Chapter 7

LDPC Codes and Differential Modulations

7.1 Introduction

This chapter investigates the use of differentially encoded (DE) modulations concatenated with low-density parity-check (LDPC) codes. The chapter is logically divided in two parts.

In the first part of this chapter (from Section 7.2 till Section 7.5), we discuss the approach to the design of DE-LDPC coded scheme originally proposed in [79]. Adopting the optimization technique described in Section 5.6, we show a method to design good LDPC codes for DE modulations. We analyze the optimized codes, gaining insights into their graph structures and highlighting the differences between LDPC codes for DE modulations and standard LDPC codes, i.e., optimized for transmission over a *memoryless* channel. We consider the concatenation of an LDPC code with a differential modulator for both phase shift keying (PSK) and quadrature amplitude modulation (QAM). At the receiver side, we make an extrinsic information transfer (EXIT) chart-based system performance evaluation, as described in Section 5.4. We compare the performance of codes optimized for DE modulations with the performance of standard LDPC codes. We show that LDPC codes optimized for DE modulations significantly outperform standard LDPC codes when concatenated with DE modulations. Vice versa, the obtained optimized codes are shown to be tailored *specifically* for the particular DE modulation format and the considered receiver scheme: in other words, while they perform well if used jointly with DE, they perform poorly with memoryless modulation schemes. This will be shown to depend on the presence of a large fraction of degree-2

M. Franceschini, G. Ferrari and R. Raheli, *LDPC Coded Modulations,*
DOI: 10.1007/978-3-540-69457-1_7, © Springer-Verlag Berlin Heidelberg 2009

variable nodes.

In the second part of this chapter (Section 7.6 and Section 7.7), we discuss an iterative detection/decoding scheme based on the concatenation of an outer soft-output differential detector and an inner LDPC decoder. The outer detector makes use of a detection strategy, referred to as *detection by multiple trellises* and originally introduced in [108], to perform trellis-based detection over realistic channels. More precisely, we consider channels with unknown parameters and apply the concept of detection by multiple trellises using parallel forward-backward (FB) algorithms (see Chapter 2 for more details). The key idea of our approach consists, *first*, in properly quantizing the channel parameters and, *then*, in considering replicated *coherent* FB algorithms operating on parallel trellises, one per hypothetical quantized value.

7.2 Serial Concatenation of LDPC Codes with PSK and DE-PSK

Consider the transmission side of an LDPC coded modulation scheme described in Chapter 5 and shown in Figure 7.1. As a representative coded modulation (CM) for the transmission system in Figure 7.1, we first consider DE-PSK. For coherent detection, the corresponding CM-SISO module implements, with very low complexity, the FB algorithm. The performance of the considered systems, first studied through an EXIT chart-based analysis, is evaluated in terms of bit error rate (BER) versus bit SNR E_b/N_0, where E_b is the average received energy per bit and N_0 is the one-sided AWGN power spectral density. In all the considered simulations and optimizations, Gray mapping over the PSK constellation is used.

In Figure 7.2 (a), EXIT charts are shown for a regular rate-1/2 $(3, 6)$ LDPC code, characterized by the degree distributions $\lambda(x) = x^2$ and $\rho(x) = x^5$. This code without DE, mapped to a quaternary PSK (QPSK) modulation format, is characterized by a good tradeoff, between complexity and performance, for transmission over an AWGN channel. The EXIT curves are computed

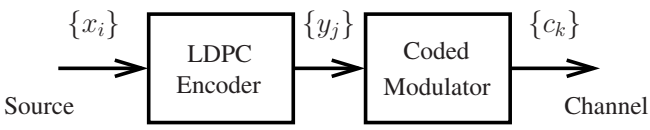

Figure 7.1: System model: transmitter side.

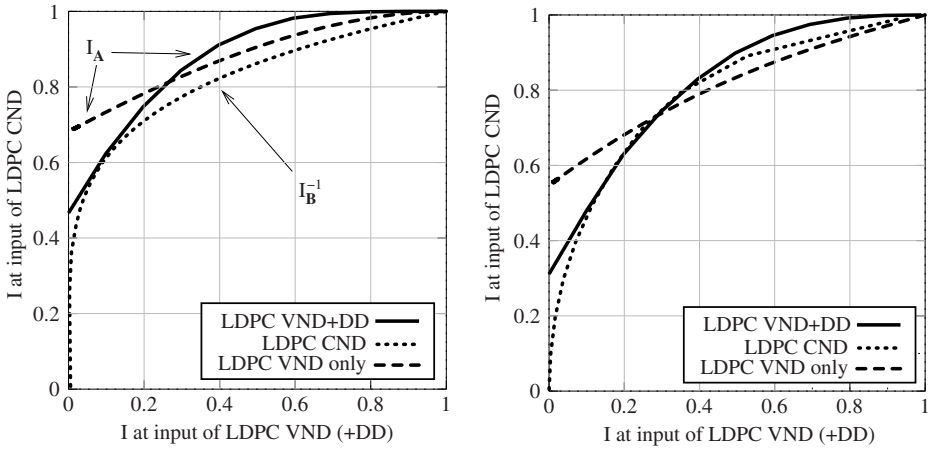

Figure 7.2: EXIT chart-based analysis of a system with serial concatenation of an LDPC code and QPSK: (a) EXIT chart of a $(3, 6)$ regular LDPC code concatenated with a QPSK with DE ($E_b/N_0 = 2.5$dB: tunnel is near *pinch-off*) and QPSK without DE (tunnel is *open*); (b) EXIT chart of an optimized rate 1/2 LDPC code concatenated with a QPSK with DE ($E_b/N_0 = 0.8$dB: tunnel is at *pinch-off*) and QPSK without DE (tunnel is *closed*).

at $E_b/N_0 = 2.5$ dB: the solid curve is the EXIT curve of block **A** (LDPC variable node detector, VND, and differential detector, DD) and the dotted curve is the EXIT curve of block **B** (LDPC check node detector, CND)—for more details on the VND and CND, see Section 5.3. Note that the SNR does not influence the EXIT curve relative to the LDPC CND (the dotted one in Figure 7.2). It is easy to see that the system is at pinch-off: convergence at this and lower values of E_b/N_0 is not possible. The dashed curve represents the EXIT curve of the single LDPC VND: this corresponds to the QPSK system *without* DE, i.e., LDPC BICM. It can be immediately seen that at $E_b/N_0 = 2.5$ dB the tunnel, relative to a transmission scheme without DE, is open. The EXIT chart-based analysis then predicts that, for a bit SNR slightly lower than 2.5 dB, the system with DE does not converge as opposed to the system without DE, which instead converges.

We now apply the optimization technique presented in Section 5.6, forcing the optimization algorithm to use check and variable nodes with specified degree values. As representative values, check nodes of degree 3, 4, 8, and 15, and variable nodes of degree 2, 3 and 4 have been used (these are reasonable

choices, but the approach is general). After a few steps, the optimized degree distributions converge to the following:

$$\begin{array}{llll} \rho_3 = 0.3157 & \rho_4 = 0.2259 & \rho_8 = 0.0273 & \rho_{15} = 0.4311 \\ \lambda_2 = 0.5473 & \lambda_3 = 0.0116 & \lambda_4 = 0.4411. & \end{array}$$

Figure 7.2 (b) shows the EXIT curves for this optimized code ensemble for $E_b/N_0 = 0.8$ dB: the solid and dashed curves correspond to block **A** and the dotted curve to block **B**. It is immediate to recognize that the tunnel is at pinch-off. The dashed curve in Figure 7.2 (b) is the EXIT curve of the LDPC VND only (i.e., without DD): the tunnel is "heavily" closed, predicting that the system with DE should perform significantly better than the single LDPC code without DE. Note that the convergence SNR threshold predicted by the results in Figure 7.2 (b) is around 0.9 dB.

In order to closely approximate the degree distributions obtained with the optimization technique, we chose to design LDPC codes with codeword length equal to 6000 binary symbols. In Figure 7.3, the performance of both optimized and regular $(3, 6)$ LDPC codes with and without DE is shown. As introduced in Section 5.3, we denote the maximum number of iterations between blocks **A** and **B** as N_i, and to the maximum number of standard LDPC final decoding iterations (between VND and CND) as N_{LDPC}. For DE systems, these maximum numbers of iterations are $N_i = 30$ and $N_{\text{LDPC}} = 30$, whereas for non DE systems a maximum number of 100 standard LDPC iterations is allowed—this makes the complexities of the two different systems very similar. It can be observed that, for a regular $(3, 6)$ LDPC code, while good performance is obtained without DE (curve marked with diamonds), the introduction of DE shifts the BER curve to the right, with an SNR loss of about 1.2 dB (curve marked with squares). When the LDPC code is optimized for DE, i.e., block **A** includes a CM-SISO module based on the FB algorithm relative to the DE modulator, it is possible to see the inversion of performance between the system with and without DE, as predicted by the EXIT chart-based analysis. In other words, the use of the LDPC code optimized for DE, in the system with DE (curve marked with triangles) leads to good performance, i.e., it behaves as the regular $(3, 6)$ LDPC code without DE (curve with diamonds). On the other hand, the use of the LDPC code optimized for DE, in the system *without* DE, i.e., LDPC BICM (curve with crosses) causes unsatisfactory performance, with an SNR loss of more than 2 dB at a BER equal to 10^{-3}, and low curve slope.

It is also possible to use the CM-SISO module, i.e., the DD, only once and then pass the obtained reliability values to a standard LDPC decoder: the

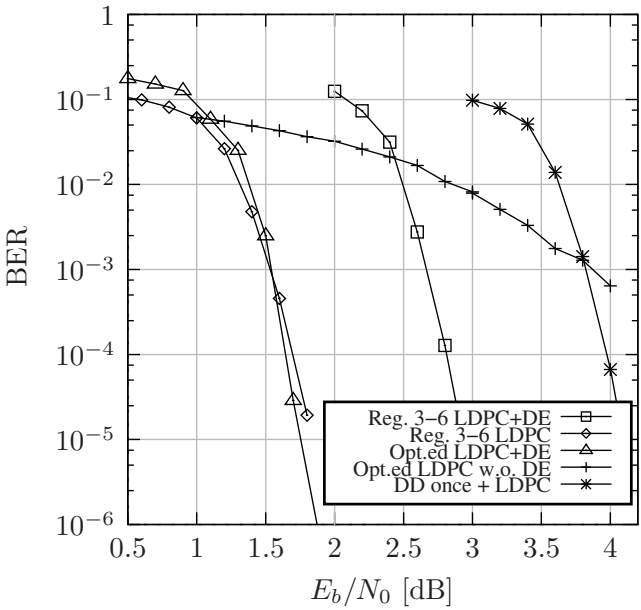

Figure 7.3: BER performance of the four communication schemes considered in Figure 7.2 (a) and (b).

corresponding performance, obtained considering a maximum number of 100 LDPC iterations and using the previous regular $(3, 6)$ LDPC code, is given by the curve marked with stars. It is easy to recognize that the absence of iterations between the CM-SISO block and the LDPC VND leads to a loss of about 1.2 dB with respect to the system with iterative detection/decoding. This can be interpreted noting that the standard LDPC decoder is based on the assumption that a memoryless channel is used, as discussed in Chapters 3–5. When a DE (and the corresponding CM-SISO module) is present, however, the messages passed to the LDPC decoder are significantly correlated. This implies that a large amount of information is embedded in the interdependence of the messages due to the presence of the DE and CM-SISO. The LDPC decoder does not exploit this correlation, thus causing a non-negligible performance degradation. Note that the performance degradation is not due to an ill-conditioned interaction between the correlation structure of the messages and that of the LDPC codewords because the adopted LDPC code is randomly generated, i.e., unstructured.

7.3 Optimized LDPC Codes for PSK

In order to understand the features and limits of the described technique, several optimizations have been carried out, for both a system using DE-PSK and a system using PSK without DE. The set of allowed variable node degrees is $\{2, 3, \ldots, 12\}$, in order to limit the maximum degree and to enable the construction of codes without short cycles. The set of check node degrees is $\{3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ for rate-1/2 codes and $\{3, 6, 8, 9, 12, 16, 20, 24\}$ for rate-3/4 codes.

LDPC codes with codeword length equal to 12000 have been designed to match the obtained optimized code ensembles. Each LDPC code has been concatenated with both a PSK modulator and a DE-PSK modulator. In both cases, Monte Carlo simulations have been performed. For the DE scheme, the maximum number of decoding iterations is $N_i = 30$ and $N_{\mathrm{LDPC}} = 30$, whereas for the scheme *without* DE, a maximum number of 100 standard LDPC iterations is allowed. The decoding process stops if a valid codeword is found earlier. In Figure 7.4, the BER curves relative to three LDPC codes optimized for the presence of a DE-PSK modulator are shown: the solid curves are relative to an LDPC code with rate $R = 1/2$ designed for DE-QPSK, the dashed curves are relative to an LDPC code with rate $R = 1/2$ designed for DE-8PSK, and the dotted curves are relative to an LDPC code with rate $R = 3/4$ designed for DE-8PSK. For each LDPC code, the BER curve which exhibits a "cliff" (i.e., the steepest point) at low SNR corresponds to the system for which the LDPC code has been optimized, i.e., the system with DE-PSK (curves marked "with 'DE"); the other curve represents, instead, the performance of the same LDPC code employed in a BICM scheme using PSK modulation with Gray mapping (curves marked "without DE"). For each case, the SNR value corresponding to the capacity limit for the considered coded modulation is shown as a vertical line. The capacity limit for QPSK with code rate 1/2, i.e., with a spectral efficiency of 1 bit per channel use, is 0.17 dB; the capacity limit for 8PSK with code rate 1/2, i.e., with a spectral efficiency equal to 1.5 bit per channel use, is 1.27 dB; the capacity limit for 8PSK with code rate 3/4, i.e., with spectral efficiency equal to 2.25 bit per channel use, is 3.66 dB. All the DE-PSK systems in Figure 7.4 are operating with about $1 \div 1.5$ dB SNR gap to capacity. In other words, the optimized codes guarantee near-capacity performance, even without an exact phase reference.

In Figure 7.5, the performance of LDPC codes optimized for a memoryless PSK modulator is analyzed, both in the presence and absence of DE. For each code, the curve which exhibits a cliff at low SNR corresponds to a system which uses a memoryless PSK modulator (curves marked "without DE"), while

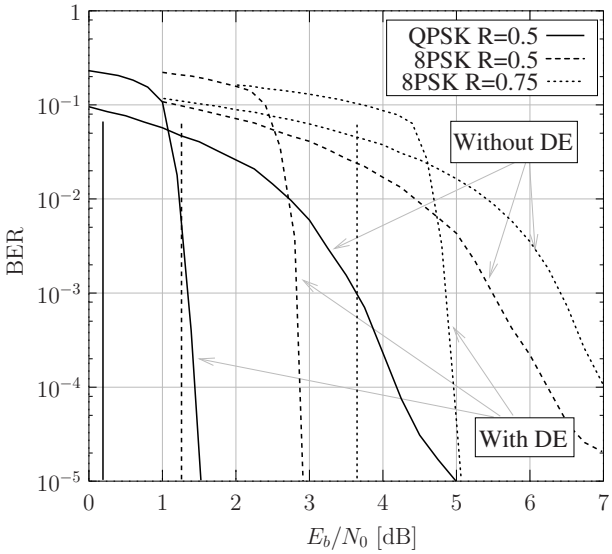

Figure 7.4: Simulated BER for 3 LDPC codes optimized for DE-PSK. Each code is analyzed both with and without DE. In each case, the SNR corresponding to the capacity bound is shown as a vertical line.

the other curve represents the performance of the same code concatenated with a DE-PSK modulator (curves marked "with DE"). The system without DE shows a performance advantage, in terms of SNR corresponding to the cliff of the BER curve, of about 1.5 dB with respect to a system with DE. However, it is important to note that LDPC codes optimized for and used with a memoryless PSK modulator exhibit higher "error floor" with respect to that obtained when the same LDPC codes are used with DE-PSK. The presence of the BER floor in the memoryless PSK modulator is due to the nature of the used code, which contains a small amount of *short* cycles. On the other hand, the absence of the floor in the DE-PSK case can be associated with the fact that the DE-PSK modulator can be interpreted as a rate-1 recursive encoder. As shown in [68], the presence of a rate-1 recursive encoder can reduce short error patterns, responsible for the BER curve flattening, by exploiting the so-called *interleaving gain*.

In Figure 7.6, the coefficients $\{\rho_j\}$ and $\{\lambda_i\}$ of several optimized LDPC codes are shown. Different code ensembles with equal constraints are obtained considering different initial seeds of the pseudo-random number generator embedded in the random walk-based optimization algorithm. The code ensembles

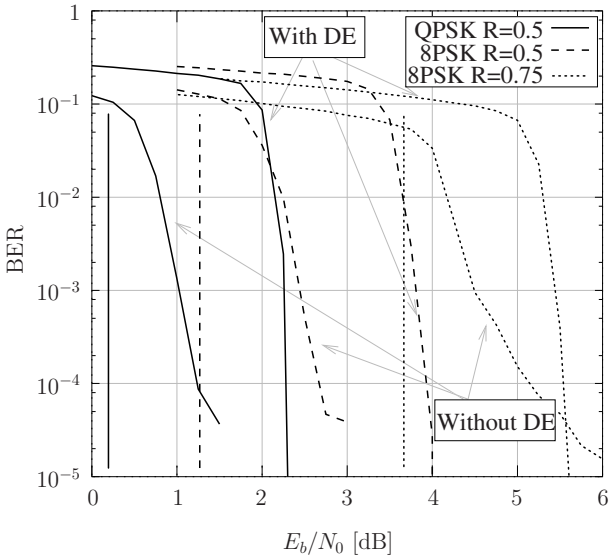

Figure 7.5: Simulated BER for 3 LDPC codes optimized for a memoryless channel. Each code is concatenated with MPSK both with and without DE. In each case, the SNR corresponding to the capacity bound is shown as a vertical line.

in Figure 7.6 (a) and (b) are optimized for DE-QPSK with rate 1/2. The algorithm operates over a limited parametric space, i.e., only a small set of possible node degrees are allowed: the set of variable node degrees is $\{2, 3, \ldots, 12\}$ and the set of check node degrees is $\{3, 4, \ldots, 12\}$. These sets of values makes it possible to design codes without short cycles and reasonable codeword length. The variable node degree distributions $\{\lambda_i\}$ in Figure 7.6 (c) correspond to realizations of rate-1/2 LDPC codes optimized for transmission with BICM PSK. The check node degree distributions appear to give little information, due to the optimization algorithm "residual noise." This is not surprising since, as stated in [13], the performance of LDPC codes exhibit little dependence on the check node degree distribution. Focusing our attention on the coefficients $\{\lambda_i\}$, it is possible to observe that degree-2 variable nodes show a characteristic behavior: in the LDPC code ensembles optimized for DE-QPSK, $\lambda_2 > 0.5$ and $\lambda_2 >> \lambda_i$, $i > 2$. Very similar results, in terms of variable node degree distributions with a predominance of λ_2, were obtained also for LDPC codes optimized for rate-1/2 DE-8PSK and rate-3/4 DE-8PSK. In the LDPC code ensembles optimized for a PSK modulator, λ_2 is still higher than the

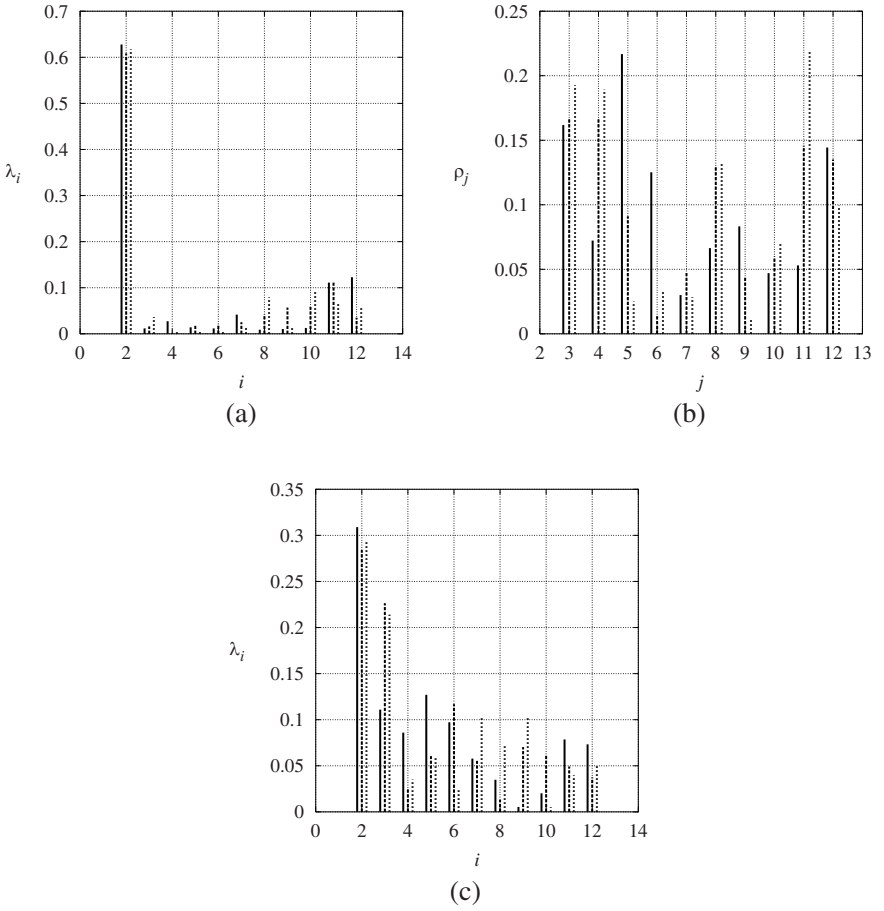

Figure 7.6: Bar diagrams of degree distribution coefficients of three realizations of optimized LDPC code ensembles. In (a) and (b), variable and check node degree distributions for three LDPC codes optimized for rate-1/2 DE-QPSK are shown, respectively. In (c), the variable node degree distributions of three LDPC codes optimized for rate-1/2 QPSK are shown.

other coefficients, but not as high as for DE schemes (see Figure 7.6 (c)).

In [13], a stability condition on λ_2 (inequality (5.30) in Chapter 5) for a standard LDPC decoding algorithm is provided. According to this condition, in order for the BER to approach zero, a necessary condition is $\lambda_2 < \epsilon$, where ϵ is a parameter which depends on the channel and $\rho(x)$. As a reference value, in [13] the authors consider $\epsilon \simeq 0.4$ for a rate-1/2 standard LDPC code. The above discussed results show that this condition is violated if a CM is inserted between the LDPC code and the channel, allowing, in the case of DE-PSK, higher values of λ_2. Note that an LDPC code with a high value of λ_2 is a code whose majority of variable nodes have degree 2, and this corresponds to code graphs with a smaller number of edges (for a given code rate and codeword length). In fact, if ℓ is the total number of edges in the graph and N is the length of the LDPC codeword, it holds

$$N = \ell \sum_i \frac{\lambda_i}{i} .$$

Considering two codes with equal codeword length, variable node degree distributions $\{\lambda_i\}$ and $\{\lambda'_i\}$ and number of edges ℓ and ℓ', respectively, the ratio between the number of edges in the code graph can be written as

$$\frac{\ell'}{\ell} = \frac{\sum_i \frac{\lambda_i}{i}}{\sum_i \frac{\lambda'_i}{i}} . \tag{7.1}$$

If we substitute in (7.1) the degree distributions obtained optimizing for AWGN and for DE, respectively, we obtain a reduction of the edges in the graph of about 20%. Since the computational cost of the decoding algorithm for an LDPC code is proportional to the number of edges in the graph, it follows that LDPC codes optimized for DE-PSK have the pleasant side effect of allowing decoding with lower complexity.

It is generally believed that degree-2 variable nodes exhibit weaker error protection than higher-order variable nodes [13, 34]. However, considering Figure 7.6 (a), one notices that the presence of a large percentage of degree-2 variable nodes is associated with an increase of the fractions of high-degree variable nodes. A possible intuitive interpretation of this behavior is as follows. While a standard LDPC decoder exploits all the available information from the very first iteration, in the considered iterative detector/decoder for LDPC coded modulations the information made available at the "channel input" of the LDPC VND by the CM-SISO block increases with the iterations. This is possible since, at every iteration, the VND passes information to the *a priori*

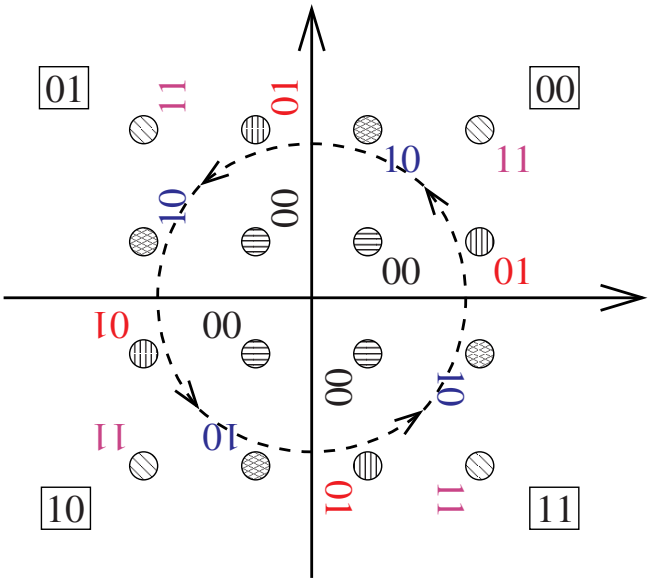

Figure 7.7: Pictorial representation of a DE-16QAM modulation format.

input of the CM-SISO block (see Figure 5.3). Therefore, the critical part of the decoding algorithm corresponds to the first iterations, when the information from CM-SISO block is limited. High-degree variable nodes seem to help the convergence of the iterative algorithm in the first iterations.

7.4 LDPC Codes for DE-QAM

In this subsection, the transmission of LDPC codes concatenated with 16QAM and DE-16QAM is considered. In Figure 7.7, a pictorial representation of a DE-16QAM modulation format is shown. One can observe that two of the four bits at its input are encoded by a Gray mapped DE-QPSK modulator: the obtained point is used to rotate, by an angle equal to a multiple of $\pi/2$, a first-quadrant 16QAM constellation point selected by the other two bits (one bit per dimension).

In Figure 7.8, the BER performance for two communication systems with 16QAM, with and without DE, is shown. Both systems use the same rate-7/8 LDPC code with codeword length 65536. For reference purposes, a vertical dash-dotted line is also shown in correspondence to the capacity SNR, equal to approximately 6.16 dB. The LDPC code is chosen from an ensemble of

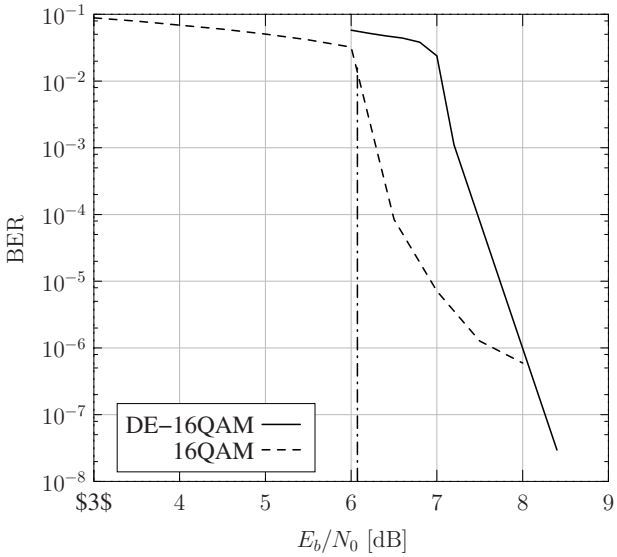

Figure 7.8: BER of a rate-7/8 LDPC code optimized for DE-16QAM and concatenated with DE-16QAM (solid line) and with a 16QAM memoryless modulator (dashed line). The vertical (dash-dotted) line indicates the SNR corresponding to the capacity limit for 16-QAM at the considered code rate.

codes optimized for the presence of DE-16QAM modulator. For DE-QAM, the maximum numbers of iterations are $N_i = 30$ and $N_{\text{LDPC}} = 30$, whereas for QAM without DE a maximum number of 100 standard LDPC iterations is allowed. The solid curve corresponds to a system with a DE-16QAM and the dashed curve corresponds to the system with a Gray mapped 16QAM.

The results in Figure 7.8 show that the code designed for DE-16QAM performs better if used *without* DE. A possible interpretation of this result is that the iteration gain of the DE-QAM SISO module, i.e., the gain enabled by allowing the VND to pass messages to the CM-SISO, is very low. In other words, from an EXIT chart point of view, the considered DE-QAM modulator is similar to a memoryless, Gray-mapped QAM modulator, and this implies that good codes for DE-QAM may also be good codes for QAM. However, the memory introduced by the DE and the corresponding CM-SISO block leads to strong sub-optimality of the processing at the LDPC VND and CND, which assume an underlying memoryless channel. Another immediately noticeable fact is that the BER curve relative to the system without DE is characterized by an error floor, whereas the curve relative to the system with DE does not

show any floor in the considered BER range. The floor in the QAM case can be attributed to the presence of a small amount of short cycles in the code graph, which is typical of *random* LDPC codes. Moreover, the differential encoder can be reinterpreted as a rate-1 recursive encoder (at least for the bits which select the quadrant) which, as observed for DE-PSK, is likely to reduce short error patterns.

7.5 LDPC Codes for DE-PSK with Noncoherent Detection

Since the design method described in Section 5.6 can take into account the particular channel, as well as the modulation format and the detection algorithm used in the CM-SISO block, the optimization has been carried out also for LDPC codes concatenated with DE-PSK with noncoherent detection. In the presence of phase uncertainty, the received observation can be modeled as

$$r_k = c_k e^{j\theta} + n_k \tag{7.2}$$

where θ is a random variable constant over the transmitted block and uniformly distributed over $[0, 2\pi)$. While coherent detection can be based on the standard FB algorithm in the CM-SISO module, noncoherent maximum a posteriori (MAP) symbol detection requires some approximations. Following the approach in [109, 110], one can derive a detection algorithm based on a quantization of the phase rotation introduced by the channel. First, the *a posteriori* probability (APP) are computed by the FB algorithm conditionally on one hypothetical channel phase value; then, the conditional APPs are averaged over all possible phase values. The *a posteriori* symbol probability can be written as

$$\begin{aligned} P\{a_k|\boldsymbol{r}\} &\propto P\{a_k\}p(\boldsymbol{r}|a_k) \\ &= P\{a_k\}\int_\theta p(\boldsymbol{r}|a_k,\theta)p_\theta(\theta)\mathrm{d}\theta \end{aligned} \tag{7.3}$$

where $\boldsymbol{r}$ is the vector of all received observations and $\propto$ means that the left member is equal to the right member times a constant independent of a_k. In (7.3), $p(\boldsymbol{r}|a_k,\theta)$ can be interpreted as the extrinsic information generated by a coherent FB algorithm, which assumes a phase rotation θ. The integral in (7.3) can be approximated as a sum over a *properly chosen* discrete set $\mathcal{P}$ of quantized phase values, obtaining:

$$P\{a_k|\boldsymbol{r}\} \mathrel{\widetilde{\propto}} P\{a_k\}\sum_{\theta\in\mathcal{P}} p(\boldsymbol{r}|a_k,\theta)P(\theta) \tag{7.4}$$

where $\widetilde{\propto}$ denotes an approximation in the relationship described by symbol $\propto$.

Since DE-PSK is insensitive to rotations of the received signal by multiples of $2\pi/M$, where M is the cardinality of the PSK symbols, the set of phases can be a subset of $[0, 2\pi/M)$ [109]. We then choose two possible sets: the first includes 8 equally spaced points (in [110], this is shown to lead to negligible performance degradation), i.e., $\mathcal{P} = \{0, \frac{1}{8}\frac{2\pi}{M}, \ldots, \frac{7}{8}\frac{2\pi}{M}\}$, and the second includes 4 equally spaced points, i.e., $\mathcal{P} = \{0, \frac{1}{4}\frac{2\pi}{M}, \frac{2}{4}\frac{2\pi}{M}, \frac{3}{4}\frac{2\pi}{M}\}$. The optimization algorithm is then run over the same set of node degrees as in the previous subsection. The difference between the degree distributions of LDPC code ensembles optimized for DE-PSK and noncoherent detection and those relative to coherent detection is not noticeable. This is true even if the number of quantization levels used for the computation of (7.4) is reduced to two. An intuitive explanation of this fact is that DE is a technique which makes the communication system insensitive to phase uncertainties, so that the introduction of a further, possibly continuous, phase uncertainty cannot induce a severe system change. Moreover, theoretical results show that, asymptotically, the performance of a noncoherent system approaches that of a coherent system [26, 111–113].

In Figure 7.9, the performance of optimized LDPC codes for DE-PSK with coherent and noncoherent detection is compared. The considered LDPC codes are optimized for DE-QPSK (with rate 1/2) and DE-8PSK (with rates equal to 1/2 and 3/4, respectively); the length of the codeword is 12000 and the maximum allowed numbers of inner and outer iterations are $N_i = 30$ and $N_{\text{LDPC}} = 30$, respectively. The considered numbers of discrete phase values are 8 (curves marked by a triangle) and 4 (curves marked by a square). The curves relative to coherent detection are marked by a circle. It is clear that the phase uncertainty introduces a limited performance loss, as long as the phase quantization is sufficiently fine. Moreover, the results in Figure 7.9 show that, while an 8-level phase quantization introduces negligible performance loss, a 4-level quantization introduces a performance loss of about 0.4 dB. Further analysis on the described noncoherent detection algorithm shows that the number of quantization levels can be reduced to a minimum number of 2, causing a performance loss of about 1.7 dB with respect to coherent detection.

7.6 Detection by Multiple Trellises

In the previous sections, LDPC codes were designed for AWGN and noncoherent channel, i.e., an AWGN channel with a constant and unknown phase uncertainty. Practical channels are often influenced by a number of parame-

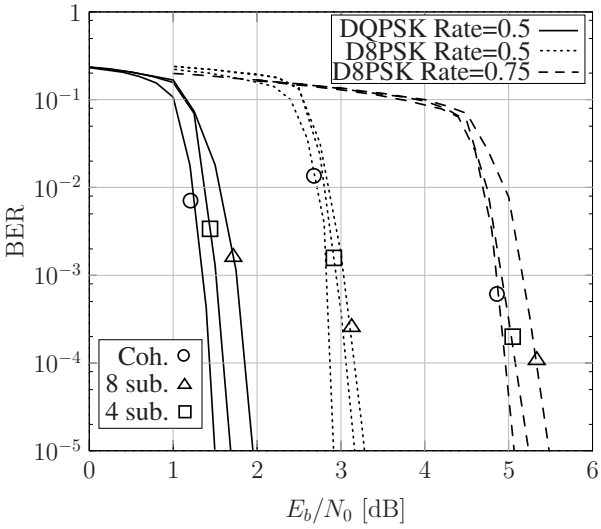

Figure 7.9: BER of LDPC codes optimized for DE-QPSK (rate 1/2) and DE-8PSK (rates 1/2 and 3/4) optimized both for AWGN channel and noncoherent channel.

ters, which in general may vary with time. The problem of designing an effective CM-SISO algorithm in the presence of time-varying parameters is often nontrivial and pursueing optimal solutions might entail a significant computational burden. In this section, a family of CM-SISO algorithms accounting for time varying parameters in the channel/system model are described, with particular emphasis on the phase uncertain channel and the fading channel.

In order to set the problem under study and present the mathematical notation, we begin by reviewing a modified version of the FB algorithm suitable for generic finite-memory channels affected by time-invariant stochastic parameters (see also Chapter 2 for more details on the design of FB algorithms for channels with memory). Afterwards, we will describe the extension to time-varying parameters and analyze two different multi-trellis SISO algorithms.

7.6.1 Time-Invariant Parameters

Let us assume that the channel output is observed for a period of $K+1$ symbol intervals. The channel can be completely described by the following

joint probability density function (pdf)

$$p(\boldsymbol{r}_0^K, \xi | \boldsymbol{a}_0^K) \tag{7.5}$$

where $\boldsymbol{r}_0^K$ (or, simply, $\boldsymbol{r}$) is the vector of the observables $(r_0, \ldots, r_K)$, $\xi \in \mathcal{D}_\xi$ is a stochastic constant channel parameter independent of the transmitted data, $\mathcal{D}_\xi$ is the domain of the channel parameter, and $\boldsymbol{a}_0^K$ is the vector of information symbols a_k transmitted through this channel. Note that (7.5) can take into account possible coding of the information symbol sequence $\{a_k\}$ into a code sequence $\{c_k\}$. We remark that the parameter ξ could be either a scalar parameter or a vector parameter, i.e., ξ could represent a whole set of parameters.

The APP of an information symbol a_k can be expressed as follows:

$$\begin{aligned} P\{a_k | \boldsymbol{r}_0^K\} &\propto p(\boldsymbol{r}_0^K | a_k) P\{a_k\} \\ &= P\{a_k\} \int_{\mathcal{D}_\xi} p(\boldsymbol{r}_0^K | a_k, \xi) . p(\xi) \, \mathrm{d}\xi \end{aligned} \tag{7.6}$$

If, conditionally on the parameter realization ξ, the channel has finite memory [26], the conditional pdf $p(\boldsymbol{r}|a_k, \xi)$ can be computed via a standard FB algorithm [17, 65]. This is possible whenever the transmission system can be modeled as a finite state machine (FSM) whose input and output are, respectively, the information symbol a_k and a random variable (RV) whose statistics depend only on the FSM state and the input symbol (see Chapter 2).

A simple approximation for the computation of the integral in (7.6) is obtained by performing the following finite sum:

$$P\{a_k | \boldsymbol{r}\} \mathrel{\widetilde{\propto}} P\{a_k\} \sum_{i=1}^{L} p\left(\boldsymbol{r} | a_k, \xi^{(i)}\right) p(\xi^{(i)}) \tag{7.7}$$

where $\{\xi^{(1)}, \ldots, \xi^{(L)}\}$ is a set of hypothetical quantized values for the channel parameter whose actual values and number L are chosen to obtain the desired accuracy in the numerical integration in (7.6). This corresponds to running L standard FB algorithms in parallel, each one associated with a value $\xi^{(i)}$, $i = 1, \ldots, L$, and computing a weighted average of their outputs to obtain a quantity approximately proportional to the APP.[1]

In the following, we denote the *forward state metrics* computed during the forward recursion of an FB algorithm as $\{\alpha^{(i)}(s_k)\}$, where the superscript

[1] A detailed explanation of the FB algorithm can be found in Chapter 2; further references include [17, 25, 65].

i refers to the FB algorithm associated with the quantized parameter value $\xi^{(i)}$ and s_k denotes the state of the FSM in the corresponding trellis diagram. In particular, we assume that $s_k \in \{0, \ldots, \Xi - 1\}$, where Ξ is the number of states characterizing each trellis. Similarly, we denote the *backward state metrics* associated with the i-th trellis diagram as $\{\beta^{(i)}(s_k)\}$.

Several practical scenarios can be cast within the model described by (7.5), (7.6) and (7.7). In particular, as useful examples, we will consider phase-uncertain and flat fading channels.

Phase-Uncertain Channel

In a communication scenario where the channel introduces a time-invariant phase rotation, the stochastic channel parameter ξ can be equivalently modeled as a phase rotation θ of the transmitted symbol sequence. The discrete-time equivalent observation can be expressed as

$$r_k = c_k e^{j\theta} + n_k \tag{7.8}$$

where r_k is the received observable, c_k is the (possibly encoded) transmitted symbol, and n_k is a (noise) sample of a sequence of independent and identically distributed (i.i.d.) zero mean Gaussian RVs.

Flat Fading Channel

The generic observation model given by (7.5) applies directly to a flat fading channel, provided that ξ has the proper statistical distribution. In particular, in a scenario with unresolvable multipath, ξ corresponds to a fading coefficient f and the channel input-output relation can be expressed as follows:

$$r_k = f\, c_k + n_k \tag{7.9}$$

where, in the case of Rayleigh fading, f has a complex circularly-symmetric Gaussian distribution with zero mean.

7.6.2 Time-Varying Parameters

The idea of detection by multiple trellises stems from an extension of the previous *static-parameter* approach to a scenario with *time-varying* channel parameters.

In order to obtain insights on the impact of the presence of a time-varying parameter, let us consider a useful case study where the channel parameter process $\{\xi_k\}$ is discrete and block constant. Let us assume that ξ_k is uniformly

distributed over the set $\{\xi^{(1)}, \ldots, \xi^{(L)}\}$ and constant over blocks of length $N < K$. In other words,

$$\xi_{lN+i} = \xi_{lN+j} \qquad \forall i, j \in \{0, \ldots, N-1\}$$

and the realizations $\{\xi_k\}$ are i.i.d. from block to block, i.e.,

$$p(\xi_{lN}, \xi_{nN}) = p(\xi_{lN})p(\xi_{nN}) = \frac{1}{L^2} \qquad \forall l \neq n\,.$$

As a consequence, the process $\{\xi_k\}$ is a time-varying Markov chain, characterized by an $L \times L$ transition matrix $P_k = (p_{ij}^{(k)})$ at the k-th epoch such that

$$p_{ij}^{(k)} = \begin{cases} \delta_{ij} & \text{if} \quad k \neq N-1 \mod N \\ \dfrac{1}{L} & \text{if} \quad k = N-1 \mod N \end{cases}$$

where δ_{ij} denotes the Kronecker delta. We further assume that the information sequence $\{a_k\}$ is encoded into a code symbol sequence $\{c_k\}$ by means of an FSM. Considering that the observed sequence of length K comprises more than one length-N block with constant channel parameter, the application of a MAP detection strategy to this scenario leads to a time-varying trellis. In Figure 7.10, a representative time-varying trellis for this illustrative block-constant discrete parameter channel is shown. Within a block, i.e., for $N-1$ consecutive time epochs, the trellis structure consists of L "coherent" trellises, each assuming knowledge of ξ, one for each hypothetical quantized value of ξ. In the sections of the various trellis diagrams connecting the states at the end of a block with the states at the beginning of the next block, each state in each coherent trellis is connected with the corresponding state in all the other coherent trellises. In other words, each coherent trellis is connected with any other trellis by the non-zero probability of variation of the parameter value.

A general formulation can be obtained considering an extension of the standard FB algorithm to a channel whose statistics at epoch k are a function of the state ξ_k of a Markov chain. Assume that, given $\{\xi_k\}$, the modulator-channel pair can be described by an FSM, in the sense that the observable statistics are functions of the state σ_k of an FSM whose input is the information symbol sequence $\{a_k\}$. Moreover, let us assume that (i) $\{a_k\}$ and $\{\xi_k\}$ are independent and (ii), given $\{a_k\}$ and $\{\xi_k\}$, the observables are independent. Following the guidelines in [26,110,114,115], it can be shown that the *a posteriori* probability of the symbol a_k can be computed as follows:

$$P\{a_k|\boldsymbol{r}\} = \sum_{(\sigma_k, \sigma_{k+1}):a_k} \beta_{k+1}(\sigma_{k+1})\, \alpha_k(\sigma_k)\, \gamma_k(\sigma_k, \sigma_{k+1}, a_k) \tag{7.10}$$

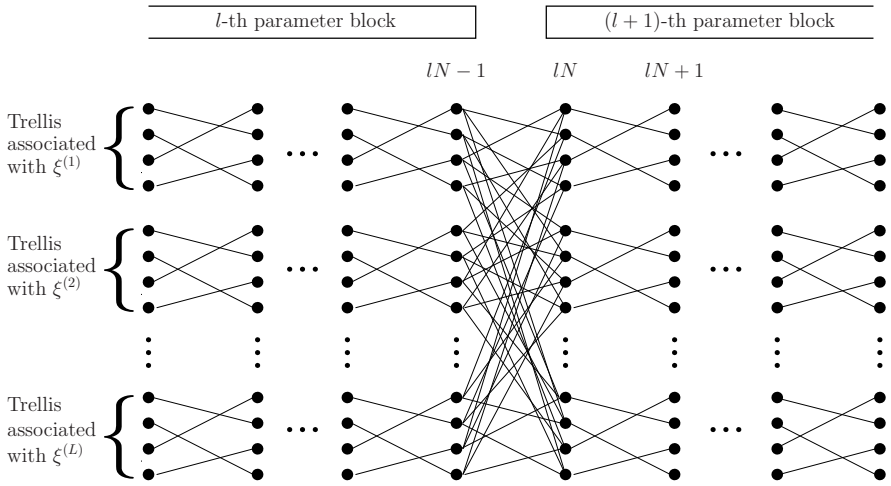

Figure 7.10: Time-varying trellis for detection on block-constant discrete parameter channel.

where, as before, $\boldsymbol{r}$ denotes the vector of the observables and $\sigma_k = (s_k, \xi_k)$ is the (extended) state of the system, the notation $(\sigma_k, \sigma_{k+1}) : a_k$ denotes "the set of all (σ_k, σ_{k+1}) pairs compatible with the input symbol a_k" and the branch "metric" $\gamma_k(\sigma_k, \sigma_{k+1}, a_k)$ is defined as[2]

$$\gamma_k(\sigma_k, \sigma_{k+1}, a_k) = p(r_k|a_k, \xi_k, s_k) \cdot P\{a_k\} \cdot P\{\xi_{k+1}|\xi_k\} \tag{7.11}$$

in which $P\{\xi_{k+1}|\xi_k\}$ is the transition probability between the Markov chain states ξ_k and ξ_{k+1}, and $p(r_k|a_k, \xi_k, s_k)$ is the channel statistical description, i.e., the observable PDF given the data sequence and the channel parameter ξ_k computed at the observation value r_k. The forward and backward "metrics" $\alpha_k(\sigma_k)$ and $\beta_k(\sigma_k)$ are obtained by the following recursions:

$$\begin{aligned} \alpha_k(\sigma_k) &= \sum_{(\sigma_{k-1}, a_{k-1}):\sigma_k} \alpha_{k-1}(\sigma_{k-1})\, \gamma_{k-1}(\sigma_{k-1}, \sigma_k, a_{k-1}) \\ \beta_k(\sigma_k) &= \sum_{(\sigma_{k+1}, a_k):\sigma_k} \beta_{k+1}(\sigma_{k+1})\, \gamma_k(\sigma_k, \sigma_{k+1}, a_k)\,. \end{aligned}$$

The FB algorithm in (7.10) operates on a trellis with a number of states equal to the number Ξ of states of the modulator-channel FSM times the number L

[2]Strictly speaking, $\log \gamma_k(\sigma_k, \sigma_{k+1}, a_k)$ is a metric.

of states of the channel parameter Markov chain. This can be interpreted as a "super-trellis" comprising L trellises, each with Ξ states.

As special case, if the Markov chain $\{\xi_k\}$ is time varying and the transition matrix differs from the identity matrix only at time epochs $k = Nl$, with $l \in \mathbb{N}$, it can be easily shown that the forward and backward recursions in the above extended FB algorithm are equivalent to the computation of L independent forward and backward recursions in the Ξ-state trellises for $N-1$ time epochs. Every N time epochs, the recursions involve, in general, all trellises. This corresponds to a block-constant discrete parameter ξ_k, which has been discussed in Section 7.6.2 assuming uniform distribution of the parameter realization. The corresponding super-trellis is pictorially exemplified in Figure 7.10.

Applying the above general formulation, the forward and backward metrics $\alpha_k(s_k, \xi_k)$ and $\beta_k(s_k, \xi_k)$ are functions of the "extended" state $\sigma_k = (s_k, \xi_k)$. They can be computed recursively by running L separate coherent FB algorithms, one for each parameter value. Every N time epochs, in general, $\alpha_k(s_{k+1}, \xi_{k+1})$ and $\beta_k(s_k, \xi_k)$ depend on all forward and backward metrics in all coherent trellises, respectively, i.e., a "mix" of the forward and backward metrics in the coherent FB algorithms is performed. The above considerations can be equivalently drawn by following the guidelines in [110], where a Markov chain model for the channel phase is assumed.

At this point, the idea underlying detection by multiple trellises can be outlined. As for a constant channel parameter ξ, several coherent FB algorithms, characterized by forward and backward metrics $\alpha_k^{(i)}(s_k) = \alpha_k(s_k, \xi^{(i)})$ and $\beta_k^{(i)}(s_k) = \beta_k(s_k, \xi^{(i)})$, respectively, are run independently. The difference with respect to the time-invariant channel parameter case is that every N time epochs, the forward (backward) metrics in the different trellises are properly "mixed" to account for the possible variation of the channel parameter. In the following, we will refer to N as "inter-mix interval."

The idea of considering parallel trellises which occasionally "talk" to each other is appealing, since it is likely to allow both low-complexity and parallel processing. In this sense, performing detection by multiple trellises can be equivalently interpreted as an instance of the *divide et impera* approach to tackle complicated problems with limited complexity.

We remark that the "mixing strategy" should be tailored for the specific communication scenario at hand. Nevertheless, some general considerations can be drawn:

- If ξ is time invariant, the quantity $p(\boldsymbol{r}|a_k, \xi^{(i)})$, computed via a coherent FB algorithm, is expected to be maximum in correspondence to the value

$\xi^{(i)}$ closest to the true[3] channel parameter ξ. In fact, numerical analyses carried out in several scenarios showed that the forward and backward state metrics $\{\alpha_k^{(i)}(s_k)\}$ and $\{\beta_k^{(i)}(s_k)\}$ exhibit an exponential decay in the probability domain. as a function of the epoch k. This is due to the fact that, denoting by $\boldsymbol{\alpha}_k^{(i)}$ the vector of the forward metrics at epoch k in the i-th trellis, the forward recursion can be equivalently expressed as

$$\boldsymbol{\alpha}_k^{(i)} = \Gamma_{k-1}^{(i)} \boldsymbol{\alpha}_{k-1}^{(i)} \tag{7.12}$$

where $\Gamma_k^{(i)}$ is a matrix whose elements are the pdfs of the observable r_k conditioned on every possible transitions in the i-th coherent trellis. In particular, as expected, the decay exponent is higher (i.e., decay is slower) in the FB algorithm associated with the phase value $\xi^{(i)}$ which is closest to the *true* channel parameter ξ, leading to state metrics $\{\alpha_k^{(i)}(s_k)\}$ and $\{\beta_k^{(i)}(s_k)\}$ relatively much larger than those computed by the j-th FB algorithm with $j \neq i$.

- If ξ is time varying, we expect that $\{\alpha_k^{(i)}(s_k)\}$ and $\{\beta_k^{(i)}(s_k)\}$ will try to *adapt* to the parameter changes. This adaptiveness is limited by the fact that state metrics exhibit a "low-pass filter" behavior, i.e., they have *memory* and can change only slowly. This is due to the recursive structure of the metric computation algorithm (7.12). In other words, the FB metric computation process can be equivalently described as a recursive time-varying vector filtering.

- While in standard applications an FB algorithm is insensitive to a possible multiplication of all forward or backward state metrics by a constant, in the algorithm underlying (7.7), the relative weights of different trellises are important. Accordingly, the multi-trellis SISO algorithm turns out to be insensitive to a *normalization* of the metrics *only if* this normalization is carried out, at a given epoch, over all forward or backward state metrics of all parallel FB algorithms.

In the following, two possible "mix" strategies are described.

[3]Depending on the symmetry structure of the modulation code, i.e., the law encoding the information symbols a_k into the transmitted symbols c_k, there can be a *set* of ξ values which are optimal, in the sense that they are undistinguishable at the receiver. This may occur, for example, in differential M-PSK transmitted over a phase uncertain channel, where phase rotations of the observed sequences by multiples of $2\pi/M$ cannot be distinguished [109,110].

Multi-Trellis SISO Algorithm 1

At each length-N interval, i.e., at epochs $k = lN$, $l \in \mathbb{N}$, one could manipulate the forward metrics $\{\alpha_k^{(i)}(s_k)\}$ (and, similarly, the backward metrics $\{\beta_k^{(i)}(s_k)\}$) according to the following rule:

$$\alpha_k^{(i)}(s_k) \longleftarrow \sum_{j=1}^{L} \alpha_k^{(j)}(s_k) \quad i = 1, \ldots, L \quad \forall s_k \tag{7.13}$$

where the notation "$\longleftarrow$" represents the assignment of a new value. This corresponds to averaging, for every given state s_k, the metrics relative to all quantized phase values: in other words, the metrics associated with a given state in the various trellises are averaged. We will refer to this algorithm as Algorithm 1. This is the exact APP computation algorithm for the channel with block-constant parameter described at the beginning of Section 7.6.2, if the observables are independent (conditionally on the parameter and the data sequence).

Multi-Trellis SISO Algorithm 2

Assume that the channel is slowly time varying, i.e., assume that ξ_k can exhibit small changes at adjacent epochs. If a suitable manipulation of $\{\alpha_k^{(i)}(s_k)\}$ and $\{\beta_k^{(i)}(s_k)\}$ is allowed only at epochs $k = lN$, with $l \in \mathbb{N}$, the possible transitions of the parameter from one quantization interval to another, occurring amid the block, should be taken into account. Heuristically, it was discovered in [108] that the impact of slow parameter changes within the block can be limited by performing a normalization of the forward state metrics $\{\alpha_k^{(i)}(s_k)\}$ (and, similarly, of the backward state metrics $\{\beta_k^{(i)}(s_k)\}$) as follows:

$$\alpha_k^{(i)}(s_k) \longleftarrow \frac{\alpha_k^{(i)}(s_k)}{\sum_{s'_k} \alpha_k^{(i)}(s'_k)} \quad i = 1, \ldots, L \quad \forall s_k \,. \tag{7.14}$$

where s'_k is a dummy state in the summation, running over all Ξ states of a coherent trellis. This corresponds to a normalization of the state metrics within each FB algorithm, i.e., trellis by trellis, as opposed to a normalization amongst all trellises (as considered in Algorithm 1). We will refer to this algorithm as Algorithm 2.

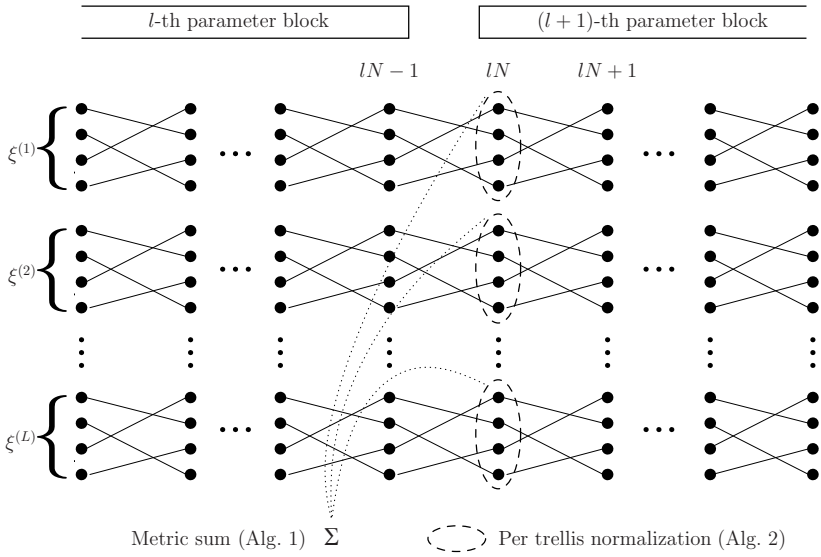

Figure 7.11: Pictorial exemplification of the metric mixes in the two considered algorithms.

Metric Mix in the Algorithms: a Comparison

The manipulations corresponding to (7.13) and (7.14) can be interpreted as a combining or *mixing* of the metrics $\{\alpha_k^{(i)}(s_k)\}$ (similarly for the metrics $\{\beta_k^{(i)}(s_k)\}$). Figure 7.11 gives a pictorial description of the described algorithmic family, highlighting the *metric mix* for both Algorithms 1 and 2. Each depicted trellis diagram is associated with a coherent FB algorithm which assumes a given channel parameter $\xi^{(i)}$, $i = 1, \ldots, L$. The metric mix for Algorithm 1 is shown to "manipulate" the metrics of all trellises summing all metrics on a per-state basis, whereas the metric mix for Algorithm 2 "manipulates" each trellis independently of the other trellises, performing a per-trellis normalization. The mixing epochs $\{lN\}$, i.e., the beginning of the blocks, refer to the forward metric computation. The backward metric computation mix is performed at epochs $\{lN-1\}$.

In both Algorithms 1 and 2, the value of L, i.e., the number of quantized values of the channel parameter, must be chosen considering its impact on both performance and complexity. In particular, by increasing L the performance of the described detection algorithms can be improved, even though

for sufficiently large value of L the performance improvement becomes negligible. On the other hand, it can be shown that the complexity of the detection algorithms increases linearly with L [108].

7.7 LDPC Coded Schemes with Detection by Multiple Trellises

7.7.1 Phase-Uncertain Channels

In this subsection, a phase-uncertain channel is considered. First, the algorithms introduced in Section 7.6 are specialized to this type of channel. Then, these algorithms are analyzed and numerical results are given to characterize their performance.

In Section 7.6.1, the model for a channel introducing a time-invariant phase rotation θ is given. In this case, the APP of an information symbol a_k is given by (7.6). Assuming that θ is uniformly distributed, i.e., $p_\theta(\vartheta) = 1/2\pi$ for $\vartheta \in [0, 2\pi)$ (and 0 otherwise), expression (7.7) specializes to the following:

$$P\{a_k|\boldsymbol{r}\} \mathrel{\widetilde{\propto}} P\{a_k\} \sum_{i=1}^{L} p(\boldsymbol{r}|a_k, \vartheta^{(i)}) \tag{7.15}$$

where $\{\vartheta^{(1)}, \ldots, \vartheta^{(L)}\}$ is a set of L *properly chosen* phase values [79]. This detection approach for channels with a block-constant random phase was used in [109].

If we assume a slowly varying channel phase (i.e., the bandwidth of the channel parameter process is small compared with the receiver filter bandwidth), the discrete-time observable can be modeled as in (7.8) by incorporating a time-varying phase process $\{\theta_k\}$[4]:

$$r_k = c_k e^{j\theta_k} + n_k \ . \tag{7.16}$$

where $|c_k| = 1$ since DE-QPSK is considered and n_k is a discrete-time complex AWGN process with $\mathrm{Var}\{n_k\} = (RE_\mathrm{b}/N_0)^{-1}$, in which R is the system spectral efficiency in bits per channel use. By suitably modeling the stochastic process $\{\theta_k\}$, one could try to develop an *exact* APP algorithm. Since we do not want to rely on the exact knowledge of the channel parameter statistics, which is seldom available at the receiver, we resort to the multi-trellis SISO algorithms described in Section 7.6.2.

[4]This discrete-time model can be obtained from the continuous-time multiplicative model assuming that the phase process has a bandwidth much smaller than the signal bandwidth.

In this section, we assume that transmission over an AWGN channel is affected by a Wiener phase noise process $\{\theta_k\}$ described by the following recursive relation:

$$\theta_k = \theta_{k-1} + w_k \mod 2\pi \tag{7.17}$$

where $\{w_k\}$ is a sequence of i.i.d zero mean Gaussian variables. The standard deviation of w_k, denoted as σ_θ, is representative of the phase noise intensity.

The adopted LDPC coded modulation scheme uses a regular (3,6) LDPC code. The codeword length is set to 6000 bits. The decoder structure is that described in Chapter 5. The number of inner and outer final iterations is $N_i = 30$ and $N_{\text{LDPC}} = 30$, respectively.

In Figure 7.12, the performance of the described schemes is shown in terms of BER versus SNR. The performance for transmission over an AWGN channel without phase noise, considering an ideal coherent FB algorithm as inner detector, is shown as a reference. The remaining curves show the performance obtained with the considered algorithms. In particular, the curves marked as "Alg1" and "Alg2" correspond to the performance of the schemes with Algorithms 1 and 2, respectively. For each algorithm, several values of the phase noise standard deviation σ_θ (given in degrees in the figure legend) are considered. In each case, the inter-mix interval N is heuristically optimized. The results in Figure 7.12 show that, even in the presence of a significant phase noise (for instance, $\sigma_\theta = 10°$), it is possible to "blindly" process the metrics of the trellises while still achieving an SNR loss as limited as 1 dB. Heuristically, the optimum value of N turns out to be inversely proportional to σ_θ. The results in Figure 7.12 show that Algorithm 2 entails better performance than Algorithm 1. In particular, for very strong phase noise, i.e., $\sigma_\theta = 10°$, Algorithm 1 suffers an SNR penalty larger than 1 dB with respect to Algorithm 2. This is due to the fact that Algorithm 1 completely erases the phase information every N time epochs, whereas Algorithm 2 performs only a "trellis balancing" as described in Section 7.6.2.

In Figure 7.13, a direct comparison between the performance (in terms of BER as a function of the SNR) with Algorithm 1 and Algorithm 2, for a fixed value of the inter-mix distance $N = 15$, and several values of σ_θ, is shown. The value $N = 15$ optimizes the system performance at $\sigma_\theta = 5°$, as shown in Figure 7.12. The remaining system and simulation parameters are those of Figure 7.12. The BER curves show clearly that for values of the phase noise parameter σ_θ lower than or equal to 5°, decoding convergence is guaranteed for approximately the same value of SNR, whereas if $\sigma_\theta > 5°$ convergence is not guaranteed any longer, i.e., an error floor may appear. In particular, the error floor characterizing the BER curve corresponding to Algorithm 2

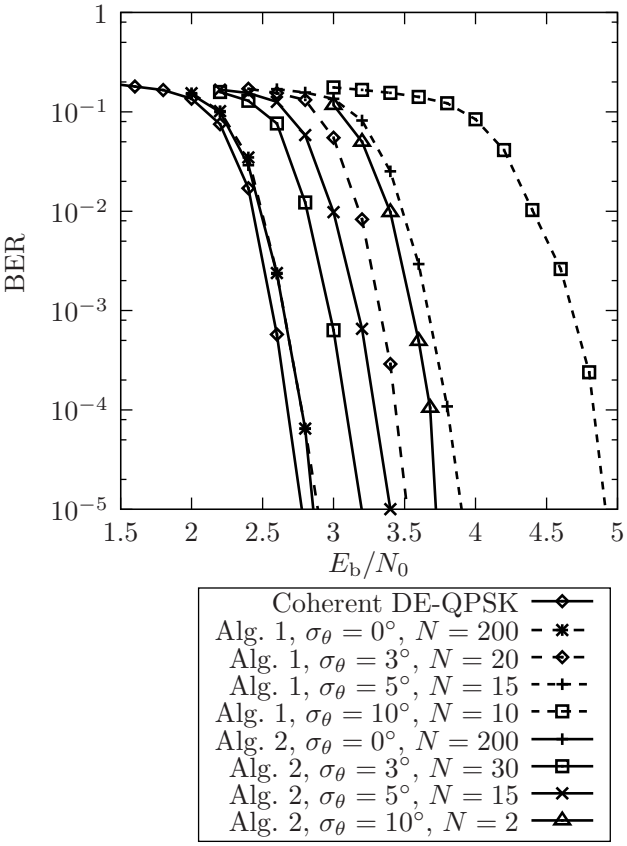

Figure 7.12: BER performance of LDPC coded DE-QPSK schemes based on algorithms 1 and 2.

with $\sigma_\theta = 10°$ is due to the fact that, in order to cope with a strong phase noise, Algorithm 2 needs a very small inter-mix interval N, as clearly shown in Figure 7.12. From the results in Figure 7.13, one can conclude that the described algorithms are *blind* with respect to the phase noise intensity *as long as* this intensity is lower than the value considered in the algorithm design.

7.7.2 Flat Fading Channels

In this section, a flat fading channel is considered. First, we derive the FB algorithm assuming a Markov chain model for the fading channel. Then, we specialize the algorithm introduced in Section 7.6 to the case of flat fading

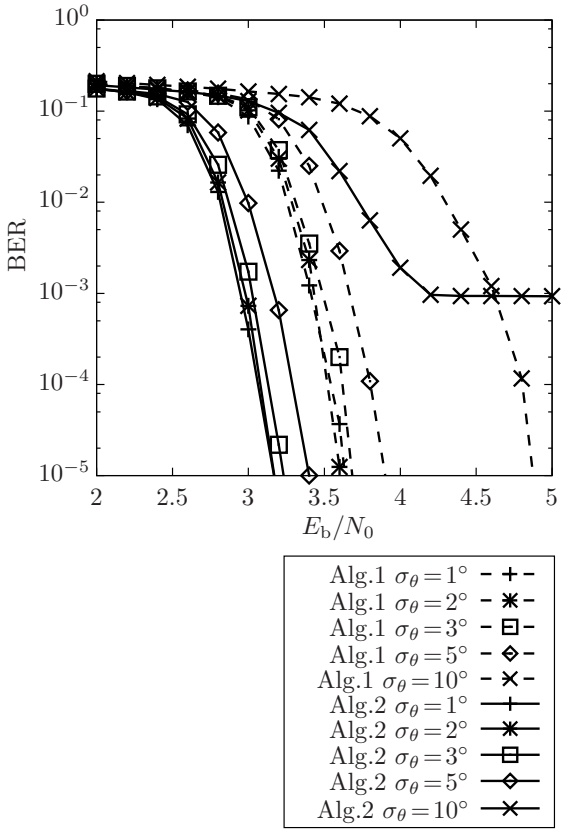

Figure 7.13: BER performance, as a function of the SNR, of the considered Algorithms 1 and 2. Several values of σ_θ are considered and $N = 15$.

channel, highlighting its similarities with the Markov chain-based approach. Finally, the algorithms are analyzed and their performance is characterized through numerical results.

The time-invariant flat fading model given in (7.9) can be extended to a more realistic model with time-varying flat fading. Accordingly, the discrete-time observable can be expressed as

$$r_k = f_k\, c_k + n_k \tag{7.18}$$

where $\{f_k\}$ is the fading process.[5] In the presence of Rayleigh fading, each realization f_k can be modeled as a zero-mean complex circularly symmetric

[5]We remark that this discrete-time model can be obtained from the continuous-time

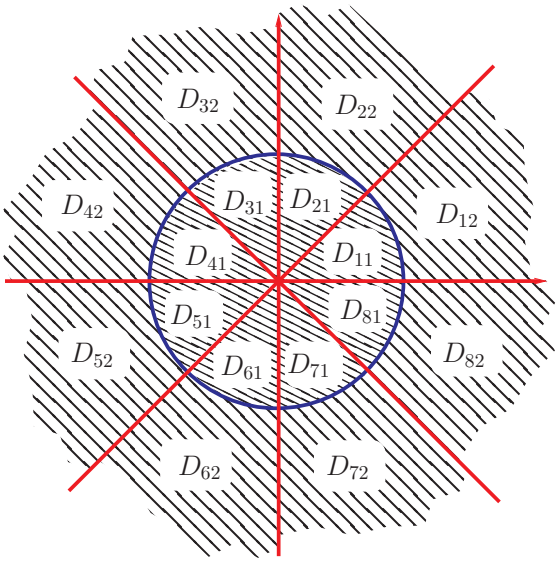

Figure 7.14: Partitioning of the fading complex plane into fading regions.

Gaussian RV. We assume that the fading process $\{f_k\}$ is modeled according to Clarke [116, 117], with zero mean, unit variance and autocorrelation function $R_f(n) = J_0(2\pi n f_{\mathrm{D}} T)$, where $J_0(\cdot)$ is the zero-th order Bessel function and $f_{\mathrm{D}}T$ is the maximum normalized Doppler shift, which characterizes the speed of the fading process.

We now outline the derivation of a simple first-order Markov chain model which approximately describes the evolution of the complex fading process. Several papers deal with Markov chain modeling of the fading process—for more details, we refer the reader to [118, 119] and references therein. We first partition the complex plane into N_{phase} angular sectors $[2\pi\frac{i-1}{N_{\mathrm{phase}}}, 2\pi\frac{i}{N_{\mathrm{phase}}})$, $i = 1, \ldots, N_{\mathrm{phase}}$. Then, we further split each sector into N_{ampl} "ring-shaped" regions. As a consequence, the complex plane is split into $N_{\mathrm{phase}}N_{\mathrm{ampl}}$ subdomains $\{D_{ij}\}$ where D_{ij} denotes the domain corresponding to the i-th phase sector and the j-th ring-shaped region. In Figure 7.14, an illustrative example with $N_{\mathrm{phase}} = 8$ angular sectors and $N_{\mathrm{ampl}} = 2$ ring-shaped regions is shown.

By associating the fading regions with states, it is possible to describe the evolution of the fading process through the use of a Markov chain. In

multiplicative fading model assuming that the fading process has a bandwidth much smaller than the signal bandwidth.

general, considering a first-order Markov modeling for the fading process,[6] the total number of fading states is $L = N_{\text{ampl}} N_{\text{phase}}$. The probabilities of transition through different fading states can be computed by proper numerical integrations. For example, in order to evaluate the probability of transition from the region D_{ij} to the region D_{kl}, one can follow the method in [118], which is accurate as long as the first-order Markov chain modeling of the fading process holds and, in turns, corresponds to a scenario where the fading process is sufficiently *slow* [119].

Since the fading process is modeled through a Markov chain whose state corresponds to the current fading subregion D_{ij}, it is possible to derive a proper FB algorithm for the computation of the APPs of the transmitted symbols $\{a_k\}$. A general formulation accounting for a finite-memory channel depending on a generic process $\{\xi_k\}$ modeled by a Markov chain can be found in [108].

In the following, we will assume that the symbols $\{a_k\}$ are quaternary and encoded by a DE-QPSK encoder before transmission. The channel parameter ξ_k corresponds to the fading region $\hat{f}_k \in \{D_{ij}\} \quad i = 1, \ldots, N_{\text{phase}}$, $j = 1, \ldots, N_{\text{ampl}}$. The extended state described in Section 7.6.2 here is $\sigma_k = (s_k, \tilde{f}_k)$, where s_k is the DE-QPSK encoder state at epoch k, and the fading region $\tilde{f}_k = D_{ij}$ for some i, j, has been substituted to the generic parameter ξ_k.

The two essential ingredients needed for actual implementation of the Markov chain-based SISO algorithm in a scenario with fading are the transition probability $P\{\tilde{f}_{k+1} | \tilde{f}_k\}$ between the Markov chain states $\tilde{f}_k$ and $\tilde{f}_{k+1}$, obtained by suitably modeling the fading Markov chain, and the conditional PDF of the observable $p(r_k | a_k, \tilde{f}_k, s_k)$, given by the following expressions:

$$\begin{aligned} p(r_k | a_k, \tilde{f}_k, s_k) &= \frac{p(r_k, \tilde{f}_k | a_k, s_k)}{p\{\tilde{f}_k\}} \\ &= \frac{\displaystyle\int_{\tilde{f}_k} p(r_k | f, a_k, s_k) p_f(f) \mathrm{d}f}{\displaystyle\int_{\tilde{f}_k} p_f(f) \mathrm{d}f} \end{aligned} \tag{7.19}$$

where the independence between the fading process and the DE-QPSK coded data sequence c_k is exploited, $p(r_k | f, a_k, \sigma_k)$ is a Gaussian PDF (with mean $f c_k$), and $p_f(f)$ is the PDF of the fading coefficient.

[6] We remark that the considered approach can easily be extended to higher-order Markov models of the fading process, at the expense of an increased number of fading states.

The concept of detection by multiple trellises can be now directly applied to a fading channel. In particular, as for the phase-uncertain channel, if the channel is characterized by block-constant fading, Algorithm 1 is an optimum solution. In order to simplify the metric computation, the integral in (7.19) will be approximated by a finite sum of simple Gaussian metrics. However, it was observed that this can lead to numerical problems at high SNR, where the noise variance becomes small. To overcome this problem, one may increase the accuracy of the numerical integration techniques used to compute (7.19) or prevent the variance of the Gaussian pdfs to become too small and trigger numerical problems.

Observe that every concatenated scheme with a powerful error correction code is characterized by a bad BER performance below a given SNR threshold and an operational BER performance beyond this threshold.[7] If the detection algorithm assumes a given, fixed, SNR value, one is guaranteed to obtain optimal performance only when the actual SNR value equals the assumed value. The BER of the fixed-SNR receiver as a function of the SNR, is still expected to be monotonically decreasing. Therefore, if the assumed SNR is fixed to guarantee an operational BER at that very SNR value, the fixed-SNR algorithm will guarantee operational BER beyond this SNR as well. As a consequence, we chose to fix the variance of the Gaussian metric, i.e., the SNR assumed by the detection algorithm, and to make it independent of the actual noise variance. This allows to overcome numerical problems and leads to a completely blind detection algorithm, which does not need either knowledge of fading or noise statistics.

Unlike commonly considered in the literature, where the fading process used in the simulations is generated according to the considered Markov chain model, in the following the fading process used in the simulations is generated according to a realistic Clarke model.

In order to verify the effectiveness of the described detection by multiple trellises, we consider applications to DE-QPSK, both uncoded and coded by a regular (3,6) LDPC code with codeword length 32000—this length allows to counteract long fades. The code should, in fact, "observe" a received sequence long enough to accurately describe the statistics of the channel, i.e., to exploit its ergodicity. We performed simulations considering $N_{\text{ampl}} = 2$ and $N_{\text{phase}} = 16$ and considering Algorithm 1 and the above described simplified metric scheme. Algorithm 2, in the case of fading channel, exhibits

[7] In actual systems, the transition from bad BER performance to operational BER is not perfectly sharp, i.e., it happens within a small SNR region, usually referred to as *waterfall region*.

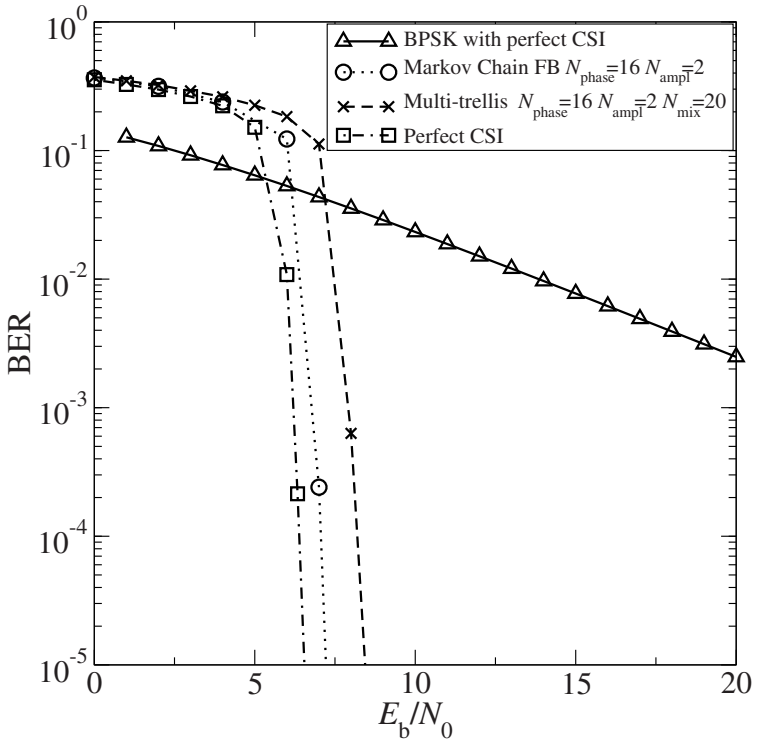

Figure 7.15: BER performance, as a function of the SNR, in a scenario with a flat Rayleigh fading channel. Various schemes are considered: (i) BPSK with perfect CSI, (ii) LDPC coded QPSK with Markov chain-based FB detection, (iii) LDPC coded QPSK with multi-trellis SISO detection, and (iv) LDPC coded QPSK with perfect CSI.

unacceptable performance, and, therefore, is not shown. This is due to the fact that the mix operation in Algorithm 2 normalizes independently every trellis thus assigning large weights to trellises characterized by incorrect fading *amplitudes*. The considered normalized Doppler rate $f_{\mathrm{D}}T$ is equal to 0.01, corresponding to a moderately fast fading channel. The obtained results are shown in Figure 7.15. The multi-trellis curve is obtained assuming a noise variance value corresponding to an SNR of about 7 dB. The inter-mix interval is heuristically optimized by trial and error and set to 20. In every LDPC coded scheme, a number of inner iterations and final LDPC decoder iterations equal to $N_i = 30$ and $N_{\mathrm{LDPC}} = 30$, respectively, is used. The Markov

chain-based algorithm presented at the beginning of this subsection is also investigated and its performance is shown. As a reference, the performance of (i) the described concatenated scheme and (ii) an uncoded BPSK signaling, both considering perfect CSI, is also shown. As one can immediately see, the performance loss incurred by the use of the described detection by multiple trellises can be quantified at about 1 dB in comparison with the Markov chain model performance and 1.8 dB compared with the perfect CSI scenario.

7.8 Concluding Remarks

In this chapter, LDPC coded modulation for differential modulation schemes has been considered. The two main aspects of LDPC coded modulation design are the design of the LDPC code and the selection/design of the proper modulation and corresponding CM-SISO. The first important consideration is that using standard LDPC codes for memoryless channels in a DE scheme leads to a significant performance loss, which can be avoided using properly designed LDPC codes. A second consideration arises from the analysis of the performance of LDPC coded modulation schemes with multiple trellises detection. Multiple trellis detection is a suboptimal detection which cannot be used in the absence of coding since its use in an uncoded scheme leads to a significant error floor. Nevertheless, in an LDPC coded modulation scheme this remarkable sub-optimality becomes negligible. This highlights the fact that a suboptimal CM-SISO scheme cannot be characterized by its performance in the absence of coding: the use of a concatenated LDPC coded modulation scheme allows powerful simplifications which might be catastrophic in an uncoded scenario.