

Research Article

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

Wim J. van Houtum^{1,2} and Frans M. J. Willems²

¹ *Catena Radio Design, Science Park Eindhoven, Ekkersrijt 5228, 5692 EG Son en Breugel, The Netherlands*

² *Department of Electrical Engineering, Technische Universiteit Eindhoven, Den Dolech 2, Postbus 513, 5600 MB Eindhoven, The Netherlands*

Correspondence should be addressed to Wim J. van Houtum, wvhoutum@catena.nl

Received 6 July 2011; Revised 14 October 2011; Accepted 17 October 2011

Academic Editor: Chirn Chye Boon

Copyright © 2012 W. J. van Houtum and F. M. J. Willems. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We investigate iterative trellis decoding 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 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; however, also their gain of 0.7 dB, relative to two-symbol differential detection (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 (COST207) channel show that we get an improvement of 2.4 dB at a Doppler frequency of 10 Hz.

1. Introduction

1.1. Problem Description. Digital audio broadcasting (DAB) systems, DAB+ systems, and terrestrial-digital multimedia broadcasting (T-DMB) systems use orthogonal frequency division multiplexing (OFDM), for which every OFDM-subcarrier is modulated by $\pi/4$ -Differentially Encoded-Quaternary PSK (DE-QPSK) [1].

Commonly used *classical* DAB receivers perform non-coherent 2SDD with soft-decision Viterbi decoding [2]. Noncoherent detection schemes like 2SDD are not optimal and can be improved by multisymbol differential detection (MSDD), which is a maximum likelihood procedure for finding a block of information symbols after observing a block of received symbols [3]. For very large numbers of observations, the performance of MSDD approaches the performance of ideal coherent detection of DE-QPSK, which

is given in, for example, [4–6]. Noncoherent MSDD can also be used if channel coding is applied in a noniterative way, see [7, 8].

If MSDD is combined with iterative (turbo) processing (parallel concatenated systems were first described by Berrou et al. [9], serial concatenation was developed by Benedetto and coinvestigators [10–12]), 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 [13–18], as well as for time-varying channels in [19–23]. 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

[19, 20], and for the 2D-case [24–26], or secondly by directly calculating the *a posteriori* probabilities of the information symbols as in [17, 18, 22], and for the 2D case [27–29].

We focus in the present paper on 2D processing, that is, in both the frequency and time domains. 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 [24] on 2D phase-estimate methods can be regarded as an extension of the results of Hoeher and Lodge in [20] to the multicarrier case. Park et al. in [25] improved the hard-decision approach of ten Brink et al. by considering soft-decision. Both [24, 25] rely on pilot symbols, which are not present in DAB-transmission [1] unfortunately. Blind channel estimation techniques were proposed by Sanzi and Necker in [26]. They proposed a combination of the iterative scheme of ten Brink in [24] and a fast converging blind channel estimator based on higher-order asymmetrical modulation schemes, which are not used within a DAB-transmission [1].

To obtain a posteriori probabilities of the information symbols in a 2D setting, May, Rohling, and Haase in [27–29] considered iterative decoding schemes for multicarrier modulation with the soft-output Viterbi algorithm (SOVA, [30]). The SOVA was used for differential detection 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 innerdecoder, actually ignoring the differential encoding. For the incoherent case, they used a transition metric for the SOVA innerdecoder 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 [5] for the coherent case, but also in [18] for the incoherent case.

To reduce complexity we accept a small performance loss due to channel-phase discretization (see, e.g., Peleg et al. [17] and Chen et al. [22]) 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 [17]. 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 [31] in a 2D setting, see also [32]. We will consider 2D blocks and trellis decomposition. 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, but also since 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 allow us to estimate the unknown channel-phase efficiently. This phase is related to subtrellises of which we can determine the a posteriori probabilities. With these probabilities we are able to chose a dominant subtrellis, which results in a significant complexity reduction.

Franceschini et al. [33] also use the idea of trellis-decomposition and subtrellises (multiple trellises), to focus on estimating channel parameters. Variation of these parameters is tackled by applying the so-called intermix 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 Section 2.1, why we cannot track the channel phase.

1.2. Paper Outline. In this paper, we will focus both on iterative and noniterative decoding techniques for DAB-like systems. In the next section, we will give a short outline of the DAB system. In Section 3, we will start our analysis by considering noniterative methods for the single-carrier case and introduce trellis-decomposition with a dominant subtrellis approach. In Section 4, we expand our single-carrier methods to the multicarrier case and introduce a 2D-block approach for demodulation. Iterative methods based on serial concatenation of convolutional codes (SCCC) and trellis-decomposition with two dominant subtrellis approaches are considered in Section 5. In Section 6, we generalize our iterative methods for the single-carrier case to the multicarrier case with 2D-block demodulation. Results of applying our approach to a practical case are shown in Section 7. Finally, Section 8 draws conclusions on our decoding procedures based on 2D blocks and trellis-decomposition for DAB-like systems.

2. Description of a Digital Audio Broadcasting (DAB) System

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, $\pi/4$ -DE-QPSK modulation followed by OFDM, see Figure 1. Time multiplexing of the transmitted services allows the receiver to perform *per service symbol processing* [1], see Figure 2 where 1536 is the number of “active” OFDM subcarriers for a DAB-transmission in Mode-I [1]; 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 [33] 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

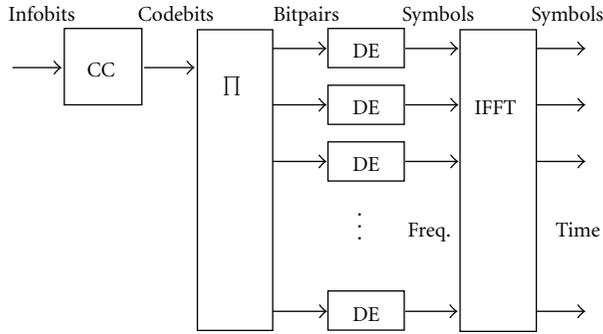


FIGURE 1: DAB convolutional encoder, interleaver, differential encoders, and multicarrier modulator with an inverse fast Fourier transform (IFFT).

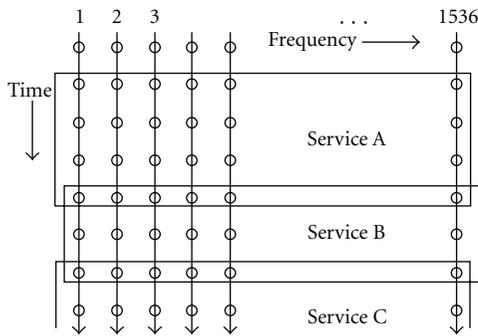


FIGURE 2: Three services mapped onto consecutive OFDM symbols. Note that there is an overlap between the service since differential modulation is used.

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.

2.2. Convolutional Coding and Interleaving. The convolutional code that is used within DAB has basic code-rate $R_c = 1/4$, 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 et al. [34]. The time and frequency interleavers in DAB perform bit and bit-pair interleaving, respectively. As a result the code-bits leaving the convolutional encoder are permuted and partitioned over the subcarriers of a number of subsequent OFDM symbols (in subsequent frames). The bits for each subcarrier 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, that is, labels that correspond to adjacent phase differences differ only in a single bit position.

2.3. Differential Modulation in Each Subcarrier. For each subcarrier, $\pi/4$ -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

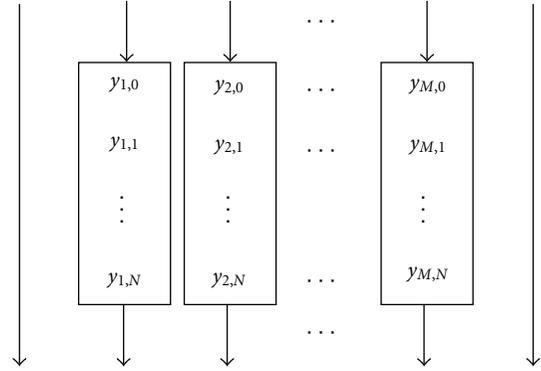


FIGURE 3: A 2D block of symbols out of an OFDM stream. We are interested in M adjacent sequences of $N + 1$ subsequent symbols, where each such sequence corresponds to one of M adjacent subcarriers.

that is to be transmitted via this subcarrier. 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, that is,

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

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

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

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

and $s_{m,n}$ is the n -th differentially encoded symbol corresponding to the m -th subcarrier, or equivalently the m -th element in the n -th frequency-domain OFDM-symbol, see Figure 1. Note, that the IFFT is a computationally efficient inverse discrete Fourier transform (IDFT) for values of B that are powers of 2. To prevent Intersymbol interference (ISI) resulting from multipath reception, a cyclic prefix of length L_{cp} is added to the sequence $\tilde{\mathbf{s}}_n$. This leads to the sequence $\tilde{\mathbf{s}}_n = (\tilde{s}_{B-L_{cp}+1,n}, \dots, \tilde{s}_{B,n}, \tilde{s}_{1,n}, \tilde{s}_{2,n}, \dots, \tilde{s}_{B,n})$ 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 Figure 3.

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

$$r_{m,n} = \frac{1}{\sqrt{B}} \sum_{t=1}^B \tilde{r}_{t,n} e^{-j2\pi(m-1)(t-1)/B}, \quad \text{for } m = 1, 2, \dots, B. \quad (3)$$

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

$$r_{m,n} = |h| e^{j\phi} s_{m,n} + n_{m,n}, \quad (4)$$

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.

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:

$$r_n = e^{j\phi} s_n + n_n, \quad \text{for } n = 0, 1, \dots, N, \quad (5)$$

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, for example, Peleg et al. [17] and Chen et al. [22], we may assume that the channel-phase is discrete and uniform over 32 levels, which are uniformly spaced over $[0, 2\pi)$, hence

$$\Pr\left\{\phi = \frac{\pi l}{16}\right\} = \frac{1}{32}, \quad \text{for } l = 0, 1, 2, \dots, 31. \quad (6)$$

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

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

subcarriers, is equivalent to DE-QPSK. To see this, we define for $n = 1, 2, \dots, N$

$$a_n = b_n e^{-j\pi/4}, \quad (7)$$

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

$$\begin{aligned} x_n &= s_n e^{-jn\pi/4}, \\ y_n &= r_n e^{-jn\pi/4}, \\ w_n &= n_n e^{-jn\pi/4}. \end{aligned} \quad (8)$$

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

$$\begin{aligned} x_n &= b_n s_{n-1} e^{-jn\pi/4} \\ &= b_n e^{-j\pi/4} s_{n-1} e^{-j(n-1)\pi/4} = a_n x_{n-1}, \\ y_n &= (e^{j\phi} s_n + n_n) e^{-jn\pi/4} = e^{j\phi} x_n + w_n. \end{aligned} \quad (9)$$

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.

3. Detection and Decoding: Single-Carrier Case, Noniterative

We will start by considering the single-carrier case. For some single subcarrier, 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.

3.1. *Trellis Representation, Subtrellises, Decomposition.* In this section, we will focus on noniterative 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

$$\Pr\{z_0 = e^{j\pi l/16}\} = \frac{1}{32}, \quad \text{for } l = 0, 1, \dots, 31, \quad (10)$$

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

$$\begin{aligned} z_n &= a_n z_{n-1}, \\ \text{where } \Pr\{a_n = e^{jp\pi/2}\} &= \frac{1}{4}, \quad \text{for } p = 0, 1, 2, 3. \end{aligned} \quad (11)$$

The variables z_n for $n = 0, 1, \dots, N$ can now be regarded as states in a trellis, and the *independent uniformly distributed (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 Figure 4.

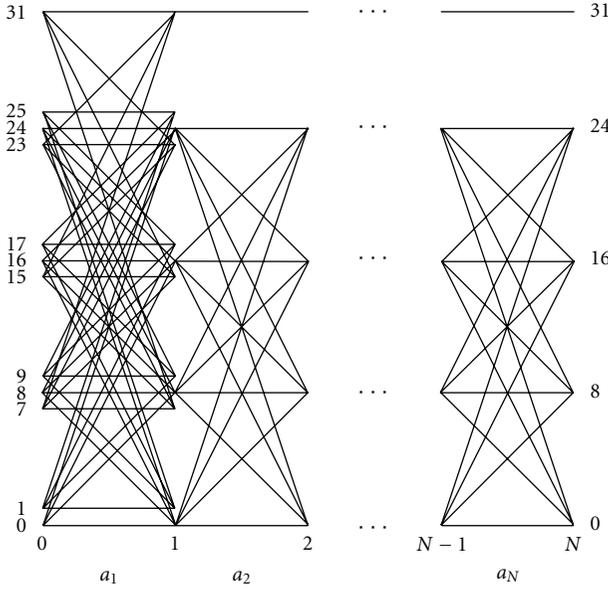


FIGURE 4: Trellis representation of the states z_0, z_1, \dots, z_N and the differentially encoded symbols a_1, a_2, \dots, a_N in the incoherent case. An edge between two subsequent states indicates that a transition between these states is possible. Note that the trellis can be decomposed into eight unconnected subtrellises.

If we would use the standard BCJR algorithm for computing the a posteriori symbol probabilities in the trellis in Figure 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 normalizations. (In 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.)

An important observation for our investigations is that the trellis can be seen to consist of eight subtrellises $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_7$, that are *not connected to each other*. A similar observation was made by Chen et al. [22]. We will discuss connections between our work on the trellis decomposition and that of [22] later.

Subtrellis \mathcal{T}_s consists of states $z_n \in \mathcal{Z}_s = \{e^{j\pi l/16}, l = s + 8p, p = 0, 1, 2, 3\}$, for $s = 0, 1, \dots, 7$. Figure 4 shows the entire subtrellis \mathcal{T}_0 , and the first section of subtrellis \mathcal{T}_1 and of subtrellis \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 subtrellis, we can write that

$$\gamma_n(z_n) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|y_n - z_n|^2}{2\sigma^2}\right). \quad (12)$$

3.2. Forward-Backward Algorithm, Subtrellises. In this subsection, we would like to focus on computing the a posteriori symbol probabilities $\Pr\{a_n \mid y_0, y_1, \dots, y_N\}$ 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 subtrellises. Computing the a posteriori probabilities for each subtrellis is simple and can be done without performing the BCJR algorithm, as was demonstrated by Colavolpe [5]. 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.

3.2.1. Forward Recursion. In our forward pass, we focus on subtrellis \mathcal{T}_s , for some $s \in \{0, 1, \dots, 7\}$. For that subtrellis we find out how to compute all the α 's in that subtrellis 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

$$\alpha_n(z_n) = \sum_{(z_{n-1}, a_n) \rightarrow z_n} \alpha_{n-1}(z_{n-1}) \frac{1}{4} \gamma_{n-1}(z_{n-1}), \quad (13)$$

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 $K_s(n) \triangleq \sum_{z_n \in \mathcal{Z}_s} (1/4) \gamma_n(z_n)$, then we have*

$$\alpha_n(z_n) = \frac{1}{32} \prod_{i=0}^{n-1} K_s(i). \quad (14)$$

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 $\alpha_{n-1}(z_{n-1}) = (1/32) \prod_{i=0}^{n-2} K_s(i)$ for $z_{n-1} \in \mathcal{Z}_s$, then from (13) we obtain

$$\begin{aligned} \alpha_n(z_n) &= \sum_{(z_{n-1}, a_n) \rightarrow z_n} \left(\frac{1}{32} \prod_{i=0}^{n-2} K_s(i) \right) \frac{1}{4} \gamma_{n-1}(z_{n-1}) \\ &= \frac{1}{32} \prod_{i=0}^{n-2} K_s(i) \sum_{z_{n-1} \in \mathcal{Z}_s} \frac{1}{4} \gamma_{n-1}(z_{n-1}) \\ &= \frac{1}{32} \prod_{i=0}^{n-1} K_s(i), \end{aligned} \quad (15)$$

for all $z_n \in \mathcal{Z}_s$. \square

3.2.2. Backward Recursion. Also in the backward pass we first focus only on subtrellis \mathcal{T}_s for some s . In this subtrellis, 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

$$\beta_n(z_n) = \sum_{a_{n+1}} \frac{1}{4} \gamma_n(z_n) \beta_{n+1}(z_n a_{n+1}), \quad (16)$$

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

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

$$\beta_n(z_n) = \gamma_n(z_n) \prod_{i=n+1}^N K_s(i), \quad (17)$$

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

$$\begin{aligned} \beta_n(z_n) &= \sum_{a_{n+1}} \frac{1}{4} \gamma_n(z_n) \left(\gamma_{n+1}(z_n a_{n+1}) \prod_{i=n+2}^N K_s(i) \right) \\ &= \gamma_n(z_n) \sum_{z_{n+1} \in \mathcal{Z}_s} \frac{1}{4} \gamma_{n+1}(z_{n+1}) \prod_{i=n+2}^N K_s(i) \\ &= \gamma_n(z_n) \prod_{i=n+1}^N K_s(i), \end{aligned} \quad (18)$$

for all $z_n \in \mathcal{Z}_s$. \square

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

$$\begin{aligned} \Pr\{a_n\} p(\mathbf{y} | a_n) &= \sum_{z_{n-1} \in \mathcal{Z}} \alpha_{n-1}(z_{n-1}) \frac{1}{4} \gamma_{n-1}(z_{n-1}) \beta_n(z_{n-1} a_n) \\ &= \sum_{s=0}^7 \sum_{z_{n-1} \in \mathcal{Z}_s} \alpha_{n-1}(z_{n-1}) \frac{1}{4} \gamma_{n-1}(z_{n-1}) \beta_n(z_{n-1} a_n) \\ &= \sum_{s=0}^7 \frac{1}{32} \prod_{i=0}^{n-2} K_s(i) \prod_{j=n+1}^N K_s(j) \\ &\quad \cdot \sum_{z_{n-1} \in \mathcal{Z}_s} \frac{1}{4} \gamma_{n-1}(z_{n-1}) \gamma_n(z_{n-1} a_n). \end{aligned} \quad (19)$$

If we consider the ‘‘middle’’ term in (19), then we see that

$$\begin{aligned} &\sum_{a_n} \left(\sum_{z_{n-1} \in \mathcal{Z}_s} \frac{1}{4} \gamma_{n-1}(z_{n-1}) \gamma_n(z_{n-1} a_n) \right) \\ &= \sum_{z_{n-1} \in \mathcal{Z}_s} \frac{1}{4} \gamma_{n-1}(z_{n-1}) \sum_{a_n} \gamma_n(z_{n-1} a_n) \\ &= 4K_s(n-1)K_s(n). \end{aligned} \quad (20)$$

From this we may conclude that

$$p(\mathbf{y}) = \sum_{s=0}^7 \Pr\{s\} p(\mathbf{y} | s) = \sum_{s=0}^7 \frac{1}{8} \prod_{i=0}^N K_s(i), \quad (21)$$

with

$$\begin{aligned} \Pr\{s\} &= \frac{1}{8}, \quad \text{for } s = 0, 1, \dots, 7, \\ p(\mathbf{y} | s) &= \prod_{i=0}^N K_s(i). \end{aligned} \quad (22)$$

Now observing that

$$\Pr\{s | \mathbf{y}\} = \frac{(1/8) \prod_{i=0}^N K_s(i)}{\sum_{s=0}^7 (1/8) \prod_{i=0}^N K_s(i)}, \quad (23)$$

$$\Pr\{a_n | \mathbf{y}, s\} = \frac{\sum_{z_{n-1} \in \mathcal{Z}_s} (1/4) \gamma_{n-1}(z_{n-1}) \gamma_n(z_{n-1} a_n)}{4K_s(n-1)K_s(n)}, \quad (24)$$

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

$$\Pr\{a_n | \mathbf{y}\} = \sum_{s=0}^7 \Pr\{s | \mathbf{y}\} \Pr\{a_n | \mathbf{y}, s\}. \quad (25)$$

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 subtrellis, where the weighting-coefficients are the a posteriori subtrellis probabilities $\Pr\{s | \mathbf{y}\}$. An a posteriori subtrellis 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 (25) has three tasks, first the eight weighting coefficients (23) have to be computed, then for each of the eight subtrellises 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 (25) has to be done. Computing the weighting coefficient requires for each subtrellis $s \in \{0, 1, \dots, 7\}$ 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 [5] technique to each subtrellis. As in Colavolpe each such a posteriori symbol probability is based on only two received symbols γ_{n-1} and γ_n as is shown in (24). This avoids the use of the BCJR method in full generality and leads to significant complexity reductions, that is, 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.

3.4. Dominant Subtrellis Approach. Equation (25) shows how the exact a posteriori symbol probabilities can be determined. If the a posteriori subtrellis probabilities are such that one of the probabilities dominates the other ones then weighting (25) can be approximated by

$$\Pr\{a_n | \mathbf{y}\} \approx \Pr\{a_n | \mathbf{y}, \hat{s}\}, \quad \text{with } \hat{s} = \arg \max_s \Pr\{s | \mathbf{y}\}. \quad (26)$$

Observe that this approach involves the computations of the a posteriori symbol probabilities, as described in (24), only for the dominant subtrellis \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.

3.5. *Simulations.* We use in our simulations, just like Peleg et al. [17], the de facto industry standard $R_c = 1/2$ 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 [1, Section 11.1.2, page 131]. The DAB, DAB+, and T-DMB bit-reversal time interleaver and block frequency interleaver are modeled by a bitwise 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 [17].

The demodulator calculates, for each OFDM-subcarrier, the a posteriori probability given by (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, that is,

$$\begin{aligned} b_1 b_2 & \quad 00 \quad 01 \quad 11 \quad 10, \\ a(b_1 b_2) & \quad 1 \quad e^{j\pi/2} \quad e^{j\pi} \quad e^{j3\pi/2}, \end{aligned} \quad (27)$$

that the desired metrics related to transmission n , that is, the log-likelihood ratios (LLRs) [30], can be expressed as

$$\lambda_n^1 = \ln \left(\frac{e^{m(\pi)} + e^{m(3\pi/2)}}{e^{m(0)} + e^{m(\pi/2)}} \right), \quad \lambda_n^2 = \ln \left(\frac{e^{m(\pi/2)} + e^{m(\pi)}}{e^{m(0)} + e^{m(3\pi/2)}} \right), \quad (28)$$

with symbol metric

$$m(\phi) = \ln \left(\Pr \{ a_n = e^{j\phi} \mid \mathbf{y} \} \right), \quad (29)$$

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

Figure 5 shows the Bit-Error Rate (BER) performance with the 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 $E_b/N_0 = 1/(2\sigma^2)$. The demodulator operates according to (25).

We will compare the performance of this demodulator with that of two well-known procedures described in the literature: firstly, to “classical” DQPSK [35, Section 4.5-5, page 224], that is, two-symbol differential detection (2SDD). This leads to a posteriori symbol probabilities as in (9) in Divsalar and Simon [3], that is, to

$$\Pr \{ a_n \mid y_n, y_{n-1} \} \propto I_0 \left(\frac{1}{\sigma^2} | y_n a_n^* + y_{n-1} | \right), \quad \text{for } a_n \in \mathcal{A}, \quad (30)$$

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 derotated, that is, $\tilde{\mathbf{y}} = \mathbf{y}e^{-j\phi}$. Then the a posteriori symbol probabilities are given by

$$\Pr \{ a_n \mid \tilde{\mathbf{y}} \} \propto \sum_{x_{n-1} \in \mathcal{A}} \exp \left(\frac{1}{\sigma^2} \Re \{ x_{n-1}^* (\tilde{y}_n a_n^* + \tilde{y}_{n-1}) \} \right), \quad (31)$$

for $a_n \in \mathcal{A}$,

as described by Colavolpe [5]. Note that (31) is similar to (24) for $s = 0$.

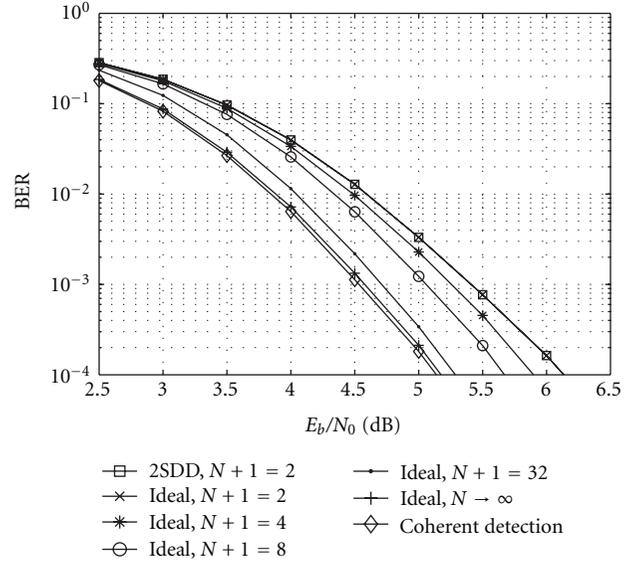


FIGURE 5: Bit-error performance for LLRs computed as in (25), that is, ideal LLRs, for different trellis-lengths.

The simulation results, which are shown in Figure 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, Figure 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, that is, 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 Figure 6, we turn to the dominant subtrellis 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 (25) and the approximated LLRs based on the dominant-subtrellis a posteriori probabilities specified in (26). It can be seen from Figure 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.

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 subcarrier 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

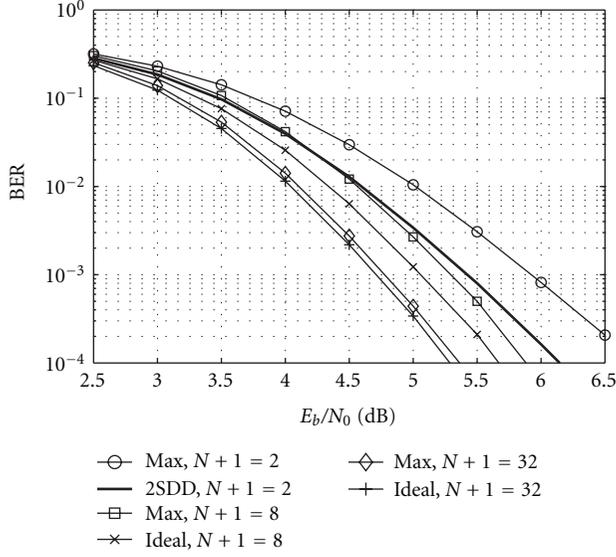


FIGURE 6: Bit-error performance for LLRs computed as in (25), that is, ideal LLRs, and approximated LLRs computed as in (26), for different trellis-lengths.

$T_c \approx 32T_s$, 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 Section 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 Figure 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) subcarriers 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 subcarriers, see Figure 3. This will be the subject of the next section.

4. Detection and Decoding: Multicarrier Case, Noniterative

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 (25), also

if we would concentrate on what was received over several subcarriers.

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

$$x_{m,n} = a_{m,n}x_{m,n-1} \quad (32)$$

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

$$y_{m,n} = e^{j\phi}x_{m,n} + w_{m,n}, \quad (33)$$

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 subcarrier 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 subtrellis probabilities:

$$\Pr\{s \mid \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} = \frac{\Pr\{s\}p(\mathbf{y}_1 \mid s)p(\mathbf{y}_2 \mid s) \cdots p(\mathbf{y}_M \mid s)}{\sum_{s=0}^7 \Pr\{s\}p(\mathbf{y}_1 \mid s)p(\mathbf{y}_2 \mid s) \cdots p(\mathbf{y}_M \mid s)}, \quad (34)$$

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

$$p(\mathbf{y}_m \mid s) = \prod_{i=0}^N K_{m,s}(i), \quad (35)$$

where $K_{m,s}(n) \triangleq \sum_{z_n \in \mathcal{Z}_s} (1/4)\gamma_{m,n}(z_n)$. 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 subtrellis, we can write that

$$\gamma_{m,n}(z_n) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|y_{m,n} - z_n|^2}{2\sigma^2}\right). \quad (36)$$

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

$$\Pr\{a_{m,n} \mid \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} = \sum_{s=0}^7 \Pr\{s \mid \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \Pr\{a_{m,n} \mid \mathbf{y}_m, s\}, \quad (37)$$

where

$$\Pr\{s \mid \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} = \frac{(1/8) \prod_{m=1}^M \prod_{i=0}^N K_{m,s}(i)}{\sum_{s=0}^7 (1/8) \prod_{m=1}^M \prod_{i=0}^N K_{m,s}(i)}, \quad (38)$$

$$\Pr\{a_{m,n} \mid \mathbf{y}_m, s\} = \frac{\sum_{z_{n-1} \in \mathcal{Z}_s} (1/4)\gamma_{m,n-1}(z_{n-1})\gamma_{m,n}(z_{n-1}a_n)}{4K_{m,s}(n-1)K_{m,s}(n)}, \quad (39)$$

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

This suggests that the demodulator first determines the a posteriori subtrellis probabilities (weighting coefficients) using (38), for which the first $8 \times M \times (N + 1)$ K -factors have

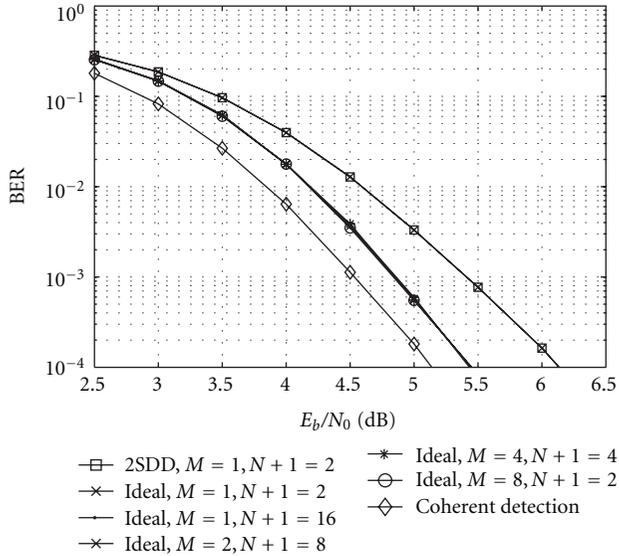


FIGURE 7: Bit-error performance with ideal LLRs for a decomposed multi-carrier trellis for different values of M and N but with a fixed block-size of $M(N + 1) = 16$.

to be computed. Using the weighting coefficients, the convex combination in (37) 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 [5] method for each subcarrier and for each subtrellis, where again such a posteriori symbol probability is based on only the two received symbols $y_{m,n-1}$ and $y_{m,n}$ as is shown in (39). Again the BCJR method in full generality is not needed, and the number of required multiplications and normalizations per trellis section is the same as in the single carrier case.

Equation (37) shows how the exact a posteriori symbol probabilities can be determined. Just like in the single-carrier case, if the a posteriori subtrellis probabilities are such that one of the probabilities dominates the other, ones then weighting (37) can be approximated as follows:

$$\Pr\{a_{m,n} | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \approx \Pr\{a_{m,n} | \mathbf{y}_m, \hat{s}\}, \quad (40)$$

with

$$\hat{s} = \arg \max_s \Pr\{s | \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\}. \quad (41)$$

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

4.2. Simulations. In the previous section, we analyzed and simulated the single subcarrier 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 $E_b/N_0 = 1/(2\sigma^2)$. The BER performance for the ideal LLRs, based on a posteriori probabilities computed as in (37), is shown in Figure 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 (37). The performance of 2SDD and coherently detected DE-QPSK are shown as reference curves.

From Figure 7, it can be observed that a 2D decomposition with a shortest possible trellis-length of $N + 1 = 2$ and $M = 8$ adjacent subcarriers performs identical to the largest trellis-length $N + 1 = 16$ and $M = 1$ subcarrier that is, the single-carrier case. Intermediate cases also have an identical performance.

We do not show the results of the dominant subtrellis approach for the multi-carrier case here, since these results are identical to the corresponding results for the single-carrier case shown in Figure 6 in the previous section.

4.3. Conclusion Noniterative Decoding. Our investigations for the noniterative 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 subcarriers. A next question is whether we can do better than this. In the literature, see, for example, Peleg et al. in [17] and Chen et al. [22], 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.

5. Detection and Decoding: Single-Carrier Case, Iterative

In the following two sections, we consider iterative decoding procedures. Peleg and Shamai [13] 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.

5.1. Serial Concatenation. 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 Figure 8. Serially concatenated turbo codes were proposed by Benedetto and Montorsi [10] and later investigated in more detail in Benedetto et al. [12]. Iterating between the DPSK-demodulator and convolutional decoder for the incoherent case was first suggested (for a single carrier) by Peleg and Shamai [13]. Hoeher and Lodge [20] 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. [22].

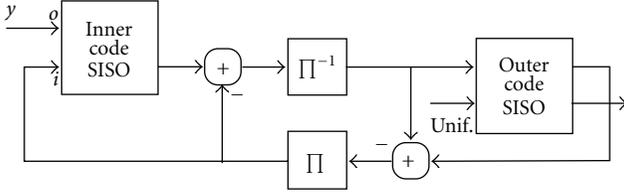


FIGURE 8: Structure of the receiver.

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.

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. [17] we focus on the entire trellis \mathcal{T} . Note, however, that our trellis is different from that of [17], in which tracking of small channel phase variations is made possible by adding “side-step” transitions. We do not have such transitions in our trellis and, therefore, our trellis can be decomposed in eight unconnected subtrellises. In the next subsection, we take advantage of this decomposition; however, first we will consider the undecomposed 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

$$\alpha_n(z_n) = \sum_{(z_{n-1}, a_n) \rightarrow z_n} \alpha_{n-1}(z_{n-1}) \Pr\{a_n\} \gamma_{n-1}(z_{n-1}), \quad (42)$$

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

$$\beta_n(z_n) = \sum_{a_{n+1}} \gamma_n(z_n) \Pr\{a_{n+1}\} \beta_{n+1}(z_n a_{n+1}), \quad (43)$$

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

$$\begin{aligned} & \Pr\{a_n\} p(\mathbf{y} | a_n) \\ &= \sum_{z_{n-1} \in \mathcal{Z}} \alpha_{n-1}(z_{n-1}) \Pr\{a_n\} \gamma_{n-1}(z_{n-1}) \beta_n(z_{n-1} a_n). \end{aligned} \quad (44)$$

This expression also tells us how the resulting extrinsic information can be determined. It can be checked, see Benedetto and Montorsi [10], that multiplying by the factors $\Pr\{a_n\}$ in the a posteriori information (44) 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 Section 5.5.

Using the standard BCJR algorithm for computing the extrinsic symbol probabilities in the trellis in Figure 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.

5.3. Trellis Decomposition. Here we investigate whether we can decompose the entire trellis for the case where the a priori probabilities are nonuniform. We are interested in decomposing (44) in such a way that we can write

$$\Pr\{a_n | \mathbf{y}\} = \sum_s \Pr\{s, a_n | \mathbf{y}\} = \sum_s \Pr\{s | \mathbf{y}\} \Pr\{a_n | \mathbf{y}, s\}, \quad (45)$$

for all $a_n \in \mathcal{A}$. The question now is how to compute the a posteriori subtrellis probabilities $\Pr\{s | \mathbf{y}\}$ for $s = 0, 1, \dots, 7$.

It can be shown that

$$\Pr\{s, \mathbf{y}\} = \frac{\sum_{z_0 \in \mathcal{Z}_s} \beta_0(z_0)}{32}, \quad (46)$$

and therefore

$$\Pr\{s | \mathbf{y}\} = \frac{\sum_{z_0 \in \mathcal{Z}_s} \beta_0(z_0)/32}{\sum_{s'=0}^7 \sum_{z_0 \in \mathcal{Z}_{s'}} \beta_0(z_0)/32}. \quad (47)$$

Now for each subtrellis, we can determine the a posteriori symbol probabilities using

$$\begin{aligned} \Pr\{a_n | \mathbf{y}, s\} &\propto \Pr\{a_n, s\} p(\mathbf{y} | a_n, s) \\ &= \Pr\{a_n, s\} \sum_{z_{n-1} \in \mathcal{Z}_s} \alpha_{n-1}(z_{n-1}) \gamma_{n-1}(z_{n-1}) \beta_n(z_{n-1} a_n), \end{aligned} \quad (48)$$

and by omitting the factor $\Pr\{a_n, s\}$ in (48), 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 (43), $\gamma_n(z_n)$ can be put in front of the summation sign. Then for all subtrellises \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 subtrellis (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 (45) 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.

5.4. Dominant Subtrellis Approaches. To achieve a complexity reduction, we investigate a method that is based on finding, at the start of a new iteration, the dominant subtrellis first and then do the forward-backward processing for demodulation only in this dominant subtrellis.

Finding the dominant subtrellis for an iteration is done based on the a posteriori subtrellis probabilities $\Pr\{s | \mathbf{y}\}$ that are computed before starting this iteration. Now assuming that one of the a posteriori subtrellis probabilities dominates the other ones, we can write

$$\Pr\{a_n | \mathbf{y}\} \approx \Pr\{a_n | \mathbf{y}, \hat{s}\}, \quad \text{with } \hat{s} = \arg \max_s \Pr\{s | \mathbf{y}\}. \quad (49)$$

This approach involves the computations of the a posteriori symbol probabilities (and corresponding extrinsic information) as described in (42), (43), and (44) only for the dominant subtrellis \hat{s} . Computing the a posteriori subtrellis 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 subtrellis $\mathcal{T}_{\hat{s}}$, 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 subtrellis ($4 \times 4 \times 2 = 32$ 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 subtrellis only once, before starting with the iterations. Since before starting the iterations the a priori probabilities $\Pr\{a_n\} = 1/4$, that is, are all equal, the analysis in Section 3.4 applies. The a posteriori subtrellis probabilities can be computed as in (23). Now we do the iterations only in the subtrellis 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 subtrellises.

5.5. Simulations. We simulated the Peleg method described in [17] and determined the BER versus the signal-to-noise ratio $E_b/N_0 = 1/(2\sigma^2)$. This BER performance is shown in Figure 9 for trellis lengths practically infinite, that is, $N \rightarrow \infty$ and ideal LLRs are based on the a posteriori probability given by (44). 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 Figure 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 (45). As expected, the performance of this approach shows no differences with the Peleg approach in (44). From Figure 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. Figure 9 also shows that improvement decreases with the number of iterations

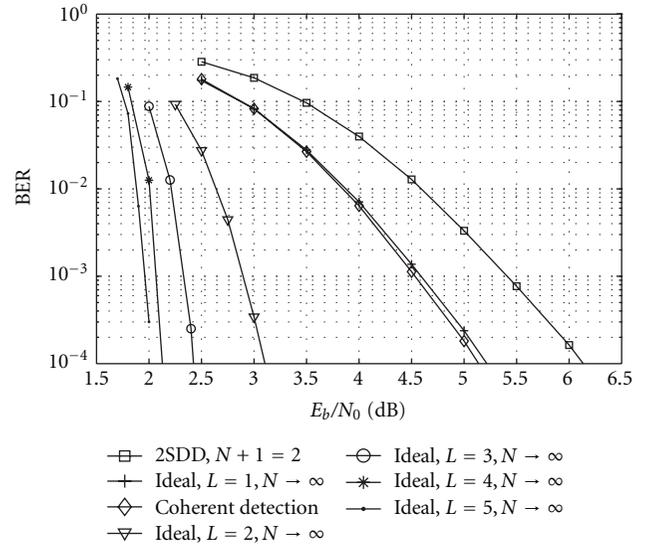


FIGURE 9: Bit error performance of the Peleg method for trellis length $N \rightarrow \infty$ and up to $L = 5$ iterations.

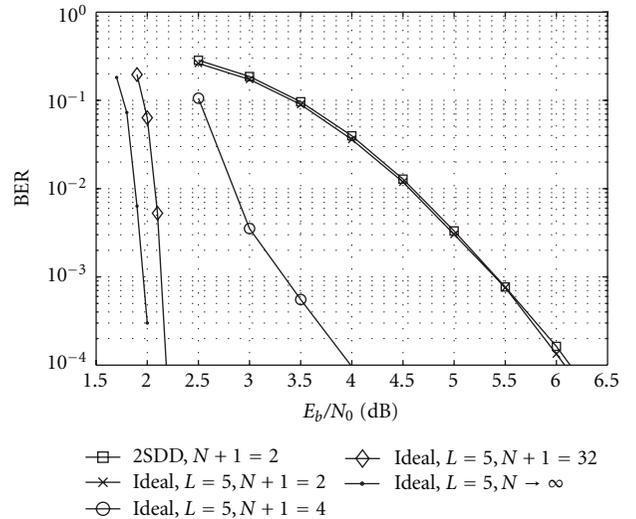


FIGURE 10: Bit-error performance for the Peleg method for different values of N and $L = 5$ iterations.

and that the first iteration yields the largest improvement. Similar results were obtained by Peleg et al. [17].

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 Figure 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 (44) or (45) and the approximated LLRs based on the a posteriori (extrinsic) probability given by (49). The results are shown

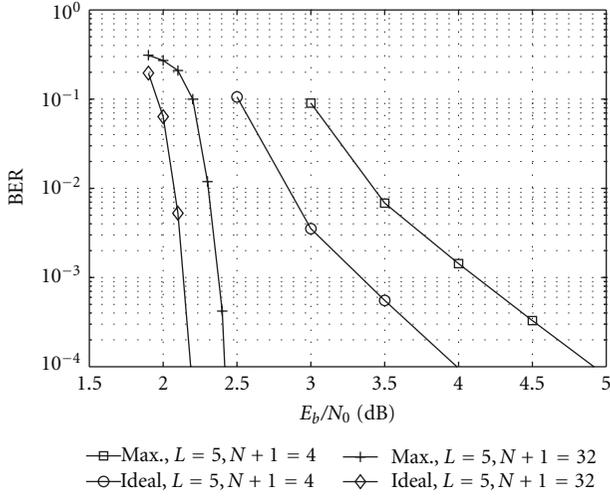


FIGURE 11: Bit-error performance with dominant subtrellis approach for different values N and $L = 5$ iterations.

in Figure 11. We can conclude from Figure 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 subtrellis before starting with the iteration process, is less than 0.3 dB.

6. Detection and Decoding: Multicarrier Case, Iterative

6.1. Trellis Decomposition. Just like in the noniterative multicarrier case, we do the processing based on trellis decomposition and focus on the computation of the a posteriori subtrellis probabilities:

$$\Pr\{s \mid \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} = \frac{\Pr\{s\} p(\mathbf{y}_1 \mid s) p(\mathbf{y}_2 \mid s) \dots p(\mathbf{y}_M \mid s)}{\sum_{s=0}^7 \Pr\{s\} p(\mathbf{y}_1 \mid s) p(\mathbf{y}_2 \mid s) \dots p(\mathbf{y}_M \mid s)}. \quad (50)$$

Note that $\Pr\{s\} = 1/8$ for $s = 0, 1, \dots, 7$ and therefore it follows from (46) that

$$p(\mathbf{y}_m \mid s) = \frac{\sum_{z_0 \in \mathcal{Z}_s} \beta_{m,0}(z_0)}{4}, \quad (51)$$

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 (37) that is,

$$\Pr\{a_{m,n} \mid \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} = \sum_{s=0}^7 \Pr\{s \mid \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \Pr\{a_{m,n} \mid \mathbf{y}_m, s\}, \quad (52)$$

where $\Pr\{a_{m,n} \mid \mathbf{y}_m, s\}$ is computed as given by (48) 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.

This suggests that, for each iteration, the demodulator first determines the a posteriori subtrellis probabilities using (50), for which first a backward pass in each of the M trellises corresponding to the subcarriers is needed.

Using the weighting coefficients, the convex combination in (52) leads to the a posteriori symbol probabilities. Finding the a posteriori symbol probabilities $\Pr\{a_{m,n} \mid \mathbf{y}_m, s\}$ should be done in the standard way, taking into account that the backward passes were already carried out.

6.2. Dominant Subtrellis Approach. Equation (52) 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 subtrellis probabilities are such that one of the probabilities dominates the other ones, then convex combination (52) can be approximated as follows:

$$\Pr\{a_{m,n} \mid \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\} \approx \Pr\{a_{m,n} \mid \mathbf{y}_m, \hat{s}\}, \quad (53)$$

with

$$\hat{s} = \arg \max_s \Pr\{s \mid \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_M\}. \quad (54)$$

Again this approach involves, in each iteration, the computations of the a posteriori symbol probabilities only for the dominant subtrellis \hat{s} .

If we compute the dominant subtrellis only before the start of the iteration process, we obtain a significant complexity reduction since the analysis in Section 3.4 applies. Moreover, all iterations are done in the initially chosen subtrellis. The methods described here will be evaluated in the next subsection.

6.3. Simulations. We have seen before that in the noniterative 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 subcarriers M if the frequency selectivity allows this. Note that keeping $N + 1$ small also has advantages related to service symbol processing [1]. Here the situation is slightly different as is demonstrated in Figure 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 (52) and the approximated LLRs based on the a posteriori (extrinsic) probabilities given by (53), see Figure 13. We can observe from Figure 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 subtrellis before starting to iterate, is roughly 0.3 dB.

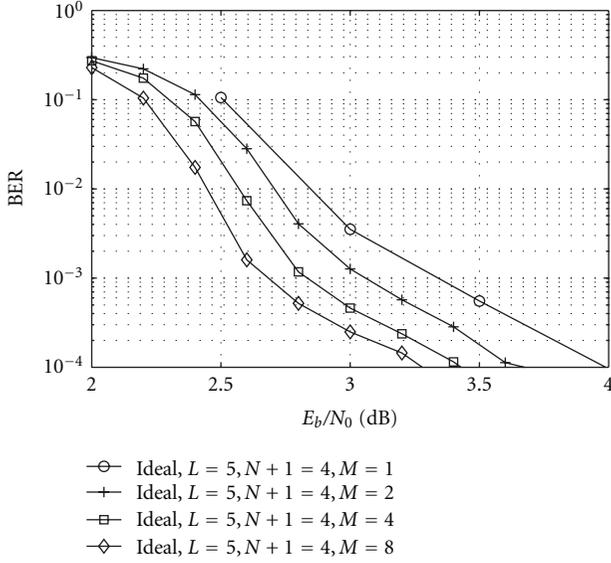


FIGURE 12: Bit-error performance for the multi-carrier case for different values M , where $N + 1 = 4$ and for $L = 5$ iterations.

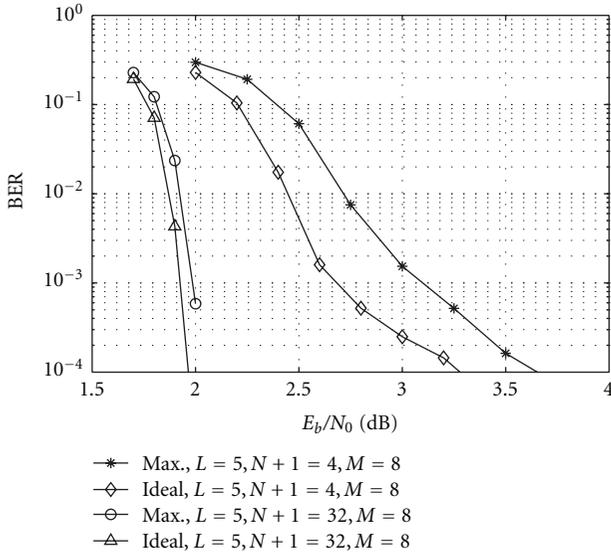


FIGURE 13: Bit-error performance with dominant subtrellis approach for different values N , $M = 8$ and $L = 5$ iterations.

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 (Typical Urban 6 taps) channel model defined in [36], which is commonly used to test DAB, DAB+, or T-DMB transmission. Two maximum Doppler frequencies are chosen, that is, $f_d = 10$ and 20 Hz, representing DAB transmission (in Band-III) movement speeds between transmitter and receiver of ≈ 45 and ≈ 90 km/h, respectively.

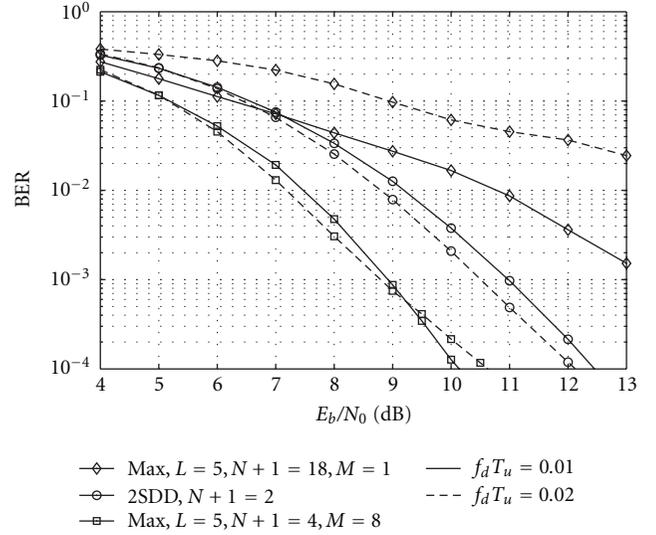


FIGURE 14: Bit error performance for the TU-6 COST-207 channel with the solid lines for $f_d T_u = 0.01$, the dashed lines for $f_d T_u = 0.02$, $N + 1 = (4, 18)$, $M = (8, 1)$, and $L = 5$ iterations.

We use our methods for DAB transmission in Mode-I, where the inverse subcarrier spacing $T_u = 1$ ms and where the cyclic-prefix period $T_{cp} = 246 \mu\text{s}$ [1].

Now, with these settings, the *normalized Doppler rate* $f_d T_u$ is 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 [2], that is, the channel impulse response length τ_m must satisfy $\tau_m \leq T_{cp}$. Within the DAB-system, $T_{cp} \triangleq (63/256)T_u < T_u/4$ [1] and therefore the coherence-bandwidth $B_c \approx 1/\tau_m > 4(1/T_u)$, which is at least 4 OFDM-subcarriers. For Doppler frequency $f_d = 20$ Hz, the coherence-time $T_c \approx 1/(2f_d) = 25$ 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. [22], that is,

$$\widehat{|h|^2} = \max \left(\frac{1}{M(N+1)} \sum_{m=1, M, n=0, N} |y_{m,n}|^2 - 2\sigma^2, 0 \right). \quad (55)$$

The results of our simulations with the TU-6 model are shown in Figure 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 subtrellis approach, where we have chosen the dominant subtrellis 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 Figure 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.

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.

We have used trellis decomposition methods that allows us to estimate the unknown channel-phase modulo $\pi/2$. This channel phase relates to subtrellises of which we can determine the a posteriori probabilities. Using these probabilities we can weigh the contributions of all the subtrellises to compute the a posteriori symbol probabilities. We can also use these probabilities to chose a dominant subtrellis for providing us with these a posteriori symbol probabilities. Working with dominant subtrellises results in significant complexity reductions. A second important advantage of trellis-decomposition is that it allows us to process in an efficient way several subcarriers simultaneously.

We have first investigated noniterative methods. The advantage of these methods is that forward-backward procedures turned out to be extremely simple since we could use Colavolpe processing [5]. The drawback of these noniterative 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 noniterative AWGN case show that (a) trellis-lengths of $N + 1 \geq 32$ are required and (b) that 2D dominant subtrellis 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 subtrellis 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 subcarrier) no reliable communication is possible, but that (b) with a modest trellis-length $N + 1 = 4$ and $M = 8$ subcarriers, 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.

Acknowledgments

The authors would like to acknowledge Professor J. W. M. Bergmans, the management and the technical staff of Catena Radio Design B. V., and NXP-Research Eindhoven for their support to accomplish this work. Moreover, They thank the anonymous reviewers for their comments and valuable suggestions.

References

- [1] 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), v1.4.1, June 2006.
- [2] G. Zimmermann, M. Rosenberger, and S. Dostert, "Theoretical bit error rate for uncoded and coded data transmission in digital audio broadcasting," in *Proceedings of the IEEE International Conference on Communications : Converging Technologies for Tomorrow's Applications (ICC '96)*, vol. 1, pp. 297–301, June 1996.
- [3] D. Divsalar and M. K. Simon, "Multiple-symbol differential detection of MPSK," *IEEE Transactions on Communications*, vol. 38, no. 3, pp. 300–308, 1990.
- [4] G. E. Bottomley, H. Arslan, R. Ramesh, and G. Brismark, "Coherent MAP detection of DQPSK bits," *IEEE Communications Letters*, vol. 4, no. 11, pp. 354–356, 2000.
- [5] G. Colavolpe, "Classical coherent receivers for differentially encoded M-PSK are optimal," *IEEE Communications Letters*, vol. 8, no. 4, pp. 211–213, 2004.
- [6] W. J. Van Houtum and F. M. J. Willems, "A-posteriori symbol probabilities and log-likelihood ratios for coherently detected $\pi/4$ -DE-QPSK," *IEEE Communications Letters*, vol. 15, no. 2, pp. 160–162, 2011.
- [7] D. Divsalar, M. K. Simon, and M. Shahshahani, "The performance of trellis-coded MDPSK with multiple symbol detection," *IEEE Transactions on Communications*, vol. 38, no. 9, pp. 1391–1403, 1990.
- [8] 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," *IEEE Transactions on Communications*, vol. 42, no. 1, pp. 76–89, 1994.
- [9] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near shannon limit errorcorrecting coding and decoding: turbo-codes. 1," in *Proceedings of the IEEE International Conference on Communications (ICC '93)*, vol. 2, pp. 1064–1070, Geneva, Switzerland, May 1993.
- [10] S. Benedetto and G. Montorsi, "Iterative decoding of serially concatenated convolutional codes," *Electronics Letters*, vol. 32, no. 13, pp. 1186–1188, 1996.
- [11] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," Tech. Rep. 42-126, TDA Progress, August 1996, pp.1–26, .
- [12] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Transactions on Information Theory*, vol. 44, no. 3, pp. 909–926, 1998.

- [13] M. Peleg and S. Shamai, "Iterative decoding of coded and interleaved noncoherent multiple symbol detected DPSK," *Electronics Letters*, vol. 33, no. 12, pp. 1018–1020, 1997.
- [14] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "A soft-input soft-output APP module for iterative decoding of concatenated codes," *IEEE Communications Letters*, vol. 1, no. 1, pp. 22–24, 1997.
- [15] S. ten Brink, J. Speidel, and R. H. Yan, "Iterative demapping for QPSK modulation," *Electronics Letters*, vol. 34, no. 15, pp. 1459–1460, 1998.
- [16] I. D. Marsland and P. T. Mathiopoulos, "On the performance of iterative noncoherent detection of coded M-PSK signals," *IEEE Transactions on Communications*, vol. 48, no. 4, pp. 588–596, 2000.
- [17] M. Peleg, S. Shamai, and S. Galán, "Iterative decoding for coded noncoherent MPSK communications over phase-noisy AWGN channel," *IEE Proceedings: Communications*, vol. 147, no. 2, pp. 87–95, 2000.
- [18] G. Colavolpe, G. Ferrari, and R. Raheli, "Noncoherent iterative (turbo) decoding," *IEEE Transactions on Communications*, vol. 48, no. 9, pp. 1488–1498, 2000.
- [19] P. Hoeher and J. Lodge, "Iterative differential PSK demodulation and channel decoding," in *Proceedings of the IEEE International Symposium on Information Theory*, p. 425, August 1998.
- [20] P. Hoeher and J. Lodge, "'Turbo DPSK': iterative differential PSK demodulation and channel decoding," *IEEE Transactions on Communications*, vol. 47, no. 6, pp. 837–843, 1999.
- [21] L. H. J. Lampe and R. Schober, "Iterative decision-feedback differential demodulation of bit-interleaved coded MDPSK for flat Rayleigh fading channels," *IEEE Transactions on Communications*, vol. 49, no. 7, pp. 1176–1184, 2001.
- [22] R.-R. 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," *IEEE Transactions on Communications*, vol. 51, no. 10, pp. 1676–1689, 2003.
- [23] V. Pauli, L. Lampe, and R. Schober, "'Turbo DPSK' using soft multiple-symbol differential sphere decoding," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1385–1398, 2006.
- [24] S. ten Brink, F. Sanzi, and J. Speidel, "Two-dimensional iterative APP channel estimation and decoding for OFDM systems," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '00)*, vol. 2, pp. 741–745, 2000.
- [25] S. Y. Park, Y. G. Kim, C. G. Kang, and D. E. Kang, "Iterative receiver with joint detection and channel estimation for OFDM system with multiple receiver antennas in mobile radio channels," in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '01)*, vol. 5, pp. 3085–3089, 2001.
- [26] F. Sanzi and M. C. Necker, "Totally blind APP channel estimation with higher order modulation schemes," in *Proceedings of the 58th IEEE Vehicular Technology Conference (VTC '03)*, vol. 2, pp. 1167–1171, October 2003.
- [27] T. May and H. Rohling, "Turbo decoding of convolutional codes in differentially modulated OFDM transmission systems," in *Proceedings of the 49th IEEE Vehicular Technology Conference (VTC '99)*, vol. 3, pp. 1891–1895, July 1999.
- [28] H. Rohling and T. May, "OFDM systems with differential modulation schemes and turbo decoding techniques," in *Proceedings of the International Zurich Seminar on Broadband Communications*, pp. 251–255, 2000.
- [29] P. Haase and H. Rohling, "Iterative detection of differentially modulated APSK signals in an OFDM transmission system," in *Proceedings of the The 5th International Symposium on Wireless Personal Multimedia Communications*, vol. 2, pp. 711–714, October 2002.
- [30] J. Hagenauer and P. Hoeher, "Viterbi algorithm with soft-decision outputs and its applications," in *Proceedings of the IEEE Global Telecommunications Conference and Exhibition (GLOBECOM '89)*, vol. 3, pp. 1680–1686, 1989.
- [31] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (Corresp.)," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, 1974.
- [32] W. J. van Houtum and F. M. J. Willems, "Joint and iterative detection and decoding of differentially encoded COFDM systems," in *Proceedings of the 17th IEEE International Conference on Telecommunications (ICT '10)*, pp. 36–43, April 2010.
- [33] M. Franceschini, G. Ferrari, and R. Raheli, "Detection by multiple trellises," *IEEE Transactions on Communications*, vol. 57, no. 3, pp. 726–737, 2009.
- [34] J. Hagenauer, N. Seshadri, and C. E. W. Sundberg, "The performance of rate-compatible punctured convolutional codes for digital mobile radio," *IEEE Transactions on Communications*, vol. 38, no. 7, pp. 966–980, 1990.
- [35] J. G. Proakis and M. Salehi, *Digital Communications*, Higher Education Series, McGraw-Hill, 5th edition, 2008.
- [36] "COST 207, digital land mobile radio communications," Tech. Rep., Office for Official Publications of the European Communities, Luxembourg, UK, 1989.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

