding to systematic encoding, and after that we will describe the modified BCJR procedure.

The systematic structure is shown in Fig. 5.9. Note that y_{bk} and y_{pk} are the channel outputs resulting from the systematic symbols b_k and parity symbols p_k respectively. The transition probabilities (likelihoods) are as in (5.10). Moreover we define $\underline{y}_k \triangleq (y_{bk}, y_{pk})$. Based on this systematic probabilistic structure we can write for the “numerator” for

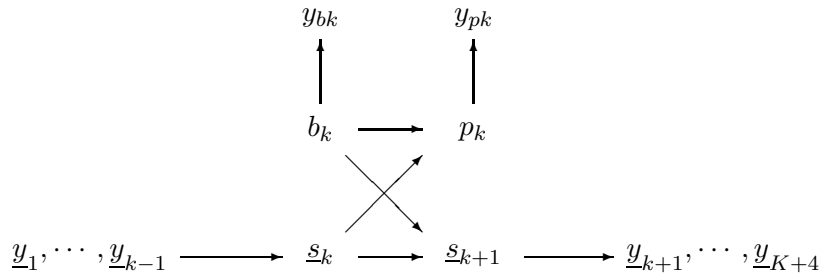


Figure 5.9: Probabilistic structure corresponding to a systematic finite-state encoder.

$b = 0, 1$ and all k that

$$\begin{aligned}
& P(\underline{y}_1, \dots, \underline{y}_{k-1}, B_k = b, \underline{y}_k, \underline{y}_{k+1}, \dots, \underline{y}_{K+4}) \\
&= \sum_{\underline{s}, p, \underline{s}'} P(\underline{y}_1, \dots, \underline{y}_{k-1}, \underline{S}_k = \underline{s}, B_k = b, P_k = p, \\
&\quad Y_{b_k} = y_{b_k}, Y_{p_k} = y_{p_k}, \underline{S}_{k+1} = \underline{s}', \underline{y}_{k+1}, \dots, \underline{y}_{K+4}) \\
&= \sum_{\underline{s}} P(\underline{y}_1, \dots, \underline{y}_{k-1}, \underline{S}_k = \underline{s}) P(B_k = b) \\
&\quad \cdot P(Y_{b_k} = y_{b_k} | B_k = b) P(Y_{p_k} = y_{p_k} | P_k = \pi(\underline{s}, b)) \\
&\quad \cdot P(\underline{y}_{k+1}, \dots, \underline{y}_{K+4} | \underline{S}_{k+1} = \sigma(\underline{s}, b)), \tag{5.23}
\end{aligned}$$

where $\pi(\underline{s}, b)$ is the parity output and $\sigma(\underline{s}, b)$ the next-state if \underline{s} is the current state and b the encoder input (and systematic output). Substituting α 's and β 's and re-arranging terms we obtain

$$\begin{aligned}
& P(\underline{y}_1, \dots, \underline{y}_{k-1}, B_k = b, \underline{y}_k, \underline{y}_{k+1}, \dots, \underline{y}_{K+4}) \\
&= P(B_k = b) P(Y_{b_k} = y_{b_k} | B_k = b) \\
&\quad \cdot \sum_{\underline{s}} \alpha_k(\underline{s}) P(Y_{p_k} = y_{p_k} | P_k = \pi(\underline{s}, b)) \beta_{k+1}(\sigma(\underline{s}, b)). \tag{5.24}
\end{aligned}$$

In this expression we can distinguish between three different factors. The first factor $P(B_k = b)$ is the a-priori probability of the message digit, the second factor $P(Y_{b_k} = y_{b_k} | B_k = b)$ is the systematic channel likelihood, and the third factor $\sum_{\underline{s}} \alpha_k(\underline{s}) P(Y_{p_k} = y_{p_k} | P_k = \pi(\underline{s}, b)) \beta_{k+1}(\sigma(\underline{s}, b))$ is the so-called extrinsic probability. We will see later that this extrinsic probability will play a crucial role in the turbo-decoding process.

Just like the α - and β -recursions for feedforward codes, which are given in (5.18) and

(5.19), the recursions for the systematic case are given by

$$\begin{aligned}
 \alpha_{k+1}(\underline{s}') &= \sum_{\underline{s}, b \rightarrow \underline{s}'} \alpha_k(\underline{s}) P(B_k = b) P(Y_{b_k} = y_{b_k} | B_k = b) \\
 &\quad P(Y_{p_k} = y_{p_k} | P_k = \pi(\underline{s}, b)), \\
 \beta_k(\underline{s}) &= \sum_b P(B_k = b) P(Y_{b_k} = y_{b_k} | B_k = b) \\
 &\quad P(Y_{p_k} = y_{p_k} | P_k = \pi(\underline{s}, b)) \beta_{k+1}(\sigma(\underline{s}, b)). \tag{5.25}
 \end{aligned}$$

Now that we know how the BCJR method should be adapted to systematic codes we can turn to turbo-decoding. First we will discuss a related problem however.

5.10.2 One systematic symbol in two codes

The idea behind turbo decoding is that information resulting from the first decoding procedure is handed over to the second decoding procedure. The second decoding procedure subsequently hands over information to a third procedure, etcetera. To find out what kind of information should be exchanged between the decoders, we consider a simple problem in which a single systematic symbol is part of two systematic codes, see Fig. 5.10.

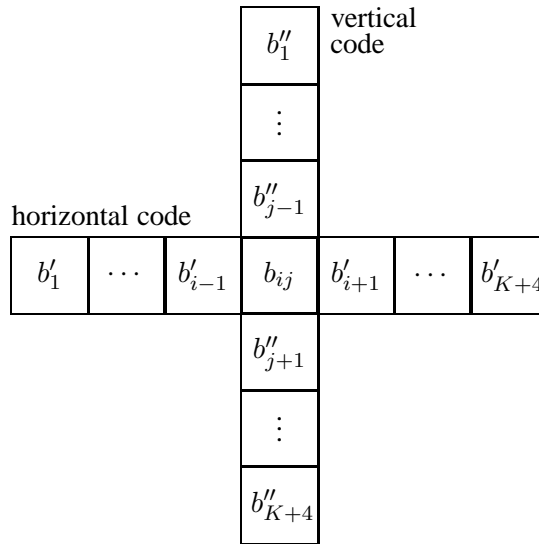


Figure 5.10: A symbol b_{ij} contained in two different systematic codes.

Consider a pair of integers i, j . The binary message symbols $b'_1, b'_2, \dots, b'_{i-1}, b'_{ij}, b'_{i+1}, \dots, b'_{K+4}$ are the systematic symbols of a first (horizontal) convolutional code. The symbols $b''_1, b''_2, \dots, b''_{j-1}, b'_{ij}, b''_{j+1}, \dots, b''_{K+4}$ are the systematic symbols of a second (vertical) convolutional code. Note that symbol b_{ij} is part of

both codes. The rate of both codes is $1/2$, hence the first and second code produce parity symbols $p'_1, p'_2, \dots, p'_{K+4}$ and $p''_1, p''_2, \dots, p''_{K+4}$ respectively. Note that there are two parity symbols for b_{ij} . The length of both codes is $K + 4$, just like the length of the component codes of our turbo code. Note however that in our turbo code all message symbols are in two codes.

We are interested only in decoding the overlap-symbol b_{ij} . To determine the a-posteriori probabilities for the overlap-symbol b_{ij} we can need

$$\begin{aligned}
& P(\underline{y}'_1, \dots, \underline{y}'_{i-1}, \underline{y}''_1, \dots, \underline{y}''_{j-1}, B_{ij} = b, y_{b_{ij}}, y_{p'_{ij}}, y_{p''_{ij}}, \\
& \quad \underline{y}'_{i+1}, \dots, \underline{y}'_{K+4}, \underline{y}''_{j+1}, \dots, \underline{y}''_{K+4}) \\
&= \sum_{\underline{s}', \underline{s}''} \alpha'_i(\underline{s}') \alpha''_j(\underline{s}'') P(B_{ij} = b) P(Y_{b_{ij}} = y_{b_{ij}} | B_{ij} = b) \\
& \quad \cdot P(Y_{p'_{ij}} = y_{p'_{ij}} | P'_{ij} = \pi(\underline{s}', b)) \beta'_{i+1}(\sigma(\underline{s}', b)) \\
& \quad \cdot P(Y_{p''_{ij}} = y_{p''_{ij}} | P''_{ij} = \pi(\underline{s}'', b)) \beta''_{j+1}(\sigma(\underline{s}'', b)) \\
&= P(B_{ij} = b) \cdot P(Y_{b_{ij}} = y_{b_{ij}} | B_{ij} = b) \tag{5.26} \\
& \quad \cdot \left[\sum_{\underline{s}'} \alpha'_i(\underline{s}') P(Y_{p'_{ij}} = y_{p'_{ij}} | P'_{ij} = \pi(\underline{s}', b)) \beta'_{i+1}(\sigma(\underline{s}', b)) \right] \\
& \quad \cdot \left[\sum_{\underline{s}''} \alpha''_j(\underline{s}'') P(Y_{p''_{ij}} = y_{p''_{ij}} | P''_{ij} = \pi(\underline{s}'', b)) \beta''_{j+1}(\sigma(\underline{s}'', b)) \right].
\end{aligned}$$

Again we can recognize an a-priori probability factor, a systematic channel probability factor, the extrinsic probability factor corresponding to the horizontal code, and another extrinsic probability factor related to the vertical code. When we focus only on the decoding based on the vertical code, the first extrinsic probability factor can be considered as an adjustment factor for the a-priori probability of the vertical code. The extra extrinsic probability factor should have been determined before, by decoding based on the horizontal code.

5.10.3 All systematic symbols of a vertical code are in different horizontal codes

Next consider a slightly more complex problem where we have a systematic vertical code that we want to decode, for which all the systematic symbols are also in a horizontal code, see Fig. 5.11. The vertical code and all horizontal codes have length $K + 4$ again. The question is now how to set this decoding procedure.

Following and extrapolating the procedure developed in the previous subsection, it is obvious that first extrinsic probability factors for all systematic symbols b_1, b_2, \dots, b_{K+4} should be obtained from $K + 4$ horizontal BCJR-decoding actions. These horizontal extrinsic probability factors should then be used to adjust the a-priori probabilities of all the systematic symbols b_1, b_2, \dots, b_{K+4} . A BCJR procedure should be done based on

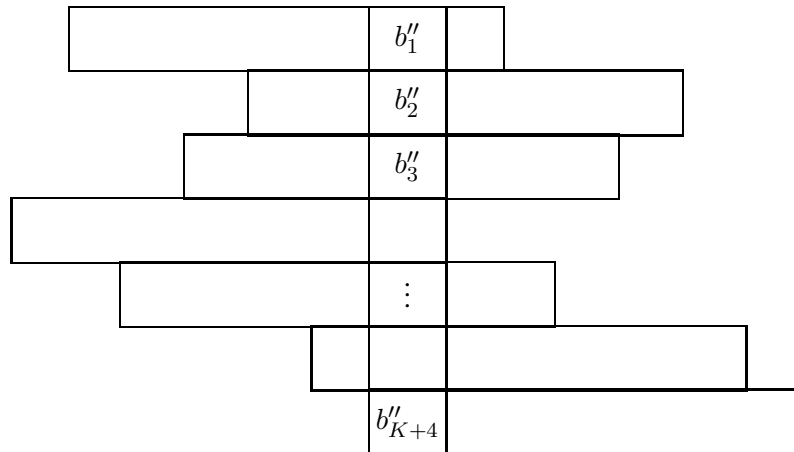


Figure 5.11: All symbols in the vertical code belong to different horizontal codes.

these adjusted a-priori probability factors. This procedure then leads to the a-posteriori symbol probabilities which can be used to make (optimum) decisions on these symbols.

5.10.4 An additional layer of vertical codes

We continue now by considering a setting in which we have a systematic vertical code that we want to decode, for which all the systematic symbols are also in different horizontal codes. The other systematic symbols in these horizontal codes are all in different vertical codes again, see Fig. 5.12. Again we are interested in the decoding procedure that leads to the (a-posteriori probabilities of the) systematic symbols in the layer-three vertical code. It will be clear that an optimal procedure starts with BCJR-decoding of the layer-one vertical codes. These decoding operations result in extrinsic probability factors that are used to adjust the corresponding a-priori probabilities of the symbols in the horizontal codes in layer two. Next all these horizontal codes are BCJR-decoded, which results in the extrinsic factors that are used to adjust the a-priori probabilities of the systematic symbols $b''_1, b''_2, \dots, b''_{K+4}$ in the final vertical code. Finally a BCJR-procedure based on these adjusted a-priori probability factors gives the final result, i.e., the symbols in the layer-three vertical code.

From the discussions so far, it will be clear that similar procedures apply to systems with more than three layers. What is still missing however, is the procedure for decoding a turbo code.

5.10.5 Decoding a turbo code

Observe first that the systematic input sequence b_1, b_2, \dots, b_{K+4} can be decoded in two different ways: (i) using the channel-output streams

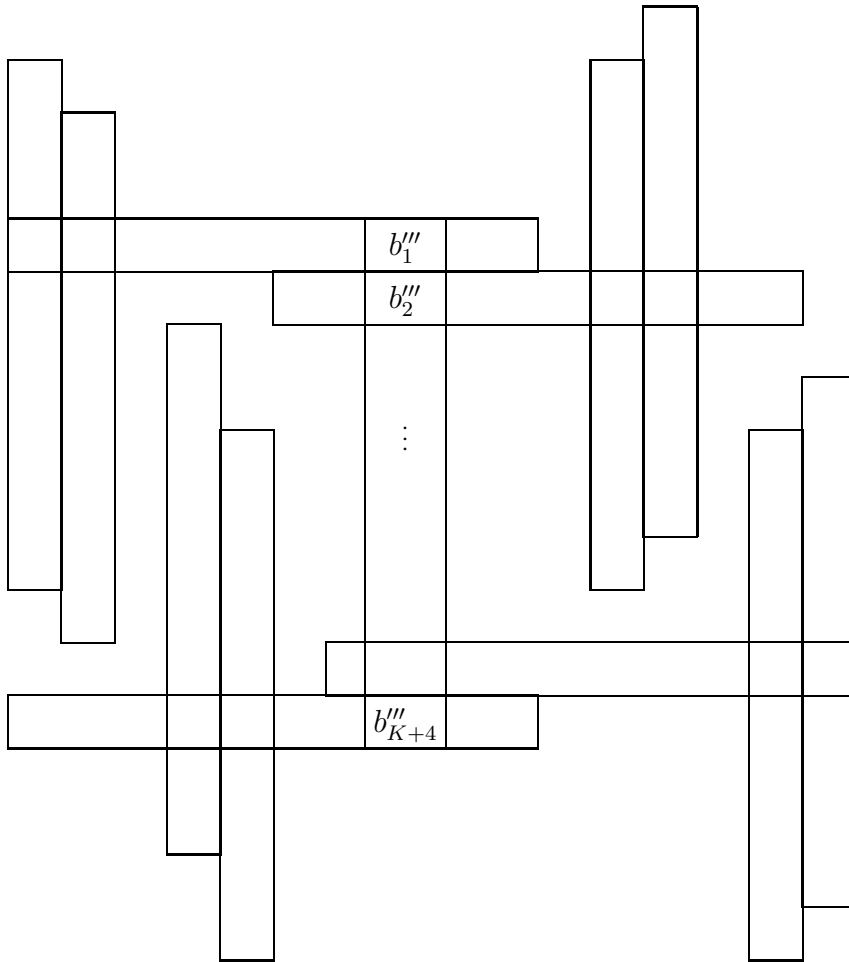


Figure 5.12: Three layers of codes.

$(y_{b_1}, y_{p'_1}), (y_{b_2}, y_{p'_2}), \dots, (y_{b_{K+4}}, y_{p'_{K+4}})$ for decoding the first (horizontal) code, or (ii) using the channel-output streams $(y_{b_1}, y_{p''_1}), (y_{b_2}, y_{p''_2}), \dots, (y_{b_{K+4}}, y_{p''_{K+4}})$ for decoding the second (vertical) code. A turbo decoder iterates between these horizontal and vertical decoding actions. It starts by doing a horizontal BCJR decoding action. Next it performs a vertical BCJR action. It would be nice if for this action information resulting from the first horizontal action could be used somehow. Note that this situation resembles the one described in subsection 5.10.3, however here all message symbols are in the same horizontal code, and not in different ones as in subsection 5.10.3. Since an interleaver is used, the difference can be ignored however. The behavior of a BCJR algorithm is mainly determined by the dependencies of the variables that “are close together,” i.e., in neighboring trellis sections. Dependencies of variables that are far apart do not really matter. The interleaver guarantees that there is little dependency between extrinsic information factors, which are close in the vertical trellis. As a result the horizontal decoder can pass on interleaved extrinsic information of all the message digits to the vertical decoder (see Fig. 5.13).

The next operation is a horizontal BCJR action, and again we want to find out what kind of information should be used coming from the last vertical action. Just as in subsection 5.10.4 we can use the extrinsic information resulting from that last vertical action. It should be noticed that the extrinsic information should be de-interleaved (see Fig. 5.13). Again the use of the interleaver guarantees that the fact, that there is only one vertical and one horizontal code, can be ignored.

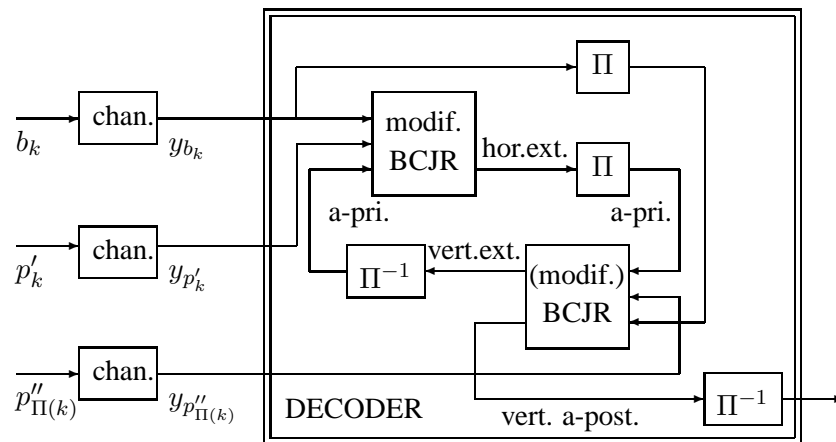


Figure 5.13: Turbo-decoder architecture. The decoder outputs a-posteriori probabilities of the message digits.

In this way we continue. The extrinsic information produced in a certain decoding action is (de-)interleaved and used to adjust the a-priori information in the next decoding action. Since we assume that the dependence between extrinsic factors is zero, the suggested procedure is actually suboptimal. Nevertheless its performance turns out to be excellent in practise.

The architecture of a turbo-decoder is given in Fig. 5.13. There are a couple of remarks that we want to add here. All a-priori probabilities are equal to $1/2$. Adjusting such a-priori probabilities with extrinsic factors is actually the same as using these extrinsic factors as a-priori probabilities. Moreover we see that in the iterations extrinsic information circles around between the two BCJR-decoders, and these decoders are modified for computing extrinsic information only. In the last decoding action the decoder should be complete and compute a-posteriori message digit probabilities. In the literature, computing extrinsic information, is described often as an adjustment of the a-posteriori probabilities, i.e., dividing them by a-priori probability and systematic channel likelihood. It is better to compute the extrinsic information directly however.

5.11 Terminating two trellises

How can we terminate the two trellises simultaneously? To see how this can be done, observe that each weight-one input sequence results in some final state-pair, a state in which the first encoder ends and a state in which the second encoder ends. For an arbitrary input sequence the final state-pair is the sum of such pairs by linearity. Now using a Gram-Schmidt procedure, we can select a set S of four weight-one input sequences that spans the space of the sixteen possible final state-pairs. The symbols at the positions corresponding to the set S of weight-one input sequences can be used to compensate every state-pair resulting from the symbols at the remaining positions.

5.12 Performance turbo code

To determine the performance of our turbo code we have done a simulation. We have decoded 2048 blocks of 4096 symbols. For each block a new interleaver was chosen at random, where all permutations have the same probability. We have considered up to 16 iterations, i.e., 32 BCJR-actions. From these simulations it can be concluded that a signal-to-noise ratio of roughly -1.1 dB is needed to obtain an error probability of 10^{-5} . Since the Shannon limit corresponding to rate $1/3$ is -2.31 dB, our turbo code is only 1.2 dB from the Shannon limit. By choosing better and more complex component codes and larger interleavers, we can even get closer to the Shannon limit. However our relatively simple turbo code has already bridged the largest part of the eight dB gap that separates uncoded systems from the Shannon limit.

Fig. 5.14 shows that for 16 iterations the observed bit-error rate curve seems to flatten off at signal to noise ratios larger than -1.2 dB. This so-called floor effect is a consequence of the fact that the minimum distance of our turbo-code is not very large. The codes that we will discuss in the next section behave better in that respect, see Fig. 5.19.

5.13 Serial concatenation, encoder structure

So far we have discussed turbo coding, a technique based on parallel concatenation. In the non-iterative case, serial concatenation is a topic that is better studied and better under-

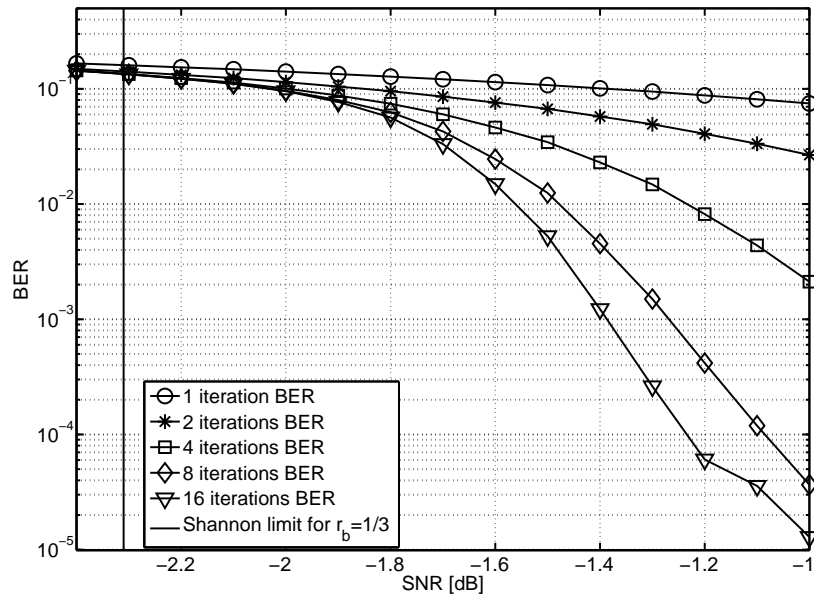


Figure 5.14: Bit-error rate of our turbo code. Also the corresponding Shannon limit is shown.

stood than parallel concatenation. Forney [33] showed that serial concatenation can lead to powerful codes that are relatively easy to decode. Moreover these codes can be used to correct burst-errors by spreading out bursts over several codewords.

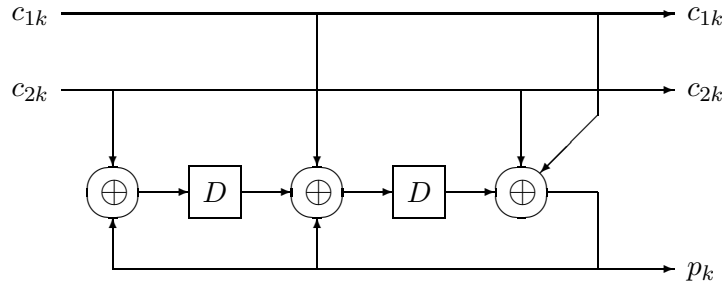


Figure 5.15: A systematic (recursive) rate-2/3 convolutional encoder.

A question that pops up now is whether a turbo-like decoding procedure and performance is possible also for serially concatenated convolutional codes. Benedetto and Montorsi [9] proposed a method demonstrating that indeed this is true. Their serial code consist of a recursive rate-2/3 systematic, see Fig. 5.15, inner code, i.e., the code closest to the channel, and an outer code which is feedforward with rate 1/2, see Fig. 5.16. The two codes are connected to each other by an interleaver. Although in [9] this interleaver permutes the code bits that leave the outer encoder, we assume here that the code-bit pairs that are produced by this encoder are permuted, before being handed over to the inner encoder, see Fig. 5.16.

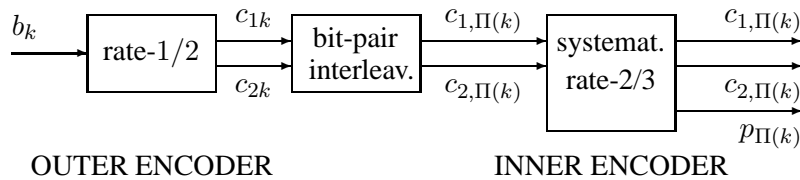


Figure 5.16: Two serially concatenated convolutional codes and a bit-pair interleaver.

5.14 Serial case, decoding procedure

Just like for the turbo-case it is a challenge to find out how to set up the iterative decoding process for the serial setting. The problem is to determine what kind of information should circle around between subsequent decoding actions. Where in the turbo-case both the decoders were jointly working on improving the quality of their systematic message digit a-posteriori probabilities, it is intuitively clear to have, in the serial case, both the inner

and outer decoder working on improving the quality of the a-posteriori probabilities of the code-bit pairs (c_{1k}, c_{2k}) . This follows from the fact that these pairs form the connection between both encoders. In the next subsection we will find out what information one decoder needs to provide for the other.

5.14.1 Two connected codes again

Just like in the parallel case the outer and inner code are connected. Now it is not a systematic symbol b_{ij} which is input to two encoders, but now it is a code-bit pair \underline{c}_{ij} which is output of the outer encoder and, after interleaving, input of the inner encoder, see Fig. 5.17. Also here four message digits are used to force both encoders back to the all-zero state.

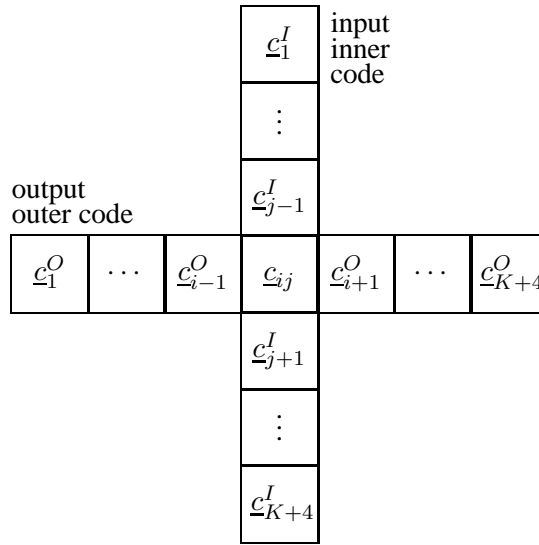


Figure 5.17: A pair \underline{c}_{ij} which is both output of the outer encoder and input for the inner encoder.

The corresponding joint probability of the code-bit pair \underline{c}_{ij} and all received channel outputs, can therefore be written as

$$\begin{aligned}
 &P(\underline{C}_{ij} = \underline{c}, \dots) \\
 &= \sum_{\underline{z}^O, b \rightarrow \underline{c}} \sum_{\underline{z}^I} \alpha_i^O(\underline{z}^O) P(B_i = b) \alpha_j^I(\underline{z}^I) \\
 &\quad \cdot P(\underline{y}_j | \gamma^I(\underline{z}^I, \underline{c})) \beta_{j+1}^I(\sigma^I(\underline{z}^I, \underline{c})) \beta_{i+1}^O(\sigma^O(\underline{z}^O, b)), \quad (5.27)
 \end{aligned}$$

where superscripts O and I refer to the outer and inner code respectively. The a-priori probabilities $P(B_i = 0) = P(B_i = 1) = 1/2$, i.e., uniform. An output triple \underline{y}_j consists

of three values $(y_{c_{1j}}, y_{c_{2j}}, y_{p_j})$. We can now rewrite this equation in two different ways, focussing either on the outer decoder and on the inner decoder.

5.14.2 Two perspectives

First we consider the actions of the inner decoder. Now

$$P^I(\underline{C}_{ij} = \underline{c}) = \sum_{\underline{s}^I} \alpha_j^I(\underline{s}^I) P_x^O(\underline{c}) P(\underline{y}_j | \gamma^I(\underline{s}^I, \underline{c})) \beta_{j+1}^I(\sigma^I(\underline{s}^I, \underline{c})) \quad (5.28)$$

where extrinsic information $P_x^O(\underline{c})$

$$P_x^O(\underline{c}) = \sum_{\underline{s}^O, b \rightarrow \underline{c}} \alpha_i^O(\underline{s}^O) P(B_i = b) \beta_{i+1}^O(\sigma^O(\underline{s}^O, b)), \quad (5.29)$$

should be provided by the outer decoder. This extrinsic information acts as a-priori information for the inner decoder. Observe that this inner decoder computes the a-posteriori probabilities of the input symbols \underline{c} of the inner encoder, which is what we have seen before in the parallel concatenated scheme.

Next we concentrate on the actions of the outer decoder. We can write

$$P^O(\underline{C}_{ij} = \underline{c}) = \sum_{\underline{s}^O, b \rightarrow \underline{c}} \alpha_i^O(\underline{s}^O) P(B_i = b) P_x^I(\underline{c}) \beta_{i+1}^O(\sigma^O(\underline{s}^O, b)), \quad (5.30)$$

where the extrinsic information $P_x^I(\underline{c})$

$$P_x^I(\underline{c}) = \sum_{\underline{s}^I} \alpha_j^I(\underline{s}^I) P(\underline{y}_j | \gamma^I(\underline{s}^I, \underline{c})) \beta_{j+1}^I(\sigma^I(\underline{s}^I, \underline{c})) \quad (5.31)$$

should come from the other decoder, the inner decoder. Note that this extrinsic information can be regarded as likelihood information for the outer decoder. Observe that the outer decoder computes the a-posteriori probabilities of the output symbols \underline{c} of the outer encoder, which is different from what we have seen before in the parallel concatenated scheme.

5.14.3 Decoding procedure

The observations made in the previous subsection lead to the following decoding procedure, see Fig. 5.18.

First the inner decoder computes, as in (5.28) the a-posteriori (joint) probabilities $P^I(\underline{C}_{ij} = \underline{c})$, assuming that there is no knowledge about the $P_x^O(\underline{c})$ yet, hence the extrinsic probabilities $P_x^O(\underline{c})$ are all equal. Then the extrinsic probabilities $P_x^I(\underline{c})$ should be computed as in (5.31). Observe however that in this first step these are identical to the a-posteriori probabilities $P^I(\underline{c})$. This extrinsic information is now de-interleaved and passed on to the outer decoder.

The outer decoder computes, as in (5.30), the a-posteriori probabilities $P^O(\underline{C}_{ij} = \underline{c})$ in which the de-interleaved extrinsic information $P_x^I(\underline{c})$ is used. Next the outer decoder

must compute the extrinsic probabilities $P_x^O(\underline{c})$ based on (5.29). Note that these extrinsic probabilities are equal to the a-posteriori probabilities divided by the incoming extrinsic probabilities $P_x^I(\underline{c})$. The resulting extrinsic probabilities are interleaved and then handed over to the inner decoder.

The inner decoder computes, as in (5.28) the a-posteriori probabilities $P^I(C_{ij} = \underline{c})$, using the interleaved extrinsic probabilities $P_x^O(\underline{c})$. The extrinsic probabilities $P_x^I(\underline{c})$ are now computed as in (5.31), hence we must divide the a-posteriori probabilities by the incoming interleaved extrinsic probabilities $P_x^O(\underline{c})$. The computed extrinsic information is de-interleaved and serves as input for the outer decoder. Etc.

A last decoding action is done by the outer decoder, that computes similar to (5.30) a-posteriori probabilities $P^O(B_i = b)$ of the message bits. We can write for $b = 0, 1$

$$P^O(B_i = b) = \sum_{\underline{x}^O} \alpha_i^O(\underline{x}^O) P(B_i = b) P_x^I(\gamma^O(\underline{x}^O, b)) \beta_{i+1}^O(\sigma^O(\underline{x}^O, b)). \quad (5.32)$$

Normalizing now yields the a-posteriori probabilities.

Although the description above suggests that outgoing extrinsic probabilities are computed by dividing computed a-posteriori probabilities by incoming extrinsic probabilities, the BCJR algorithm can also be modified such that extrinsic probabilities are computed directly. Fig. 5.18 shows how the extrinsic information circles around between inner and outer decoder.

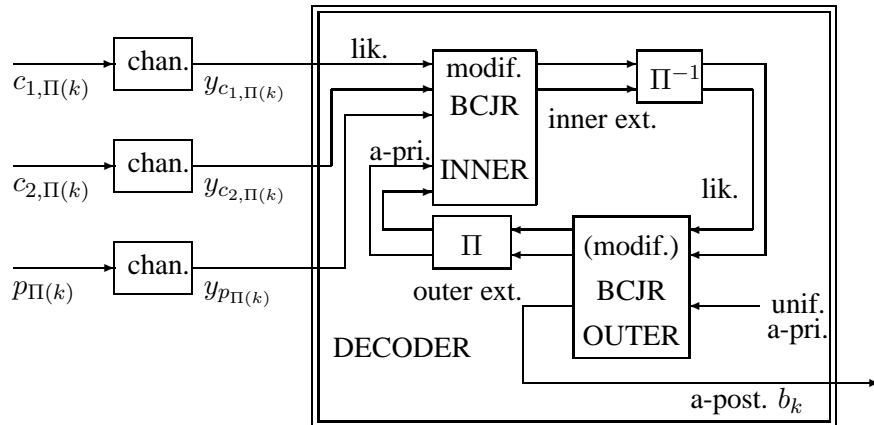


Figure 5.18: Decoder architecture for a serially concatenated code.

5.15 Performance serially concatenated code

Also for our serially concatenated code we have performed simulations. Again we have decoded 2048 blocks of 4096 symbols, and again for each block a new interleaver was chosen at random. Just like for the turbo case (parallel concatenation) we have considered

up to 16 iterations, i.e., 32 BCJR-actions. Also these simulations demonstrate, see Fig. 5.19, that a signal-to-noise ratio of roughly -1.3 dB results in a bit-error probability of 10^{-5} . Therefore our serially concatenated code is only one dB from the Shannon limit. Using better codes and larger interleavers we can get closer to the Shannon limit.

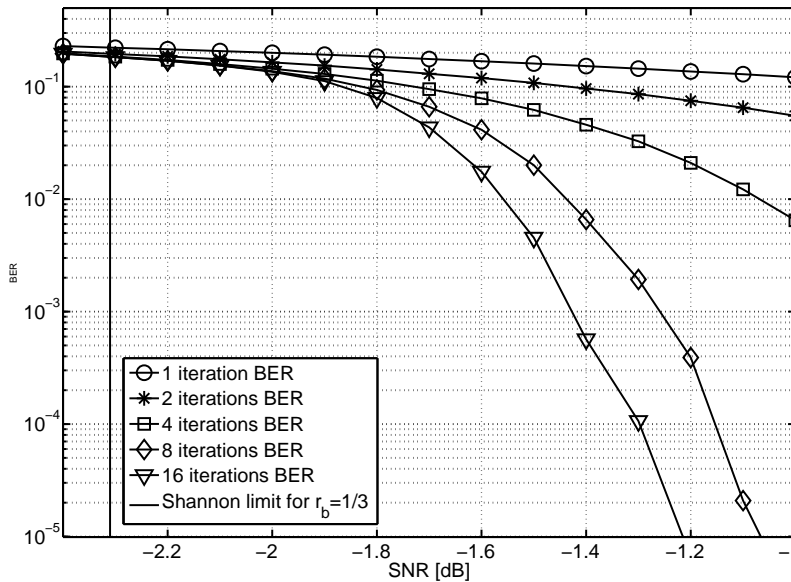


Figure 5.19: Bit-error rate of our serially concatenated code.

5.16 Conclusion

We have first discussed channel capacity and the resulting Shannon limit. Without coding a signal-to-noise ratio, which is roughly 8dB larger than the Shannon limit, is needed for reliable transmission. We have discussed simple parallel and serially concatenated convolutional codes, and the corresponding decoding procedures, that achieve the Shannon limit within roughly one dB. These procedures are based on modified versions of the BCJR algorithm [4].

Our approach follows Gallager [32] who investigated iterative procedures for decoding LDPC codes. The notion of a parity-check tree, see Fig. 6 in [32], is quite similar to the setting shown in Fig. 5.12, where we discuss codes intersecting with other codes that intersect again with other codes, etc.. Just like Gallager we ignore the fact that some of these codes are not independent but actually the same.

We have focussed only on methods based on the BCJR [4] method. These methods are called probabilistic decoding methods in Costello and Forney [28]. This latter paper gives

an excellent and extensive overview of the development of coding theory, and contains many references for readers who want to learn more on this topic.

Chapter 6

Two-Dimensional Iterative Processing for DAB Receivers Based on Trellis Decomposition

It is well known that iterative (turbo) decoding procedures approach channel capacity, e.g., in the AWGN setting. Moreover, in the literature a number of encouraging results on serial concatenation of convolutional encoding followed by differential encoding with turbo-like decoding techniques for single-carrier systems are found, also referred to as Turbo-DPSK. For those reasons the focus of this chapter is to analyse the application of Turbo-DPSK for DAB receivers¹.

6.1 Outline

We investigate trellis decoding and iterative techniques for DAB, with the objective of gaining from processing 2D-blocks in an OFDM scheme, that is, blocks based on the time and frequency dimension, and from trellis decomposition.

Trellis-decomposition methods allow us to estimate the unknown channel-phase since this phase relates to the sub-trellises. We will determine a-posteriori sub-trellis probabilities, and use these probabilities for weighting the a-posteriori symbol probabilities resulting from all the sub-trellises. Alternatively we can determine a dominant sub-trellis from the a-posteriori sub-trellis probabilities and use the a-posteriori symbol probabilities corresponding to this dominant sub-trellis. This dominant sub-trellis approach results in a significant complexity reduction.

We will investigate both iterative and non-iterative methods. The advantage of non-iterative methods is that their forwardbackward procedures are extremely simple; how-

¹This chapter is based on the paper published as *W.J. van Houtum and F.M.J. Willems, "Two-Dimensional Iterative Processing for DAB Receivers Based on Trellis-Decomposition", Journal of Electrical and Computer Engineering, no. 394809, 15 pages, 2012.*

ever, also their gain of 0.7 dB, relative to 2SDD at a BER of 10^{-4} , is modest. Iterative procedures lead to the significantly larger gain of 3.7 dB at a BER of 10^{-4} for five iterations, where a part of this gain comes from 2D-processing. Simulations of our iterative approach applied to the TU-6 (COST-207) channel show that we get an improvement of 2.4 dB at a Doppler frequency of 10Hz.

6.1.1 Problem description

DAB systems, DAB+ systems, and T-DMB systems use OFDM, for which every OFDM-subcarrier is modulated by $\pi/4$ -DE-QPSK [25].

Commonly used *classical* DAB receivers perform non-coherent 2SDD with soft-decision Viterbi decoding [79]. Non-coherent detection schemes like 2SDD are not optimal and can be improved by MSDD, which is a maximum likelihood procedure for finding a block of information symbols after observing a block of received symbols [22]. For very large numbers of observations, the performance of MSDD approaches the performance of ideal coherent detection of DE-QPSK, which is given in, e.g., [15], [19], [74]. Non-coherent MSDD can also be used if channel coding is applied in a non-iterative way, see [23], [21]. If MSDD is combined with iterative (turbo) processing (parallel concatenated systems were first described by Berrou et al. [11], serial concatenation was developed by Benedetto and co-investigators [9], [8], [7]), it needs to be improved to get a more acceptable complexity. We were motivated by a number of encouraging results on serial concatenation of convolutional encoding followed by differential encoding with turbo-like decoding techniques, also referred to as *Turbo-DPSK*. Turbo-DPSK was investigated for single-carrier transmission on AWGN channels in [57], [6], [72], [49], [58], and [20], as well as for time-varying channels in [41], [42], [46], [18], and [53]. The main objective of these papers was to reduce the complexity of the inner decoder. Two main methods can be distinguished, first an explicit estimation of the channel phase followed by coherent detection, see [41] and [42], and for the 2D-case [71], [52], and [65], or secondly by directly calculating the *a-posteriori* probabilities of the information symbols as in [58], [20], [18], and for the 2D case [51], [64], and [36].

We focus in the present paper on 2D processing, i.e., in both the frequency- and time-domain. We will propose methods based on iteratively demodulating and decoding blocks of received symbols in a DAB-transmission stream. First we will however summarize other 2D approaches, that are relevant to our work.

The work of ten Brink et al. in [71] on 2D phase-estimate methods can be regarded as an extension of the results of Hoeher and Lodge in [42] to the multi-carrier case. Park et al. in [52] improved the hard-decision approach of ten Brink et al. by considering soft-decision. Both [71] and [52] rely on pilot symbols, which are not present in DAB-transmission [25] unfortunately. Blind channel estimation techniques were proposed by Sanzi and Necker in [65]. They proposed a combination of the iterative scheme of ten Brink in [71] and a fast converging blind channel estimator based on higher order asymmetrical modulation schemes, which are not used within a DAB-transmission [25].

To obtain *a-posteriori* probabilities of the information symbols in a 2D setting, May, Rohling, and Haase in [51], [64], and [36], considered iterative decoding schemes for multi-carrier modulation with the SOVA, [37]. The SOVA was used for differential detec-

tion as well as for decoding of the convolutional code. They used in the coherent setting an estimate of the phase based on a block of three by three received symbols, which are adjacent in time and frequency direction. They proposed, for the coherent case, to use only the current received symbol to obtain a symbol-metric for the SOVA inner-decoder, actually ignoring the differential encoding. For the incoherent case, they used a transition-metric for the SOVA inner-decoder based on the current and previously received symbol. These a-posteriori detection schemes produce approximations of the a-posteriori probabilities. Procedures that focus on efficient computation of exact values can be found in [19] for the coherent case, but also in [20] for the incoherent case.

To reduce complexity we accept a small performance loss due to channel-phase discretization (see, e.g., Peleg et al. [58] and Chen et al. [18]) in this contribution, but apart from that we determine the exact a-posteriori probabilities of the information symbols in a 2D setting. Our starting point will be the techniques proposed by Peleg et al. in [58]. We discretize the channel phase into a number of equispaced values, but do not allow the “side-step” transitions that were proposed by Peleg et al. to track small channel-phase variations. Then we calculate, in an efficient way, the a-posteriori probabilities of the information symbols using the BCJR-algorithm [4] in a 2D setting, see also [73]. We will consider 2D-blocks and trellis decomposition. Each 2D-block consist of a number of adjacent subcarriers of a number of subsequent OFDM symbols. Focussing on 2D-blocks was motivated by the fact that the channel coherence-time is typically limited to a small number of OFDM symbols, but also since DAB-transmissions use time-multiplexing of services, which limits the number of OFDM symbols in a codeword. Extension in the sub-carrier direction is required then to get reliable phase estimates. The trellis-decomposition method allow us to estimate the unknown channel-phase efficiently. This phase is related to sub-trellises of which we can determine the a-posteriori probabilities. With these probabilities we are able to chose a dominant sub-trellis, which results in a significant complexity reduction.

Franceschini et al. [31] also use the idea of trellis-decomposition and sub-trellises (multiple trellises), to focus on estimating channel parameters. Variation of these parameters is tackled by applying so-called inter-mix intervals, in which special manipulations (mix-metric techniques) on the forward and backward metrics are performed. Since we cannot track channel variations here, we apply a 2D-approach which is based on the assumption that there are independent channel realizations within distinct blocks. We will explain later, in Subsection 6.2.1, why we cannot track the channel phase.

6.2 Description of a digital audio broadcasting (DAB) system

Since we use in this chapter some specific symbol notations and definitions we will, for clearness, introduce the notations and definitions by discussing the particular system-blocks where their functionality is represented by these symbols.

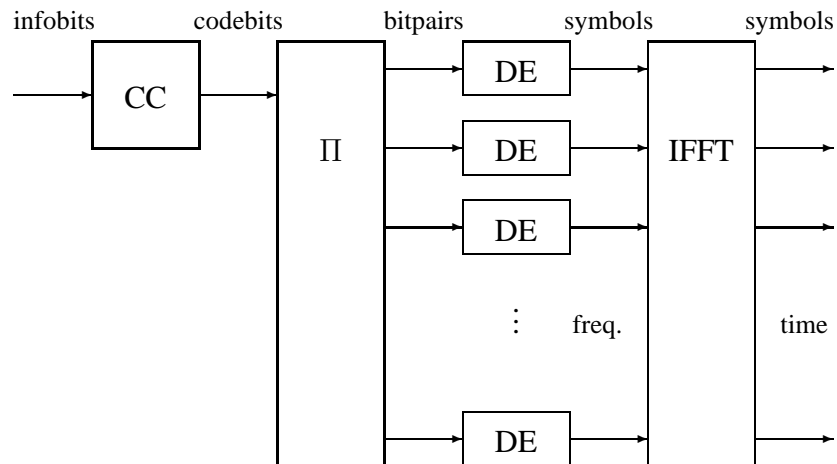


Figure 6.1: DAB convolutional encoder, interleaver, differential encoders, and multi-carrier modulator with an IFFT.

6.2.1 Overview

Terrestrial digital broadcasting systems like DAB, DAB+, and T-DMB, all members of the “DAB-family”, comprise a combination of convolutional coding (CC), interleaving, $\frac{\pi}{4}$ -DE-QPSK modulation followed by OFDM, see Fig. 6.1. Time multiplexing of the transmitted services allows the receiver to perform *per service symbol processing* [25], see Fig. 6.2 where 1536 is the number of “active” OFDM-subcarriers for a DAB-transmission in Mode-I [25], hence the receiver can decode a certain service without having to process the OFDM symbols that do not correspond to this service.

Consequently, only at particular time instants within a DAB transmission-frame a small number (usually up to four) of OFDM-symbols need to be processed. This results in “idle-time” for the demodulation and decoding processes.

Note, that due to this “idle-time” the mix-metric techniques of [31] cannot be applied to DAB-receivers. However, if all the transmitted services are decoded, and there is no idle-time, mix-metric techniques could be a valuable extension to the 2D iterative processing methods based on trellis-decomposition that we will develop here.

In the following subsections we will describe the transmit processes (convolutional encoding, differential modulation, and OFDM) in more detail.

6.2.2 Convolutional coding and interleaving

The convolutional code that is used within DAB has basic code-rate $R_c = \frac{1}{4}$, constraint length $K = 7$ and generator polynomials $g_0 = 133$, $g_1 = 171$, $g_2 = 145$, and $g_3 = 133$. Larger code-rates can be obtained via puncturing of the mother code, see Hagenauer [38]. The time- and frequency-interleavers in DAB perform bit and bit-pair interleaving, re-

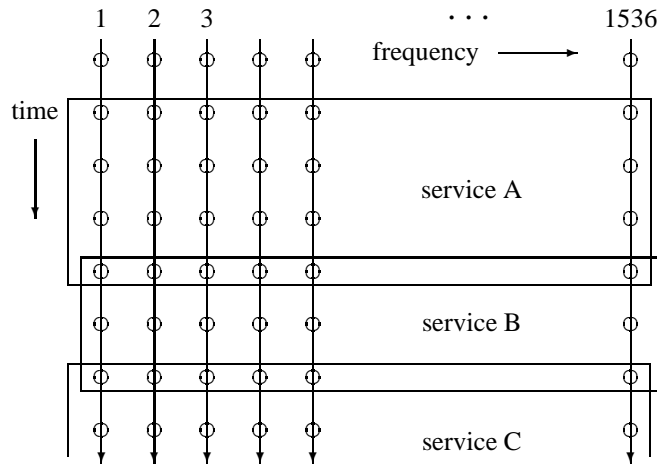


Figure 6.2: Three services mapped onto consecutive OFDM symbols. Note that there is overlap between the service since differential modulation is used.

spectively. As a result the code-bits leaving the convolutional encoder are permuted and partitioned over the sub-carriers of a number of subsequent OFDM-symbols (in subsequent frames). The bits for each sub-carrier are grouped in pairs, and each of such pair is mapped onto a phase (difference) that therefore can assume four different values. The mapping that is used here is based on the Gray principle, i.e., labels that correspond to adjacent phase differences differ only in a single bit-position.

6.2.3 Differential modulation in each subcarrier

For each sub-carrier $\frac{\pi}{4}$ -DE-QPSK modulation is applied. A sequence $\mathbf{b} = (b_1, b_2, \dots, b_N)$ consisting of N symbols (phase differences) b_n for $n = 1, 2, \dots, N$ carries the information that is to be transmitted via this sub-carrier. The symbols $b_n, n = 1, 2, \dots, N$ assume values in the (offset) alphabet $\mathcal{B} = \{e^{j(p\pi/2 + \pi/4)}, p = 0, 1, 2, 3\}$. The transmitted sequence $\mathbf{s} = (s_0, s_1, \dots, s_N)$ of length $N + 1$ follows from \mathbf{b} by applying differential phase modulation, i.e.,

$$s_n = b_n s_{n-1}, \text{ for } n = 1, 2, \dots, N, \quad (6.1)$$

where for the first symbol $s_0 \triangleq 1$.

6.2.4 OFDM in DAB

OFDM in DAB is realized using a B -point complex IFFT, where B is 256, 512, 1024, or 2048. To compute the n -th time-domain OFDM-symbol $\tilde{\mathbf{s}}_n = (\tilde{s}_{1,n}, \tilde{s}_{2,n}, \dots, \tilde{s}_{B,n})$, we

determine

$$\tilde{s}_{t,n} = \frac{1}{\sqrt{B}} \sum_{m=1}^B s_{m,n} e^{j2\pi \frac{(t-1)(m-1)}{B}}, \text{ for } t = 1, 2, \dots, B, \quad (6.2)$$

and $s_{m,n}$ is the n -th differentially encoded symbol corresponding to the m -th sub-carrier, or equivalently the m -th element in the n -th frequency-domain OFDM-symbol, see Fig. 6.1. Note, that the IFFT is a computationally efficient IDFT for values of B that are powers of 2. To prevent ISI resulting from multi-path reception, a cyclic prefix of length L_{cp} is added to the sequence \tilde{s}_n . This leads to the sequence $\bar{s}_n = (\tilde{s}_{B-L_{cp}+1,n}, \dots, \tilde{s}_{B,n}, \tilde{s}_{1,n}, \tilde{s}_{2,n}, \dots, \tilde{s}_{B,n})$ that is finally transmitted.

We assume that the channel is slowly varying with an impulse response shorter than the cyclic-prefix length. Moreover we assume that the channel coherence-bandwidth and coherence-time span multiple OFDM-subcarriers and multiple OFDM-symbols. Therefore, the channel-phase and gain might be assumed to be fixed for a number of adjacent subcarriers and consecutive symbols. This is the assumption on which we base our investigations. The channel phase and gain are assumed constant (yet unknown to the receiver) over a $2D$ -block of symbols, see Fig. 6.3.

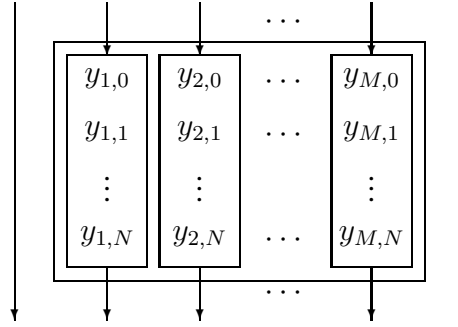


Figure 6.3: A $2D$ -block of symbols out of an OFDM stream. We are interested in M adjacent sequences of $N + 1$ subsequent symbols, where each such sequence corresponds to one of M adjacent sub-carriers.

The receiver, in the case of perfect synchronization, removes the (received version of the) cyclic prefix, and then applies a B -point complex FFT on the time-domain received sequence $\tilde{\mathbf{r}}_n = (\tilde{r}_{1,n}, \tilde{r}_{2,n}, \dots, \tilde{r}_{B,n})$ which results in the B received symbols

$$r_{m,n} = \frac{1}{\sqrt{B}} \sum_{t=1}^B \tilde{r}_{t,n} e^{-j2\pi \frac{(m-1)(t-1)}{B}}, \text{ for } m = 1, 2, \dots, B. \quad (6.3)$$

OFDM reception can be regarded as parallel matched-filtering corresponding to B complex orthogonal waveforms, one for each subcarrier. This results in a channel model, holding for a $2D$ -block of symbols, that is given by

$$r_{m,n} = |h|e^{j\phi}s_{m,n} + n_{m,n}, \quad (6.4)$$

for some subsequent values of n and m , where the channel gain $|h|$ and phase ϕ are unknown to the receiver. It should be noted that a phase rotation proportional to m , due to a time-delay, is removed by linear phase correction (LPC). This technique modifies the phase of each OFDM subcarrier with an appropriate rotation based on the starting position (time-delay) of the FFT-window within the OFDM symbol. In practise this delay can be determined quite accurately.

In the next subsection we focus on a single subcarrier.

6.2.5 Incoherent reception, channel gain known to receiver

The sequence \mathbf{s} that is transmitted via a certain subcarrier is now observed by the receiver as sequence $\mathbf{r} = (r_0, r_1, \dots, r_N)$. Note that compared to the previous subsection we have dropped the subscript m here. Since it is relatively easy to estimate the channel gain, we assume here that it is perfectly known to the receiver and to ease our analysis we take it to be one. The received sequence now relates to the transmitted sequence \mathbf{s} as follows

$$r_n = e^{j\phi}s_n + n_n, \text{ for } n = 0, 1, \dots, N, \quad (6.5)$$

where we assume that n_n is circularly symmetric complex Gaussian with variance σ^2 per component. Basically we assume that the random channel-phase ϕ is real-valued and uniform over $[0, 2\pi)$. This channel phase is fixed over all $N + 1$ transmissions, and unknown to the receiver.

Accepting a small performance loss as in, e.g., Peleg et al. [58] and Chen et al. [18] we may assume that the channel-phase is discrete, and uniform over 32 levels which are uniformly spaced over $[0, 2\pi)$, hence

$$\Pr\{\phi = \pi l/16\} = 1/32, \text{ for } l = 0, 1, 2, \dots, 31. \quad (6.6)$$

We will first study the situation in which we consider a uniformly chosen channel phase in a single sub-carrier. Later we will also investigate the setting in which a uniformly chosen channel phase is moreover constant over a number of (adjacent) sub-carriers.

6.2.6 Equivalence between DE-QPSK and $\pi/4$ -DE-QPSK

It is well-known and straightforward to show that the $\pi/4$ -DE-QPSK modulation, which is performed in each of the sub-carriers, is equivalent to DE-QPSK. To see this, we define for $n = 1, 2, \dots, N$

$$a_n = b_n e^{-j\pi/4}, \quad (6.7)$$

and for $n = 0, 1, \dots, N$

$$\begin{aligned} x_n &= s_n e^{-jn\pi/4}, \\ y_n &= r_n e^{-jn\pi/4}, \\ w_n &= n_n e^{-jn\pi/4}. \end{aligned} \quad (6.8)$$

It now follows that $a_n \in \mathcal{A} = \{e^{jp\pi/2}, p = 0, 1, 2, 3\}$, $x_0 = 1$, and

$$\begin{aligned} x_n &= b_n s_{n-1} e^{-jn\pi/4} \\ &= b_n e^{-j\pi/4} s_{n-1} e^{-j(n-1)\pi/4} = a_n x_{n-1}. \\ y_n &= (e^{j\phi} s_n + n_n) e^{-jn\pi/4} = e^{j\phi} x_n + w_n. \end{aligned} \quad (6.9)$$

Now we may conclude that also $x_n \in \mathcal{A}$ for all $n = 0, 1, \dots, N$, and that w_n , just like n_n , is circularly symmetric complex Gaussian with variance σ^2 per component. Moreover since \mathbf{b} is Gray-coded with respect to the interleaved code-bits, so is $\mathbf{a} = (a_1, a_2, \dots, a_N)$. From now on we will therefore focus on DE-QPSK.

6.3 Detection and decoding, single-carrier case, non-iterative

We will start by considering the single-carrier case. For some single sub-carrier we will discuss DE-QPSK modulation with incoherent reception. Based on trellis decoding techniques we will determine the a-posteriori symbol probabilities under the assumption that the (quantized) channel phase is uniform and unknown to the receiver. We also assume the transmitted symbols to be independent of each other and uniform.

6.3.1 Trellis representation, sub-trellises, decomposition

In this section we will focus on non-iterative detection. We start our analysis by noting that if we define $z_n = x_n e^{j\phi}$ for $n = 0, 1, \dots, N$, then, since $x_0 = 1$ and ϕ is uniform over $\{\pi l/16, l = 0, 1, \dots, 31\}$ it follows that

$$\Pr\{z_0 = e^{jl\pi/16}\} = 1/32, \text{ for } l = 0, 1, \dots, 31, \quad (6.10)$$

and $z_n \in \mathcal{Z} \triangleq \{e^{jl\pi/16}, l = 0, 1, \dots, 31\}$. Moreover, for $n = 1, 2, \dots, N$,

$$\begin{aligned} z_n &= a_n z_{n-1}, \text{ where} \\ \Pr\{a_n = e^{jp\pi/2}\} &= 1/4, \text{ for } p = 0, 1, 2, 3. \end{aligned} \quad (6.11)$$

The variables z_n for $n = 0, 1, \dots, N$ can now be regarded as states in a trellis and the iud symbols a_1, a_2, \dots, a_N correspond to transitions between states. The resulting graphical representation of our trellis can be found in Fig. 6.4.

If we would use the standard BCJR algorithm for computing the a-posteriori symbol probabilities in the trellis in Fig. 6.4, we have to do 32×4 multiplications in the forward pass, 32×4 multiplications in the backward pass, and $4 \times 32 \times 2$ multiplications and 4 normalizations in the combination pass, per trellis section, if the a-priori probabilities are all equal. In total this is 512 multiplications and 4 normalizations per trellis section. We suggest to focus only on multiplications and normalizations in this paper, since additions have a smaller complexity than multiplications and normalizations².

An important observation for our investigations is that the trellis can be seen to consist of eight sub-trellises $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_7$, that are *not connected to each other*. A similar observation was made by Chen et al. [18]. We will discuss connections between our work on the trellis decomposition and that of [18] later.

Sub-trellis \mathcal{T}_s consists of states $z_n \in \mathcal{Z}_s = \{e^{jl\pi/16}, l = s + 8p, p = 0, 1, 2, 3\}$, for $s = 0, 1, \dots, 7$. Fig. 6.4 shows the entire sub-trellis \mathcal{T}_0 , and the first section of sub-trellis \mathcal{T}_1 and of sub-trellis \mathcal{T}_7 .

Note that for the likelihood $\gamma_n(z_n)$ corresponding to some state $z_n \in \mathcal{Z}$ for $n = 0, 1, \dots, N$ in the trellis \mathcal{T} or in a sub-trellis we can write that

$$\gamma_n(z_n) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|y_n - z_n|^2}{2\sigma^2}\right). \quad (6.12)$$

6.3.2 Forward-backward algorithm, sub-trellises

In this sub-section we would like to focus on computing the a-posteriori symbol probabilities $\Pr\{a_n | y_0, y_1, \dots, y_N\}$ for all $n = 1, 2, \dots, N$ and all values $a_n \in \mathcal{A}$. It will be demonstrated that it is a relatively simple exercise to do this. We will show that the resulting a-posteriori probability is a *convex combination* of the a-posteriori probabilities corresponding to the eight sub-trellises. Computing the a-posteriori probabilities for each sub-trellis is simple and can be done without performing the BCJR algorithm, as was demonstrated by Colavolpe [19]. The coefficients of the convex combination do not depend on the trellis section index n , and are quite easy to determine as we will show here.

Forward recursion

In our forward pass we focus on sub-trellis \mathcal{T}_s , for some $s \in \{0, 1, \dots, 7\}$. For that sub-trellis we find out how to compute all the α 's in that sub-trellis first. Starting from $\alpha_0(z_0) = 1/32$ for all $z_0 \in \mathcal{Z}_s$ we can compute the α 's recursively from

$$\alpha_n(z_n) = \sum_{(z_{n-1}, a_n) \rightarrow z_n} \alpha_{n-1}(z_{n-1}) \frac{1}{4} \gamma_{n-1}(z_{n-1}), \quad (6.13)$$

²In the log-domain, multiplications and normalizations are replaced by additions, and additions are typically approximated by maximizations. This would more or less suggest to consider multiplications, normalizations, as well as additions, but for reasons of simplicity we neglect the additions here.

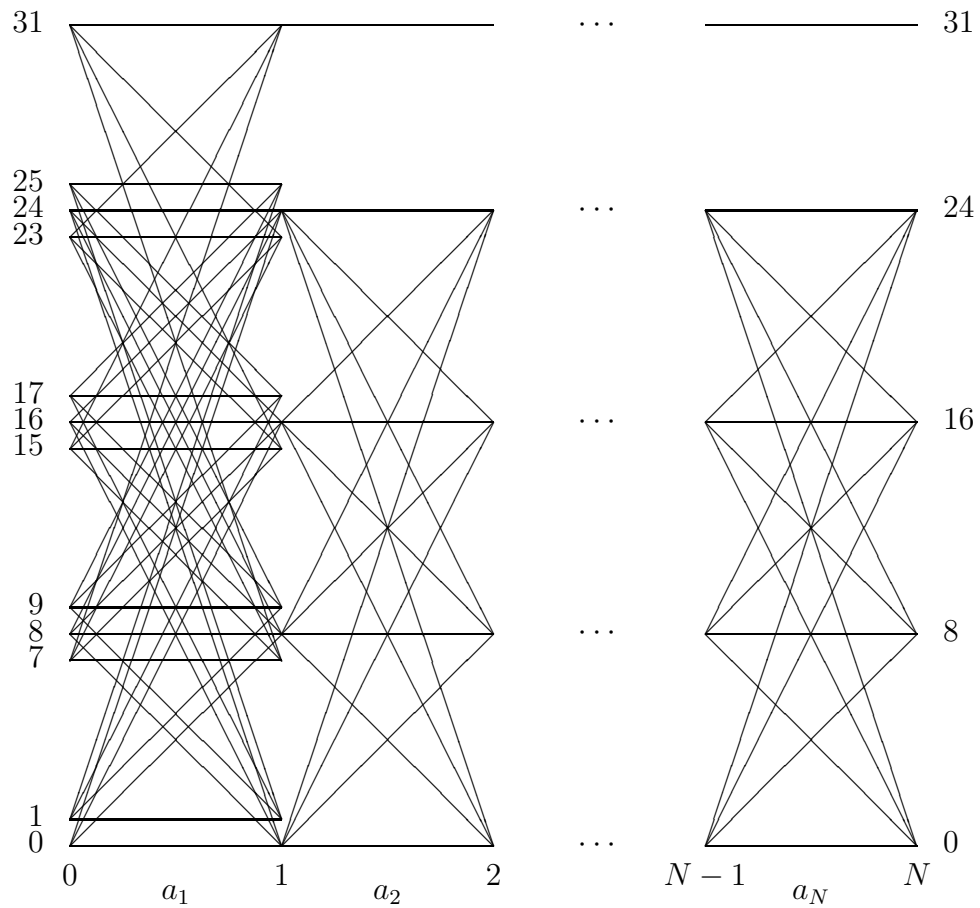


Figure 6.4: Trellis representation of the states z_0, z_1, \dots, z_N and the differentially encoded symbols a_1, a_2, \dots, a_N in the incoherent case. An edge between two subsequent states indicates that a transition between these states is possible. Note that the trellis can be decomposed into eight unconnected sub-trellises.

for $n = 1, 2, \dots, N$ and $z_n \in \mathcal{Z}_s$. The notation $(z, a) \rightarrow z'$ stands for all states z and symbols a that lead to next state z' .

Lemma 1. *If for $n = 0, 1, \dots, N$ we define $K_s(n) \triangleq \sum_{z_n \in \mathcal{Z}_s} \frac{1}{4} \gamma_n(z_n)$ then we have that*

$$\alpha_n(z_n) = \frac{1}{32} \prod_{i=0}^{n-1} K_s(i). \quad (6.14)$$

for $n = 0, 1, \dots, N$ and $z_n \in \mathcal{Z}_s$

Proof. Our proof is based on induction. Clearly for $n = 0$ the result holds. Now assume that $\alpha_{n-1}(z_{n-1}) = \frac{1}{32} \prod_{i=0}^{n-2} K_s(i)$ for $z_{n-1} \in \mathcal{Z}_s$, then from (6.13) we obtain

$$\begin{aligned} \alpha_n(z_n) &= \sum_{(z_{n-1}, a_n) \rightarrow z_n} \left(\frac{1}{32} \prod_{i=0}^{n-2} K_s(i) \right) \frac{1}{4} \gamma_{n-1}(z_{n-1}) \\ &= \frac{1}{32} \prod_{i=0}^{n-2} K_s(i) \sum_{z_{n-1} \in \mathcal{Z}_s} \frac{1}{4} \gamma_{n-1}(z_{n-1}) \\ &= \frac{1}{32} \prod_{i=0}^{n-1} K_s(i), \end{aligned} \quad (6.15)$$

for all $z_n \in \mathcal{Z}_s$. □

Backward recursion

Also in the backward pass we first focus only on sub-trellis \mathcal{T}_s for some s . In this sub-trellis we would like to compute the β 's. Taking $\beta_N(z_N) = \gamma(z_N)$ for $z_N \in \mathcal{Z}_s$ we can compute all other β 's from

$$\beta_n(z_n) = \sum_{a_{n+1}} \frac{1}{4} \gamma_n(z_n) \beta_{n+1}(z_n a_{n+1}), \quad (6.16)$$

where again $n = 0, 1, \dots, N-1$ and $z_n \in \mathcal{Z}_s$.

Lemma 2. *Based on definition of $K_s(n) \triangleq \sum_{z_n \in \mathcal{Z}_s} \frac{1}{4} \gamma_n(z_n)$ for all $n = 0, 1, \dots, N$ we get*

$$\beta_n(z_n) = \gamma_n(z_n) \prod_{i=n+1}^N K_s(i), \quad (6.17)$$

for $n = 0, 1, \dots, N$ and all $Z_n \in \mathcal{Z}_s$.

Proof. Again our proof is based on induction. Note first that for $n = N$ the result holds. Now assume that $\beta_{n+1}(z_{n+1}) = \gamma_{n+1}(z_{n+1}) \prod_{i=n+2}^N K_s(i)$, for $z_{n+1} \in \mathcal{Z}_s$. Then

$$\begin{aligned}
 \beta_n(z_n) &= \sum_{a_{n+1}} \frac{1}{4} \gamma_n(z_n) \left(\gamma_{n+1}(z_n a_{n+1}) \prod_{i=n+2}^N K_s(i) \right) \\
 &= \gamma_n(z_n) \sum_{z_{n+1} \in \mathcal{Z}_s} \frac{1}{4} \gamma_{n+1}(z_{n+1}) \prod_{i=n+2}^N K_s(i) \\
 &= \gamma_n(z_n) \prod_{i=n+1}^N K_s(i), \tag{6.18}
 \end{aligned}$$

for all $z_n \in \mathcal{Z}_s$. □

6.3.3 Combination

To determine the a-posteriori symbol probability for symbol value $a_n \in \mathcal{A}$ we compute the joint probability and density

$$\begin{aligned}
 &\Pr\{a_n\} p(\mathbf{y}|a_n) \\
 &= \sum_{z_{n-1} \in \mathcal{Z}} \alpha_{n-1}(z_{n-1}) \frac{1}{4} \gamma_{n-1}(z_{n-1}) \beta_n(z_{n-1} a_n) \\
 &= \sum_{s=0}^7 \sum_{z_{n-1} \in \mathcal{Z}_s} \alpha_{n-1}(z_{n-1}) \frac{1}{4} \gamma_{n-1}(z_{n-1}) \beta_n(z_{n-1} a_n) \\
 &= \sum_{s=0}^7 \frac{1}{32} \prod_{i=0}^{n-2} K_s(i) \prod_{j=n+1}^N K_s(j) \\
 &\quad \cdot \sum_{z_{n-1} \in \mathcal{Z}_s} \frac{1}{4} \gamma_{n-1}(z_{n-1}) \gamma_n(z_{n-1} a_n), \tag{6.19}
 \end{aligned}$$

If we consider the “middle” term in (6.19) then we see that

$$\begin{aligned}
 &\sum_{a_n} \left(\sum_{z_{n-1} \in \mathcal{Z}_s} \frac{1}{4} \gamma_{n-1}(z_{n-1}) \gamma_n(z_{n-1} a_n) \right) \\
 &= \sum_{z_{n-1} \in \mathcal{Z}_s} \frac{1}{4} \gamma_{n-1}(z_{n-1}) \sum_{a_n} \gamma_n(z_{n-1} a_n) \\
 &= 4K_s(n-1)K_s(n). \tag{6.20}
 \end{aligned}$$

From this we may conclude that

$$p(\mathbf{y}) = \sum_{s=0}^7 \Pr\{s\}p(\mathbf{y}|s) = \sum_{s=0}^7 \frac{1}{8} \prod_{i=0}^N K_s(i), \quad (6.21)$$

with

$$\begin{aligned} \Pr\{s\} &= 1/8, \text{ for } s = 0, 1, \dots, 7 \\ p(\mathbf{y}|s) &= \prod_{i=0}^N K_s(i). \end{aligned} \quad (6.22)$$

Now observing that

$$\Pr\{s|\mathbf{y}\} = \frac{\frac{1}{8} \prod_{i=0}^N K_s(i)}{\sum_{s=0}^7 \frac{1}{8} \prod_{i=0}^N K_s(i)}, \text{ and} \quad (6.23)$$

$$\Pr\{a_n|\mathbf{y}, s\} = \frac{\sum_{z_{n-1} \in \mathcal{Z}_s} \frac{1}{4} \gamma_{n-1}(z_{n-1}) \gamma_n(z_{n-1} a_n)}{4K_s(n-1)K_s(n)} \quad (6.24)$$

for $s \in \{0, 1, \dots, 7\}$ and $a_n \in \mathcal{A}$, we can write that

$$\Pr\{a_n|\mathbf{y}\} = \sum_{s=0}^7 \Pr\{s|\mathbf{y}\} \Pr\{a_n|\mathbf{y}, s\}. \quad (6.25)$$

The right-hand side of this equation can be interpreted as a convex combination of a-posteriori symbol probabilities $\Pr\{a_n|\mathbf{y}, s\}$, one for each sub-trellis, where the weighting-coefficients are the a-posteriori sub-trellis probabilities $\Pr\{s|\mathbf{y}\}$. An a-posteriori sub-trellis probability is the conditional probability that the discrete channel phase modulo 8 equals s for some $s = 0, 1, \dots, 7$ given \mathbf{y} .

The demodulator that operates according to (6.25) has three tasks, first the eight weighting coefficients (6.23) have to be computed, then for each of the eight sub-trellises for all symbol values $a_n \in \mathcal{A}$ and all $n \in \{1, 2, \dots, N\}$ the a-posteriori symbol probabilities have to be computed. Finally the weighting (6.25) has to be done. Computing the weighting coefficient requires for each sub-trellis $s \in \{0, 1, \dots, 7\}$ the computation of the factors $K_s(n)$ for $n = 0, 1, \dots, N$. These factors should then be multiplied and normalized to form $\Pr\{s|\mathbf{y}\}$. For these computations 8 multiplications per trellis section are needed. Computing the a-posteriori symbol probabilities $\Pr\{a_n|\mathbf{y}, s\}$ can be done efficiently by applying the Colavolpe [19] technique to each sub-trellis. As in Colavolpe each such a-posteriori symbol probability is based on only two received symbols y_{n-1} and y_n as is shown in (6.24). This avoids the use of the BCJR method in full generality and leads to significant complexity reductions, i.e., only $8 \times 4 \times 4 = 128$ multiplications and $8 \times 4 = 32$ normalizations are needed per trellis section. The weighting operation requires $8 \times 4 = 32$ multiplications, and therefore in total this approach leads to $8 + 128 + 32 = 168$ multiplications and 32 normalizations, which is considerably less than what we need for full BCJR.

6.3.4 Dominant sub-trellis approach

Equation (6.25) shows how the *exact* a-posteriori symbol probabilities can be determined. If the a-posteriori sub-trellis probabilities are such that one of the probabilities dominates the other ones then weighting (6.25) can be approximated by

$$\Pr\{a_n|\mathbf{y}\} \approx \Pr\{a_n|\mathbf{y}, \hat{s}\}, \text{ with } \hat{s} = \arg \max_s \Pr\{s|\mathbf{y}\}. \quad (6.26)$$

Observe that this approach involves the computations of the a-posteriori symbol probabilities, as described in (6.24), only for the dominant sub-trellis \hat{s} . This requires $4 \times 4 = 16$ multiplications and 4 normalizations only per trellis section. Together with the computation of the weighting coefficients $8 + 16 = 24$ multiplications and 4 normalizations are necessary. Therefore this reduces the number of multiplications with respect to full weighting by a factor of seven.

6.3.5 Simulations

We use in our simulations, just like Peleg et al. [58], the de-facto industry standard $R_c = \frac{1}{2}$ convolutional code with generator polynomials $g_0 = 133$ and $g_1 = 171$, which is equal to the convolutional code with puncturing index $PI = 8$ of Table 29 in [25, Sec. 11.1.2, p.131]. The DAB, DAB+, and T-DMB bit-reversal time interleaver and block frequency interleaver is modeled by a bit-wise uniform block interleaver generated for each simulated code block of bits, hence, any permutation of the coded bits is a permissible interleaver and is selected with equal probability, as is done in [58].

The demodulator calculates, for each OFDM-subcarrier, the a-posteriori probability given by (6.25) for $N + 1 = 2, 4, 8,$ and 32 . The demodulator is followed by a convolutional decoder which needs as input soft-decision information about the coded bits. Now, it follows from Gray mapping, i.e.,

$$\begin{array}{c|cccc} b_1 b_2 & 00 & 01 & 11 & 10 \\ \hline a(b_1 b_2) & 1 & e^{j\pi/2} & e^{j\pi} & e^{j3\pi/2} \end{array},$$

that the desired metrics related to transmission n , i.e., the LLRs [37] can be expressed as

$$\lambda_n^1 = \ln \left(\frac{e^{m(\pi)} + e^{m(3\pi/2)}}{e^{m(0)} + e^{m(\pi/2)}} \right), \lambda_n^2 = \ln \left(\frac{e^{m(\pi/2)} + e^{m(\pi)}}{e^{m(0)} + e^{m(3\pi/2)}} \right), \quad (6.27)$$

with symbol metric

$$m(\phi) = \ln \left(\Pr\{a_n = e^{j\phi}|\mathbf{y}\} \right), \quad (6.28)$$

and where λ_n^1 corresponds to bit b_1 and λ_n^2 to bit b_2 .

Fig. 6.5 shows the BER performance with so-called ideal LLRs for a decomposed trellis for trellis-length $N + 1 = 2, 4, 8,$ and 32 . On the horizontal axis is the signal-to-noise ratio $E_b/N_0 = \frac{1}{2\sigma^2}$. The demodulator operates according to (6.25).

We will compare the performance of this demodulator with that of two well-known procedures described in the literature. Firstly, to ‘‘classical’’ DQPSK [61, Sec. 4.5-5, p. 224],

i.e., 2SDD. This leads to a-posteriori symbol probabilities as in (9) in Divsalar and Simon [22], i.e., to

$$\Pr\{a_n|y_n, y_{n-1}\} \propto I_0\left(\frac{1}{\sigma^2}|y_n a_n^* + y_{n-1}|\right), \text{ for } a_n \in \mathcal{A}, \quad (6.29)$$

where $I_0(\cdot)$ is the zeroth order modified Bessel function of the first kind. Secondly, we will compare our results to coherently detected DE-QPSK. We assume that the received sequence is perfectly de-rotated, i.e., $\tilde{\mathbf{y}} = \mathbf{y}e^{-j\phi}$. Then the a-posteriori symbol probabilities are given by

$$\Pr\{a_n|\tilde{\mathbf{y}}\} \propto \sum_{x_{n-1} \in \mathcal{A}} \exp\left(\frac{1}{\sigma^2}\Re\{x_{n-1}^*(\tilde{y}_n a_n^* + \tilde{y}_{n-1})\}\right), \text{ for } a_n \in \mathcal{A}, \quad (6.30)$$

as described by Colavolpe [19]. Note that (6.30) is similar to (6.24) for $s = 0$.

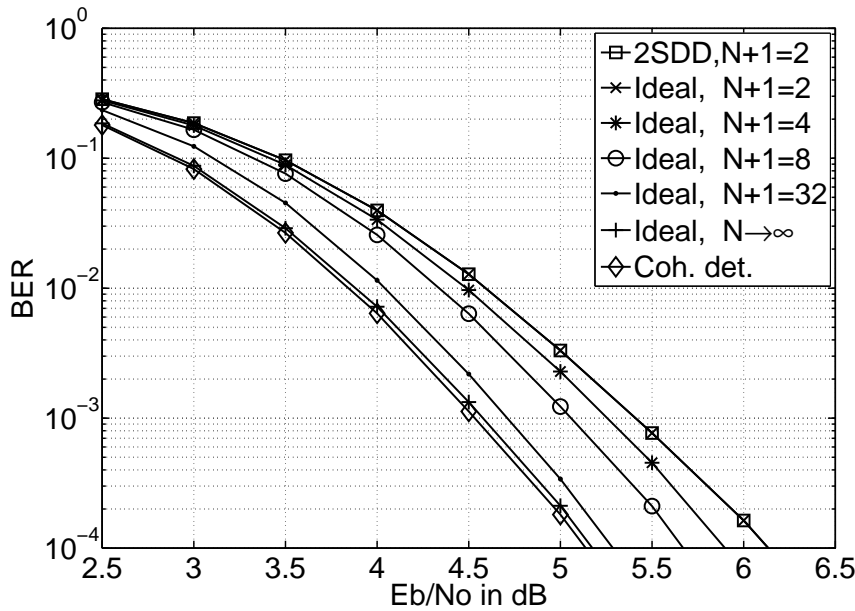


Figure 6.5: Bit-error performance for LLRs computed as in (6.25), i.e., ideal LLRs, for different trellis-lengths.

The simulation results, which are shown in Fig. 6.5, demonstrate that the BER performance curves of 2SDD and trellis length $N + 1 = 2$ are practically identical as we expect. Moreover, the coherent-detection curve and the curve for very large trellis-sizes

($N \rightarrow \infty$) are very close. The small performance loss is due to discretizing the channel-phase with 32 levels. Furthermore, Fig. 6.5 shows that: (a) larger values of $N + 1$ result in performance closer to the coherent-detection performance, and, (b) for $N + 1 = 32$ ideally computed LLRs for a decomposed trellis perform quite close to coherent detection, i.e., the difference in signal-to-noise ratio (E_b/N_0) is less than 0.15 dB at a BER of 10^{-4} . Next, in Fig. 6.6, we turn to the dominant sub-trellis approach, which is denoted by “Max” in the legend. We compare for trellis-length $N + 1 = 2, 8,$ and 32 , the difference in performance between ideal LLRs based on the a-posteriori probabilities given by (6.25) and the approximated LLRs based on the dominant-sub-trellis a-posteriori probabilities specified in (6.26). It can be seen from Fig. 6.6 that: (a) for larger $N + 1$ the difference between the exact and approximated LLRs becomes smaller, and, (b) for $N + 1 = 32$ the difference between the ideal LLRs and the approximated LLRs, is less than 0.1 dB.

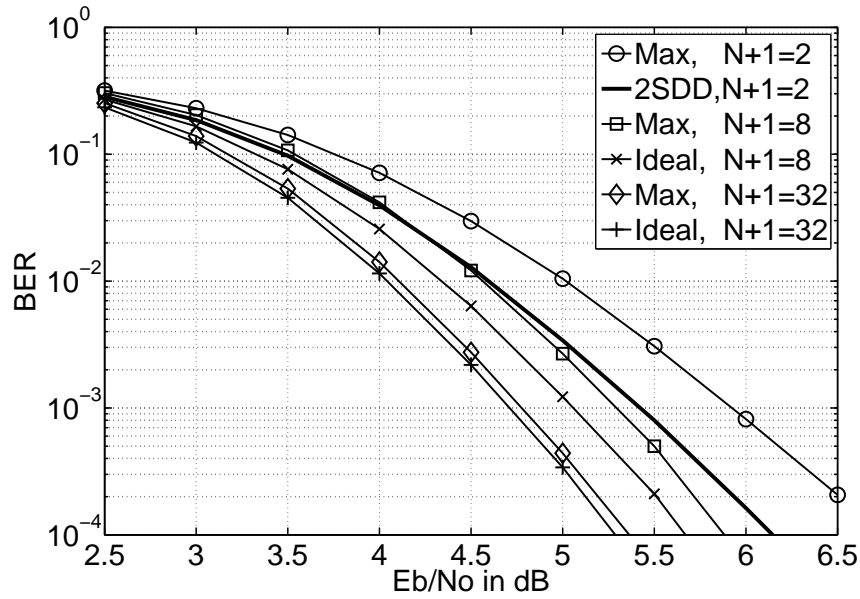


Figure 6.6: Bit-error performance for LLRs computed as in (6.25), i.e., ideal LLRs, and approximated LLRs computed as in (6.26), for different trellis-lengths.

6.3.6 Some conclusions

Our simulations demonstrate, that for trellis length $N + 1 = 32$, the ideal LLRs and the approximated LLRs have a performance quite close to that of coherent detection. The difference in signal-to-noise ratio is less than 0.25 dB at a BER of 10^{-4} for the dominant-

trellis approach. Therefore, if we focus on a BER of 10^{-4} for obtaining an acceptable performance, with single sub-carrier transmission, we need a trellis-length $N + 1 \geq 32$. With a trellis-length of $N + 1 \geq 32$ received symbols, the channel coherence-time needs to be in the order of $T_c \approx 32T_s$, where T_s is the OFDM symbol time. This imposes quite a strong restriction on the time-varying behavior of the channel. In practice the channel may not be coherent so long, and therefore focussing on trellis-length $N + 1 = 32$ might not be realistic. We will discuss this effect in more detail in Sec. 6.7, where we study a typical urban channel. There is a second reason for arguing that large values of N are undesirable. DAB-systems support, for complexity reduction, *per service symbol processing*. In such services, typically, at most $N + 1 \leq 4$ subsequent OFDM symbols are contained in a single convolutionally encoded word, see Fig. 6.2, and this does not match to processing more than four OFDM symbols in a demodulation trellis.

After having concluded that we cannot make N too large, it makes sense to investigate the possibility of using a number of (adjacent) sub-carriers to jointly determine the a-posteriori symbol probabilities for the corresponding DE-QPSK streams. Instead of using a single trellis with length $N + 1 = 32$ we could find out whether a similar performance can be obtained with a 2D-block of $M = 8$ trellises of length $N + 1 = 4$ corresponding to adjacent sub-carriers, see Fig. 6.3. This will be the subject of the next section.

6.4 Detection and decoding, multi-carrier case non-iterative

6.4.1 Demodulation procedures

We have seen, that the trellis-length $N + 1$ needs to be as large as possible. For obtaining an acceptable performance, it must be larger than 32. This may not always be true. Therefore we want to investigate the question of jointly decoding a block (2D) of received symbols. It would again be nice if we could decompose the computation of the a-posteriori probabilities as in (6.25), also if we would concentrate on what was received over several sub-carriers.

Now we assume that in each subcarrier $m = 1, 2, \dots, M$ a sequence

$\mathbf{a}_m = (a_{m,1}, a_{m,2}, \dots, a_{m,N})$ is conveyed using differential encoding. For the components of the transmitted sequence $\mathbf{x}_m = (x_{m,0}, x_{m,1}, \dots, x_{m,N})$ we can write

$$x_{m,n} = a_{m,n}x_{m,n-1}, \quad (6.31)$$

and $x_{m,0} = 1$. We assume that the channel phase is constant over the block of symbols, therefore

$$y_{m,n} = e^{j\phi}x_{m,n} + w_{m,n}, \quad (6.32)$$

where $\phi \in \{l\pi/16, l = 0, 1, \dots, 31\}$ and uniform just as before, and the noise variables $w_{m,n}$ are circularly complex Gaussian with variance σ^2 per component. The output sequence corresponding to sub-carrier m is denoted by $\mathbf{y}_m = (y_{m,0}, y_{m,1}, \dots, y_{m,N})$.

Just like in the single carrier case we can determine the a-posteriori sub-trellis probabilities.

$$\begin{aligned} & \Pr\{s|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \\ &= \frac{\Pr\{s\}p(\mathbf{y}_1|s)p(\mathbf{y}_2|s) \cdots p(\mathbf{y}_M|s)}{\sum_{s=0}^7 \Pr\{s\}p(\mathbf{y}_1|s)p(\mathbf{y}_2|s) \cdots p(\mathbf{y}_M|s)}, \end{aligned} \quad (6.33)$$

where $\Pr\{s\} = 1/8$ for $s = 0, 1, \dots, 7$ and

$$p(\mathbf{y}_m|s) = \prod_{i=0}^N K_{m,s}(i), \quad (6.34)$$

where $K_{m,s}(n) \triangleq \sum_{z_n \in \mathcal{Z}_s} \frac{1}{4} \gamma_{m,n}(z_n)$. Note that for the likelihood corresponding to some state z_n for $n = 0, 1, \dots, N$ in the trellis \mathcal{T} or in a sub-trellis we can write that

$$\gamma_{m,n}(z_n) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|y_{m,n} - z_n|^2}{2\sigma^2}\right). \quad (6.35)$$

Now the a-posteriori symbol probability for $a_{m,n} \in \mathcal{A}$ can be written as

$$\begin{aligned} & \Pr\{a_{m,n}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \\ &= \sum_{s=0}^7 \Pr\{s|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \Pr\{a_{m,n}|\mathbf{y}_m, s\}. \end{aligned} \quad (6.36)$$

where

$$\begin{aligned} & \Pr\{s|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \\ &= \frac{\frac{1}{8} \prod_{m=1}^M \prod_{i=0}^N K_{m,s}(i)}{\sum_{s=0}^7 \frac{1}{8} \prod_{m=1}^M \prod_{i=0}^N K_{m,s}(i)}, \end{aligned} \quad (6.37)$$

and

$$\Pr\{a_{m,n}|\mathbf{y}_m, s\} = \frac{\sum_{z_{n-1} \in \mathcal{Z}_s} \frac{1}{4} \gamma_{m,n-1}(z_{n-1}) \gamma_{m,n}(z_{n-1} a_n)}{4K_{m,s}(n-1)K_{m,s}(n)} \quad (6.38)$$

for $s \in \{0, 1, \dots, 7\}$ and $a_{m,n} \in \mathcal{A}$.

This suggests that the demodulator first determines the a-posteriori sub-trellis probabilities (weighting coefficients) using (6.37), for which first $8 \times M \times (N + 1)$ K -factors have to be computed. Using the weighting coefficients the convex combination in (6.36) then leads to the a-posteriori symbol probabilities. Finding the a-posteriori symbol probabilities $\Pr\{a_{m,n}|\mathbf{y}_m, s\}$ again can be done using the Colavolpe [19] method for each sub-carrier and for each sub-trellis, where again such an a-posteriori symbol probability

is based on only the two received symbols $y_{m,n-1}$ and $y_{m,n}$ as is shown in (6.38). Again the BCJR method in full generality is not needed, and the number of required multiplications and normalizations per trellis section are the same as in the single carrier case. Equation (6.36) shows how the exact a-posteriori symbol probabilities can be determined. Just like in the single-carrier case, if the a-posteriori sub-trellis probabilities are such that one of the probabilities dominates the other ones then weighting (6.36) can be approximated as follows:

$$\Pr\{a_{m,n}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \approx \Pr\{a_{m,n}|\mathbf{y}_m, \hat{s}\}, \quad (6.39)$$

with

$$\hat{s} = \arg \max_s \Pr\{s|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\}.$$

Again this approach involves the computations of the a-posteriori symbol probabilities only for the dominant sub-trellis \hat{s} . The resulting number of multiplications and normalizations per trellis section is the same as for the single carrier case.

6.4.2 Simulations

In the previous section we analyzed and simulated the single sub-carrier approach. Here we will discuss the simulations corresponding to the multi-carrier method. We will again study the coded BER versus the signal-to-noise ratio $E_b/N_0 = \frac{1}{2\sigma^2}$. The BER performance for the ideal LLRs, based on a-posteriori probabilities computed as in (6.36) is shown in Fig. 6.7 with a fixed block-size of $M(N+1) = 16$. This fixed block-size is realized by the parameter pair values $(M, N+1) = (1, 16), (2, 8), (4, 4),$ and $(8, 2)$. The detector operates according to (6.36). The performance of 2SDD and coherently detected DE-QPSK are shown as reference curves.

From Fig. 6.7 can be observed that, a 2D decomposition with a shortest possible trellis-length of $N+1 = 2$ and $M = 8$ adjacent sub-carriers performs identical to the largest trellis-length $N+1 = 16$ and $M = 1$ sub-carrier, i.e., the single-carrier case. Intermediate cases also have an identical performance.

We do not show the results of the dominant sub-trellis approach for the multi-carrier case here, since these results are identical to the corresponding results for the single-carrier case shown in Fig. 6.6 in the previous section.

6.4.3 Conclusion non-iterative decoding

Our investigations for the non-iterative 2D-case, show that we are very close to the performance of coherent detection of DE-QPSK even for small values of the trellis length $N+1$, by processing simultaneously several sub-carriers. A next question is whether we can do better than this. In the literature, see, e.g., Peleg et al. in [58] and Chen et al. [18], it is demonstrated that iterative decoding techniques lead to good results for differential encoding. Therefore, in the sequel of this paper we study iterative decoding techniques for DAB-like streams, with a special focus again on 2D blocks.

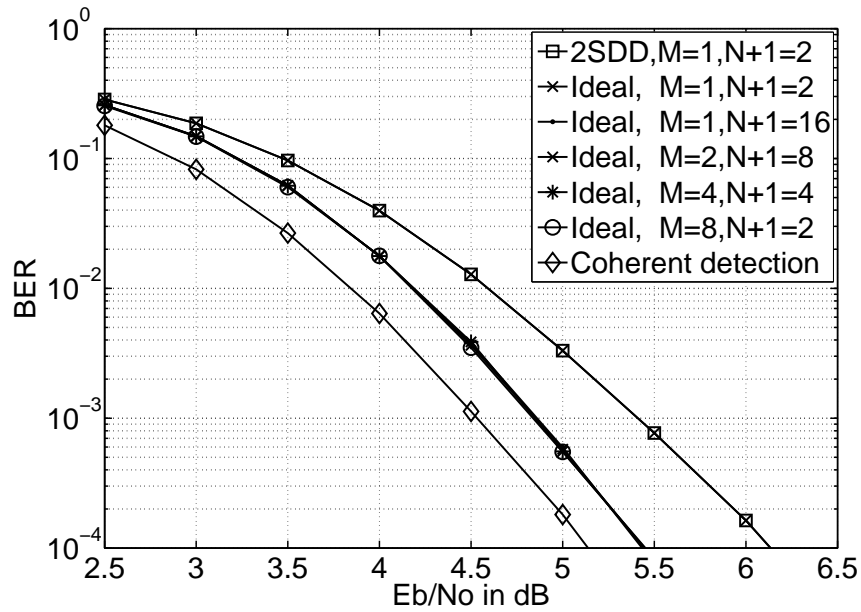


Figure 6.7: Bit-error performance with ideal LLRs for a decomposed multi-carrier trellis for different values of M and N but with a fixed block-size of $M(N + 1) = 16$.

6.5 Detection and decoding, single-carrier case, iterative

In the following two sections we consider iterative decoding procedures. Peleg and Shamai [57] first demonstrated that iterative techniques could increase the performance of the demodulation procedures of DE-QPSK streams significantly. We specialize their approach to DAB systems and solve a problem connected to the, in practise quite small, length of the trellises for each subcarrier, by turning to 2D-blocks for iterative demodulation.

6.5.1 Serial concatenation

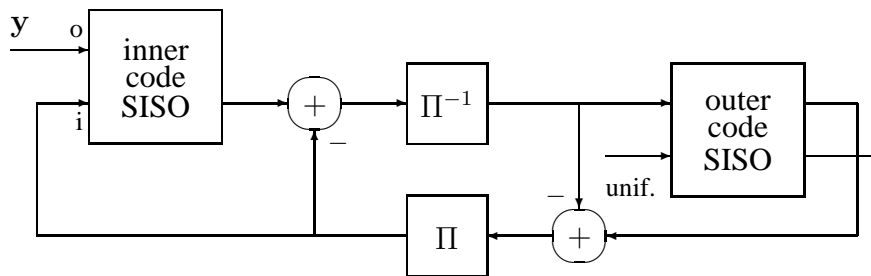


Figure 6.8: Structure of the receiver.

In the current section we will investigate iterative decoding procedures for DAB-like systems which are based on convolutional encoding, interleaving, and DE-QPSK modulation. If we consider DE-QPSK modulation as the inner coding method and convolutional encoding as the outer code, then it is obvious that we can apply techniques developed for serially concatenated coding systems here, see Fig. 6.8. Serially concatenated turbo codes were proposed by Benedetto and Montorsi [9] and later investigated in more detail in Benedetto, Divsalar, Montorsi, and Pollara [7]. Iterating between the DPSK-demodulator and convolutional decoder for the incoherent case was first suggested (for a single carrier) by Peleg and Shamai [57]. Hoeher and Lodge [42] also applied iterative techniques to the incoherent case, but focussed on channel estimation, to be able to use coherent detection. For an overview of related results, all for the single-carrier case, see Chen et al. [18].

We will start in this section by considering the single carrier case and our aim is again to find out what we can gain from decomposing the trellis used in the demodulator into a part that corresponds to the channel phase and a part that relates to differential encoding. In the section that follows we will consider the multi-carrier setting.

6.5.2 Peleg approach

In this subsection we investigate the Forward Backward procedures where we drop the assumption that the symbols $a_n, n = 1, 2, \dots, N$ are uniformly distributed. Interleaving should still guarantee the independence of the symbols however.

Just like Peleg et al. [58] we focus on the entire trellis \mathcal{T} . Note however that our trellis is different from that of [58], in which tracking of small channel phase variations is made possible by adding “side-step” transitions. We don’t have such transitions in our trellis and therefore our trellis can be decomposed in eight unconnected sub-trellises. In the next sub-section we take advantage of this decomposition, however first we will consider the un-decomposed trellis.

Again starting from $\alpha_0(z_0) = 1/32$ for all $z_0 \in \mathcal{Z}$ we can compute the α ’s recursively from

$$\alpha_n(z_n) = \sum_{(z_{n-1}, a_n) \rightarrow z_n} \alpha_{n-1}(z_{n-1}) \Pr\{a_n\} \gamma_{n-1}(z_{n-1}), \quad (6.40)$$

for $n = 1, 2, \dots, N$ and $z_n \in \mathcal{Z}$. Also in the backward pass we consider the entire trellis \mathcal{T} . Taking $\beta_N(z_N) = \gamma(z_N)$ for $z_N \in \mathcal{Z}$ we can compute all other β ’s from

$$\beta_n(z_n) = \sum_{a_{n+1}} \gamma_n(z_n) \Pr\{a_{n+1}\} \beta_{n+1}(z_n a_{n+1}), \quad (6.41)$$

where again $n = 0, 1, \dots, N - 1$ and $z_n \in \mathcal{Z}$.

To determine the a-posteriori symbol probability for symbol value $a_n \in \mathcal{A}$ we compute the joint probability and density

$$\begin{aligned} \Pr\{a_n\} p(\mathbf{y}|a_n) \\ = \sum_{z_{n-1} \in \mathcal{Z}} \alpha_{n-1}(z_{n-1}) \Pr\{a_n\} \gamma_{n-1}(z_{n-1}) \beta_n(z_{n-1} a_n), \end{aligned} \quad (6.42)$$

This expression also tells us how the resulting extrinsic information can be determined. It can be checked, see Benedetto et al. [9], that multiplying by the factors $\Pr\{a_n\}$ in the a-posteriori information (6.42) should be omitted for obtaining extrinsic information. The extrinsic information is now further processed by the convolutional decoder. The results of the iterative procedure are discussed in subsection 6.5.5.

Using the standard BCJR algorithm for computing the extrinsic symbol probabilities in the trellis in Fig. 6.4, since a-priori symbol probabilities are non-uniform now, leads to $32 \times 4 \times 2$ multiplications in the forward pass, $32 \times 4 \times 2$ multiplications in the backward pass, and $32 \times 4 \times 2$ multiplications and 4 normalizations in the combination pass for computing extrinsic information, per trellis section. In total this is 768 multiplications and 4 normalizations per trellis section per iteration. In the next subsection we investigate the decomposition of the demodulation trellis.

6.5.3 Trellis decomposition

Here we investigate whether we can decompose the entire trellis for the case where the a-priori probabilities are non-uniform. We are interested in decomposing (6.42) in such a way that we can write

$$\Pr\{a_n|\mathbf{y}\} = \sum_s \Pr\{s, a_n|\mathbf{y}\} = \sum_s \Pr\{s|\mathbf{y}\} \Pr\{a_n|\mathbf{y}, s\}, \quad (6.43)$$

for all $a_n \in \mathcal{A}$. The question now is how to compute the a-posteriori sub-trellis probabilities $\Pr\{s|\mathbf{y}\}$ for $s = 0, 1, \dots, 7$.

It can be shown that

$$\Pr\{s, \mathbf{y}\} = \sum_{z_0 \in \mathcal{Z}_s} \beta_0(z_0)/32, \quad (6.44)$$

and therefore

$$\Pr\{s|\mathbf{y}\} = \frac{\sum_{z_0 \in \mathcal{Z}_s} \beta_0(z_0)/32}{\sum_{s'=0}^7 \sum_{z_0 \in \mathcal{Z}_{s'}} \beta_0(z_0)/32}. \quad (6.45)$$

Now for each sub-trellis we can determine the a-posteriori symbol probabilities using

$$\begin{aligned} \Pr\{a_n|\mathbf{y}, s\} &\propto \Pr\{a_n, s\} p(\mathbf{y}|a_n, s) \\ &= \Pr\{a_n, s\} \sum_{z_{n-1} \in \mathcal{Z}_s} \alpha_{n-1}(z_{n-1}) \gamma_{n-1}(z_{n-1}) \beta_n(z_{n-1} a_n), \end{aligned} \quad (6.46)$$

and by omitting the factor $\Pr\{a_n, s\}$ in (6.46) the corresponding extrinsic information. Note that this approach requires a backward pass through the entire trellis \mathcal{T} first to find the weighting probabilities $\Pr\{s|\mathbf{y}\}$, for $s = 0, 1, \dots, 7$. This requires $32 \times (4+1) = 160$ multiplications per trellis section observing that in (6.41) $\gamma_n(z_n)$ can be put in front of the summation sign. Then for all sub-trellises \mathcal{T}_s we do a forward pass (requiring $8 \times 4 \times 4 \times 2 = 256$ multiplications per section) and then combine the results to obtain the extrinsic symbol probabilities $\Pr\{a_n|\mathbf{y}, s\}$ for that sub-trellis (for which we need $8 \times 4 \times 4 \times 2 = 256$ multiplications and $8 \times 4 = 32$ normalizations per section). Finally these probabilities have to be weighted as in (6.43) which requires $8 \times 4 = 32$ multiplications. In total this results in 704 multiplications and 32 normalizations per iteration. It should be noted that decomposition of the trellis, does not result in a significant complexity reduction with respect to the Peleg approach. In the next subsection we will discuss an approach that gives a relevant complexity reduction however.

6.5.4 Dominant sub-trellis approaches

To achieve a complexity reduction we investigate a method that is based on finding, at the start of a new iteration, the dominant sub-trellis first and then do the forward-backward processing for demodulation only in this dominant sub-trellis.

Finding the dominant sub-trellis for an iteration is done based on the a-posteriori sub-trellis probabilities $\Pr\{s|\mathbf{y}\}$ that are computed before starting this iteration. Now assuming that one of the a-posteriori sub-trellis probabilities dominates the other ones we can write

$$\Pr\{a_n|\mathbf{y}\} \approx \Pr\{a_n|\mathbf{y}, \hat{s}\}, \text{ with } \hat{s} = \arg \max_s \Pr\{s|\mathbf{y}\}. \quad (6.47)$$

This approach involves the computations of the a-posteriori symbol probabilities (and corresponding extrinsic information) as described in (6.40), (6.41), and (6.42) only for the dominant sub-trellis \hat{s} . Computing the a-posteriori sub-trellis probabilities for each iteration and then focussing only on the forward pass and combination computations, is less complex than following the Peleg procedure. For the best sub-trellis $\mathcal{T}_{\hat{s}}$ we do a forward pass ($4 \times 4 \times 2 = 32$ multiplications per trellis-section) and then we combine the results to obtain the a-posteriori (actually extrinsic) symbol probabilities $\Pr\{a_n|\mathbf{y}, \hat{s}\}$ for that sub-trellis ($4 \times 4 \times 2 = 32$ multiplications and 4 normalizations per section). In total we now need 224 multiplications and 4 normalizations per trellis section per iteration.

A second approach involves choosing the dominant sub-trellis only once, before starting with the iterations. Since before starting the iterations the a-priori probabilities $\Pr\{a_n\} = 1/4$, i.e., are all equal, the analysis in subsection 6.3.4 applies. The a-posteriori sub-trellis probabilities can be computed as in (6.23). Now we do the iterations only in the sub-trellis that was chosen initially. This approach requires 84 multiplications and 4 normalizations per trellis section per iteration, and is therefore essentially less complex than the Peleg technique. In our simulations we will only use this last technique when we address dominant sub-trellises.

6.5.5 Simulations

We simulated the Peleg method described in [58] and determined the BER versus the signal-to-noise ratio $E_b/N_0 = \frac{1}{2\sigma^2}$. This BER performance is shown in Fig. 6.9 for trellis-lengths practically infinite, i.e., $N \rightarrow \infty$, and ideal LLRs based on the a-posteriori probability given by (6.42). The BER performance is shown for $L = 1, 2, \dots, 5$ iterations, where $L = 1$ stands for no iterations. Note that since we are using ideal LLRs and infinite trellis-lengths, the corresponding curves shown in Fig. 6.9 can be regarded as target curves for the iterative (single-carrier) case. In addition, also here, 2SDD and coherently detected DE-QPSK curves are shown for reference. Not in the figure are the curves corresponding to the approach based on decomposing the trellis and using weighting as in (6.43). As expected, the performance of this approach shows no differences with the Peleg approach in (6.42). From Fig. 6.9 it can be seen that for a BER = 10^{-4} the improvement in required signal to noise ratio is ≈ 4.1 dB after $L = 5$ iterations. Fig. 6.9 also shows that improvement decreases with the number of iterations and that the first iteration yields the largest improvement. Similar results were obtained by Peleg et al. [58].

To see how the performance in the iterative case depends on the trellis length N we simulated the Peleg approach for $N + 1 = 2, 4$, and 32, for $L = 5$ iterations. The results are in Fig. 6.10. It can be seen that the “iterative coding gain” increases, as expected, with N and that, for $N + 1 = 32$, the performance is already quite close to that of $N \rightarrow \infty$. Finally we compared for $N + 1 = 4$ and 32, the difference in BER between the exact LLRs based on the a-posteriori (extrinsic) probability given by (6.42) or (6.43), and the approximated LLRs based on the a-posteriori (extrinsic) probability given by (6.47). The

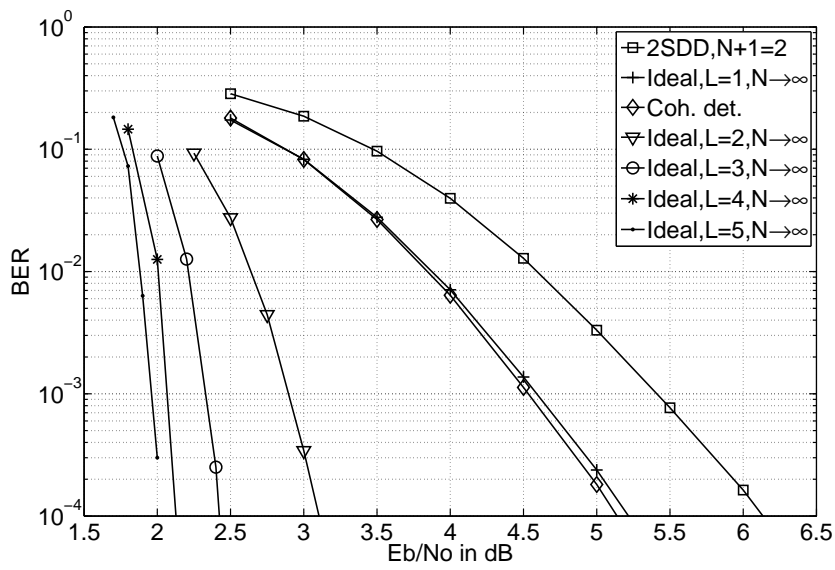


Figure 6.9: Bit-error performance of the Peleg method for trellis length $N \rightarrow \infty$ and up to $L = 5$ iterations.

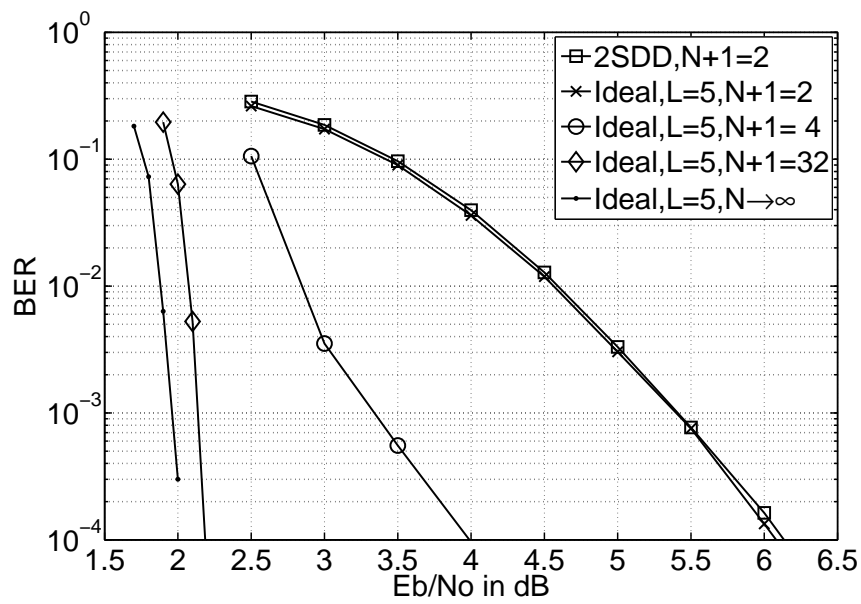


Figure 6.10: Bit-error performance for the Peleg method for different values of N and $L = 5$ iterations.

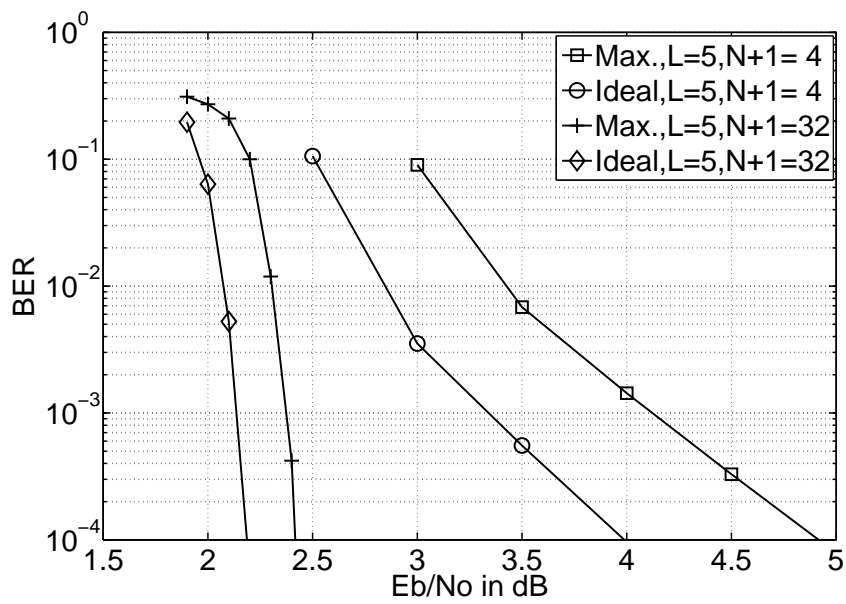


Figure 6.11: Bit-error performance with dominant sub-trellis approach for different values N and $L = 5$ iterations.

results are shown in Fig. 6.11. We can conclude from Fig. 6.11 that for larger $N + 1$ the difference in performance between the exact and approximated LLRs becomes smaller and that for $N + 1 = 32$ the difference between the ideal LLRs and the approximation versions, by selecting the dominant sub-trellis before starting with the iteration process, is less than 0.3 dB.

6.6 Detection and decoding, multi-carrier case, iterative

6.6.1 Trellis decomposition

Just like in the non-iterative multi-carrier case we do the processing based on trellis decomposition, and focus on the computation of the a-posteriori sub-trellis probabilities:

$$\begin{aligned} & \Pr\{s|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \\ &= \frac{\Pr\{s\}p(\mathbf{y}_1|s)p(\mathbf{y}_2|s) \dots p(\mathbf{y}_M|s)}{\sum_{s=0}^7 \Pr\{s\}p(\mathbf{y}_1|s)p(\mathbf{y}_2|s) \dots p(\mathbf{y}_M|s)}. \end{aligned} \quad (6.48)$$

Note that $\Pr\{s\} = 1/8$ for $s = 0, 1, \dots, 7$ and therefore it follows from (6.44) that

$$p(\mathbf{y}_m|s) = \sum_{z_0 \in \mathcal{Z}_s} \beta_{m,0}(z_0)/4, \quad (6.49)$$

for each subcarrier $m = 1, 2, \dots, M$.

Now the a-posteriori symbol probability for $a_{m,n} \in \mathcal{A}$ can be written as in (6.36), i.e.,

$$\begin{aligned} & \Pr\{a_{m,n}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \\ &= \sum_{s=0}^7 \Pr\{s|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \Pr\{a_{m,n}|\mathbf{y}_m, s\}, \end{aligned} \quad (6.50)$$

where $\Pr\{a_{m,n}|\mathbf{y}_m, s\}$ is computed as given by (6.46) for $s \in \{0, 1, \dots, 7\}$ and $a_{m,n} \in \mathcal{A}$. From these a-posteriori probabilities we can compute the extrinsic information that is needed by the convolutional decoder. Computing extrinsic information is actually a little bit easier since it involves less multiplications.

Thus suggests that, for each iteration, the demodulator first determines the a-posteriori sub-trellis probabilities using (6.48), for which first a backward pass in each of the M trellises corresponding to the sub-carriers is needed.

Using the weighting coefficients, the convex combination in (6.50) leads to the a-posteriori symbol probabilities. Finding the a-posteriori symbol probabilities $\Pr\{a_{m,n}|\mathbf{y}_m, s\}$ should be done in the standard way, taking into account that the backward passes were already carried out.

6.6.2 Dominant sub-trellis approach

Equation (6.50) shows how the exact a-posteriori symbol probabilities can be determined, in each iteration. Just like in the single-carrier case, if the a-posteriori sub-trellis probabilities are such that one of the probabilities dominates the other ones, then convex combination (6.50) can be approximated as follows:

$$\Pr\{a_{m,n}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \approx \Pr\{a_{m,n}|\mathbf{y}_m, \hat{s}\}, \quad (6.51)$$

with

$$\hat{s} = \arg \max_s \Pr\{s|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\}. \quad (6.52)$$

Again this approach involves, in each iteration, the computations of the a-posteriori symbol probabilities only for the dominant sub-trellis \hat{s} .

If we compute the dominant sub-trellis only before the start of the iteration process, we obtain a significant complexity reduction since the analysis in subsection 6.3.4 applies. Moreover all iterations are done in the initially chosen sub-trellis. The methods described here will be evaluated in the next subsection.

6.6.3 Simulations

We have seen before that in the non-iterative multi-carrier case the performance was more or less determined by the size $M(N+1)$ of the block. If the channel cannot be assumed to be constant for large values of $N+1$ we can always increase the number of sub-carriers M , if the frequency selectivity allows this. Note that keeping $N+1$ small also has advantages related to service symbol processing [25]. Here the situation is slightly different as is demonstrated in Fig. 6.12. Increasing M has a positive effect on the performance, however since the trellis-length $N+1$ remains constant (and is quite small), the effect of iterating is limited. We see however that by increasing M from 1 to 8 we get an improvement of roughly 0.7 dB.

Finally we compare for $N+1 = 4$ and 32, for $M = 8$, the difference between the performance of exact LLRs based on the a-posteriori (extrinsic) probabilities given by (6.50) and the approximated LLRs based on the a-posteriori (extrinsic) probabilities given by (6.51), see Fig. 6.13. We can observe from Fig. 6.13 that, as expected, the larger $N+1$ is, the smaller the difference between the exact and approximated LLRs becomes. For $N+1 = 4$ the difference between the ideal LLRs and the approximation, by selecting the dominant sub-trellis before starting to iterate, is roughly 0.3 dB.

6.7 Performance for TU-6 channel model

So far we have used AWGN channels with unknown channel phase and fixed (unit) gain in our analysis and simulations. To investigate the performance in practise we have used the TU-6 channel model defined in [1], which is commonly used to test DAB, DAB+, or T-DMB transmission. Two maximum Doppler frequencies are chosen, i.e., $f_d = 10$ and 20

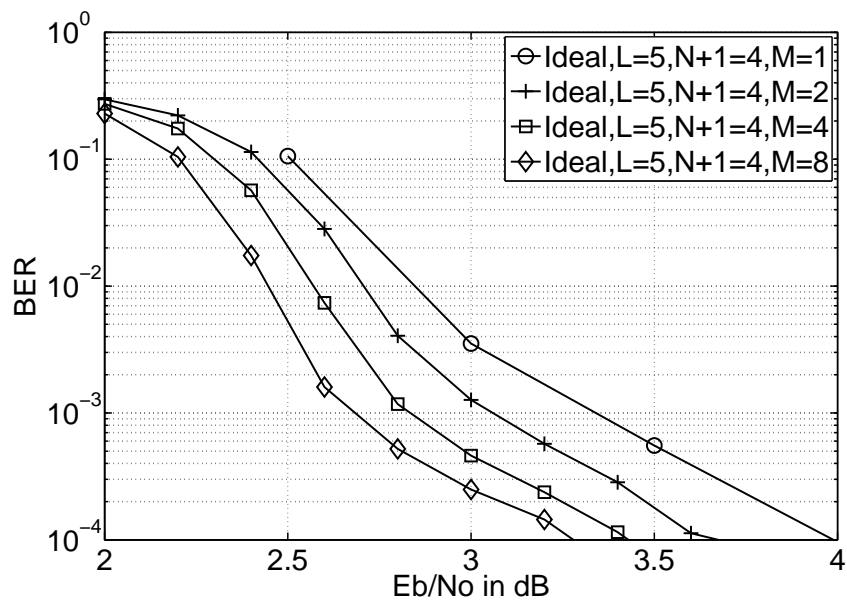


Figure 6.12: Bit-error performance for the multi-carrier case for different values M , where $N + 1 = 4$ and for $L = 5$ iterations.

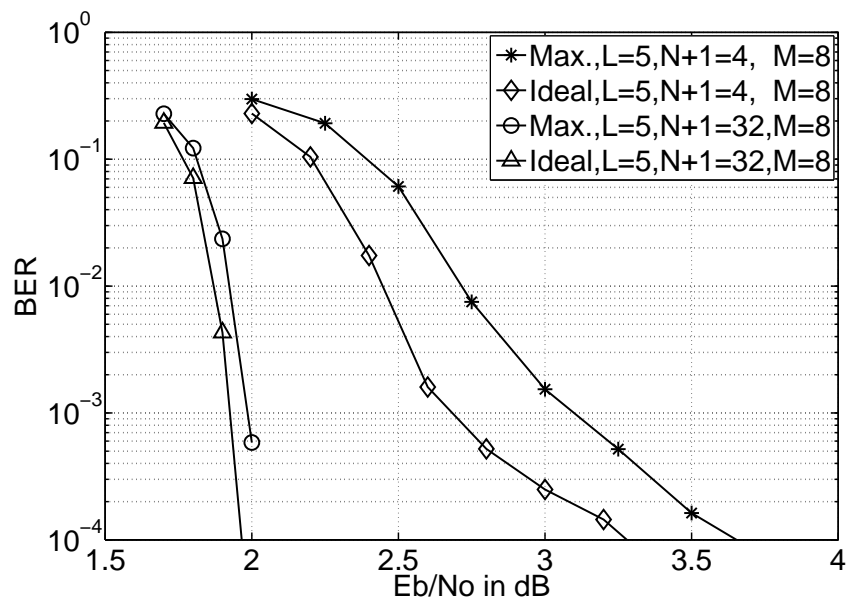


Figure 6.13: Bit-error performance with dominant sub-trellis approach for different values N , $M = 8$ and $L = 5$ iterations.

Hz, representing DAB transmission (in Band-III) movement-speeds between transmitter and receiver of ≈ 45 and ≈ 90 km/h, respectively.

We use our methods for DAB transmission in Mode-I, where the inverse subcarrier-spacing $T_u = 1$ ms and where the cyclic-prefix period $T_{cp} = 246 \mu\text{s}$ [25].

Now, with these settings, the *normalized Doppler rate* $f_d T_u$ are 0.01 and 0.02 respectively.

Note that to prevent ISI in an OFDM-scheme, the delay differences on separate propagation paths need to be less than the cyclic-prefix period [79], i.e., the channel impulse response length τ_m must satisfy $\tau_m \leq T_{cp}$. Within the DAB-system $T_{cp} \triangleq \frac{63}{256} T_u < \frac{T_u}{4}$ [25] and therefore the coherence-bandwidth $B_c \approx \frac{1}{\tau_m} > 4 \frac{1}{T_u}$, which is at least 4 OFDM-subcarriers. For Doppler frequency $f_d = 20$ Hz the coherence-time $T_c \approx \frac{1}{2f_d} = 25$ ms, which is ≈ 20 OFDM-symbols (including cyclic prefix).

The channel gain representative for a 2D-block, where it is assumed to be constant, is estimated similar to (8) in Chen et al. [18], i.e.,

$$|\widehat{h}|^2 = \max \left(\frac{1}{M(N+1)} \sum_{m=1, M, n=0, N} |y_{m,n}|^2 - 2\sigma^2, 0 \right). \quad (6.53)$$

The results of our simulations with the TU-6 model are shown in Fig. 6.14, where the solid-lines show the results for $f_d T_u = 0.01$ and the dashed-lines for $f_d T_u = 0.02$. We have results for $N+1 = 18$ with $M = 1$ and for $N+1 = 4$ with $M = 8$. We considered iterative procedures with $L = 5$ iterations. In our simulations we used the dominant sub-trellis approach, where we have chosen the dominant sub-trellis before starting the iterations.

The value for $N+1 = 4$ might be seen as a representative frame-size for services broadcasted by the DAB-family in transmission Mode-I. In this mode $N+1 = 18$ is the maximum possible number of interleaved OFDM symbols. Note that $N+1 = 18$ is close to the coherence-time of our TU-6 channel for a Doppler frequency of 20 Hz.

It can be concluded from Fig. 6.14 that for $N+1 = 18$ and $M = 1$, reliable transmission is not possible for the TU-6 channel with movement speeds of ≈ 45 km/h and ≈ 90 km/h. For the 2D-decomposition approach however with $N+1 = 4$ and $M = 8$ there is a considerable improvement of roughly 2.4 and 1.6 dB for 10 and 20 Hz, respectively, in required signal-to-noise ratio possible, compared to 2SDD.

6.8 Conclusions

We have investigated decoding procedures for DAB like systems, focussing on trellis decoding and iterative techniques, with a special focus on obtaining an advantage from considering 2D-blocks and trellis decomposition. These 2D-blocks consist of the intersection of a number of subsequent OFDM symbols and a number of adjacent subcarriers. The idea to focus on blocks was motivated by the fact that the channel coherence time is typically limited to a small number of OFDM symbols, but also since per service symbol processing is used which limits the number of OFDM symbols in a codeword.

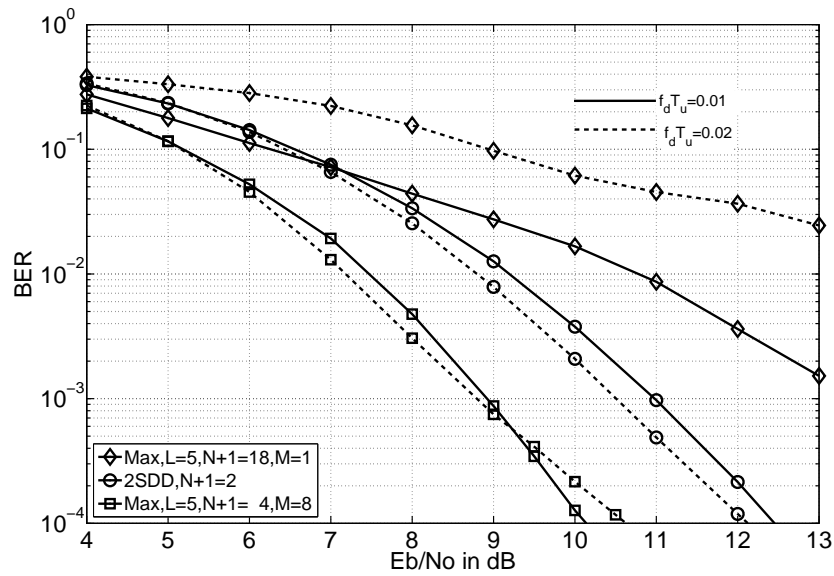


Figure 6.14: Bit-error performance for the TU-6 COST-207 channel with the solid-lines for $f_d T_u = 0.01$, the dashed-lines for $f_d T_u = 0.02$, $N + 1 = (4, 18)$, $M = (8, 1)$ and $L = 5$ iterations.

We have used trellis decomposition methods that allow us to estimate the unknown channel-phase modulo $\pi/2$. This channel phase relates to sub-trellises of which we can determine the a-posteriori probabilities. Using these probabilities we can weight the contributions of all the sub-trellises to compute the a-posteriori symbol probabilities. We can also use these probabilities to chose a dominant sub-trellis for providing us with these a-posteriori symbol probabilities. Working with dominant sub-trellises results in significant complexity reductions. A second important advantage of trellis-decomposition is that it allows us to process in an efficient way several sub-carriers simultaneously.

We have first investigated non-iterative methods. The advantage of these methods is that forward-backward procedures turned out to be extremely simple since we could use Colavolpe processing [19]. The drawback of these non-iterative methods is however that their gain, relative to the standard 2SDD technique, is modest. Iterative procedures result in a significantly larger gain however. In this context we must emphasize that part of this gain comes from the fact that we can do 2D-processing.

Simulations for the non-iterative AWGN case show that (a) trellis-lengths of $N + 1 \geq 32$ are required and (b) that 2D dominant sub-trellis processing with $M(N + 1) = 32$ outperforms 2SDD by 0.7 dB at a BER of 10^{-4} .

For the iterative AWGN case with $L = 5$ iterations, simulations show that 2D dominant sub-trellis processing with $M(N + 1) = 32$ where $N + 1 = 32$ and $M = 1$ outperforms 2SDD by 3.7 dB at a BER of 10^{-4} . However, simulations also reveal that with $M(N + 1) = 32$ where $N + 1 = 4$ and $M = 8$ the iterative coding gain is reduced to 2.5 dB, which is caused by the smaller value of $N + 1$.

On the other hand, iterative simulations for a practical setting (i.e., the TU-6 model) show that (a) with trellis-length $N + 1 = 18$ and $M = 1$ (one sub-carrier) no reliable communication is possible but that (b) with a modest trellis-length $N + 1 = 4$ and $M = 8$ sub-carriers, the iterative coding advantage is maintained and that the gain is roughly 2.4 dB for 10 Hz Doppler frequency, and 1.6 dB for 20 Hz.

Chapter 7

A Practical DAB System with 2D-Block-Based Iterative Decoding

In the previous chapter, Chapter 6, it was shown that 2D-block-based iterative decoding procedures for DAB receivers can improve the performance significantly. However it was also discussed in Chapter 6 that complexity reduction of the inner decoder is of crucial importance to realize such a DAB receiver. Therefore the focus of the current chapter is on the practical realization of a DAB receiver that operates with iterative decoding techniques¹.

7.1 Outline

In Section 7.2 complexity reduction of the inner decoder is investigated. This complexity reduction is realized by choosing, based on a-posteriori sub-trellis probabilities, in two different ways a dominant sub-trellis. In the first approach, a method is investigated that is based on finding, at the start of a new iteration, the dominant sub-trellis first and then do the forward-backward processing for demodulation only in this dominant sub-trellis. The second approach involves choosing the dominant sub-trellis only once, before starting with the iterations. In Section 7.3, an implementation of a MAP channel-phase estimator based on the second dominant sub-trellis approach is described. In addition, in Section 7.4, an implementation of a channel-gain estimator based on the received symbols within a 2D-block is discussed. Finally, in Section 7.5, a real-time and bit-true DAB-receiver is sketched. This DAB receiver operates according to the proposed 2D-block-based iterative decoding procedure within a dominant sub-trellis obtained by the

¹Section 7.2 has been submitted as *W.J. van Houtum and F.M.J. Willems, "Complexity Reduction for Non-Coherent Iteratively Detected DE-QPSK Based on Trellis-Decomposition", to ETT Eur. Trans. on Telecomm.*

second method. The performance improvements of this DAB receiver are evaluated for various numbers of iterations, block-sizes, and Doppler-frequencies.

7.2 Complexity reduction for non-coherent iteratively detected DE-QPSK based on trellis-decomposition

7.2.1 Abstract

In this section, we investigate complexity reduction for non-coherent iterative detection of DE-QPSK applied to DAB receivers. We use 2D blocks in an OFDM scheme and trellis decomposition to calculate, iteratively, the a-posteriori probabilities of the information symbols. The 2D-blocks are based on the time- and frequency dimension. By turning to 2D-blocks for iterative demodulation we solve a problem connected to the, in practise quite small, length of the trellises for each subcarrier. Furthermore, the trellis-decomposition method allows us to estimate the unknown channel phase efficiently. This phase is related to sub-trellises of which we can determine the a-posteriori probabilities. With these probabilities we are able to choose, in two different ways, a dominant sub-trellis, which results in a significant complexity reduction of the inner decoder. In our first approach, we investigate a method which is based on finding, at the start of a new iteration, the dominant sub-trellis first and then do the forward-backward processing for demodulation only in this dominant sub-trellis. This method reduces the number of multiplications with more than a factor of 3, the normalizations with a factor 8 and, introduces after five iterations ≈ 0.05 dB loss in performance on the COST-207 TU-6 channel. Our second approach involves choosing the dominant sub-trellis only once, before starting with the iterations. This results in a reduction of the number of multiplications and normalizations with more than a factor of 8 and introduces ≈ 0.5 dB loss in performance.

7.2.2 Introduction

We will investigate inner decoder complexity reduction of iterative decoding procedures for DAB systems, which are based on convolutional encoding, interleaving, and DE-QPSK [25]. Just as in [75], we consider DE-QPSK as inner coding method and convolutional encoding as outer coding method of a serially concatenated coding system. We were motivated by encouraging results for single-carrier transmissions on performance improvements and inner-decoder complexity reduction in [58, 20, 18], and [31]. To reduce the complexity of the inner decoder and accepting only a small performance loss (see, e.g., Peleg et al. [58] and Chen et al. [18]), we will focus on the techniques proposed by Peleg et al. in [58]. We also discretize the phase of the desired signal into several equispaced values, but do not allow “side-step” transitions to track small channel phase variations. However we will focus on 2D-blocks consisting of a number of subsequent OFDM symbols and a number of adjacent sub-carriers. Focussing on 2D-blocks was motivated by the fact that the channel coherence-time is typically limited to a small number of OFDM symbols, but also since DAB-transmissions use time-multiplexing of services, which limits the number of OFDM symbols in a codeword [25]. As was shown in [75],

extension in the subcarrier direction is then required to get reliable phase estimates. Using these phase-estimates we calculate, in an efficient way, the a-posteriori probabilities of the information symbols using forward-backward processing in full generality [4] where we consider 2D-blocks and trellis decomposition, see again [75]. Moreover, with the application of 2D-blocks for iterative demodulation we solve a problem connected to the, in practise quite small, length of the trellises for each subcarrier. Unfortunately, as was also shown in [75], using 2D-blocks and trellis decomposition in a straightforward way does not result in significant complexity reduction compared to the approach of Peleg and Shamai in [58]. However, in this contribution, we exploit the fact that the trellis-decomposition method does allow us to estimate the unknown channel-phase efficiently. This phase is related to sub-trellises of which we can determine the a-posteriori probabilities. With these probabilities we are able to choose, in two different ways, a dominant sub-trellis, which results in a significant complexity reduction of the inner decoder. In our first approach, we investigate a method that is based on finding, at the start of a new iteration, the dominant sub-trellis first and then do the forward-backward processing for demodulation only in this dominant sub-trellis. Our second approach chooses the dominant sub-trellis only once, before starting with the iterations. We will study for both methods their complexity and, by doing simulations, their performances.

7.2.3 Trellis-decomposition for DAB receivers

For our further analysis, just as in [75], we are interested in M aligned sequences of $N + 1$ subsequent OFDM symbols, i.e., 2D blocks of symbols, where we assume that in each OFDM sub-carrier $m = 1, 2, \dots, M$ a sequence $\mathbf{a}_m = (a_{m,1}, a_{m,2}, \dots, a_{m,N})$ is conveyed using differential encoding. For the components of the transmitted sequence $\mathbf{x}_m = (x_{m,0}, x_{m,1}, \dots, x_{m,N})$ we can write

$$x_{m,n} = a_{m,n}x_{m,n-1}, \quad (7.1)$$

where we assume that $x_{m,0}$ for each subcarrier $m = 1, 2, \dots, M$ is known and $x_{m,n} \in \mathcal{A} = \{e^{jp\pi/2}, p = 0, 1, 2, 3\}$. We assume that the channel phase is constant over the 2D block of symbols, therefore

$$y_{m,n} = e^{j\phi}x_{m,n} + w_{m,n}, \quad (7.2)$$

where the noise variables $w_{m,n}$ are circularly complex Gaussians with variance σ^2 per component. Note that a phase rotation proportional to m , due to a time-delay, is removed by linear phase correction [75]. Accepting a small performance loss we may assume that the channel-phase is discrete, and uniform over 32 levels which are uniformly spaced over $[0, 2\pi)$, hence

$$\Pr\{\phi = \pi l/16\} = 1/32, \text{ for } l = 0, 1, 2, \dots, 31. \quad (7.3)$$

The output sequence corresponding to OFDM sub-carrier m is denoted by $\mathbf{y}_m = (y_{m,0}, y_{m,1}, \dots, y_{m,N})$ and we define $z_n = x_{m,n}e^{j\phi}$ for $n = 0, 1, \dots, N$, then, since $x_{m,0}$ is chosen from \mathcal{A} for all m and ϕ is uniform over $\{\pi l/16, l = 0, 1, \dots, 31\}$ it follows that

$$\Pr\{z_0 = e^{jl\pi/16}\} = 1/32, \quad \text{for } l = 0, 1, \dots, 31, \quad (7.4)$$

and $z_n \in \mathcal{Z} \triangleq \{e^{jl\pi/16}, l = 0, 1, \dots, 31\}$. Moreover, for $n = 1, 2, \dots, N$

$$\begin{aligned} z_n &= a_{m,n}z_{n-1}, \text{ where} \\ \Pr\{a_{m,n} = e^{jp\pi/2}\} &= 1/4, \text{ for } p = 0, 1, 2, 3. \end{aligned} \quad (7.5)$$

The variables z_n for $n = 0, 1, \dots, N$ can now be regarded as states in a trellis \mathcal{T} and the symbols $a_{m,1}, a_{m,2}, \dots, a_{m,N}$ correspond to transitions between states for OFDM sub-carrier m . Just like Peleg et al. [58] we focus on the entire trellis \mathcal{T} . Note however that our trellis is different from that of [58], in which tracking of small channel phase variations is made possible by adding ‘‘side-step’’ transitions. We don’t have such transitions in our trellis and therefore our trellis can be decomposed in eight unconnected sub-trellises. Thus, as in [75], an important observation for our investigations is that the trellis \mathcal{T} can be seen to consist of eight sub-trellises $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_7$, that are *not connected to each other*, see also [18]. Moreover, sub-trellis \mathcal{T}_s consists of states $z_n \in \mathcal{Z}_s = \{e^{jl\pi/16}, l = s + 8p, p = 0, 1, 2, 3\}$, for $s = 0, 1, \dots, 7$. Note that for the likelihood $\gamma_{m,n}(z_n)$ corresponding to some state $z_n \in \mathcal{Z}_s$ for $n = 0, 1, \dots, N$ in a sub-trellis \mathcal{T}_s we can write

$$\gamma_{m,n}(z_n) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|y_{m,n} - z_n|^2}{2\sigma^2}\right). \quad (7.6)$$

In the next sub-section we take advantage of this decomposition.

7.2.4 A-posteriori symbol probabilities

Since we do processing based on trellis decomposition, we focus, as in [75], on the computation of the a-posteriori sub-trellis probabilities

$$\begin{aligned} \Pr\{s|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} & \\ &= \frac{\Pr\{s\}p(\mathbf{y}_1|s)p(\mathbf{y}_2|s) \dots p(\mathbf{y}_M|s)}{\sum_{s=0}^7 \Pr\{s\}p(\mathbf{y}_1|s)p(\mathbf{y}_2|s) \dots p(\mathbf{y}_M|s)}. \end{aligned} \quad (7.7)$$

Note that $\Pr\{s\} = 1/8$ for $s = 0, 1, \dots, 7$ and it is shown by (51) in [75] that the probabilities $p(\mathbf{y}_m|s)$ for each subcarrier $m = 1, 2, \dots, M$ can be obtained by performing a backward pass. Now, as was shown by (52) in [75], the a-posteriori symbol probability is a *convex combination* of the a-posteriori probabilities corresponding to the sub-trellises and the a-posteriori symbol probabilities of each sub-trellis $\Pr\{a_{m,n}|\mathbf{y}_m, s\}$, i.e.,

$$\begin{aligned} \Pr\{a_{m,n}|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} & \\ &= \sum_{s=0}^7 \Pr\{s|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \Pr\{a_{m,n}|\mathbf{y}_m, s\}, \end{aligned} \quad (7.8)$$

where $\Pr\{a_{m,n}|\mathbf{y}_m, s\}$ is computed as given by (48) in [75] for $s \in \{0, 1, \dots, 7\}$ and $a_{m,n} \in \mathcal{A}$. As stated in [75], from these a-posteriori probabilities we can compute, by omitting $\Pr\{a_{m,n}, s\}$, the extrinsic information that is needed by the convolutional decoder.

This suggests that, for each iteration, the demodulator first determines the a-posteriori sub-trellis probabilities using (7.7), for which first a backward pass in each of the M trellises corresponding to the sub-carriers is needed (requiring $8 \times 4 \times (4 + 1) = 160$ multiplications per section). Then we find the a-posteriori symbol probabilities $\Pr\{a_{m,n} | \mathbf{y}_m, s\}$ in the standard way, taking into account that the backward passes were already carried out. Thus, for all sub-trellises \mathcal{T}_s we do a forward pass (requiring $8 \times 4 \times 4 \times 2 = 256$ multiplications, see (42) in [75]) and then combine the results for that sub-trellis (requiring $8 \times 4 \times 4 \times 2 = 256$ multiplications and $8 \times 4 = 32$ normalizations per section). Finally these probabilities have to be weighted as in (7.8) (requiring $8 \times 4 = 32$ multiplications). In total this results in 704 multiplications and 32 normalizations per iteration. We suggest to focus only on multiplications and normalizations, since additions have a smaller complexity than multiplications and normalizations².

Trellis-decomposition requires the forward-backward processing in full generality [4] and does not result in significant complexity reduction compared to the approach of Peleg and Shamai in [58]. However, in the next two sub-sections we will introduce two methods based on choosing a *dominant sub-trellis* resulting, for both methods, in relevant complexity reduction.

First method

Finding the dominant sub-trellis, at the start of a new iteration, is based on the a-posteriori sub-trellis probabilities $\Pr\{s | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\}$ obtained from the backward passes. Assuming that one of the a-posteriori sub-trellis probabilities dominates the other ones we can write

$$\Pr\{a_{m,n} | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \approx \Pr\{a_{m,n} | \mathbf{y}_m, \hat{s}\}, \quad (7.9)$$

with

$$\hat{s} = \arg \max_s \Pr\{s | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\}. \quad (7.10)$$

Now, for the best sub-trellis $\mathcal{T}_{\hat{s}}$, we do a forward pass (requiring $4 \times 4 \times 2 = 32$ multiplications per section) and combine the results to obtain the a-posteriori (actually extrinsic) symbol probabilities $\Pr\{a_{m,n} | \mathbf{y}_m, \hat{s}\}$ for that sub-trellis (requiring $4 \times 4 \times 2 = 32$ multiplications and 4 normalizations per section). In total we now need 224 multiplications and 4 normalizations per section per iteration, which is roughly 32% of the complexity of the weighting approach given by (7.8). In Section 7.2.4 we will evaluate, by doing simulations, the performances of this first method. Moreover, in the next section we will introduce a second method, which reduces the complexity even further.

Second method

A second approach involves choosing the dominant sub-trellis only once, before starting with the iterations. Since before starting the iterations the *a-priori* probabilities are all

²In the log-domain, multiplications and normalizations are replaced by additions, and additions are typically approximated by maximizations, but for reasons of simplicity we neglect the additions here.

equal, computing the a-posteriori sub-trellis probabilities for each sub-trellis is straightforward. It is shown in Section 3 in [75], that for computing the a-posteriori sub-trellis probabilities the BCJR-algorithm [4] is not required. Now, as with our first method, we do the iterations only in the sub-trellis that was chosen initially, requiring for the forward pass $4 \times 4 \times 2 = 32$, for the backward pass $4 \times (4 + 1) = 20$, and for the combining-pass $4 \times 4 \times 2 = 32$ multiplications plus 4 normalizations per section. Resulting in 84 multiplications and 4 normalizations per section per iteration, which is roughly 12% of the complexity of doing the weighted approach given by (7.8). Next, we will demonstrate, by simulations, the performances of both dominant sub-trellis techniques.

Simulations

To investigate the performance we have used the TU-6 channel model defined in [1], which is commonly applied to test DAB transmission. A maximum Doppler frequency of $f_d = 20$ Hz is chosen, representing a DAB transmission (in Band-III) movement-speed between transmitter and receiver of ≈ 90 km/h. The channel gain representative for a 2D-block, in which it is assumed to be constant, is estimated similar to (8) in Chen et al. [18].

We simulated, both for a perfectly known channel gain $|h|$ as well as for the estimated channel gain $|\widehat{h}|$, the weighting approach given by (7.8) for $L = 1$ and $L = 5$ iterations ($L = 1$ stands for no iterations), the first and second dominant sub-trellis method conform (7.9) for $L = 5$, and determined the BER versus the signal-to-noise ratio $E_b/N_0 = \frac{1}{2\sigma^2}$. The BER performance with a perfectly known channel gain $|h|$ is shown in Fig. 7.1 and with an estimated channel gain $|\widehat{h}|$ in Fig. 7.2.

From Fig. 7.1 can be seen that, after five iterations, our first and second method introduces ≈ 0.05 dB and ≈ 0.5 dB loss, respectively, relative to the weighting method.

Fig. 7.2 shows that with an estimated channel gain $|\widehat{h}|$ the performance of our first method is practically identical to that of the weighting method and that our second method introduces ≈ 0.6 dB loss in performance at a BER = 10^{-4} .

7.2.5 Conclusions

We have investigated trellis decoding and iterative techniques for DAB systems with the objective to reduce receiver complexity by applying 2D-block-based processing and trellis decomposition. These 2D-blocks consist of the intersection of a number of subsequent OFDM symbols and a number of adjacent subcarriers. The idea to focus on blocks was motivated by the fact that the channel coherence time is typically limited to a small number of OFDM symbols, but also since per service symbol processing is used which limits the number of OFDM symbols in a codeword. Moreover, by turning to 2D-blocks for iterative demodulation we solve a problem connected to the, in practise quite small, length of the trellises for each subcarrier. Trellis decomposition methods allowed us to estimate the unknown channel-phase since it relates to the sub-trellises of which we can determine the a-posteriori probabilities. Using these probabilities we can weight the contributions of all the sub-trellises to compute the a-posteriori symbol probabilities. We can also use

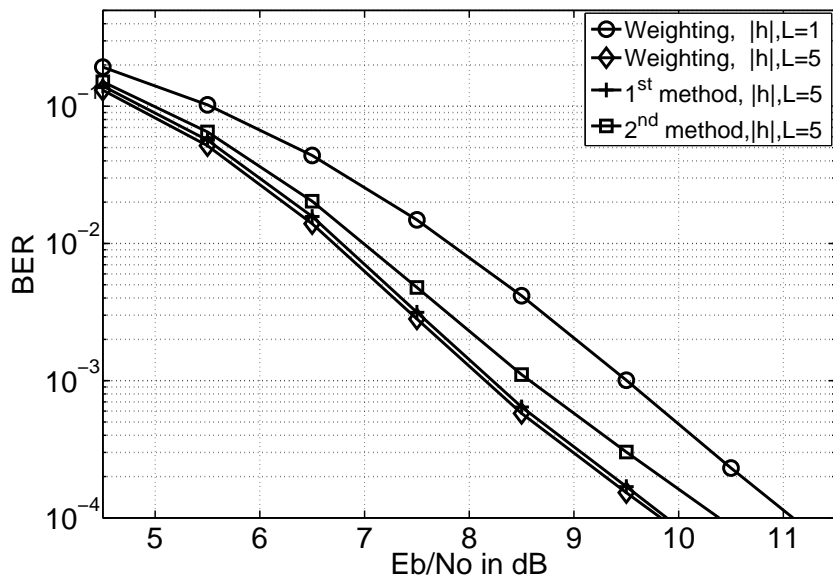


Figure 7.1: BER-performance for the TU-6 COST-207 channel with a perfectly known channel gain $|h|$ for the weighting, the 1st and, the 2nd method.

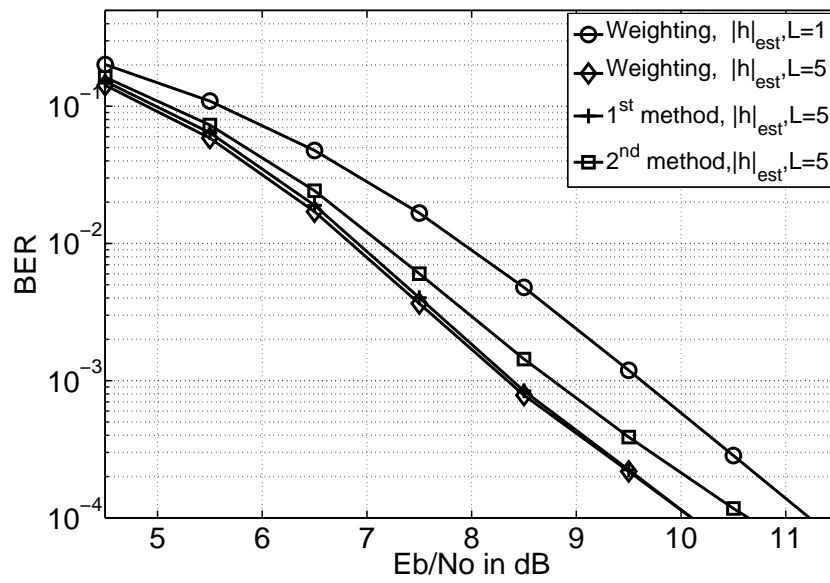


Figure 7.2: BER-performance for the TU-6 COST-207 channel with an estimated channel gain $|\hat{h}|$ for the weighting, the 1st and, the 2nd method.

these probabilities to chose a dominant sub-trellis for providing us with these a-posteriori symbol probabilities. Moreover, we showed that dominant sub-trellis approaches result in a significant complexity reduction of the inner decoder. We investigated two dominant sub-trellis approaches. Our first approach was based on finding, at the start of a new iteration, the dominant sub-trellis. Our second approach chooses the dominant sub-trellis only once, before starting with the iterations. The first method reduces the number of multiplications with more than a factor of 3, the normalizations with a factor 8 and introduces, after five iterations, ≈ 0.05 dB loss in performance on the COST-207 TU-6 channel. The second method results in reducing the number of multiplications and normalizations with more than a factor of 8 and results, after five iterations, ≈ 0.5 dB loss in performance.

7.3 MAP channel-phase estimator

7.3.1 Introduction

In this section we introduce a MAP channel-phase estimator, which operates according to the second dominant sub-trellis method of Section 7.2. This MAP channel-phase estimator determines the estimated channel phase $\hat{\phi} \in \{s\pi/16, s = 0, 1, \dots, 7\}$. Note that similar techniques of phase quantization were discussed by Peleg et al. in [58], and by Chen et al. in [18]. For our analysis of the phase estimator we assume that the channel gain $|h|$ is perfectly known at the receiver. Note that in the next section we will discuss a channel-gain estimator. We denote by \mathbf{x}_m the sequence $(x_{m,0}, x_{m,1}, \dots, x_{m,N})$ of transmitted DE-QPSK symbols and by \mathbf{y}_m the sequence $(y_{m,0}, y_{m,1}, \dots, y_{m,N})$ of received DE-QPSK symbols corresponding to sub-carrier m . We compute the dominant sub-trellis only before the start of the iteration process. In this case the information symbols can be assumed to be iud, i.e., $\Pr\{a_{m,n} = \frac{1}{4}\}$, and we use the trellis-decomposition analysis for the non-iterative multi-carrier setting discussed in Section 6.4. Hence we write for (6.37)

$$\begin{aligned} \Pr\{s|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M, |h|\} &= \frac{\frac{1}{8} \prod_{m=1}^M \prod_{n=0}^N K_{m,s}(n, |h|)}{\sum_{s=0}^7 \frac{1}{8} \prod_{m=1}^M \prod_{n=0}^N K_{m,s}(n, |h|)} \\ &\propto \prod_{m=1, M, n=0, N} K_{m,s}(n, |h|), \end{aligned} \quad (7.11)$$

where $K_{m,s}(n, |h|) \triangleq \sum_{z_n \in \mathcal{Z}_s} \frac{1}{4} \gamma_{m,n}(z_n, |h|)$ for $n = 0, 1, \dots, N$ and

$$\gamma_{m,n}(z_n, |h|) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|y_{m,n} - |h|z_n|^2}{2\sigma^2}\right). \quad (7.12)$$

Now an estimate of the channel phase is obtained with the MAP decision rule

$$\begin{aligned} \hat{\phi} &= \frac{\pi}{16} \arg \max_s \Pr\{s|\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M, |h|\} \\ &= \frac{\pi}{16} \arg \max_s \left(\prod_{m=1, M, n=0, N} K_{m,s}(n, |h|) \right). \end{aligned} \quad (7.13)$$

Hence with this MAP decision rule we select the dominant sub-trellis corresponding to the second method discussed in Section 7.2.

7.3.2 Complexity reduction

The a-posteriori sub-trellis probability given by (7.11) is proportional to the product of $M(N + 1)$ factors $K_{m,s}(n, |h|)$. We will first show that $K_{m,s}(n, |h|)$ is the sum of four exponentials. To avoid exponential and logarithmic operations we could select one dominant exponential instead of computing the entire sum [15]. To be able to do so we write, using $\phi_s \triangleq \frac{s\pi}{16}$

$$\begin{aligned}
 K_{m,s}(n, |h|) &= \sum_{x_n \in \mathcal{A}} \frac{1}{4} \gamma_{m,n}(x_n e^{j\phi_s}, |h|) \\
 &\propto \sum_{x_n \in \mathcal{A}} \exp \left\{ -\frac{|y_{m,n} - |h|x_n e^{j\phi_s}|^2}{2\sigma^2} \right\} \\
 &\propto \exp \left(\frac{\Re(q_{m,n}^s)}{\sigma^2} \right) + \exp \left(\frac{\Im(q_{m,n}^s)}{\sigma^2} \right) \\
 &\quad + \exp \left(-\frac{\Re(q_{m,n}^s)}{\sigma^2} \right) + \exp \left(-\frac{\Im(q_{m,n}^s)}{\sigma^2} \right) \\
 &\propto 2 \cosh(R_{m,n}^s) + 2 \cosh(I_{m,n}^s), \tag{7.14}
 \end{aligned}$$

where

$$\begin{aligned}
 q_{m,n}^s &\triangleq |h| y_{m,n} e^{-j\phi_s}, \\
 R_{m,n}^s &\triangleq \frac{\Re(q_{m,n}^s)}{\sigma^2}, \\
 I_{m,n}^s &\triangleq \frac{\Im(q_{m,n}^s)}{\sigma^2}.
 \end{aligned}$$

Now the a-posteriori probability of the channel phase becomes

$$\Pr\{s | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M, |h|\} \propto \prod_{m=1, M, n=0, N} [2 \cosh(R_{m,n}^s) + 2 \cosh(I_{m,n}^s)]. \tag{7.15}$$

From (7.15) we conclude that the a-posteriori probability of the channel phase is the product of sums of two cosh-functions. To avoid underflow, calculations are often carried out in the logarithmic domain. Instead of (7.14) the metric

$$\mu_0 \triangleq \ln(2 \cosh(R) + 2 \cosh(I)), \tag{7.16}$$

is applied. Note that for ease of notation we have dropped the indices m, n and s . As an approximation for μ_0 , similar to (4.11) in Section 4.2.1, we apply the max-log-MAP approximation [15]

$$\mu_1 \triangleq \ln \left(\max \left(e^{|R|}, e^{|I|} \right) \right) = \max(|R|, |I|). \tag{7.17}$$

Note that the max-log-MAP approximation is based on selecting dominant exponentials, similar to, e.g., Bottomley's [15] approximation. Now the MAP decision rule given by (7.13) becomes

$$\begin{aligned}\hat{\phi} &= \frac{\pi}{16} \arg \max_s \left(\ln \left(\prod_{m=1, M, n=0, N} [2 \cosh(R_{m,n}^s) + 2 \cosh(I_{m,n}^s)] \right) \right) \\ &= \frac{\pi}{16} \arg \max_s \left(\sum_{m=1, M, n=0, N} \max(|R_{m,n}^s|, |I_{m,n}^s|) \right).\end{aligned}\quad (7.18)$$

Motivated by the improved results obtained with the piecewise-linear fitting of the logarithm of the hyperbolic cosine, see (4.15) in Section 4.2.2, we also introduce here the approximation μ_2 of μ_0

$$\mu_2 \triangleq f\left(\frac{R+I}{2}\right) + f\left(\frac{R-I}{2}\right) + 2 \ln 2, \quad (7.19)$$

where we used the same identity as given by (4.16) and approximated the $\ln(\cosh(g))$ by the piecewise-linear function as given by (4.17) in Section 4.2.2.

7.3.3 Simulations

We have performed BER simulations on the AWGN-channel to investigate the proposed MAP channel-phase estimator and the corresponding approximations. The simulations show the BER versus the signal-to-noise ratio $E_b/N_0 = \frac{1}{2\sigma^2}$, where E_b is the received signal energy of an information bit. We used bit-interleaving and Viterbi-decoding. The soft-decision bit metrics are conform (3.27). The de facto-standard $R_c = 1/2$ RCPC code, as described in Section 2.3.2, is used for the encoding. Figure 7.3 shows simulation results for the case where we estimate the channel phase with the MAP decision rule (7.13) and metric μ_0 given by (7.16). Figure 7.3 also shows simulation results if we replace metric μ_0 with the approximated metric μ_1 given by (7.17) or the improved approximation μ_2 given by (7.19). Moreover, the simulation results with a perfectly known channel phase ϕ are shown for reference purposes. From Figure 7.3 can be seen that the metrics μ_0 , μ_1 , and μ_2 achieve similar performances in combination with the MAP-decision rule given by (7.13). Figure 7.4 shows that the three channel-phase estimates result in a slight performance degradation relative to the case where the channel phase ϕ is perfectly known. Note that if we use the max-log-MAP approximation μ_1 we do not require knowledge of the noise variance and the channel gain. We would like to conclude that the max-log-MAP approximation is an appropriate method in terms of performance and complexity trade-off to perform channel-phase estimation.

Since a receiver will, in practise, not have a perfect knowledge of the channel gain $|h|$ we introduce in the next section a channel-gain estimator.

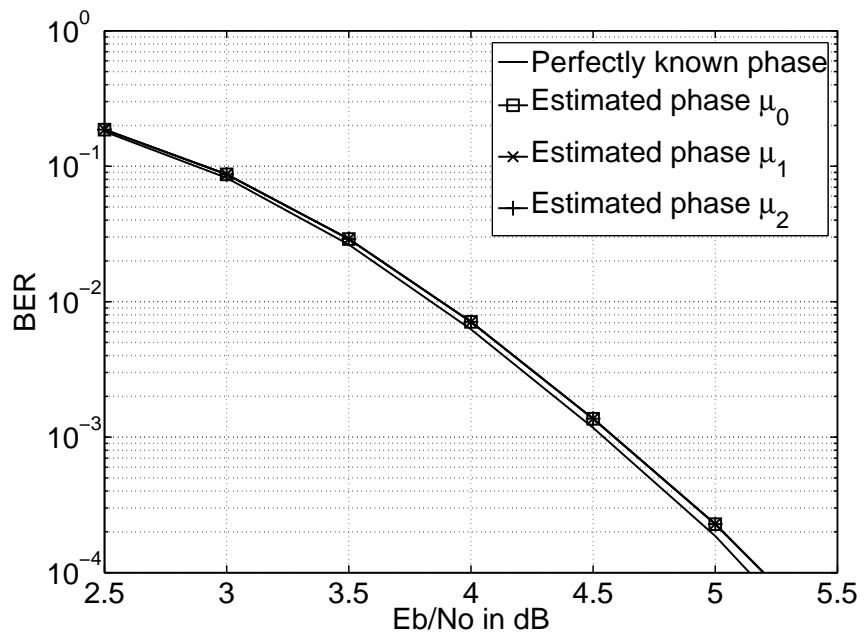


Figure 7.3: BER-performance with perfect knowledge and estimates of the channel phase.

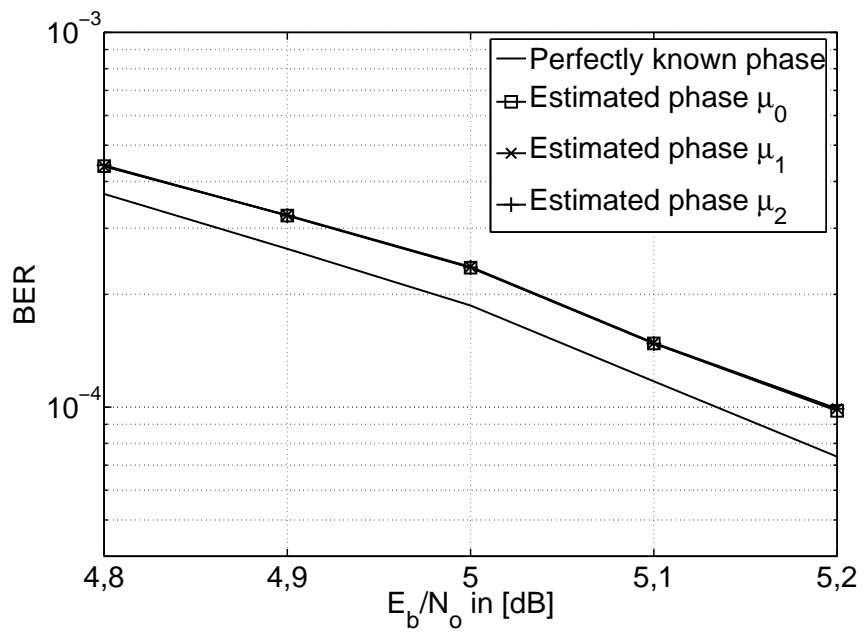


Figure 7.4: Detail of the BER-performance with perfect knowledge and estimates of the channel phase.

7.4 Channel-gain estimator

7.4.1 Introduction

For the AWGN channels that we have used in our analysis and simulations it was assumed that the channel phase was unknown to the receiver and the channel gain was fixed and equal to one. To study the performance of DAB transmissions over channels where the channel gain is not unity the channel gain $|h|$ or an estimate of the channel gain $|\widehat{h}|$ is assumed to be available at the receiver. Therefore we discuss in the sequel of this section a simple channel-gain estimator based on averaging. A similar estimator was proposed by Chen et al. in [18] and showed good performances. The channel-gain estimator is based on the received power, i.e., the expectation of the squared magnitude of the received DE-QPSK symbols within a 2D-block. We assume that the channel gain is constant within a 2D-block, and therefore the received signal for $m = 1, 2, \dots, M$ and $n = 0, 1, \dots, N$ is

$$y_{m,n} = |h|x_{m,n}e^{j\phi} + w_{m,n}, \quad (7.20)$$

where $w_{m,n}$ is circularly symmetric mean zero complex Gaussian noise with variance σ^2 in both the real and imaginary parts. The expectation of the channel gain with $|x_{m,n}|^2 = 1$ yields

$$\begin{aligned} \mathbb{E}\{|y_{m,n}|^2\} &= \mathbb{E}\{(|h|x_{m,n}e^{j\phi} + w_{m,n})(|h|x_{m,n}^*e^{-j\phi} + w_{m,n}^*)\} \\ &= \mathbb{E}\{|h|^2|x_{m,n}|^2 + 2|h|\Re\{x_{m,n}^*e^{-j\phi}w_{m,n}\} + |w_{m,n}|^2\} \\ &= \mathbb{E}\{|h|^2 + 2|h|\Re\{x_{m,n}^*e^{-j\phi}\}\Re\{w_{m,n}\} - \\ &\quad 2|h|\Im\{x_{m,n}^*e^{-j\phi}\}\Im\{w_{m,n}\} + |w_{m,n}|^2\}. \end{aligned} \quad (7.21)$$

Since $w_{m,n}$ is circularly symmetric zero-mean complex Gaussian noise with variance σ^2 per component, (7.21) becomes

$$\begin{aligned} \mathbb{E}\{|y_{m,n}|^2\} &= \mathbb{E}\{|h|^2\} + 2|h|\mathbb{E}\{\Re\{x_{m,n}^*e^{-j\phi}\}\}\mathbb{E}\{\Re\{w_{m,n}\}\} - \\ &\quad 2|h|\mathbb{E}\{\Im\{x_{m,n}^*e^{-j\phi}\}\}\mathbb{E}\{\Im\{w_{m,n}\}\} + \mathbb{E}\{|w_{m,n}|^2\} \\ &= |h|^2 + 2\sigma^2. \end{aligned} \quad (7.22)$$

Rewriting (7.22) yields

$$|h|^2 = \mathbb{E}\{|y_{m,n}|^2\} - 2\sigma^2. \quad (7.23)$$

Since we assume that the channel coherence-time and coherence-bandwidth span a limited number of OFDM-symbols and a limited number of OFDM-subcarriers the channel gain $|h|$ is only constant over a limit number of received symbols. Therefore we propose in the next section an approximation based on a finite number of received symbols. Moreover, (7.23) suggests that the channel-gain estimator requires knowledge of the noise variance σ^2 . For complexity reduction reasons we also show in the next section the performance of the channel-gain estimator when the noise-variance term is ignored.

7.4.2 Complexity reduction

Since we are interested in a practical realization of the channel-gain estimator we compute the expectation with a finite number of $M(N + 1)$ received symbols. Note that a similar approach was followed by Chen et al. in [18] for the one-dimensional case. We now get

$$\widehat{|h|_1^2} \triangleq \max \left(\frac{1}{M(N+1)} \sum_{m=1, M, n=0, N} |y_{m,n}|^2 - 2\sigma^2, 0 \right). \quad (7.24)$$

This first estimator is similar to (6.53). Note, maximizing in (7.24) prevents the estimator to become negative.

To avoid that the receiver needs to know the noise variance we propose a second estimator where the noise variance term is ignored. This leads to

$$\widehat{|h|_2^2} \triangleq \frac{1}{M(N+1)} \sum_{m=1, M, n=0, N} |y_{m,n}|^2. \quad (7.25)$$

7.4.3 Simulations

We performed BER simulations on the practical TU-6 channel [1] to evaluate the proposed channel-gain estimators. The simulations show the BER versus the signal-to-noise ratio $E_b/N_0 = \frac{1}{2\sigma^2}$, where E_b is the received signal energy of an information bit. We used bit-interleaving and Viterbi-decoding. The soft-decision bit metrics conform (3.31) where $1/\sigma^2$ was replaced by $|h|/\sigma^2$. The de facto-standard $R_c = 1/2$ RCPC code, as described in Section 2.3.2, is used for the encoding. Figure 7.5 shows the performance of both channel-gain estimators. Moreover, in Figure 7.5 the perfectly known channel gain is shown as a reference. Figure 7.6 shows that both estimators achieve similar results for several values of $M(N+1)$ around $E_b/N_0 = 12$ dB where the BER $\approx 10^{-4}$. For $M(N+1) \geq 8$, the estimators given by (7.24) and (7.25) are slightly worse than the reference. Note that the estimator given by (7.24) requires knowledge of the noise variance σ^2 .

We would like to conclude that the second estimator given by (7.25) is appropriate for channel gain estimation in terms of performance and complexity trade-off.

7.5 A realization of the proposed DAB receiver

7.5.1 Block-diagram

Figure 7.7 shows a block-diagram of the proposed DAB receiver. This DAB receiver performs decoding of 2D-blocks of received symbols and decomposes the a-posteriori symbol probabilities according to (7.8). For jointly demodulation and decoding we applied the iterative decoding procedures with SCCC and trellis decomposition discussed in Section 6.6. The channel-phase estimator discussed in Section 7.3 and the channel-gain estimator discussed in Section 7.4 are also applied to this receiver. In the next section we present real-time bench-test results of the proposed DAB receiver.

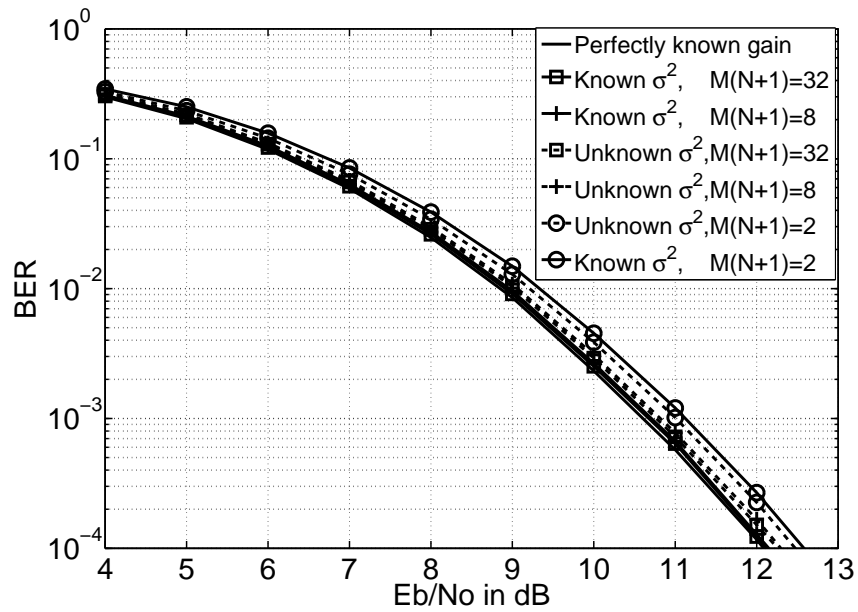


Figure 7.5: BER-performance with perfect knowledge and estimates of the channel gain for different values of $M(N + 1)$.

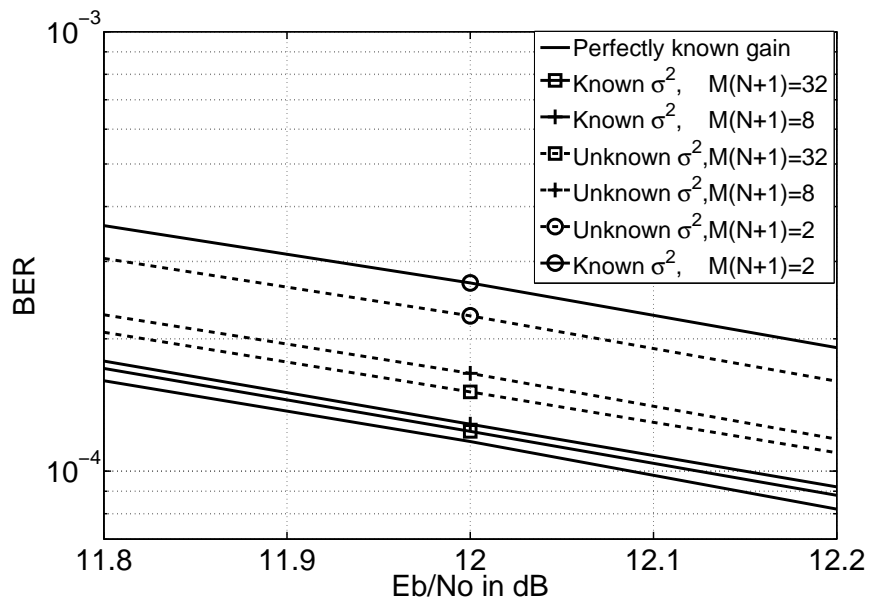


Figure 7.6: Detail of BER-performance with perfect knowledge and estimates of the channel gain for different values of $M(N+1)$.

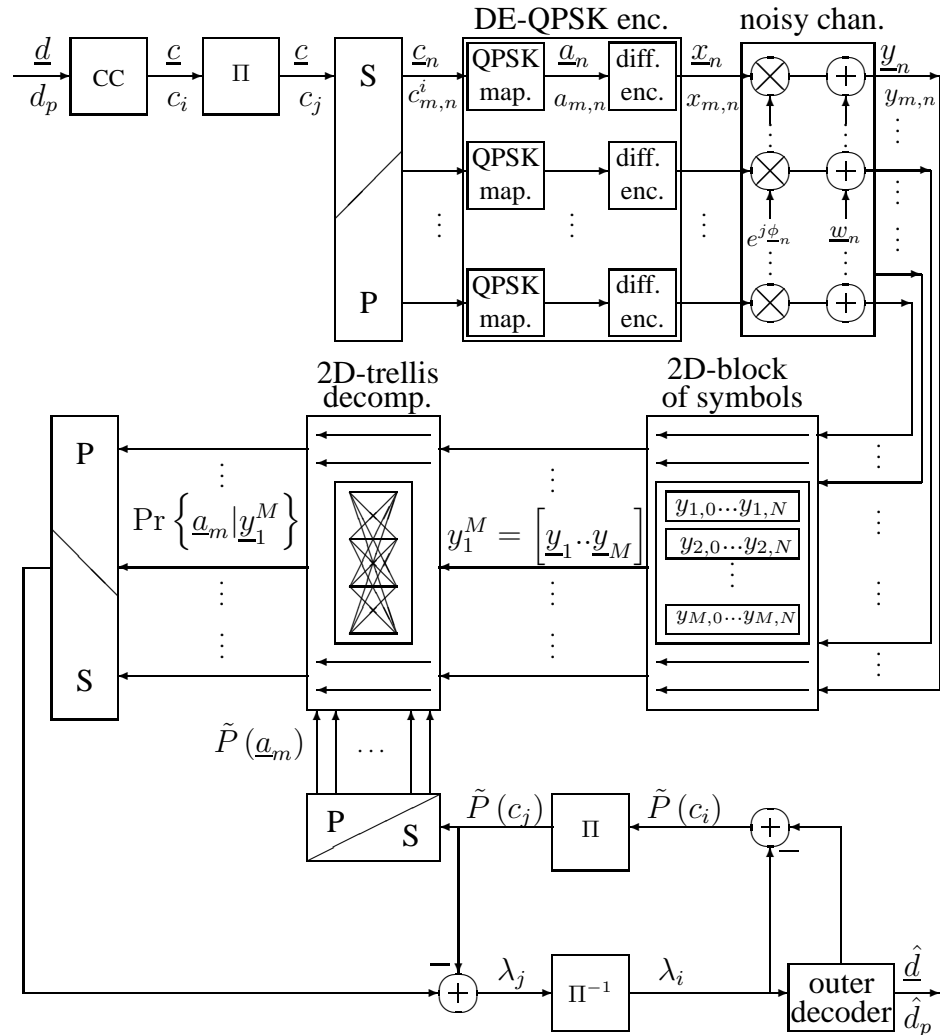


Figure 7.7: Block-diagram of the proposed DAB receiver with 2D-block-based joint and iterative demodulation and decoding.

7.5.2 Bench-test results of the proposed DAB receiver

In this section we give bench-test results of a real-time and bit-true version of the proposed DAB receiver running on a PC. We demonstrate reception improvements with this DAB receiver for the AWGN-channel and the TU-6 channel. These performance improvements are shown for several numbers of iterations, block-sizes, and Doppler-frequencies.

Real-time bench-test stimuli

For the real-time bench-tests a DAB stream with different services was created, see [25, Table 6, p.50]. The sample frequency of the generated input file for the real-time PC-application *DAB-PC* is 8.192 MSamples/s at a low-IF frequency of 2.048 MHz. The duration of the real-time bench-test file is ≈ 600 seconds to avoid "wrap-around" of the measurement data for the AWGN tests and the TU-6 tests. A BER-measurement per *SNR*-value is based on 144 or 432 seconds of real-time bench-test stimuli for the AWGN, respectively, the TU-6 channel. We will show the performance improvement of the proposed DAB receiver in dB, i.e., the decrease in required signal-to-noise ratio compared to 2SDD to obtain a $\text{BER} \approx 10^{-4}$.

AWGN channel bench-tests

The BER is measured for a duration of 144 seconds, which corresponds to $\frac{144}{96 \cdot 10^{-3}} = 1500$ DAB transmission frames in Mode-I. With a bit-rate of, for example, 128 kbit/s there will be on the average $144 \times 128 \cdot 10^3 \times 10^{-4} \approx 1843$ errors for a $\text{BER} \approx 10^{-4}$. Fig. 7.8 shows the bench-test results for the AWGN channel as function of the number of iterations. Fig. 7.8 shows a performance improvement of ≈ 2.6 dB for $L = 4$ iterations ($L = 1$ stands for no iterations) and with $M(N + 1) = 80$. Fig. 7.8 also shows that the performance is more or less determined by the size $M(N + 1)$ of the block just as we have seen in Section 6.6.3. It can be seen that a larger 2D-block size results in a higher gain. Furthermore, from Fig. 7.8 can also be seen that if the trellis-length $N + 1$ is quite small the effect of iterating is limited. This was also shown by Fig. 6.12 in Section 6.6.3.

TU-6 channel bench-tests

The BER is measured for a duration of 432 seconds, which corresponds to $\frac{432}{96 \cdot 10^{-3}} = 4500$ DAB transmission frames in Mode-I. Fig. 7.9 shows the performance improvements as a function of the movement speed in [km/h] between transmitter and receiver. The speed in [km/h] between transmitter and receiver is determined for a DAB-carrier frequency of $f_c = 240$ MHz in Band-III, i.e.,

$$V = 3.6 \left(\frac{c}{f_c} \right) f_d = 4.5 \cdot f_d, \quad (7.26)$$

where $c \triangleq 3 \cdot 10^8$ m/s is the speed of light. Fig. 7.9 shows performance improvements up to ≈ 2.4 dB and ≈ 1.7 dB for speeds of 45 km/h, respectively, 90 km/h. Also here we see that the gains are in principle determined by the size $M(N + 1)$. However, compared

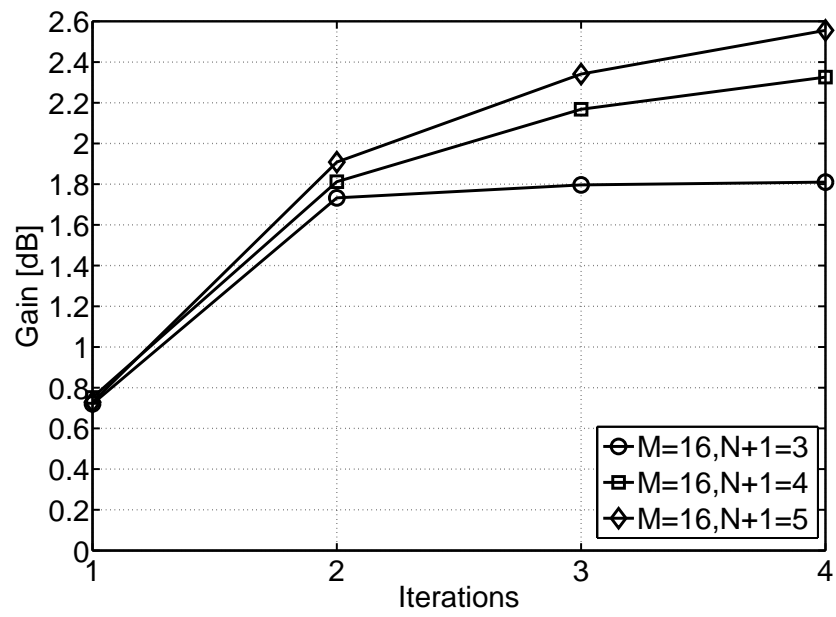


Figure 7.8: Real-time bench-test results of the performance improvement for DAB-transmissions in Mode-I on the AWGN channel.

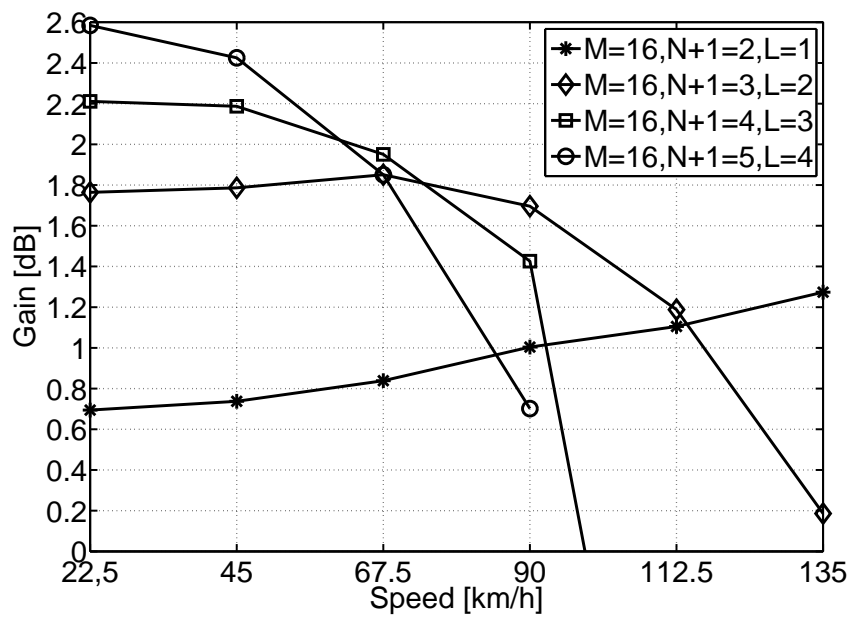


Figure 7.9: Real-time bench-test results of the iterative decoding gain for DAB-transmissions in Mode-I on the TU-6 channel.

to the AWGN case shown in Fig. 7.8, the situation is slightly different here. Increasing $N + 1$ (with a fixed M) has a positive effect on the performance, however if the speed is increasing, the coherence-time of the channel decreases, and the effect of large $N + 1$ is limited. Note that keeping $N + 1$ small has also advantages related to service symbol processing [25]. Finally, for the non-iterative case, i.e., $L = 1$ the improved DAB receiver shows at higher speeds better performances than 2SDD.

7.6 Conclusions

We investigated two dominant sub-trellis approaches to reduce the complexity of the inner decoder. Our first approach was based on finding, at the start of a new iteration, the dominant sub-trellis. Our second approach chooses the dominant sub-trellis only once, before starting with the iterations. The first dominant sub-trellis method reduces the number of multiplications with more than a factor of 3, the normalizations with a factor 8 and introduces, after five iterations, only ≈ 0.05 dB loss in performance on the COST-207 TU-6 channel. The second method results in reducing the number of multiplications and normalizations with more than a factor of 8 and results, after five iterations, ≈ 0.5 dB loss in performance.

We showed that a MAP channel-phase estimator with a max-log-MAP approximation is an appropriate method in terms of performance and complexity trade-off to perform channel phase estimation.

We observed that a channel-gain estimator based on averaging the squared magnitude of the received symbols combines simplicity with a good performance. A modified version of such an estimator that ignores the noise variance term has an acceptable performance without having knowledge of the noise variance at the receiver.

The real-time bench-tests of the proposed DAB receiver demonstrate that the performance is more or less determined by the size $M(N + 1)$ of the 2D-block. Moreover, it was shown that if the trellis-length $N + 1$ is quite small the effect of iterating is limited. Simulations for the AWGN channel show a performance improvement of ≈ 2.6 dB with $L = 4$ iterations and $M(N + 1) = 80$ for our DAB receiver. In addition, TU-6 channel simulations show performance improvements up to ≈ 2.4 dB and ≈ 1.7 dB for speeds of 45 km/h, respectively, 90 km/h. The TU-6 channel simulations also show that increasing $N + 1$ (with a fixed M) has a positive effect on the performance, however if the speed increases, the coherence-time of the channel decreases, the effect of large $N + 1$ is limited. Furthermore, with no iterations the proposed DAB receiver shows at higher speeds better performances than 2SDD.

Chapter 8

Conclusions and Recommendations

In this final chapter we present the conclusions of the investigations we carried out on two-dimensional block-based reception for differentially encoded OFDM systems. We also give some recommendations for further research.

8.1 Outline

Commonly used DAB receivers perform non-coherent 2SDD with soft-decision Viterbi decoding. It is well-known that 2SDD can be improved if the detection is based on more than two received symbols as, e.g., in non-coherent MSDD. For improving the performance of the demodulation procedures of DAB-like streams, demodulation based on 2D blocks of received symbols with a decomposed demodulation trellis is therefore proposed in this thesis.

Peleg and Shamai [58] demonstrated that iterative techniques could increase the performance of the demodulation procedures of DE-QPSK streams even further. However, they investigated demodulation procedures of DE-QPSK streams for single-carrier transmissions, i.e., the one-dimensional case. These demodulation procedures were based on complex trellises with large lengths and “side-steps” to combat phase noise. In this thesis, their approach is generalized to the 2D setting where again the decomposed demodulation trellis and, to combat phase noise, the 2D blocks of received symbols are used. In this way a problem connected to the small lengths of the trellises for each subcarrier is solved. The application of these iterative decoding techniques in DAB receivers is only feasible if their complexity can be drastically reduced. A significant complexity reduction is obtained by iterating only in a dominant sub-trellis of the decomposed demodulation trellis. In this way, a real-time and bit-true DAB receiver based on iterative decoding techniques is realized.

In the next section we will give our conclusions on iterative demodulation of DE-QPSK based on 2D blocks of received symbols with a decomposed demodulation trellis. In

Section 8.3 we give some recommendations for further work.

8.2 Conclusions

8.2.1 Introductory chapters

In Chapter 2, we described the basic elements of a DAB transmitter and a standard receiver. Moreover, we introduced simulation models that were applied to evaluate the proposed reception methods.

In Chapter 3 of the thesis, we described the state-of-the-art in non-iterative detection and decoding techniques for DE-QPSK streams with convolutional encoding. First, as a reference, coherent detection of DE-QPSK with soft-decision Viterbi decoding was studied. Then it was demonstrated that the performance of non-coherent 2SDD of DE-QPSK with soft-decision Viterbi decoding can be improved by, e.g., non-coherent MSDD. Since pilots are lacking for DAB systems, detection based on observing multiple received symbols is a technique that could lead to reception improvement for DAB receivers.

8.2.2 A-posteriori symbol probabilities and log-likelihood ratios for coherently detected $\frac{\pi}{4}$ -DE-QPSK

By applying MSDD, a DAB-receiver can approach the performance of a receiver that performs coherent detection. Therefore, we studied in Chapter 4 a-posteriori symbol probabilities and LLRs for coherently detected $\frac{\pi}{4}$ -DE-QPSK. It was demonstrated, as an extension to the results known in the literature, that an approximation of MAP symbol detection, based on selecting dominant exponentials, leads to MAP sequence detection. To improve the performance towards MAP symbol detection, a better approximation was proposed. This approximation relies on piecewise-linear fitting of the logarithm of the hyperbolic cosine and results in a performance quite close to that of MAP symbol detection. For the coded case, where the symbols are produced by convolutional encoding and Gray mapping, the LLRs were investigated. Again an approximation based on selecting dominant exponentials and an improved approximation relying on piecewise-linear fits, was proposed. The improved approximation gives a performance quite close to ideal.

8.2.3 The Shannon limit and some codes approaching it

In Chapter 5 we explained iterative decoding procedures corresponding to serially concatenated codes developed by Benedetto et al. because these decoding procedures are, later in the thesis, proposed for DAB-like streams. In that chapter we also considered parallel concatenated systems, turbo-codes, first described by Berrou et al. [11]. The iterative decoding procedures for the serially concatenated codes as well as for the turbo-codes are based on modified versions of the BCJR algorithm [4]. The approach taken in Chapter 5 to explain these iterative decoding procedures, is similar to the approach Gallager [32] followed to investigate iterative procedures for decoding LDPC codes. This way of ex-

plaining iterative decoding procedures for the serially concatenated codes as well as for the turbo-codes does not appear in the literature.

8.2.4 Two-dimensional iterative processing for DAB receivers based on trellis-decomposition

In Chapter 6 we investigated trellis decoding and iterative techniques for DAB receivers. Specifically, the concept of 2D-blocks and trellis decomposition in decoding was considered. Each 2D-block consists of a number of adjacent subcarriers of a number of subsequent OFDM symbols. The trellis-decomposition method allows for an estimation of the unknown channel phase, since this phase relates to sub-trellises. A dominant sub-trellis can be determined from the a-posteriori sub-trellis probabilities and the a-posteriori symbol probabilities corresponding to this dominant sub-trellis can be used. This dominant sub-trellis approach results in a significant complexity reduction, which is the subject of Chapter 7.

In Chapter 6, we also investigated non-iterative methods. To obtain an acceptable performance with single-carrier transmission, we need a trellis-length of $N + 1 \geq 32$. For such trellis-lengths the channel coherence-time needs to be in the order of $T_c \approx 32T_s$, where T_s is the OFDM symbol time. In practice the channel may not be coherent so long, and therefore focussing on trellis-length $N + 1 = 32$ might not be realistic. There is a second reason for arguing that large values of N are undesirable. DAB-systems support, for complexity reduction, per service symbol processing. In such services, typically, at most $N + 1 \leq 4$ subsequent OFDM symbols are contained in a single convolutionally encoded word and this does not match to processing more than four OFDM symbols in a demodulation trellis. Since we cannot make N too large, we used M adjacent sub-carriers to jointly determine the a-posteriori symbol probabilities for the corresponding DE-QPSK streams. We demonstrated that we are very close to the performance of coherent detection of DE-QPSK even for small values of the trellis length $N + 1 = 4$, by processing simultaneously $M = 8$ sub-carriers. Simulations showed that 2D dominant sub-trellis processing with $M(N + 1) = 32$ outperforms 2SDD by 0.7 dB at a BER of 10^{-4} .

Iterative decoding techniques for the single-carrier case lead to an improvement in required signal-to-noise ratio compared to 2SDD of at most ≈ 4.1 dB after $L = 5$ iterations and $N \rightarrow \infty$. Furthermore, for $N + 1 = 32$ the difference in performance between the exact and approximated LLRs, by selecting the dominant sub-trellis before starting with the iteration process, is less than 0.3 dB. In the iterative multi-carrier case we showed that increasing M has a positive effect on the performance, however when the trellis-length $N + 1$ is quite small the effect of iterating is limited. Simulations with $L = 5$ iterations showed that 2D dominant sub-trellis processing with $N + 1 = 32$ and $M = 1$ outperforms 2SDD by 3.7 dB at a BER $\approx 10^{-4}$. However, simulations also reveal that with $N + 1 = 4$ and $M = 8$ the performance improvement is reduced to 2.5 dB. The difference of 1.2 dB is caused by the smaller value of $N + 1$.

For the TU-6 channel model defined in COST-207 [1] it was shown that for $N + 1 = 18$ and $M = 1$, reliable transmission is not possible with movement speeds of ≈ 45 km/h and ≈ 90 km/h. However, for the 2D-decomposition approach with $N + 1 = 4$ and $M = 8$ we found considerable improvements of roughly 2.4 dB and 1.6 dB for Doppler

frequencies of 10 Hz, and 20 Hz, respectively.

8.2.5 A practical DAB system with 2D-block-based iterative decoding

In Chapter 7, complexity reduction of the inner decoder was investigated. This complexity reduction is realized by choosing, based on a-posteriori sub-trellis probabilities, a dominant sub-trellis (i) after each iteration or (ii) before starting with the iteration process. The first method reduces the number of multiplications by more than a factor of three, the number of normalizations by eight and, after five iterations, introducing ≈ 0.05 dB loss in performance on the COST-207 TU-6 channel. The second method results in reducing the number of multiplications and normalizations by more than a factor of eight and introduces ≈ 0.5 dB loss in performance.

We also described in that chapter, an implementation of a MAP channel-phase estimator which operates according to the second dominant sub-trellis approach. It was demonstrated that the a-posteriori probability of the channel-phase estimator is the product of sums of two cosh-functions. We showed that the max-log-MAP approximation combines an acceptable complexity with a good performance and is an appropriate method to be considered for channel phase estimation.

In addition, an implementation of a channel-gain estimator based on the power of the received DE-QPSK symbols within a 2D-block was discussed. We proposed, for complexity reduction reasons, a first channel-gain estimator only based on the received symbols within a 2D-block. It was shown that this channel-gain estimator requires the knowledge of the noise variance. For further reducing the complexity, we proposed a second channel-gain estimator which ignores the noise-variance term. We showed, by simulations for the TU-6 channel, that this simple channel-gain estimator shows good performances and is an appropriate method for channel gain estimation.

Finally, the performance improvements of a real-time and bit-true DAB-receiver are evaluated for various numbers of iterations, block-sizes, and Doppler-frequencies. This DAB receiver operates according to the proposed 2D-block-based iterative decoding procedure within a dominant sub-trellis obtained by the second method. Simulations for the AWGN channel showed a performance improvement of ≈ 2.6 dB for $L = 4$ iterations with $M(N + 1) = 80$. TU-6 channel simulations showed performance improvements up to ≈ 2.4 dB and ≈ 1.7 dB for speeds of 45 km/h and 90 km/h, respectively. The TU-6 simulation also showed that increasing $N + 1$ (with a fixed M) has a positive effect on the performance, however if the speed increases, the effect of large $N + 1$ is limited. Finally, in the non-iterative case the proposed DAB receiver shows at higher speeds better performances than 2SDD.

8.3 Recommendations

In this section we give some suggestions for further research.

8.3.1 Non-ideal conditions and more severe channel distortions

It is recommended to further investigate the performance of the proposed iterative 2D-block based detection method with trellis decomposition under more severe channel distortions. The main assumption of our proposed iterative 2D-block based detection method with trellis decomposition is that within a 2D-block the channel phase is fixed. However in some practical cases, already shown in this thesis, it can be seen that if the channel is varying quite fast, i.e., at higher Doppler frequencies, the improvements of our proposed iterative methods degrade. Therefore, it could be of interest to further investigate whether the assumption that the channel-phase is fixed within a 2D-block still holds under non-ideal conditions and more severe channel distortions. For example, in the thesis ideal synchronization is assumed, i.e., no ICI or ISI interference is introduced. As one can imagine, it could be of interest to investigate the performances if this assumption of ideal synchronization does not hold anymore. Also the influence of scaling-effects introduced by, for example, an automatic gain control (AGC) could be a subject for further investigation. The reason for this is that we assume for the estimation of the channel gain factor that the channel gain is constant within a 2D-block. In addition, more severe channel distortions could result from longer impulse responses, higher Doppler-frequencies, narrow-band interferences (ingress noise), broadband interferences (impulse noise), nonlinearities, and other impairments introduced by the radio front-end. For these situations in which the fixed channel phase assumption does not hold anymore, more sophisticated phase tracking and phase correction methods are required.

8.3.2 Implementation and (specific) optimization

A (programmable) hard-ware implementation of the real-time and bit-true DAB-receiver and its optimization are subjects for further studies. The real-time and bit-true iterative DAB-receiver, discussed in the thesis, has been realized in a PC-environment. The next step could be to realize the complete iterative DAB-receiver in a (programmable) hardware-platform targeting a specific Integrated Circuit (IC). This requires specific platform-dependent implementation optimizations.

8.3.3 Field-testing

The complete iterative implemented DAB-receiver should be tested “in the field”. From these field-tests we can learn how the proposed reception techniques for digital audio broadcasting systems are perceived.

8.3.4 Other systems

Finally, it should be recommended to investigate if other differentially encoded OFDM broadcast systems can reuse the technique that is developed here. For example, HD-radio is using differentially encoded OFDM for their reference symbols. Furthermore, the Japanese digital terrestrial television broadcast standard; Integrated Services Digital Broadcasting–Terrestrial (ISDB-T), also supports differentially encoded OFDM.

Appendix A

A-Posteriori Symbol Probabilities and LLRs for Coherent Detection.

A.1 A-posteriori symbol probabilities for coherent detection

In this appendix we derive the a-posteriori symbol probabilities for coherent detection of DE-QPSK with iud information symbols $\{a_n\}$ hence $\Pr\{a_n\} = 1/4$ for all $a_n \in \{1, e^{j\pi/2}, e^{j\pi}, e^{j3\pi/2}\}$. The channel phase ϕ takes only the values $\phi_i = \frac{\pi}{2}i$, for $i = 0, 1, 2, 3$ with equal probability $\Pr\{\phi_i\} = 1/4$. Hence the integral over ϕ in (3.22) can be replaced by a summation over ϕ_i with $\Pr\{\phi_i\} = \frac{1}{4}$. This yields

$$\Pr\{a_n|y_n, y_{n-1}\} = \sum_{x_{n-1}} \sum_{\phi} \left[\frac{1}{4} \frac{\Pr\{x_{n-1}\}}{4p(y_n, y_{n-1}) (2\pi\sigma^2)^2} \cdot \exp\left(-\frac{|y_n - e^{j\phi}x_{n-1}a_n|^2 + |y_{n-1} - e^{j\phi}x_{n-1}|^2}{2\sigma^2}\right) \right]. \quad (\text{A.1})$$

We can now replace $\exp(j\phi)x_{n-1}$ by $\exp(j\theta)$ and assume that θ is uniform over $\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$. This yields

$$\begin{aligned} \Pr\{a_n|y_n, y_{n-1}\} &= \sum_{\theta} \frac{1}{4} \frac{\exp\left(-\frac{|y_n - e^{j\theta}a_n|^2 + |y_{n-1} - e^{j\theta}|^2}{2\sigma^2}\right)}{4p(y_n, y_{n-1}) (2\pi\sigma^2)^2} \\ &= \sum_{\theta} \frac{1}{4} K_A \exp\left(\frac{1}{\sigma^2} \Re\{(y_n a_n^* + y_{n-1}) e^{-j\theta}\}\right), \quad (\text{A.2}) \end{aligned}$$

with

$$K_A \triangleq \frac{\exp\left(-\frac{\sum_{k=0}^1 |y_{n-k}|^2 + 2}{2\sigma^2}\right)}{4p(y_n, y_{n-1}) (2\pi\sigma^2)^2}.$$

Note that for $n = 1$ we have that $x_0 = 1$ instead of uniform over $\{1, e^{j\pi/2}, e^{j\pi}, e^{j3\pi/2}\}$. However note that still $e^{j\theta}$ is uniform over $\{1, e^{j\pi/2}, e^{j\pi}, e^{j3\pi/2}\}$. Now, by substitution of $\theta \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$, (A.2) becomes

$$\begin{aligned} \Pr\{a_n | y_n, y_{n-1}\} &= \frac{K_A}{4} \left\{ \exp\left(\frac{v_n}{\sigma^2}\right) + \exp\left(\frac{w_n}{\sigma^2}\right) + \exp\left(-\frac{v_n}{\sigma^2}\right) + \exp\left(-\frac{w_n}{\sigma^2}\right) \right\} \\ &= \frac{K_A}{2} \left\{ \cosh\left(\frac{v_n}{\sigma^2}\right) + \cosh\left(\frac{w_n}{\sigma^2}\right) \right\}, \end{aligned} \quad (\text{A.3})$$

$$\text{with } \begin{cases} v(a_n) \triangleq v_n = \Re\{y_n a_n^* + y_{n-1}\} \\ w(a_n) \triangleq w_n = \Im\{y_n a_n^* + y_{n-1}\}. \end{cases}$$

A.2 LLRs for coherent detection

The desired soft-decision bit metrics related to transmission n , i.e., the LLRs [37] can be expressed as

$$\lambda_n^1 = \ln\left(\frac{e^{m(\pi)} + e^{m(3\pi/2)}}{e^{m(0)} + e^{m(\pi/2)}}\right), \lambda_n^2 = \ln\left(\frac{e^{m(\pi/2)} + e^{m(\pi)}}{e^{m(0)} + e^{m(3\pi/2)}}\right), \quad (\text{A.4})$$

with symbol metrics

$$m(\phi) = \ln(\Pr\{a_n = e^{j\phi} | \mathbf{y}\}), \quad (\text{A.5})$$

where λ_n^1 corresponds to bit b_1 , λ_n^2 to bit b_2 , with Gray mapping conform

$b_1 b_2$	00	01	11	10
$a(b_1 b_2)$	1	$e^{j\pi/2}$	$e^{j\pi}$	$e^{j3\pi/2}$

Combining of (A.3) and (A.4) gives the soft-decision bit metrics

$$\lambda_n^1 = \ln\left(\frac{\cosh\left(\frac{v(\pi)}{\sigma^2}\right) + \cosh\left(\frac{w(\pi)}{\sigma^2}\right) + \cosh\left(\frac{v(\frac{3\pi}{2})}{\sigma^2}\right) + \cosh\left(\frac{w(\frac{3\pi}{2})}{\sigma^2}\right)}{\cosh\left(\frac{v(0)}{\sigma^2}\right) + \cosh\left(\frac{w(0)}{\sigma^2}\right) + \cosh\left(\frac{v(\frac{\pi}{2})}{\sigma^2}\right) + \cosh\left(\frac{w(\frac{\pi}{2})}{\sigma^2}\right)}\right), \quad (\text{A.6})$$

where

$$\begin{aligned} v(0) &= +\Re\{y_n\} + \Re\{y_{n-1}\}, \\ w(0) &= +\Im\{y_n\} + \Im\{y_{n-1}\}, \\ v\left(\frac{\pi}{2}\right) &= +\Im\{y_n\} + \Re\{y_{n-1}\}, \\ w\left(\frac{\pi}{2}\right) &= -\Re\{y_n\} + \Im\{y_{n-1}\}, \\ v(\pi) &= -\Re\{y_n\} + \Re\{y_{n-1}\}, \\ w(\pi) &= -\Im\{y_n\} + \Im\{y_{n-1}\}, \\ v\left(\frac{3\pi}{2}\right) &= -\Im\{y_n\} + \Re\{y_{n-1}\}, \\ w\left(\frac{3\pi}{2}\right) &= +\Re\{y_n\} + \Im\{y_{n-1}\}. \end{aligned}$$

In addition, with the used Gray-mapping shown by (A.4), the soft-decision bit metric λ_n^2 is obtained by interchanging $\frac{\pi}{2}$ with $\frac{3\pi}{2}$ in (A.6).

A.3 Max-log-MAP approximation of the LLRs

Applying the max-log-MAP approximation to the optimal soft-decision bit metrics given by (3.27) yields

$$\lambda_n^1 \approx \frac{1}{\sigma^2} \left\{ \max \left(|v(\pi)|, |w(\pi)|, \left| v\left(\frac{3\pi}{2}\right) \right|, \left| w\left(\frac{3\pi}{2}\right) \right| \right) - \max \left(|v(0)|, |w(0)|, \left| v\left(\frac{\pi}{2}\right) \right|, \left| w\left(\frac{\pi}{2}\right) \right| \right) \right\}, \quad (\text{A.7})$$

with

$$\begin{aligned} |v(0)| &= |\Re\{y_n\} + \Re\{y_{n-1}\}|, \\ |w(0)| &= |\Im\{y_n\} + \Im\{y_{n-1}\}|, \\ \left| v\left(\frac{\pi}{2}\right) \right| &= |\Im\{y_n\} + \Re\{y_{n-1}\}|, \\ \left| w\left(\frac{\pi}{2}\right) \right| &= |\Re\{y_n\} - \Im\{y_{n-1}\}|, \\ |v(\pi)| &= |\Re\{y_n\} - \Re\{y_{n-1}\}|, \\ |w(\pi)| &= |\Im\{y_n\} - \Im\{y_{n-1}\}|, \\ \left| v\left(\frac{3\pi}{2}\right) \right| &= |\Im\{y_n\} - \Re\{y_{n-1}\}|, \\ \left| w\left(\frac{3\pi}{2}\right) \right| &= |\Re\{y_n\} + \Im\{y_{n-1}\}|. \end{aligned}$$

Also here, the approximation for λ_n^2 is obtained by interchanging $\frac{\pi}{2}$ with $\frac{3\pi}{2}$ in (A.7).

Appendix B

A-Posteriori Symbol Probabilities and LLRs for 2SDD.

B.1 A-posteriori symbol probabilities for non-coherent 2SDD

In this section we derive the a-posteriori symbol probabilities for non-coherent 2SDD of DE-QPSK on the AWGN channel. For non-coherent 2SDD of DE-QPSK symbols it is assumed that the frequency of the carrier is known. The channel phase, i.e., the carrier phase including the phase rotation of the channel is unknown. Hence we model the channel phase ϕ as uniformly distributed over $[0, 2\pi)$, hence $p(\phi) = \frac{1}{2\pi}$, and substitute this in (3.29). This yields

$$\Pr\{a_n|y_n, y_{n-1}\} = \sum_{x_{n-1}} \frac{\Pr\{x_{n-1}\}}{4p(y_n, y_{n-1})(2\pi\sigma^2)^2} \int_{\phi=0}^{2\pi} \frac{1}{2\pi} \exp\left(-\frac{|y_n - e^{j\phi}x_{n-1}a_n|^2 + |y_{n-1} - e^{j\phi}x_{n-1}|^2}{2\sigma^2}\right) d\phi. \quad (\text{B.1})$$

We can now replace $\exp(j\phi)x_{n-1}$ by $\exp(j\theta)$ and assume that θ is uniform over $[0, 2\pi)$, thus $p(\theta) = \frac{1}{2\pi}$. This yields

$$\begin{aligned} \Pr\{a_n|y_n, y_{n-1}\} &= \frac{1}{2\pi} \int_{\theta=0}^{2\pi} \frac{\exp\left(-\frac{|y_n - e^{j\theta} a_n|^2 + |y_{n-1} - e^{j\theta}|^2}{2\sigma^2}\right)}{4p(y_n, y_{n-1}) (2\pi\sigma^2)^2} d\theta \\ &= \frac{1}{2\pi} K_A \int_{\theta=0}^{2\pi} \exp\left(\frac{1}{\sigma^2} \Re\{(y_n a_n^* + y_{n-1}) e^{-j\theta}\}\right) d\theta \\ &= \frac{1}{2\pi} K_A \int_{\theta=0}^{2\pi} \exp\left(\frac{1}{\sigma^2} (v_n \cos \theta + w_n \sin \theta)\right) d\theta. \end{aligned} \quad (\text{B.2})$$

Note that for $n = 1$ we have that $x_0 = 1$ instead of uniform over $\{1, e^{j\pi/2}, e^{j\pi}, e^{j3\pi/2}\}$. However note that still $e^{j\theta}$ is uniform over $[0, 2\pi)$. Rewriting (B.2), using [35, 3.032.2, p.213], and by substitution of $z_n = v_n + iw_n$ yields

$$\begin{aligned} \Pr\{a_n|y_n, y_{n-1}\} &= \frac{1}{\pi} K_A \int_{\theta=0}^{\pi} \exp\left(\frac{1}{\sigma^2} \sqrt{v_n^2 + w_n^2} \cos \theta\right) d\theta \\ &= \frac{1}{\pi} K_A \int_{\theta=0}^{\pi} \exp\left(\frac{1}{\sigma^2} |z_n| \cos \theta\right) d\theta, \end{aligned} \quad (\text{B.3})$$

with

$$|z_n| = |y_n a_n^* + y_{n-1}|. \quad (\text{B.4})$$

Using [35, 8.431.3, p.958] with $\nu = 0$ and substitution of (B.4), we can rewrite (B.3), and we obtain

$$\begin{aligned} \Pr\{a_n|y_n, y_{n-1}\} &= K_A I_0\left(\frac{1}{\sigma^2} |z_n|\right) \\ &= K_A I_0\left(\frac{1}{\sigma^2} |y_n a_n^* + y_{n-1}|\right), \end{aligned} \quad (\text{B.5})$$

where $I_0(\cdot)$ is the zero-th order modified Bessel function of the first kind.

B.2 LLRs for 2SDD

The desired soft-decision bit metrics related to transmission n , i.e., the LLRs can be expressed as

$$\lambda_n^1 = \ln\left(\frac{e^{m(\pi)} + e^{m(3\pi/2)}}{e^{m(0)} + e^{m(\pi/2)}}\right), \lambda_n^2 = \ln\left(\frac{e^{m(\pi/2)} + e^{m(\pi)}}{e^{m(0)} + e^{m(3\pi/2)}}\right), \quad (\text{B.6})$$

with symbol metrics

$$m(\phi) = \ln(\Pr\{a_n = e^{j\phi} | \mathbf{y}\}), \quad (\text{B.7})$$

where λ_n^1 corresponds to bit b_1 , λ_n^2 to bit b_2 , with Gray mapping conform [25], i.e.,

$$\frac{b_1 b_2}{a(b_1 b_2)} \begin{array}{c|cccc} & 00 & 01 & 11 & 10 \\ \hline & 1 & e^{j\pi/2} & e^{j\pi} & e^{j3\pi/2} \end{array}.$$

Combining (B.5) and (B.6) gives the soft-decision bit metrics

$$\lambda_n^1 = \ln \left(\frac{I_0 \left(\frac{1}{\sigma^2} |y_n e^{-j\pi} + y_{n-1}| \right) + I_0 \left(\frac{1}{\sigma^2} |y_n e^{-j3\pi/2} + y_{n-1}| \right)}{I_0 \left(\frac{1}{\sigma^2} |y_n + y_{n-1}| \right) + I_0 \left(\frac{1}{\sigma^2} |y_n e^{-j\pi/2} + y_{n-1}| \right)} \right). \quad (\text{B.8})$$

In addition, with the applied Gray-mapping, the soft-decision bit metric λ_n^2 is obtained by interchanging $\frac{\pi}{2}$ with $\frac{3\pi}{2}$ in (B.8).

References

- [1] *COST 207, Digital land mobile radio communications*, Office for Official Publications of the European Communities (1989).
- [2] Marvin K. Simon Mohamed-Slim Alouini, *Digital Communication over Fading Channels: A Unified Approach to Performance Analysis*, first ed., John Wiley & Sons, Inc., 2000, ISBN 0-0471-31779-9.
- [3] ———, *Digital Communication over Fading Channels*, second ed., John Wiley & Sons, Inc., 2005, ISBN 0-471-64953-8.
- [4] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, *Optimal decoding of linear codes for minimizing symbol error rate (corresp.)*, Information Theory, IEEE Transactions on **20** (1974), no. 2, 284–287.
- [5] G. Begin, D. Haccoun, and C. Paquin, *Further Results on High-Rate Punctured Convolutional Codes for Viterbi and Sequential Decoding*, IEEE Trans. on Communications **38** (1990), no. 11, 1922–1928.
- [6] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, *A soft-input soft-output APP module for iterative decoding of concatenated codes*, Communications Letters, IEEE **1** (1997), no. 1, 22–24.
- [7] ———, *Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding*, Information Theory, IEEE Transactions on **44** (1998), no. 3, 909–926.
- [8] ———, *Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding*, TDA Progress Report 42-126 (August 15, 1996), 1–26.
- [9] S. Benedetto and G. Montorsi, *Iterative decoding of serially concatenated convolutional codes*, Electronics Letters **32** (1996), no. 13, 1186–1188.
- [10] Jan W. M. Bergmans, *Digital Baseband Transmission and Recording*, first ed., Kluwer Academic Publishers, P.O. Box 17, 3300AA Dordrecht, The Netherlands, 1996, ISBN 0-7923-9775-4.
- [11] C. Berrou, A. Glavieux, and P. Thitimajshima, *Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1*, Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record, IEEE International Conference on, vol. 2, May 1993, pp. 1064–1070 vol.2.

- [12] E. Biglieri, J. Proakis, and S. Shamai, *Fading Channels: Information-Theoretic and Communications Aspects*, IEEE Trans. on Information Theory **44** (1998), no. 6, 2619–2692.
- [13] R.C. Bose and D.K. Ray-Chaudhuri, *Further Results on Error Correcting Binary Group Codes*, Inform. Contr. **3** (1960), 279–290.
- [14] ———, *On a Class of Error Correcting Binary Group Codes*, Inform. Contr. **3** (1960), 68–79.
- [15] G.E. Bottomley, H. Arslan, R. Ramesh, and G. Brismark, *Coherent MAP detection of DQPSK bits*, Comm. Letters, IEEE **4** (2000), no. 11, 354 – 356.
- [16] G. Caire, G. Taricco, and E. Biglieri, *Bit-interleaved coded modulation*, Information Theory, IEEE Transactions on **44** (1998), no. 3, 927–946.
- [17] M.L. Cedervall and R. Johannesson, *A fast algorithm for computing distance spectrum of convolutional codes*, Information Theory, IEEE Transactions on **35** (1989), no. 6, 1146 – 1159.
- [18] Rong-Rong Chen, R. Koetter, U. Madhow, and D. Agrawal, *Joint noncoherent demodulation and decoding for the block fading channel: a practical framework for approaching Shannon capacity*, Communications, IEEE Transactions on **51** (2003), no. 10, 1676 – 1689.
- [19] G. Colavolpe, *Classical coherent receivers for differentially encoded M-PSK are optimal*, Communications Letters, IEEE **8** (2004), no. 4, 211 – 213.
- [20] G. Colavolpe, G. Ferrari, and R. Raheli, *Noncoherent iterative (turbo) decoding*, Communications, IEEE Transactions on **48** (2000), no. 9, 1488–1498.
- [21] D. Divsalar and M. K. Simon, *Maximum-likelihood differential detection of uncoded and trellis coded amplitude phase modulation over AWGN and fading channels-metrics and performance*, Communications, IEEE Transactions on **42** (1994), no. 1, 76 – 89.
- [22] D. Divsalar and M.K. Simon, *Multiple-symbol differential detection of MPSK*, Communications, IEEE Transactions on **38** (1990), no. 3, 300–308.
- [23] D. Divsalar, M.K. Simon, and M. Shahshahani, *The performance of trellis-coded MDPSK with multiple symbol detection*, Communications, IEEE Transactions on **38** (1990), no. 9, 1391–1403.
- [24] P. Elias, *Coding for Noisy channels.*(Reprinted in *Key Papers in the Development of Coding Theory*, ed. E. Berlekamp, pp. 48-55. New York: IEEE Press, 1974.), IRE National Convention Record **3(4)** (1955), 37–46.
- [25] European Standard (Telecommunications series), *Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers*, ETSI EN 300 401 V1.4.1 (2006-06), June 2006, V1.4.1.
- [26] ———, *Digital Radio Mondiale (DRM); System Specification*, ETSI ES 201 980 V3.1.1 (2009-08), August 2009, V3.1.1.

- [27] G. Falciascecca et al., *Influence of Environmental Conditions on BER for Wide band Mobile Radio channels*, *Alta Frequenza* **57** (1988), no. 2, 75–82.
- [28] G.D. Forney and D.J. Costello, *Channel Coding: The Road to Channel Capacity*, *Proceedings of the IEEE* **95** (2007), no. 6, 1150–1177.
- [29] Jr. Forney, G.D. and G. Ungerboeck, *Modulation and coding for linear Gaussian channels*, *Information Theory, IEEE Transactions on* **44** (1998), no. 6, 2384–2415.
- [30] M.P.C. Fossorier, F. Burkert, Shu Lin, and J. Hagenauer, *On the equivalence between SOVA and max-log-MAP decodings*, *Communications Letters, IEEE* **2** (1998), no. 5, 137–139.
- [31] M. Franceschini, G. Ferrari, and R. Raheli, *Detection by multiple trellises*, *Communications, IEEE Transactions on* **57** (2009), no. 3, 726–737.
- [32] R.G. Gallager, *Low-Density Parity-Check Codes*, *IRE Trans. Inform. Theory* (1962), 21–28.
- [33] Jr. G.D. Forney, *Concatenated Codes.*, MA: MIT Press, Cambridge, 1966.
- [34] M.J.E. Golay, *Note on the Theoretical Efficiency of Information Reception with PPM*, *Proc. IRE* **37** (1949), 1031.
- [35] I. S. Gradshteyn and I. M. Ryzhik, *Table of integrals, series and products Corrected and Enlarged edition*, fourth ed., Academic Press, California, 1980, ISBN 012-294-7606.
- [36] P. Haase and H. Rohling, *Iterative detection of differentially modulated APSK signals in an OFDM transmission system*, *Wireless Personal Multimedia Communications, 2002. The 5th International Symposium on* **2** (2002), 711–714 vol.2.
- [37] Hoehner P. Hagenauer, J., *A Viterbi algorithm with soft-decision outputs and its applications*, Nov 1989, pp. 1680–1686 vol.3.
- [38] J. Hagenauer, N. Seshadri, and C.-E.W. Sundberg, *The performance of rate-compatible punctured convolutional codes for digital mobile radio*, *Communications, IEEE Transactions on* **38** (1990), no. 7, 966–980.
- [39] R.W. Hamming, *Error Detecting and Error Correcting Codes*, *The Bell System Technical Journal* **29** (1950), 147–160.
- [40] A. Hocquenghem, *Codes Correcteurs d’Erreurs (Error Correcting Codes)*, *Chiffres* **2** (1959), 147–156.
- [41] P. Hoehner and J. Lodge, *Iterative differential PSK demodulation and channel decoding*, *Information Theory, 1998. Proceedings. 1998 IEEE International Symposium on* (1998), 425–.
- [42] ———, *“Turbo DPSK”: iterative differential PSK demodulation and channel decoding*, *Communications, IEEE Transactions on* **47** (1999), no. 6, 837–843.
- [43] K. A. Schouhamer Immink, *Codes for Mass Data Storage Systems*, first ed., Shannon Foundation Publishers, The Netherlands, 1999, ISBN 90-74249-23-X.

- [44] Rolf Johannesson and Kamil sh. Zigangirov, *Fundamentals of Convolutional Coding*, first ed., IEEE series on Digital & Mobile communications, IEEE-Press, Lund University, 1999, ISBN 0-7803-3483-3.
- [45] Robert S. Kennedy, *Fading Dispersive Communication Channels*, first ed., Wiley-Interscience, September 1969, ISBN 471 46903 3.
- [46] L.H.-J. Lampe and R. Schober, *Iterative decision-feedback differential demodulation of bit-interleaved coded MDPSK for flat Rayleigh fading channels*, Communications, IEEE Transactions on **49** (2001), no. 7, 1176–1184.
- [47] J. Lassing, E.G. Strom, E. Agrell, and T. Ottosson, *Computation of the exact bit-error rate of coherent M-ary PSK with Gray code bit mapping*, Communications, IEEE Transactions on **51** (2003), no. 11, 1758 – 1760.
- [48] Pil Lee, *Computation of the Bit Error Rate of Coherent M-ary PSK with Gray Code Bit Mapping*, Communications, IEEE Transactions on **34** (1986), no. 5, 488 – 491.
- [49] I. D. Marsland and P. T. Mathiopoulos, *On the performance of iterative noncoherent detection of coded M-PSK signals*, Communications, IEEE Transactions on **48** (2000), no. 4, 588–596.
- [50] J.L. Massey, *Coding and modulation in digital communications*, IZS-74 International Zurich Seminar on Digital Communications, vol. E2, 1974, pp. 1–4.
- [51] T. May and H. Rohling, *Turbo decoding of convolutional codes in differentially modulated OFDM transmission systems*, Vehicular Technology Conference, 1999 IEEE 49th **3** (1999), 1891–1895 vol.3.
- [52] Seung Young Park, Yeun Gu Kim, Chung Gu Kang, and Dae Eop Kang, *Iterative receiver with joint detection and channel estimation for OFDM system with multiple receiver antennas in mobile radio channels*, Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE **5** (2001), 3085–3089 vol.5.
- [53] V. Pauli, L. Lampe, and R. Schober, *"turbo DPSK" using soft multiple-symbol differential sphere decoding*, Information Theory, IEEE Transactions on **52** (2006), no. 4, 1385–1398.
- [54] R. Pawula, S. Rice, and J. Roberts, *Distribution of the Phase Angle Between Two Vectors Perturbed by Gaussian Noise*, Communications, IEEE Transactions on **30** (1982), no. 8, 1828 – 1841.
- [55] R.F. Pawula, *Distribution of the phase angle between two vectors perturbed by gaussian noise II*, Vehicular Technology, IEEE Transactions on **50** (2001), no. 2, 576 –583.
- [56] M. Peleg, I. Sason, S. Shamai, and A. Elia, *On interleaved, differentially encoded convolutional codes*, Information Theory, IEEE Transactions on **45** (1999), no. 7, 2572–2582.
- [57] M. Peleg and S. Shamai, *Iterative decoding of coded and interleaved noncoherent multiple symbol detected DPSK*, Electronics Letters **33** (1997), no. 12, 1018–1020.

- [58] M. Peleg, S. Shamai, and S. Galan, *Iterative decoding for coded noncoherent MPSK communications over phase-noisy AWGN channel*, Communications, IEE Proceedings- **147** (2000), no. 2, 87–95.
- [59] John G. Proakis, *Digital Communications*, second ed., McGraw-Hill, Northeastern University, 1989, ISBN 0-07-050937-9.
- [60] ———, *Digital Communications*, fourth ed., Higher Education, McGraw-Hill, Northeastern University, 2001, ISBN 0-07-232111-3.
- [61] John G. Proakis and Masoud Salehi, *Digital Communications*, fifth ed., Higher Education, McGraw-Hill, Northeastern University, 2008, ISBN 978-0-07-126378-8.
- [62] Theodore S. Rappaport, *Wireless communications principles and practice*, first ed., IEEE Press, Prentice Hall PTR, New York, New Jersey, 1996, ISBN 0-7803-1167-1 (IEEE), ISBN 0-13-461088-1 (Prentice Hall).
- [63] P. Robertson, E. Vilebrun, and P. Hoeher, *A comparison of optimal and sub-optimal MAP decoding algorithms operating in the log domain*, ICC '95, Seattle **2** (1995), 1009 – 1013.
- [64] H. Rohling and T. May, *OFDM systems with differential modulation schemes and turbo decoding techniques*, Broadband Communications, 2000. Proceedings. 2000 International Zurich Seminar on (2000), 251–255.
- [65] F. Sanzi and M. C. Necker, *Totally blind APP channel estimation with higher order modulation schemes*, Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th **2** (2003), 1167–1171 Vol.2.
- [66] C. E. Shannon, *A Mathematical Theory of Communication*, The Bell System Technical Journal **XXVII** (1948), no. 4, 623–656.
- [67] ———, *A Mathematical Theory of Communication*, The Bell System Technical Journal **XXVII** (1948), no. 3, 379–423.
- [68] M.K. Simon, *On the bit-error probability of differentially encoded QPSK and offset QPSK in the presence of carrier synchronization*, Communications, IEEE Transactions on **54** (2006), no. 5, 806 – 812.
- [69] M.K. Simon and D. Divsalar, *On the optimality of classical coherent receivers of differentially encoded M-PSK*, Communications Letters, IEEE **1** (1997), no. 3, 67 –70.
- [70] S. Talakoub, L. Sabeti, B. Shahrava, and M. Ahmadi, *An improved Max-Log-MAP Algorithm for Turbo Decoding and Turbo Equalization*, Instrum. Meas., IEEE Trans. **56** (2007), no. 3, 1058 – 1063.
- [71] S. ten Brink, F. Sanzi, and J. Speidel, *Two-dimensional iterative APP channel estimation and decoding for OFDM systems*, Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE **2** (2000), 741–745 vol.2.
- [72] S. ten Brink, J. Speidel, and R.-H. Han, *Iterative demapping for QPSK modulation*, Electronics Letters **34** (1998), no. 15, 1459–1460.

- [73] W. J. van Houtum and F. M. J. Willems, *Joint and iterative detection and decoding of differentially encoded COFDM systems*, Telecommunications (ICT), 2010 IEEE 17th International Conference on, Apr 2010, pp. 36–43.
- [74] ———, *A-posteriori Symbol Probabilities and Log-Likelihood Ratios for Coherently Detected $\frac{\pi}{4}$ -DE-QPSK*, Communications Letters, IEEE **15** (2011), no. 2, 160–162.
- [75] ———, *Two-Dimensional Iterative Processing for DAB Receivers Based on Trellis-Decomposition*, Journal of Electrical and Computer Engineering **2012** (2012), no. 394809, 15 pages.
- [76] A.J. Viterbi, *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm*, IEEE Trans. Inform. Theory **IT-13** (1967), no. 4, 260–269.
- [77] A.J. Viterbi and J.K. Omura, *Principles of Digital Communication and Coding*, McGraw-Hill, New York, 1979.
- [78] J. M. Wozencraft and I. M. Jacobs, *Principles of Communication engineering*, Waveland Press, 1990.
- [79] G. Zimmermann, M. Rosenberger, and S. Dostert, *Theoretical bit error rate for uncoded and coded data transmission in digital audio broadcasting*, Communications, 1996. ICC 96, Conference Record, Converging Technologies for Tomorrow's Applications. 1996 IEEE International Conference on **1** (1996), 297–301 vol.1.

Acknowledgments

I would like to express many thanks to Jan Bergmans and Frans Willems for their help, advice, and enthusiasm during my research and PhD-work. A special thanks goes to Frans who helped me to make the appropriate choices and judgements. Moreover, after our discussions on the “white-board” we had both the same ideas on what the next steps would be to take. I’m convinced that this thesis would have never been possible without the kind of support that was given to me by Jan and Frans.

I am deeply indebted to my colleagues and former colleagues at Catena Radio Design, especially Jasper Siemons, Joris van der Pas, Marko Boomstra, and Anurag Bajpai. Their contribution and effort to get the ideas proven in practice was invaluable.

In the framework of the joint development programme between Catena and NXP, I especially appreciated the help of Kave Kianush and Andrew Turley in facilitating and supporting my PhD-work.

I wish to thank the many people with whom I have worked at NXP. I liked working with Semih Serbetli, Nur Engin, Erik Lambers, and also with Martin Kessel and Joerg Siemes. In our signal processing group I enjoyed the discussions with Wu Yan and Peng Zhang, my roommates at the Technical University Eindhoven.

I am grateful to the members of the defense committee, prof. Liesbet van der Perre, dr. Peter Smulders, and prof. Han Vinck, for their valuable comments on the draft thesis and their presence in the defense.

A special thanks goes to our friend Elke Bax, who was helping us out in the weekends and supported me in this way to be able to work on my thesis in the weekends.

Last, but certainly not least, a very warm and special thanks goes to my lovely wife Dorine for here unconditionally support and almost never ending patience to let me finish this thesis.

Curriculum Vitae

Wim van Houtum was born on 14-06-1964 in Sint-Oedenrode, the Netherlands. He started his career in 1980 at Philips and joined Philips Research Laboratories Eindhoven in 1982. He received the Master of Science degree from Eindhoven University of Technology in 1995. After his graduation he worked as a research scientist at Philips Research Laboratories on topics related to Digital Video Broadcasting systems. In April 1998 he moved to Philips Semiconductors Systems Laboratory Eindhoven and continued his work on Digital Video Broadcasting systems as a senior systems engineer/project leader. Later on he became a System Architect on Wireless Networking. He was several years involved in standardization of Wireless Local Area Networks. In 2000 he joined again the Philips Research Laboratories in Eindhoven to investigate diversity techniques for COFDM systems in the home environment. Since 2004 he is a senior system expert on (COFDM-based) digital radio systems at Catena Radio Design. He was one of the technical editors and reviewer of the Digital Radio Mondiale standard. He holds several patents on topics related to communication systems.

