EURASIP Journal on Applied Signal Processing 2005:6, 828–843

Trellis-Based Iterative Adaptive Blind Sequence Estimation for Uncoded/Coded Systems with Differential Precoding

Xiao-Ming Chen

Information and Coding Theory Lab, Faculty of Engineering, University of Kiel, 24143 Kiel, Germany
Email: xc@tf.uni-kiel.de

Peter A. Hoeher

Information and Coding Theory Lab, Faculty of Engineering, University of Kiel, 24143 Kiel, Germany
Email: ph@tf.uni-kiel.de

Received 1 October 2003; Revised 23 April 2004

We propose iterative, adaptive trellis-based blind sequence estimators, which can be interpreted as reduced-complexity receivers derived from the joint ML data/channel estimation problem. The number of states in the trellis is considered as a design parameter, providing a trade-off between performance and complexity. For symmetrical signal constellations, differential encoding or generalizations thereof are necessary to combat the phase ambiguity. At the receiver, the structure of the super-trellis (representing differential encoding and intersymbol interference) is explicitly exploited rather than doing differential decoding just for resolving the problem of phase ambiguity. In uncoded systems, it is shown that the data sequence can only be determined up to an unknown shift index. This shift ambiguity can be resolved by taking an outer channel encoder into account. The average magnitude of the soft outputs from the corresponding channel decoder is exploited to identify the shift index. For frequency-hopping systems over fading channels, a double serially concatenated scheme is proposed, where the inner code is applied to combat the shift ambiguity and the outer code provides time diversity in conjunction with an interburst interleaver.

Keywords and phrases: joint data/channel estimation, blind sequence estimation, iterative processing, turbo equalization.

1. INTRODUCTION

In most digital communication systems, a training sequence is inserted in each data burst for the purpose of channel estimation or for the adjustment of the taps of linear or decision-feedback equalizers. For an efficient usage of bandwidth, however, blind equalization techniques attract considerable attentions [1, 2]. Furthermore, blind detection schemes may be embedded in existing systems as an add-on in order to improve the system performance in difficult environments.

Blind linear and nonlinear equalization techniques have been investigated since the pioneering work of Sato [3]. Conventionally, blind linear equalizers exploit the higher-order statistical relationship between the data signal and the equalizer output signal. On-line adaptive algorithms based on the zero-forcing principle have been proposed in [3, 4, 5], for example. For burst-wise transmission, an iterative batch implementation of these algorithms is also possible [6], that is, the equalizer coefficients obtained at the end of one iteration are employed as the initial values in the next iteration. Based on the minimum mean-square error (MMSE) criterion, algorithms for blind identification and blind equalization have been proposed in [7, 8] for multipath fading channels. Possible drawbacks of linear blind equalizers are, depending on the algorithm, a slow convergence rate, a possible convergence to local minima, and a lack of robustness against Doppler spread, noise, and interference.

Given the equivalent discrete-time channel model, an intersymbol interference (ISI) channel can be interpreted as a nonlinear convolutional code, which can be described by means of a trellis diagram or a tree diagram. Accordingly, any trellis-based or tree-based [9] sequence estimation technique can be used to perform data estimation. As a counterpart to maximum-likelihood sequence estimation (MLSE) with known coefficients of the equivalent discrete-time channel model (which are referred to as channel coefficients in the sequel), nonlinear blind equalization techniques by means of the expectation-maximization (EM) algorithm were derived from the maximum-likelihood estimation principle in [10, 11]. Moreover, adaptive channel estimators may be combined with blind sequence estimation, as shown in [12, 13, 14, 15]. Thereby adaptive channel estimators (e.g., based on least mean square (LMS), recursive least squares (RLS) or the Kalman algorithm [16]) are implemented in parallel to a blind trellis-based equalizer. Possible equalizers may be based on the Viterbi algorithm (VA), on per-survivor processing (PSP) [17], or on the list Viterbi algorithm (LVA) [18]. For equalizers based on the VA, a single-channel estimator is recursively updated by the locally best survivor given a suitable tentative decision delay [19, Chapter 11]. With PSP, each survivor employs its own channel estimator and no decision delay is afforded. In the LVA, for each trellis state, more than one survivor is maintained. Different from the case with known channel coefficients, the number of states in the trellis should be considered as a design parameter, which provides a trade-off between complexity and performance. In order to exploit statistical properties of the multipath fading channel and to track the time variation of the channel, model-fitting algorithms were used in [20, 21], for example. In this context, channel coefficients are modeled as complex Gaussian-distributed random variables, where the covariance matrix of channel coefficients are assumed to be known at the receiver. All these techniques can be applied straightforwardly to any tree-based sequential decoding algorithm, for example, by means of the breadth-first sequential decoding algorithm as shown in [22]. In contrast to blind linear equalizers, all these trellis-based or tree-based approaches explicitly exploit the finite-alphabet property of data sequences.

The focus of this paper is on trellis-based blind sequence estimation for short burst sizes and noisy environments, where the only available channel knowledge is an upper bound on the channel order. Significant improvements with respect to acquisition and bit error rate (BER) performance are particularly obtained by incorporating on-line adaptive channel estimation into the equalizer, by performing iterative processing in the blind sequence estimator, and by using a priori information about data symbols, for example, provided by an outer soft-output channel decoder or by exploiting the residual correlation in the data sequence after the source encoder [23]. As opposed to the optimal receiver in the sense of MLSE, the reduced-complexity trellis-based blind sequence estimators considered here do not perform an exhaustive search over all possible data hypotheses. Therefore, they may converge to local minima as observed in [12, 13, 14]. In this paper, we propose different approaches to combat phase ambiguity, shift ambiguity, and other local minima of the cost function. If the channel order is overdetermined, the data sequence can be only estimated up to an unknown shift index for uncoded systems. On the other hand, for coded schemes, this shift ambiguity can be resolved by exploiting code constraints. As opposed to the common understanding that differential encoding is used just to resolve the phase ambiguity of channel and data estimation, we explicitly use the structure of the super-trellis. Besides incorporating a priori information, the proposed trellis-based blind equalizer is also able to deliver soft outputs to subsequent processing stages. Consequently, a blind turbo processor can be obtained, which is composed of an inner blind soft-input soft-output (SISO) equalizer and an outer SISO channel decoder. For blind turbo equalization of frequency-hopping systems over fading channels, we propose a novel transmitter/receiver structure with double serial concatenations. The inner concatenation is necessary to combat the shift ambiguity, while the outer concatenation exploits time diversity of channel codes in conjunction with an interburst interleaver.

In Section 2, we present the system model under investigation. Reduced-complexity trellis-based blind equalization techniques are derived from the ML joint data/channel estimation problem in Section 3, which also shows the inherent relationship between these techniques. The initialization issue and techniques to combat local minima are discussed in Section 4. A summary of the proposed adaptive blind sequence estimator and simulation results for an uncoded GSM-like system are also presented in Section 4. Taking the outer channel decoder into consideration, we propose a blind turbo equalizer in Section 5, where the effect of phase/shift ambiguity on the coded system and corresponding solutions are also investigated. After providing numerical results for coded systems, some conclusions are drawn in Section 6.

2. SYSTEM MODEL

Throughout this paper we use the complex baseband notation. In the following, $(\cdot)^T$, $(\cdot)^*$, $(\cdot)^H$, and $(\cdot)^\dagger$ stand for transpose, complex conjugate, complex conjugate and transpose, and Moore-Penrose pseudo left inverse, respectively.

2.1. Transmitter

Within this paper, the focus is on an M-ary DPSK system. The task of the differential encoder is to resolve the phase ambiguity. The output symbols of the differential encoder can be written as

$$x[k] = x[k-1]d[k], \quad x[0] = +1, \quad 1 \le k \le K, \tag{1}$$

where $d[k]$ are M-ary PSK data symbols with unit symbol energy, $x[0] = +1$ serves as a reference symbol, and K is the burst length (excluding the reference symbol). A generalization to other symmetrical signal constellations with precoding (e.g., CPM) is possible.

2.2. Channel model

The pulse shaping filter, the frequency-selective channel, the receiving filter, and the sampling can be represented by a tapped-delay-line baud-rate model. (We restrict ourselves to baud-rate sampling. An extension to fractionally spaced sampling is straightforward. The validity of the tapped-delay-line model has been discussed for an unknown channel in [24, 25].) The corresponding outputs of the equivalent discrete-time channel model can be written as

$$\begin{aligned} y[k] &= \sum_{l=0}^{L} h_l[k]x[k-l] + n[k] \\ &= \mathbf{x}^T[k]\mathbf{h}[k] + n[k], \quad 0 \le k \le K, \end{aligned} \tag{2}$$

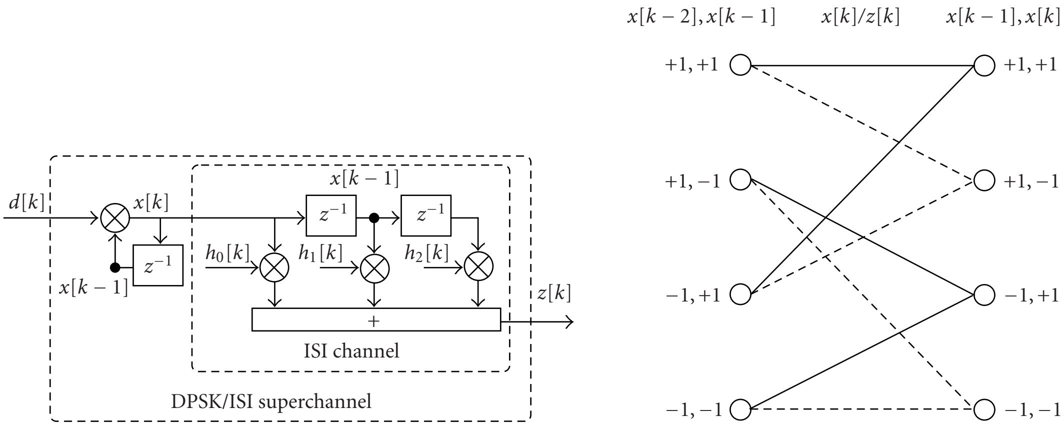


Figure 1: ISI channel model and ISI trellis for the binary case with $L = 2$.

where $\mathbf{h}[k] = [h_L[k], h_{L-1}[k], \ldots, h_0[k]]^T$ is the time-varying channel coefficient vector with normalized power, L is the effective channel memory length after suitable truncation, and $\{n[k]\}$ is assumed to be an additive white Gaussian noise (AWGN) sequence with variance σ_n^2 per sample. Moreover, $\mathbf{x}[k] = [x[k-L], \ldots, x[k]]^T$ denotes the state transitions of the kth trellis segment.

For a burst-wise transmission, the channel model can be represented in vector/matrix notation as

$$\mathbf{y} = \mathbf{X}\mathbf{h} + \mathbf{n}, \tag{3}$$

where $\mathbf{y} = [y[0], \ldots, y[K]]^T$, $\mathbf{X} = [\mathbf{x}[0], \ldots, \mathbf{x}[K]]^T$, and $\mathbf{n} = [n[0], \ldots, n[K]]^T$. Moreover, $\mathbf{h} = [h_L, \ldots, h_0]^T$ is assumed to be constant within a burst. (If the data symbols are not transmitted on a burst-by-burst basis, if the burst size is large, or if the channel is fast time varying, K may denote the length of a subburst.)

2.3. Receiver

The task of the receiver based on the maximum-likelihood sequence estimation strategy is twofold. Primarily, we are interested in an estimate of the data vector $\mathbf{d} = [d[1], d[2], \ldots, d[K]]^T$. A pseudocoherent receiver (according to the definition in [26]) must also obtain estimates of each element of $\mathbf{h}$ in amplitude and phase.

In a pseudocoherent receiver, joint data/channel estimation may be based on the ISI trellis (followed by differential decoding), or may be based on the DPSK/ISI super-trellis, which combines the differential encoding and the ISI trellis. When differential encoding is used, a receiver based on the ISI trellis followed by differential encoding is equivalent to the receiver based on the super-trellis if and only if the transmitted symbols are independent and uniformly distributed. If this is not the case, only the latter receiver can be optimal. In the following, only the latter receiver is investigated.

Figure 1 shows the ISI channel model and the corresponding ISI trellis for the case when $L = 2$ and $M = 2$. Taking the differential encoder into account, the equivalent DPSK/ISI superchannel and the corresponding DPSK/ISI super-trellis are depicted in Figure 2. Note that the number of states is not increased by differential encoding. While the data symbol after differential encoding, namely, $x[k]$, labels state transitions in the ISI trellis, the transition label changes to $d[k]$ in the DPSK/ISI super-trellis. As indicated in Figure 2, the DPSK/ISI super channel can be interpreted as a recursive encoder, which is preferable for serially concatenated turbo schemes [27]. In the following, our blind sequence estimator operates on the DPSK/ISI super-trellis. Furthermore, the differential encoder may be replaced by other recursive rate-1 precoders, which are able to combat the phase ambiguity, for example, any generalized differential encoder shown in [28]. Although only the differential encoder is considered within this paper, the proposed receiver can easily be extended to other suitable recursive precoders or modulation schemes with inherent differential encoding like CPM.

3. REDUCED-COMPLEXITY RECEIVERS DERIVED FROM THE ML JOINT DATA/CHANNEL ESTIMATOR

In this Section, reduced-complexity receivers for blind sequence estimation are derived from the ML joint data/channel estimation problem, where both data sequence and channel coefficients are unknown. Previously proposed algorithms are shown to be special cases of the proposed receiver. In the following, $\tilde{\phi}$ and $\hat{\phi}$ denote hypotheses and corresponding estimates of ϕ, respectively, where ϕ may be a scalar, a vector, or a matrix.

The ML joint data/channel estimation problem in the presence of AWGN can be formulated as

$$(\hat{\mathbf{x}}, \hat{\mathbf{h}}) = \arg\max_{\tilde{\mathbf{x}}, \tilde{\mathbf{h}}} \{p(\mathbf{y} \mid \tilde{\mathbf{x}}, \tilde{\mathbf{h}})\} = \arg\min_{\tilde{\mathbf{X}}, \tilde{\mathbf{h}}} \{\|\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{h}}\|^2\}, \tag{4}$$

where $p(\mathbf{y} \mid \tilde{\mathbf{x}}, \tilde{\mathbf{h}})$ denotes the probability density function of the received vector conditioned on data and channel

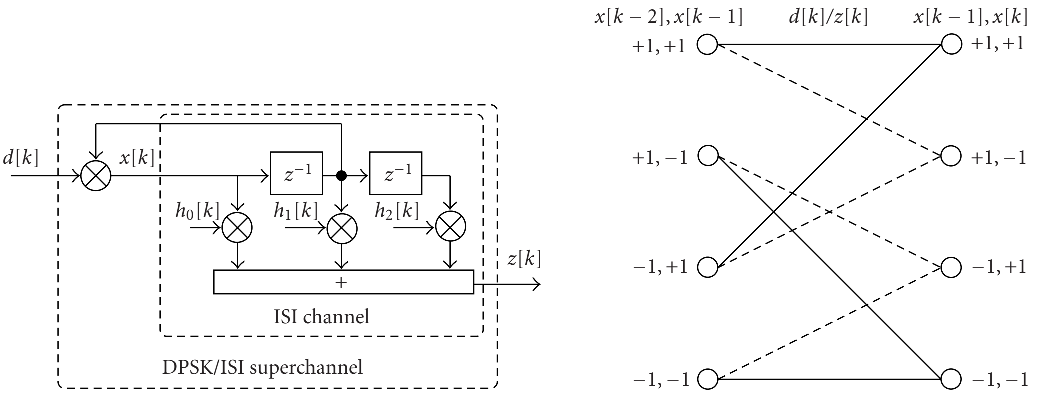


Figure 2: DPSK/ISI channel model and DPSK/ISI super-trellis for the binary case with $L = 2$.

hypotheses. The ML sequence can be written as

$$\begin{aligned}\hat{\mathbf{x}} &= \arg\min_{\tilde{\mathbf{X}}} \left\{ \arg\min_{\tilde{\mathbf{h}}} \|\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{h}}\|^2 \right\} \\ &= \arg\min_{\tilde{\mathbf{X}}} \left\{ \|\mathbf{y} - \tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\dagger}\mathbf{y}\|^2 \right\},\end{aligned} \tag{5}$$

where $\tilde{\mathbf{X}}^{\dagger}\mathbf{y}$ is the least-squares channel estimate (LS-CE) based on the data matrix hypothesis $\tilde{\mathbf{X}}$. From (5), the optimal solution for the joint estimation problem (4) necessitates performing the LS channel estimation for all possible data hypotheses. The complexity of this exhaustive search approach inhibits its applications for practical burst lengths, however.

The so-called projection matrix $\tilde{\mathbf{X}}_p \triangleq \tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\dagger}$ projects the channel output vector $\mathbf{y}$ onto the subspace spanned by the columns of $\tilde{\mathbf{X}}$, and $\tilde{\mathbf{X}}_p$ exhibits the following special properties:

$$\tilde{\mathbf{X}}_p^H = \left(\tilde{\mathbf{X}}(\tilde{\mathbf{X}}^H\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^H\right)^H = \tilde{\mathbf{X}}\tilde{\mathbf{X}}^{\dagger} = \tilde{\mathbf{X}}_p, \tag{6}$$

$$(\tilde{\mathbf{X}}e^{j\theta})_p = (\tilde{\mathbf{X}}e^{j\theta})(\tilde{\mathbf{X}}^H\tilde{\mathbf{X}})^{-1}(\tilde{\mathbf{X}}e^{j\theta})^H = \tilde{\mathbf{X}}_p, \tag{7}$$

$$\tilde{\mathbf{X}}_p\tilde{\mathbf{X}}_p = \tilde{\mathbf{X}}(\tilde{\mathbf{X}}^H\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^H\tilde{\mathbf{X}}(\tilde{\mathbf{X}}^H\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^H = \tilde{\mathbf{X}}_p, \tag{8}$$

where the matrix $\tilde{\mathbf{X}}^H\tilde{\mathbf{X}}$ is assumed to be nonsingular. Consequently, the ML joint data/channel estimator can be rewritten as

$$\hat{\mathbf{x}} = \arg\min_{\tilde{\mathbf{X}}} \left\{ \|\mathbf{y} - \tilde{\mathbf{X}}_p\mathbf{y}\|^2 \right\} = \arg\min_{\tilde{\mathbf{X}}} \left\{ -\mathbf{y}^H\tilde{\mathbf{X}}_p\mathbf{y} \right\}, \tag{9}$$

where $-\mathbf{y}^H\tilde{\mathbf{X}}_p\mathbf{y}$ can be interpreted as the path metric associated with the data hypothesis $\tilde{\mathbf{x}}$.

Equation (7) implies that there exists a phase ambiguity for symmetrical signal constellations. For example, in the binary antipodal case, $\tilde{\mathbf{x}}$ and $-\tilde{\mathbf{x}}$ are indistinguishable for the ML receiver. The phase ambiguity can be resolved by means of differential encoding or generalizations thereof.

Because the only available channel knowledge at the receiver is an upper-bounded channel order, $L_u \geq L$, the blind sequence estimator presumes the following channel model:

$$y[k] = \sum_{l=0}^{L_u} h_l x[k-l] + n[k] = \mathbf{x}^T[k]\mathbf{h} + n[k], \tag{10}$$

where we redefine $\mathbf{x}[k] \triangleq [x[k-L_u], \ldots, x[k]]^T$ and $\mathbf{h} \triangleq [h_{L_u}, \ldots, h_0]^T$. The channel model (3) is correspondingly changed with respect to $\mathbf{X}$ and $\mathbf{h}$ (with modified $\mathbf{x}[k]$ and $\mathbf{h}$) in the context of blind sequence estimation. Throughout this paper, (10) is applied for the blind sequence estimation, while (2) is suitable for equalizers with known channel coefficients. For the case $L_u = L$, (10) reduces to (2). For the case $L_u > L$, that is, the channel order is overdetermined, there exists a shift ambiguity even for the ML receiver. For the example that $L_u = L + 1$, two data sequences $\hat{x}^1[k] = x[k]$ and $\hat{x}^2[k] = x[k+1]$ are indistinguishable for the receiver due to

$$y[k] = \sum_{l=0}^{L_u} \hat{h}_l^1 x[k-l] + n[k] = \sum_{l=0}^{L_u} \hat{h}_l^2 x[k+1-l] + n[k], \tag{11}$$

where $\hat{\mathbf{h}}^1 = [h_0^1, \ldots, h_{L_u}^1]^T = [h_0, \ldots, h_L, 0]^T$ and $\hat{\mathbf{h}}^2 = [h_0^2, \ldots, h_{L_u}^2]^T = [0, h_0, \ldots, h_L]^T$. Accordingly, the transmitted data sequence can only be determined up to an unknown shift index. For the case $L_u < L$, the channel order is underdetermined, which results in residual ISI and consequently degrades the receiver performance.

A suboptimal solution of (4) can be obtained by exploring 2^{L_t+1} paths in a trellis with 2^{L_t} states (the subscript $(\cdot)_t$ abbreviates "trellis") rather than performing an exhaustive search, which takes 2^{K+1} paths into account. The memory length of the expanded trellis $L_t \geq L_u$ is a design parameter, which provides a trade-off between performance and complexity. A larger L_t results in a higher computational complexity, which implies that more paths are retained for the joint data/channel estimation. Therefore, a better performance of the receiver with a larger L_t can be expected compared to the receiver with a smaller L_t. We may define the path metrics corresponding to L_t as follows:

$$\sum_{k=0}^{K} \|\mathbf{y}[k] - \tilde{\mathbf{X}}[k] \cdot \hat{\mathbf{h}}(\tilde{\mathbf{x}}_t[k])\|^2, \tag{12}$$

where $\mathbf{y}[k] = [y[k+L_u-L_t], \ldots, y[k]]^T$ and $\tilde{\mathbf{X}}[k] = [\tilde{\mathbf{x}}[k+L_u-L_t], \ldots, \tilde{\mathbf{x}}[k]]^T$. The estimated channel coefficient vector for state transitions is denoted as $\hat{\mathbf{h}}(\tilde{\mathbf{x}}_t[k])$, where state transitions $\tilde{\mathbf{x}}_t[k] = [\tilde{x}[k-L_t],\ldots,\tilde{x}[k]]^T$ are determined by the current state $\tilde{\mathbf{s}}_t[k] = [\tilde{x}[k-L_t+1],\ldots,\tilde{x}[k]]^T$ and its predecessor $\tilde{\mathbf{s}}_t[k-1]$.

Depending on how to determine the channel coefficients $\hat{\mathbf{h}}(\tilde{\mathbf{x}}_t[k])$, different algorithms can be derived.

3.1. Two-step iterative alternating data/channel estimation

If the estimated channel coefficient vector remains unchanged over the whole burst, that is, if $\hat{\mathbf{h}}(\tilde{\mathbf{x}}_t[k]) = \hat{\mathbf{h}}$, (12) is simplified as

$$\sum_{k=0}^{K} \|\mathbf{y}[k] - \tilde{\mathbf{X}}[k]\hat{\mathbf{h}}\|^2 = (L_t - L_u + 1)\sum_{k=0}^{K} \|y[k] - \tilde{\mathbf{x}}^T[k]\hat{\mathbf{h}}\|^2. \tag{13}$$

Hence, a Viterbi equalizer with channel memory length L_t will deliver the same result as another Viterbi equalizer with channel memory length L_u, if the same estimated channel coefficients are used in both equalizers.

Given the data estimates obtained by the Viterbi equalizer, denoted as $\hat{\mathbf{x}}$, LS channel estimation can be performed as

$$\hat{\mathbf{h}} = \arg\min_{\tilde{\mathbf{h}}} \{\|\mathbf{y} - \hat{\mathbf{X}}\tilde{\mathbf{h}}\|^2\} = (\hat{\mathbf{X}}^H\hat{\mathbf{X}})^{-1}\hat{\mathbf{X}}^H\mathbf{y} = \hat{\mathbf{X}}^{\dagger}\mathbf{y}. \tag{14}$$

If the data correlation matrix $\hat{\mathbf{X}}^H\hat{\mathbf{X}}$ is rank deficient, channel estimation may be carried out using the singular value decomposition [16]. The channel estimate (14) is applied for the sequence estimation in the next iteration. This two-step alternating blind equalizer has been investigated in [29, 30] for the case $L_u = L$. A sufficiently large burst length and a priori information about the channel coefficients are necessary in [29] to get a satisfying performance. In [30], a short training sequence is afforded to get reasonable results.

If the Viterbi equalizer is replaced by a symbol-by-symbol maximum a posteriori (MAP) equalizer, we obtain a blind sequence estimator based on the EM algorithm. Applying conditional a posteriori probabilities (APPs) of state transitions $\tilde{\mathbf{x}}[k]$, denoted as $P(\tilde{\mathbf{x}}[k] \mid \mathbf{y}, \hat{\mathbf{\Theta}}^{(i)})$, the channel coefficients and the noise variance are estimated as follows [11]:

$$\begin{aligned}\hat{\mathbf{h}}^{(i+1)} &= \left(\sum_k \sum_{\tilde{\mathbf{x}}[k]} P(\tilde{\mathbf{x}}[k] \mid \mathbf{y}, \hat{\mathbf{\Theta}}^{(i)})\tilde{\mathbf{x}}^*[k]\tilde{\mathbf{x}}^T[k]\right)^{-1} \\ &\quad\times\left(\sum_k \sum_{\tilde{\mathbf{x}}[k]} P(\tilde{\mathbf{x}}[k] \mid \mathbf{y}, \hat{\mathbf{\Theta}}^{(i)})\tilde{\mathbf{x}}^*[k]y[k]\right),\end{aligned} \tag{15}$$

$$\hat{\sigma}_n^{2^{(i+1)}} = \frac{\sum_k \sum_{\tilde{\mathbf{x}}[k]} P(\tilde{\mathbf{x}}[k] \mid \mathbf{y}, \hat{\mathbf{\Theta}}^{(i)}) \left| y[k] - \tilde{\mathbf{x}}^T[k]\hat{\mathbf{h}}^{(i+1)} \right|^2}{\sum_k \sum_{\tilde{\mathbf{x}}[k]} P(\tilde{\mathbf{x}}[k] \mid \mathbf{y}, \hat{\mathbf{\Theta}}^{(i)})}, \tag{16}$$

where $\hat{\mathbf{\Theta}}^{(i)} = [\hat{\mathbf{h}}^{(i)T}, \hat{\sigma}_n^{2^{(i)}}]^T$ is the estimated channel parameter vector at the end of the ith iteration. $\hat{\mathbf{\Theta}}^{(i)}$ is considered as constant within the $(i+1)$th iteration. The conditional APPs $P(\tilde{\mathbf{x}}[k] \mid \mathbf{y}, \hat{\mathbf{\Theta}}^{(i)})$ can efficiently be evaluated using a forward and backward recursion, which can be well approximated by the max-log-APP algorithm [31] with a significantly reduced complexity.

Equation (15) essentially approximates an MMSE channel estimator conditioned on $\hat{\mathbf{\Theta}}^{(i)}$, that is,

$$\hat{\mathbf{h}}^{(i+1)} \approx \{E[\mathbf{x}^*[k]\mathbf{x}^T[k] \mid \mathbf{y}, \hat{\mathbf{\Theta}}^{(i)}]\}^{-1} E[\mathbf{x}^*[k]y[k] \mid \mathbf{y}, \hat{\mathbf{\Theta}}^{(i)}], \tag{17}$$

where the expectation is performed over the data sequence.

Using the approximations $P(\tilde{\mathbf{x}} = \hat{\mathbf{x}} \mid \mathbf{y}, \hat{\mathbf{\Theta}}^{(i)}) \approx 1$ and $P(\tilde{\mathbf{x}} \neq \hat{\mathbf{x}} \mid \mathbf{y}, \hat{\mathbf{\Theta}}^{(i)}) \approx 0$, (15) and (16) reduce to

$$\hat{\mathbf{h}}^{(i+1)} = \left(\sum_k \hat{\mathbf{x}}^*[k]\hat{\mathbf{x}}^T[k]\right)^{-1}\left(\sum_k \hat{\mathbf{x}}^*[k]y[k]\right), \tag{18}$$

$$\hat{\sigma}_n^{2^{(i+1)}} = \frac{1}{K+1}\sum_k \left| y[k] - \hat{\mathbf{h}}^{(i+1)T}\hat{\mathbf{x}}[k] \right|^2, \tag{19}$$

where (18) coincides with (14) and $\hat{\mathbf{x}}$ is obtained by means of the Viterbi algorithm using $\hat{\mathbf{h}}^{(i)}$ as channel coefficients. Therefore, the approaches proposed in [29, 30] can be regarded as simplified EM-based blind sequence estimators. While (18) and (19) can be interpreted as channel estimation based on *hard* decisions $\{\hat{x}[k]\}$, (15) and (16) offer channel estimates based on *soft* decisions $P(\tilde{\mathbf{x}}[k] \mid \mathbf{y}, \hat{\mathbf{\Theta}}^{(i)})$.

Through the iterative procedure, namely, (15) and (16), the likelihood function $p(\mathbf{y} \mid \hat{\mathbf{\Theta}}^{(i)})$ is verified to be a nondecreasing function [32]. On the other hand, as pointed out in [33], the EM solution only fulfills a necessary condition of the ML estimation, that is, the EM algorithm may converge to local maxima. Other drawbacks of the EM algorithm are its sensitivity to the initialization of unknown parameters and a possibly slow convergence. As a simplified EM algorithm, the Viterbi equalizer in conjunction with LS-CE exhibits similar drawbacks.

3.2. Trellis-based adaptive blind sequence estimation (TABSE)

In order to improve the system performance with respect to acquisition and to deal with time-varying channels, the channel coefficients $\hat{\mathbf{h}}(\tilde{\mathbf{x}}_t[k])$ are recursively estimated during the data estimation procedure.

If the estimated channel vector is independent of state transitions in the trellis, that is, if $\hat{\mathbf{h}}(\tilde{\mathbf{x}}_t[k]) = \hat{\mathbf{h}}[k]$, there is a unique channel estimator in the blind sequence estimator. The update of channel estimation is based on delayed tentative decisions of the locally best survivor. If the estimated channel vector is solely determined by the predecessor of state transitions, that is, $\hat{\mathbf{h}}(\tilde{\mathbf{x}}_t[k]) = \hat{\mathbf{h}}(\tilde{\mathbf{s}}_t[k-1])$, each state maintains a channel estimator corresponding to the PSP principle. If the estimated channel vector is determined by state transitions, the update for channel estimation is performed for each branch in the trellis, which is termed per-branch processing (PBP) [34]. While in PSP the add-compare-selection operation is done before the channel adaptation, the order of these two operations is reversed in PBP.

Another important difference of the proposed adaptive blind sequence estimator from the approaches presented in Section 3.1 lies in the evaluation of branch metrics. In the TABSE, branch metrics $\|\mathbf{y}[k] - \tilde{\mathbf{X}}[k] \cdot \widehat{\mathbf{h}}(\tilde{\mathbf{x}}_t[k])\|^2$ are evaluated based on the time-varying channel coefficients $\widehat{\mathbf{h}}(\tilde{\mathbf{x}}_t[k])$. Moreover, branch metrics $\|\mathbf{y}[k] - \tilde{\mathbf{X}}[k] \cdot \widehat{\mathbf{h}}(\tilde{\mathbf{x}}_t[k])\|^2$ are actually path metrics of short paths with length $L_t - L_u + 1$. At each time index, the blind sequence estimator traces paths in the trellis back to a certain depth for the evaluation of short-path metrics based on updated channel coefficients, which may be interpreted as extended PSP/PBP. (For the case $L_t = L_u$, it coincides with original PSP/PBP; short-path metrics are reduced to conventional branch metrics.) Using short-path metrics as branch metrics makes, on average, the difference of considered path metrics larger than using conventional branch metrics. Therefore, on average the proposed receiver delivers better data/channel estimates than standard PSP/PBP-based approaches.

Blind acquisition performances of TABSEs based on the LMS and the RLS algorithms have been explored in [12, 14, 15] for uncoded systems, respectively. For burst-wise transmission, we have investigated iterative TABSEs and soft-input soft-output counterparts thereof in [13, 35]. Details will be discussed in the sequel.

4. ITERATIVE TRELLIS-BASED ADAPTIVE BLIND SEQUENCE ESTIMATION

In this section, the initialization issue of TABSEs is firstly investigated. Afterward, we consider the problem of local minima in the context of the blind sequence estimation and propose possible solutions. Finally, a concise description of the proposed *iterative* adaptive blind sequence estimator will be given, followed by numerical results for an uncoded GSM-like system.

4.1. Initialization issue

Empirically, the central tap of linear blind equalizers is set to one, where all other taps are set to zero [2]. For the TABSE, the initial guess about the channel coefficients should be set to all-zero, if there is no a priori information available about channel coefficients. In order to obtain better initial values compared to the all-zero initialization, several algorithms have been proposed. One possibility stated in [19, Chapter 11] is to perform LS channel estimation over all possible data sequences with a short length N_s ($L_u + 1 \leq N_s \ll K$). Afterward, blind trellis-based equalization using PSP or the LVA can be performed. Due to the short length of subbursts, the probability for a singularity, equivalence, or indistinguishability of data sequences is high [14]. With increasing subburst length, the initialization can be improved at the expense of increased complexity. Another initialization strategy was introduced in [36], where a successive refinement of channel estimation is carried out over a quantized grid. For small quantization steps and a relatively long burst length, a high complexity can be expected. Therefore, we only consider the all-zero initialization in this paper.

4.2. Local minima

Because only a constrained number of paths is retained to perform joint data/channel estimation, the blind sequence estimator may converge to a wrong set of channel coefficients, corresponding to a local minimum of the cost function. An example of local minima is the shift ambiguity as observed in [12, 13, 14]. In the binary case, shift ambiguity causes channel estimates $\widehat{h}_l = \pm h_{l+\kappa}$, where $\kappa \in \{0, \pm 1, \pm 2, \ldots, \pm L_u\}$. In the absence of decision errors, the corresponding data estimates are $\widehat{x}[k] = \mp x[k - \kappa]$. The main problem related to the shift ambiguity is that κ channel coefficients are shifted out of the observation interval $L_u + 1$. To resolve this shift ambiguity, we propose to perform LS channel estimation for estimated data sequence with different shifts. Assuming $\widehat{\mathbf{X}}$ is the estimated data matrix after convergence, matrices $\widehat{\mathbf{X}}^{(m)}$ are constructed according to $\widehat{x}^{(m)}[k] = \widehat{x}[k + m]$ for $-L_u \leq m \leq L_u$. Accordingly, the shift index is estimated through the following equation (compare (5) and (14)):

$$\widehat{\kappa} = \arg\min_{m} \left\{ \left|\left|\mathbf{y} - \widehat{\mathbf{X}}^{(m)} \widehat{\mathbf{X}}^{(m)\dagger} \mathbf{y}\right|\right|^2 \right\}. \tag{20}$$

A nice feature of trellis-based blind equalization is the possibility to make use of a priori information about the data symbols and to deliver soft outputs to subsequent processing stages. Incorporating a priori information of the data symbols provides an efficient solution to combat other local minima besides the shift ambiguity.

4.3. Summary of proposed iterative TABSE

A concise description of the proposed iterative TABSE is as follows.

(1) *Initialization:* the channel coefficients are initialized to be zero: $\widehat{h}_l^{(1)}[0] = 0$, $0 \leq l \leq L_u$.

(2) *Recursive adaptive channel estimation:* in case of PSP equalization in conjunction with LMS channel estimation, the adaptive channel estimator can be written as

$$\mathbf{e}^{(i)}(\tilde{\mathbf{s}}_t[k]) = \mathbf{y}[k] - \widehat{\mathbf{X}}^{(i)}(\tilde{\mathbf{s}}_t[k])\widehat{\mathbf{h}}^{(i)}(\tilde{\mathbf{s}}_t[k-1]), \tag{21}$$

$$\widehat{\mathbf{h}}^{(i)}(\tilde{\mathbf{s}}_t[k]) = \widehat{\mathbf{h}}^{(i)}(\tilde{\mathbf{s}}_t[k-1]) + \triangle \widehat{\mathbf{X}}^{H(i)}(\tilde{\mathbf{s}}_t[k])\mathbf{e}^{(i)}(\tilde{\mathbf{s}}_t[k]), \tag{22}$$

where $\widehat{\mathbf{X}}^{(i)}(\tilde{\mathbf{s}}_t[k])$, $\widehat{\mathbf{h}}^{(i)}(\tilde{\mathbf{s}}_t[k])$, $\mathbf{e}^{(i)}(\tilde{\mathbf{s}}_t[k])$, and $\triangle$ are the tentatively decided data matrix consistent with $\tilde{\mathbf{s}}_t[k]$, the estimated channel coefficient vector, the corresponding a priori estimation error vector, and the LMS step size, respectively. Moreover, $1 \leq i \leq N_{\text{iter}}$ is the iteration index, and N_{iter} denotes the given maximum number of iterations.

(3) *Shift ambiguity compensation:* at the end of each iteration, the shift ambiguity is compensated using the estimated data sequence obtained in step (2) by means of (20). Note that (20) tends to improve the channel estimation obtained in the current iteration. The channel estimate corresponding to the best shift index is used as the initial channel estimate in the next iteration.

Table 1: Shift ambiguity in estimated channel coefficients.

Actual channel coefficients	$\Re\{h_0\}$	$\Re\{h_1\}$	$\Re\{h_2\}$	$\Re\{h_3\}$	$\Im\{h_0\}$	$\Im\{h_1\}$	$\Im\{h_2\}$	$\Im\{h_3\}$
$\mathbf{h}_1$	−0.106	−0.410	−0.104	−0.001	0.083	0.429	0.228	−0.005
$\mathbf{h}_2$	−0.094	−0.809	−0.558	0.004	−0.094	−0.156	0.137	0.005
Estimated channel coefficients	$\Re\{\widehat{h}_0\}$	$\Re\{\widehat{h}_1\}$	$\Re\{\widehat{h}_2\}$	$\Re\{\widehat{h}_3\}$	$\Im\{\widehat{h}_0\}$	$\Im\{\widehat{h}_1\}$	$\Im\{\widehat{h}_2\}$	$\Im\{\widehat{h}_3\}$
$\widehat{\mathbf{h}}_1$	−0.011	0.105	0.410	0.101	0.006	−0.084	−0.429	−0.225
$\widehat{\mathbf{h}}_2$	0.000	0.101	0.808	0.551	−0.011	0.093	0.158	−0.136

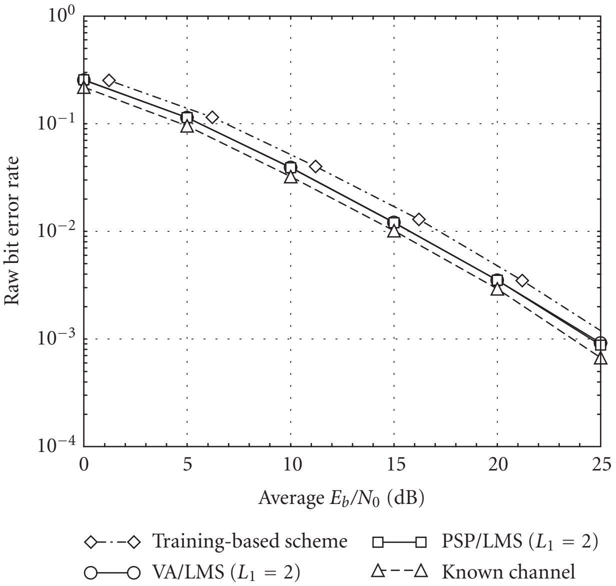


Figure 3: Raw BER versus SNR for RA channel model.

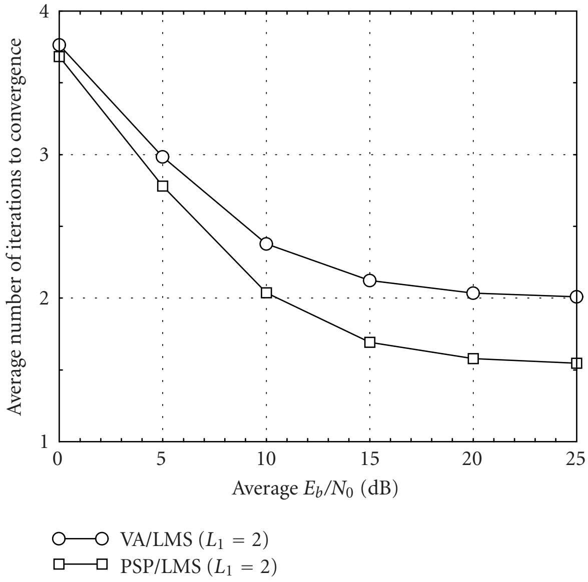


Figure 4: Average number of iterations of different algorithms to convergence for RA channel model, $N_{\text{iter}} = 10$.

(4) *Final data estimate:* steps (2) and (3) are repeated until $i = N_{\text{iter}}$ or until a convergence of the estimated data sequence is observed, which gives the final data decision.

4.4. Numerical results for uncoded transmission

The performance of the proposed blind sequence estimator was tested over a GSM-like system with burst length $K = 148$. At the transmitter, binary DPSK symbols are passed through a linearized Gaussian shaping filter, while a root-raised cosine filter is used as a receiving filter. The GSM05.05 rural area (RA) and typical urban (TU) channel models were taken into consideration. For the RA channel model, the memory length of channel model was fixed to be $L_u = 2$, while for the TU channel model $L_u = 3$ was selected.

The problem of shift ambiguity is illustrated in Table 1 for the TU channel model. The estimated channel coefficients are shifted to the right by one symbol (the phase ambiguity is uncritical due to differential encoding). Consequently, the estimated data sequences will be shifted by one symbol to the left compared to the transmitted data sequences, that is, we have a BER of around 50% for such bursts. To eliminate this effect due to shift ambiguity, for the evaluation of the BER of uncoded systems we shift the estimated data sequence by $\pm L_u$ symbols and select the one with the lowest number of errors.

For comparison, simulation results were also shown for the case of known channel coefficients and a training-based scheme (where a GSM training sequence of length 26 is used for the LS channel estimation). The signal-to-noise ratio (SNR) loss due to the training sequence was taken into account. The final decision delay in all equalizers was selected to be $2(L_u + 1)$. For the Viterbi equalizer in conjunction with an LMS adaptive channel estimator (abbreviated as VA/LMS), the tentative decision delay is selected to be 5 symbols. The step size of LMS channel estimation is selected to be $\triangle = 0.1$ in the first iteration for a fast convergence, while for remaining iterations it is chosen to be $\triangle = 0.01$ for refinement of channel estimation. For SNRs $<$ 20 dB, 10^4 quasi-static bursts were generated, that is, channel coefficients remain constant within a burst and are statistically independent from burst to burst. For SNRs $\geq$ 20 dB, the number of bursts is 10^5.

Figure 3 shows the BER performance for the RA channel model. Both VA/LMS and PSP/LMS blind sequence estimators outperform the training-based scheme and show a similar BER performance. As indicated in Figure 4, the VA/LMS receiver exhibits a slower convergence rate than the PSP/LMS

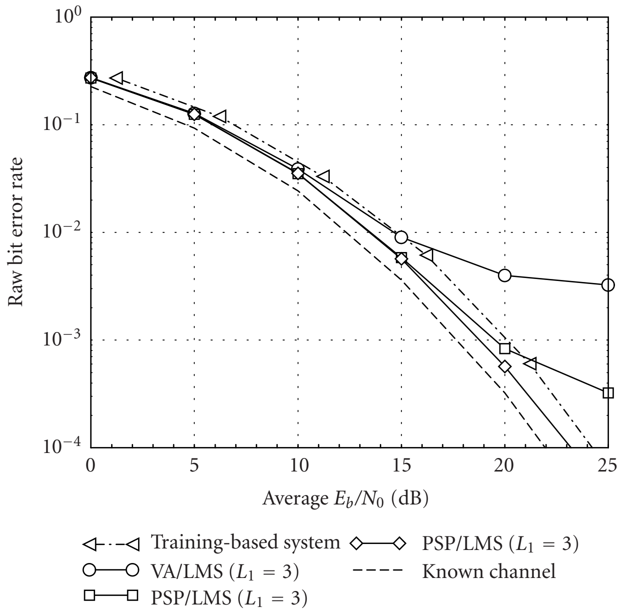


Figure 5: Raw BER versus SNR for TU channel model.

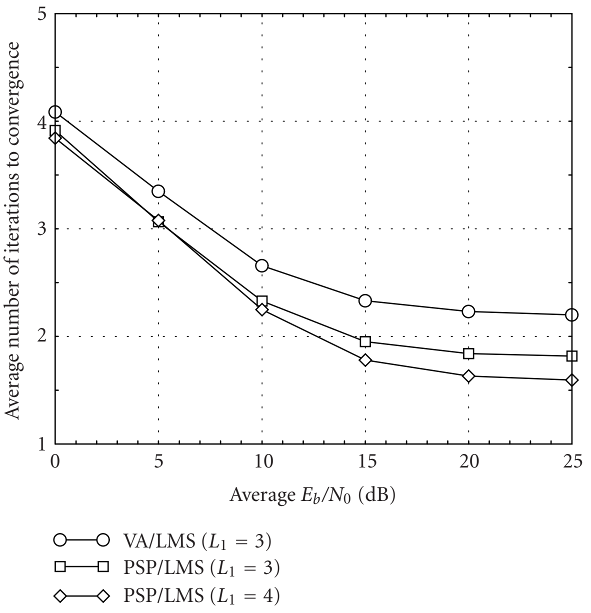


Figure 6: Average number of iterations of different algorithms to convergence for TU channel model, $N_{\text{iter}} = 10$.

with a smaller complexity. For the TU channel model, as illustrated in Figure 5, all blind equalizers under investigation outperform the training-based system for SNRs ≤ 15 dB. For PSP/LMS with $L_t = 4$, no error floor is visible. The gain of the PSP/LMS receiver with $L_t = 4$ is about 1 dB with respect to the training-based receiver, while the loss compared to the perfect channel knowledge is around 1 dB at the BER of 10^{-4}. Similar to the RA channel model, a receiver with a higher complexity shows a faster convergence rate, as illustrated in Figure 6.

5. BLIND TURBO PROCESSOR

If a priori information about data symbols is available, we may apply a MAP sequence estimator for data estimation, that is, the branch metrics in the binary case are modified as [23, 31]

$$\begin{aligned}\hat{\gamma}(\tilde{\mathbf{x}}[k]) &\\ &= -\frac{1}{\hat{\sigma}_n^2}\left| y[k] - \sum_{l=0}^{L_u}\hat{h}_l[k-1]\tilde{x}[k-l]\right|^2 + \log P(\tilde{d}[k])\\ &= -\frac{1}{\hat{\sigma}_n^2}\left| y[k] - \sum_{l=0}^{L_u}\hat{h}_l[k-1]\tilde{x}[k-l]\right|^2 + \frac{1}{2}\tilde{d}[k]L_a(d[k]),\end{aligned} \tag{23}$$

where $L_a(d[k])$ is the given or estimated log-likelihood ratio value (abbreviated as L-value in the following) of $d[k]$. (Symbol-by-symbol MAP estimation is not recommendable here due to the lack of survivors; surviving paths are necessary for channel estimation.)

The significance of (23) is a generic receiver structure, which is the same for the full range of blind equalization without a priori information (where $L_a(d[k]) = 0$ for all k) to a training-based equalizer (where $|L_a(d[k])| \to \infty$ for some k).

Besides incorporating a priori information, trellis-based blind equalizers are capable of delivering soft outputs to subsequent processing stages. Recently, blind turbo equalization techniques have been proposed in [37, 38], where the channel coefficients and the noise variance were estimated iteratively using the off-line EM algorithm (compare (15) and (16)), and in [39], where a blind channel estimator based on higher-order statistics is used. The latter technique [39] has been investigated for fading channels. Our approach is suitable for short bursts, where the unknown channel coefficients and data sequence are jointly estimated on the DPSK/ISI super-trellis. Moreover, the phase ambiguity and shift ambiguity problems are taken into consideration and solutions to combat such ambiguities are proposed, which may make our approach much more robust than related algorithms.

The overall system and the detailed turbo processor are illustrated in Figures 7 and 8, respectively.

In Figure 7, $\mathbf{u}$ and $\mathbf{d}'$ are the data vectors before and after channel encoding, respectively. Note that the "inner encoder" (represented by the DPSK/ISI super-trellis) is recursive, which is missing in the other blind turbo schemes [37, 38, 39], however.

Furthermore, $L_a(\cdot)$, $L_e^E(\cdot)$, and $L_e^D(\cdot)$ denote available a priori information, extrinsic information delivered by the SISO blind equalizer, and SISO channel decoder, respectively. Only the extrinsic information is exchanged between two SISO components.

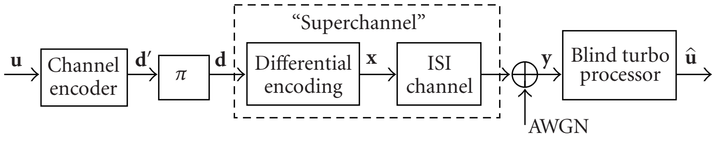


Figure 7: System model for blind turbo equalization.

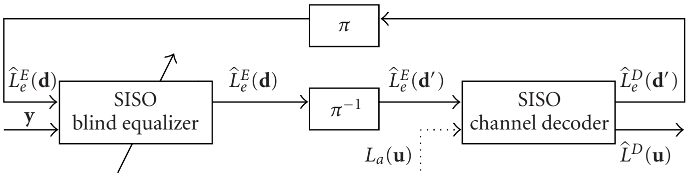

π
$\hat{L}_e^E(\mathbf{d})$
y
SISO blind equalizer
$\hat{L}_e^E(\mathbf{d})$
π^{-1}
$\hat{L}_e^E(\mathbf{d}')$
$L_a(\mathbf{u})$
SISO channel decoder
$\hat{L}_e^D(\mathbf{d}')$
$\hat{L}^D(\mathbf{u})$

Figure 8: Blind turbo processor.

In the following, we discuss the impacts of phase/shift ambiguity on the blind turbo processor, whileas novel approaches are proposed to solve these problems. The max-log-APP algorithm is used in both the blind SISO equalizer and the SISO channel decoder. For convenience, we consider the binary case with $L_t = L_u$ and assume that the estimated noise variance is equal to the true noise variance.

5.1. Estimated L-values under phase ambiguity

If $\hat{\mathbf{h}} = -\mathbf{h}$, branch metrics in SISO blind equalizer is formulated as

$$\hat{\gamma}(\tilde{\mathbf{x}}[k]) = -\frac{1}{\sigma_n^2}\left|y(k) + \sum_{l=0}^{L_u} h_l \tilde{x}[k-l]\right|^2 + \frac{1}{2}L_a(d[k])\tilde{d}[k]. \tag{24}$$

For the nonblind case with known channel coefficients, branch metrics are evaluated as

$$\gamma(\tilde{\mathbf{x}}[k]) = -\frac{1}{\sigma_n^2}\left|y(k) - \sum_{l=0}^{L_u} h_l \tilde{x}[k-l]\right|^2 + \frac{1}{2}L_a(d[k])\tilde{d}[k]. \tag{25}$$

Comparing (24) with (25), we have

$$\hat{\gamma}(\tilde{\mathbf{x}}[k]) = \gamma(-\tilde{\mathbf{x}}[k]). \tag{26}$$

Given a symmetrical initialization for the forward recursion of the max-log-APP algorithm, that is, $\hat{\alpha}(\tilde{\mathbf{s}}[-1]) = \alpha(-\tilde{\mathbf{s}}[-1])$, it is easy to verify that

$$\hat{\alpha}(\tilde{\mathbf{s}}[k]) = \alpha(-\tilde{\mathbf{s}}[k]), \quad 0 \le k \le K, \tag{27}$$

where $\hat{\alpha}(\tilde{\mathbf{s}}[k]) = \log p(\mathbf{y}_{j\le k}, \tilde{\mathbf{s}}[k] \mid \hat{\mathbf{h}})$ and $\mathbf{y}_{j\le k} = [y[0], y[1], \dots, y[k]]^T$. Similarly, the backward recursion has the same property:

$$\hat{\beta}(\tilde{\mathbf{s}}[k]) = \beta(-\tilde{\mathbf{s}}[k]), \quad 0 \le k \le K, \tag{28}$$

where $\hat{\beta}(\tilde{\mathbf{s}}[k]) = \log p(\mathbf{y}_{j\ge k+1} \mid \tilde{\mathbf{s}}[k], \hat{\mathbf{h}})$ and $\mathbf{y}_{j\ge k+1} = [y[k+1], y[k+2], \dots, y[K]]^T$. Therefore, the approximated a posteriori L-value of $d[k]$ can be obtained as

$$\begin{aligned}\hat{L}(d[k]) &= \max_{\tilde{\mathbf{s}}[k]:\tilde{d}[k]=+1}\{\hat{\alpha}(\tilde{\mathbf{s}}[k]) + \hat{\beta}(\tilde{\mathbf{s}}[k])\} \\ &\quad - \max_{\tilde{\mathbf{s}}[k]:\tilde{d}[k]=-1}\{\hat{\alpha}(\tilde{\mathbf{s}}[k]) + \hat{\beta}(\tilde{\mathbf{s}}[k])\} \\ &= \max_{\tilde{\mathbf{s}}[k]:\tilde{d}[k]=+1}\{\alpha(\tilde{\mathbf{s}}[k]) + \beta(\tilde{\mathbf{s}}[k])\} \\ &\quad - \max_{\tilde{\mathbf{s}}[k]:\tilde{d}[k]=-1}\{\alpha(\tilde{\mathbf{s}}[k]) + \beta(\tilde{\mathbf{s}}[k])\} \\ &= L(d[k]),\end{aligned} \tag{29}$$

where $\tilde{\mathbf{s}}[k] : \tilde{d}[k]$ denotes all states consistent with $\tilde{d}[k]$. Note that $\tilde{\mathbf{s}}[k]$ and $-\tilde{\mathbf{s}}[k]$ will result in the same $\tilde{d}[k]$. Hence, the correct L-values of data symbols are obtained under the condition $\hat{\mathbf{h}} = -\mathbf{h}$.

Moreover, the L-value about the reference symbol must be estimated rather than assumed to be known. Otherwise, the L-value about the first data symbol is evaluated as follows:

$$\begin{aligned}&\hat{L}(d[1] \mid x[0] = +1) \\ &\quad = \max_{\tilde{d}[1]=\tilde{x}[1]=+1}\{\hat{\alpha}(\tilde{\mathbf{s}}[1]) + \hat{\beta}(\tilde{\mathbf{s}}[1])\} \\ &\qquad - \max_{\tilde{d}[1]=\tilde{x}[1]=-1}\{\hat{\alpha}(\tilde{\mathbf{s}}[1]) + \hat{\beta}(\tilde{\mathbf{s}}[1])\} \\ &\quad = \max_{\tilde{d}[1]=\tilde{x}[1]=-1}\{\alpha(\tilde{\mathbf{s}}[1]) + \beta(\tilde{\mathbf{s}}[1])\} \\ &\qquad - \max_{\tilde{d}[1]=\tilde{x}[1]=+1}\{\alpha(\tilde{\mathbf{s}}[1]) + \beta(\tilde{\mathbf{s}}[1])\} \\ &\quad = -L(d[1]).\end{aligned} \tag{30}$$

If $\hat{L}(d[1])$ obtained in (30) is delivered to the channel decoder, it will cause error propagation during the iterative processing.

5.2. Shift ambiguity compensation

For a possible shift to the right in the channel estimation, we have

$$\begin{aligned}\hat{h}_l &= h_{l-\kappa}, \quad \kappa \le l \le L + \kappa, \\ \hat{h}_l &= 0, \quad l < \kappa \text{ or } L + \kappa < l \le L_u,\end{aligned} \tag{31}$$

where $0 \le \kappa \le L_u - L$ is the unknown shift index to be determined.

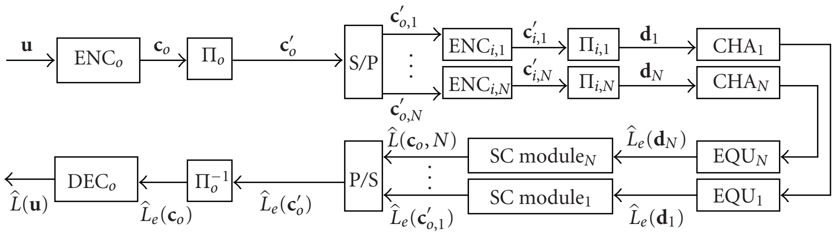


Figure 9: Proposed transmitter/receiver structure for fading channels.

We consider the very first iteration between the blind SISO equalizer and the SISO channel decoder, where no a priori information about $d[k]$ is available. Branch metrics under shifted channel coefficients are then formulated as

$$\begin{aligned}\widehat{\gamma}(\tilde{\mathbf{x}}[k]) &= -\frac{1}{\sigma_n^2}\left|y[k] - \sum_{l=0}^{L_u}\widehat{h}_l\tilde{x}[k-l]\right|^2 \\ &= -\frac{1}{\sigma_n^2}\left|y[k] - \sum_{l=0}^{L}h_l\tilde{x}[k-\kappa-l]\right|^2.\end{aligned} \tag{32}$$

For the case with correct channel coefficients, branch metrics are evaluated as follows:

$$\gamma(\tilde{\mathbf{x}}[k]) = -\frac{1}{\sigma_n^2}\left|y[k] - \sum_{l=0}^{L}h_l\tilde{x}[k-l]\right|^2. \tag{33}$$

By means of induction, the estimated L-values can be obtained as

$$\begin{aligned}\widehat{L}(d[k]) &= \max_{\tilde{\mathbf{x}}[k]:\tilde{d}[k]=+1}\left\{\widehat{\alpha}(\tilde{\mathbf{s}}[k]) + \widehat{\beta}(\tilde{\mathbf{s}}[k])\right\} \\ &\quad - \max_{\tilde{\mathbf{x}}[k]:\tilde{d}[k]=-1}\left\{\widehat{\alpha}(\tilde{\mathbf{s}}[k]) + \widehat{\beta}(\tilde{\mathbf{s}}[k])\right\} \\ &= L(d[k+\kappa]),\end{aligned} \tag{34}$$

that is, the estimated L-values of shifted channel coefficients are shifted in the reverse direction, see (31). This argument is verified in the appendix.

Because of the deinterleaver between the SISO equalizer and the SISO channel decoder (cf. Figure 8), a valid codeword is not valid any more after shifts, that is, only the L-values corresponding to the correct shift index will give reasonable soft outputs of the channel decoder. Based on this fact and on (34), we propose to shift the estimated L-values obtained from the SISO blind equalizer. Then the shift index can be estimated as

$$\widehat{\kappa} = \arg\max_{m}\left\{\sum_{n=1}^{KR}\frac{1}{KR}\left|\widehat{L}^{D(m)}(u[n])\right|\right\}, \tag{35}$$

where L-values about uncoded symbols related to shifts $m \in [-L_M, L_M]$ are denoted as $\{|\widehat{L}^{D(m)}(u[n])|\}$ and $L_M \leq L_u$ controls the range of shift search. Moreover, R denotes the code rate of the channel encoder and KR is assumed to be a positive integer. In the following, (35) is referred to as a shift compensation module (SC module).

5.3. *Double serial concatenation for fading channels*

Conventionally, for frequency-hopping systems over fading channels, an interburst interleaver is used in conjunction with a channel encoder in order to explore the time diversity of the channel code. On average, within severely faded bursts the L-values of the coded symbols have significantly smaller magnitudes than in nonfaded bursts. After deinterleaving, the L-values with small magnitudes are spread over the whole coded block. Therefore, it is easy to compensate these small L-values with the help of their "neighbors" with relatively large magnitudes. For blind turbo equalizers, a direct application of interburst interleaving is *not straightforward* because of the shift ambiguity problem. In order to combat the shift ambiguity associated with individual bursts, the shift ambiguity compensation should be carried out for individual bursts rather than for the whole coded block. Therefore, channel encoding is applied for individual bursts as shown in Figure 7, while the shift compensation is performed as presented in Section 5.2 for individual bursts. Moreover, a further outer channel encoder is introduced to exploit time diversity in conjunction with inter burst interleaving, similarly as in the conventional case with known channel coefficients. This new scheme, which has a double serially concatenated structure, is illustrated in Figure 9.

After the outer interleaver, denoted as Π_o, the coded data symbols from the outer channel encoder ENC$_o$ (with a code rate of R_o) are divided into N parallel substreams $\mathbf{c}'_{o,l}$, $1 \leq l \leq N$ by means of a serial-to-parallel converter (abbreviated as S/P). For the lth stream, we use the inner channel encoder ENC$_{i,l}$ (with a code rate of $R_{i,l}$). After the lth inner interleaver $\Pi_{i,l}$, we get the data symbols before the differential encoding, which are transmitted over a DPSK/ISI super channel CHA$_l$. In Figure 9, the additive noise is dropped for convenience. At the receiver, the shift compensation procedure is performed for each channel through an SC module. After determining the correct shift indices for individual bursts, the actual iterative processing between two SISO channel decoders and the SISO equalizers can be performed as usual. From individual SISO equalizers, the extrinsic information $\widehat{L}_e(\mathbf{d}_l)$ is delivered to corresponding SISO inner channel decoder, which delivers extrinsic information $\widehat{L}_e(\mathbf{c}'_{o,l})$ to the subsequent processing stage and also offers estimated a priori information for the SISO equalizer in the next iteration. The extrinsic information from N inner channel decoders is passed to the outer SISO channel decoder after the parallel-to-serial converter (denoted as P/S). Similarly, the outer channel decoder offers the estimated L-values about its infobits $\hat{L}(\mathbf{u})$ and also delivers the estimated a priori information for the inner channel decoders in the next iteration.

Because it is difficult to optimize the double serially concatenated system, the whole system is intuitively designed to get a compromise between the complexity and performance. Both inner and outer channel codes should be strong codes for a large time diversity and a reliable shift compensation, respectively. Within this paper, we consider rate 1/2 convolutional codes, where "strong code" means a sufficiently large memory length. On the other hand, to avoid a low bandwidth efficiency, we need a punctured code [40]. Therefore, a reasonable choice is to select an unpunctured code with a short memory length for the outer concatenation and a punctured code with a long memory length for the inner concatenation.

5.4. Overall receiver

Two scheduling strategies are possible: iterative processing between the SISO modules may be performed after a convergence of the TABSE (the receiver based on this scheduling strategy is referred to as Scheme 1), or the iterative processing is carried out directly after the all-zero initialization (the corresponding receiver is referred to as Scheme 2). Scheme 2 requires more iterations than Scheme 1 to achieve a similar performance, because in Scheme 1 the quality of soft outputs from the SISO blind equalizer are more reliable than in Scheme 2 at least at the initial phase of iterative processing. Therefore, within this paper, we only consider Scheme 1.

The overall receiver for coded systems in the jth iteration is described as follows.

(1) *Soft-output equalization:* the forward recursion is performed by means of adaptive joint data/channel estimation, where the branch metrics are evaluated as in (23). The backward recursion is carried out using the transition probabilities obtained in the forward recursion. Afterward, the L-values about data symbols before the differential encoding are evaluated to get $\{\hat{L}_e^E(d[k])\}$.

(2) *Noise variance estimation:* after the evaluation of L-values of coded data symbols, the noise variance can be estimated based on hard or soft decisions from the SISO equalizer, refer to (16) and (19), respectively. The estimated noise variance is used in the next iteration to evaluate the branch metrics (cf. (23)).

(3) *SISO channel decoding of inner codes:* the branch metrics in lth ($1 \le l \le N$) SISO inner channel decoder are calculated as ($0 \le n \le KR_{i,l} - 1$)

$$\hat{\gamma}^{(j)}\left(\tilde{\mathbf{c}}'_{i,l}[n]\right) = \sum_{k=n/R_{i,l}}^{(n+1)/R_{i,l}-1} \hat{L}_e^{(j)}\left(c'_{i,l}[k]\right)\tilde{c}'_{i,l}[k] + \hat{L}_a^{(j-1)}\left(c'_{o,l}[n]\right)\tilde{c}'_{o,l}[n], \tag{36}$$

where $\tilde{\mathbf{c}}'_{i,l}[n] = [\tilde{c}'_{i,l}[n/R_{i,l}], \ldots, \tilde{c}'_{i,l}[(n+1)/R_{i,l} - 1]]^T$ is the inner coded data symbol vector at index n and $\{\hat{L}_e^{(j)}(c'_{i,l}[k])\}$ are extrinsic information obtained from the lth SISO equalizer. Moreover, $\hat{L}_a^{(j-1)}(c'_{o,l}[n])$ denotes the estimated a priori information about coded bits $c'_{o,l}[n]$ of outer code (i.e., info-bits of inner codes) from the outer channel decoder in the $(j-1)$th iteration. The extrinsic information about coded bits $\{c'_{i,l}[k]\}$ obtained by the max-log-APP channel decoder is fed back to the lth SISO equalizer and used as estimated a priori information in the next iteration. The extrinsic information about info-bits $\{c'_{o,l}[k]\}$ is passed to the outer channel decoder after the parallel-to-serial converter. Only in the *very first* iteration, the possible shift ambiguity in the SISO equalizer is compensated by means of the proposed approach (cf. Section 5.2). L-values corresponding to the optimal shift index are delivered to the inner SISO channel decoders.

(4) *SISO channel decoding of outer code:* the branch metrics in the SISO outer channel decoder are calculated as ($0 \le n \le K\sum_{l=1}^{N} R_{i,l} \cdot R_0 - 1$)

$$\hat{\gamma}^{(j)}\left(\tilde{\mathbf{c}}_o[n]\right) = \sum_{k=n/R_o}^{(n+1)/R_o-1} \hat{L}_e^{(j)}\left(c_o[k]\right)\tilde{c}_o[k] + L_a\left(u[n]\right)\tilde{u}[n], \tag{37}$$

where $\tilde{\mathbf{c}}_o[n] = [c_o[n/R_o], \ldots, c_o[(n+1)/R_o - 1]]^T$ is the outer coded data symbol vector and $\{\hat{L}_e^{(j)}(c_o[k])\}$ are extrinsic information from the inner channel decoders. Moreover, $L_a(u[n])$ denotes the available a priori information about info-bits $u[n]$ of outer code.

(5) *Final data estimation:* steps (1)–(4) are repeated until the given number of iterations is reached. The L-values from the outer channel decoder $\hat{L}(\mathbf{u})$ deliver the hard decisions about info-bits and their corresponding reliabilities.

5.5. Numerical results for coded systems

Simulations were performed for the quasi-static TU and RA channel models using the proposed double serially concatenated scheme. The outer channel encoder is a rate $R_0 = 1/2$ convolutional code with generator polynomials $(5, 7)$. The inner codes ($N = 10$) are recursive systematic convolutional codes with the same generator polynomials $(23, 35)$, which are punctured to get a code rate of $R_{i,l} = 2/3, 1 \le l \le 10$. The puncturing table is $[1111, 0101]$, where 0 stands for the puncturing. Accordingly, the overall code rate is 1/3. No zero tailing or tail biting is applied. The code length of the outer code is 1000, while the inner codes have a code length of 150. S-random interleavers [41] are applied for the interburst interleaver ($S = 15$) and the intraburst interleavers ($S = 8$). The data sequence of length $K = 150$ from each inner channel encoder is transmitted over independently generated fading channels. The max-log-APP algorithm is applied for both SISO equalization and SISO channel decoding. The SISO equalizer is the modified TABSE based on VA/LMS, while parameters necessary for the LMS algorithm are the same as in uncoded systems. The parameter L_M in the SC modules is selected to be 1 for both channel models.

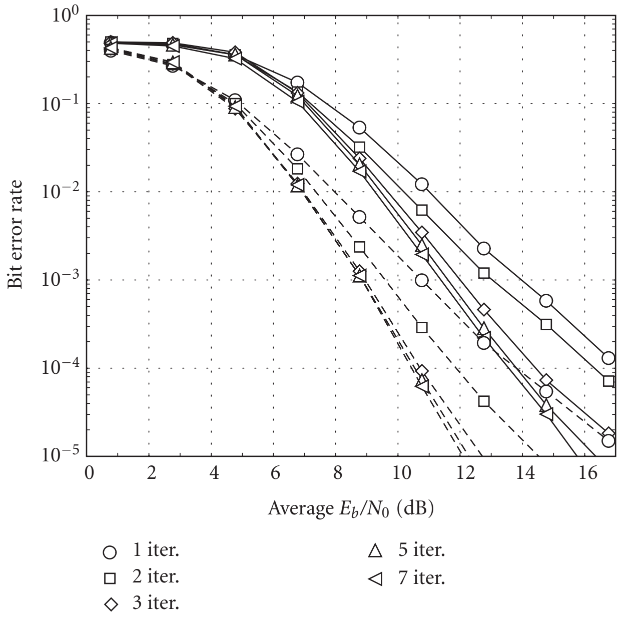


Figure 10: BER versus SNR of coded schemes for RA channel model. Solid lines and dashed lines correspond to simulation results of blind schemes and schemes with perfect channel knowledge, respectively.

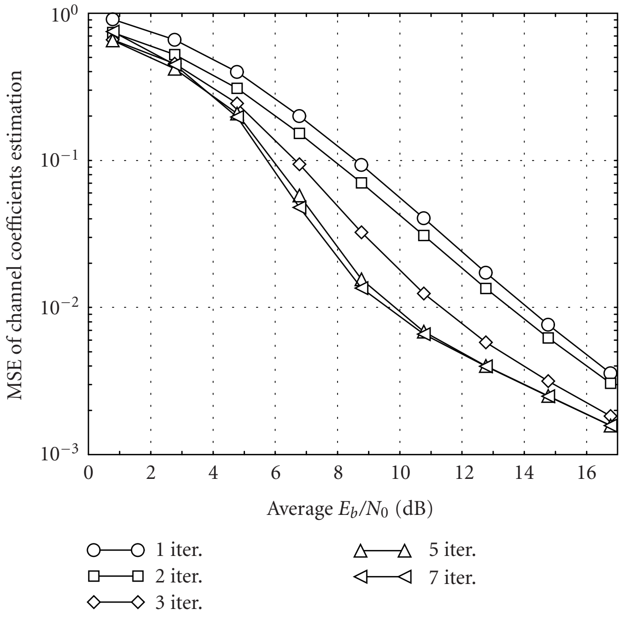


Figure 12: MSE of estimated channel coefficients versus average E_b/N_0 for different iterations, for RA channel model.

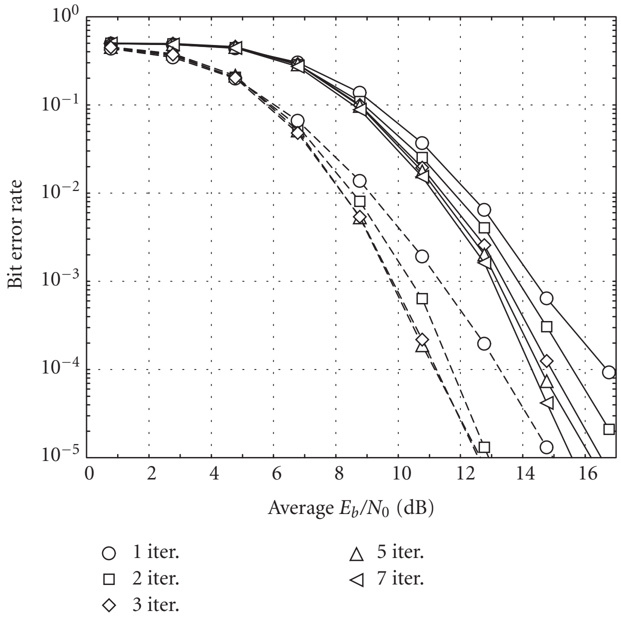


Figure 11: BER versus SNR of coded schemes for TU channel model. Solid lines and dashed lines correspond to simulation results of blind schemes and schemes with perfect channel knowledge, respectively.

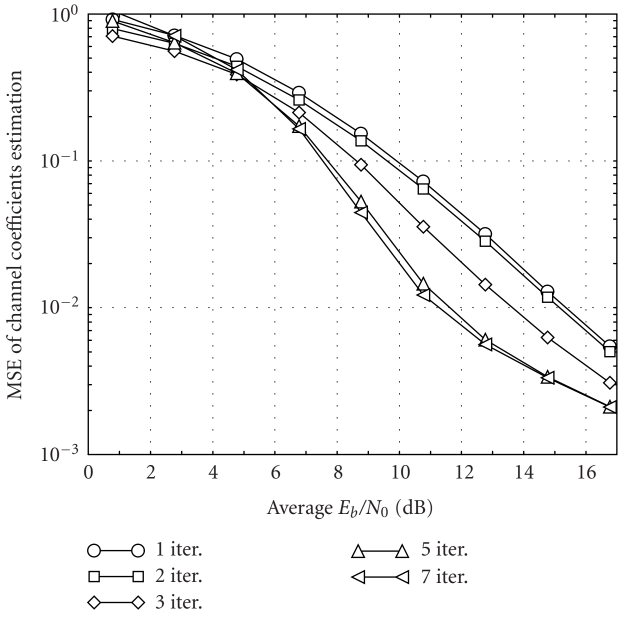


Figure 13: Decreased MSE of estimated channel coefficients through the iterative processing, for TU channel model.

As shown in Figures 10 and 11, for the systems with perfect channel knowledge (known channel coefficients and known average SNRs), the first iteration between the SISO equalizer and SISO channel decoders provides the most significant improvement. There is no further improvement after about 3 iterations for the considered SNR region. For TABSE-based turbo schemes, the system performance is improved gradually from iteration to iteration. The channel estimates are improved gradually, as shown in Figures 12 and 13, which results in a gradually increased quality of soft outputs in the SISO equalizer through the iterative processing.

6. CONCLUSIONS

Based on an approximation of a blind maximum-likelihood sequence estimator, reduced-complexity iterative adaptive trellis-based blind sequence estimators are proposed. Previously proposed blind sequence estimators can be interpreted as special cases of our proposed receiver. Moreover, the ideas of PSP/PBP are generalized by replacing conventional branch metrics by short-path metrics. The differential encoder (or generalizations thereof) is used to combat the phase ambiguity, where the resulting DPSK/ISI super-trellis is explicitly applied for SISO equalization. By means of (de-)interleaver and channel encoder, the problem of shift ambiguity due to the overdetermined channel order can be resolved efficiently. For frequency-hopping systems over frequency-selective fading channels, a double serially concatenated scheme is proposed, which can combat the shift ambiguity and explore the time diversity of channel codes in conjunction with inter-burst interleaving. Our simulation results demonstrate the potential of trellis-based adaptive blind sequence estimators for short-burst data transmission over practical fading channels, particularly in the presence of channel coding.

APPENDIX

A. L-VALUES UNDER SHIFT AMBIGUITY

In this appendix, we consider the relationship between L-values conditioned on shifted channel coefficients and correct L-values. The following conditions are presumed:

$$\begin{aligned}
\hat{h}_l &= h_{l-\kappa}, \quad \kappa \le l \le L+\kappa,\\
\hat{h}_l &= 0, \quad l<\kappa \text{ or } L+\kappa<l\le L_u,\\
\hat{\gamma}(\tilde{\mathbf{x}}[k]) &= -\frac{1}{\sigma_n^2}\left|y[k]-\sum_{l=0}^{L}h_l\tilde{x}[k-\kappa-l]\right|^2, \qquad \text{(A.1)}\\
\gamma(\tilde{\mathbf{x}}[k]) &= -\frac{1}{\sigma_n^2}\left|y[k]-\sum_{l=0}^{L}h_l\tilde{x}[k-l]\right|^2,
\end{aligned}$$

and the max-log-APP algorithm is employed.

A.1. Definitions

Firstly, we introduce some relevant definitions.

(i) A state at the time index k, which merges into the state $\mathbf{s}[k+\kappa]$ after κ steps in the forward recursion, is called a *forward-consistent state* of $\mathbf{s}[k+\kappa]$. The set of forward-consistent states of $\mathbf{s}[k+\kappa]$ at time index k is termed *forward-consistent state set* of $\mathbf{s}[k+\kappa]$ and abbreviated as $\overrightarrow{\mathcal{M}}_k(\mathbf{s}[k+\kappa])$.

Similarly, a state at time index $k+\kappa$, which merges into the state $\mathbf{s}[k]$ after κ steps in the backward recursion, is termed a *backward-consistent state* of $\mathbf{s}[k]$. The set of backward-consistent states of $\mathbf{s}[k]$ at time index $k+\kappa$ is termed *backward-consistent state set* of $\mathbf{s}[k]$ and abbreviated as $\overleftarrow{\mathcal{M}}_{k+\kappa}(\mathbf{s}[k])$.

(ii) A state transition at time index k, $\mathbf{x}[k]$, which connects two forward-consistent states of $\mathbf{s}[k+\kappa]$, is called a *forward-consistent state transition* of $\mathbf{s}[k+\kappa]$. The *forward-consistent transition set* of $\mathbf{s}[k+\kappa]$ at time index k is abbreviated as $\overrightarrow{\mathcal{Q}}_k(\mathbf{s}[k+\kappa])$.

Similarly, a state transition at time index $k+\kappa$, which connects two backward-consistent states of $\mathbf{s}[k]$, is called a *backward-consistent state transition* of $\mathbf{s}[k]$. The *backward-consistent transition set* of $\mathbf{s}[k]$ at time index $k+\kappa$ is referred to as $\overleftarrow{\mathcal{Q}}_{k+\kappa}(\mathbf{s}[k])$.

(iii) A state $\mathbf{s}_1[k]=[x_1[k-L_u+1],\dots,x_1[k]]^T$ is *κ-equivalent* to another state $\mathbf{s}_2[k]=[x_2[k-L_u+1],\dots,x_2[k]]^T$, if $x_1[k-\kappa-l]=x_2[k-\kappa-l]$, $0\le l\le L$ (for the case $L_u>\kappa+L$), or if $x_1[k-\kappa-l]=x_2[k-\kappa-l]$, $0\le l\le L-1$ (for the case $L_u=\kappa+L$). A state $\mathbf{s}_1[k]$ is *κ-shift equivalent* to another state $\mathbf{s}_2[k]$, if $x_1[k-\kappa-l]=x_2[k-l]$, $0\le l\le L$ (for the case $L_u>\kappa+L$), or if $x_1[k-\kappa-l]=x_2[k-l]$, $0\le l\le L-1$ (for the case $L_u=\kappa+L$).

A state transition $\mathbf{x}_1[k]=[x_1[k-L_u],\dots,x_1[k]]^T$ is *κ-equivalent* to another state transition $\mathbf{x}_2[k]=[x_2[k-L_u],\dots,x_2[k]]^T$, if $x_1[k-\kappa-l]=x_2[k-\kappa-l]$, $0\le l\le L$. A state transition $\mathbf{x}_1[k]$ is *κ-shift equivalent* to another state transition $\mathbf{x}_2[k]$, if $x_1[k-\kappa-l]=x_2[k-l]$, $0\le l\le L$.

(iv) For the forward recursion, we define

$$\begin{aligned}
\overrightarrow{\mathbf{s}}[k] &\triangleq \big[-x[k-L_u+1],x[k-L_u+2],\dots,x[k]\big]^T,\\
&\quad \text{if } \mathbf{s}[k]=\big[x[k-L_u+1],x[k-L_u+2],\dots,x[k]\big]^T,\\
\overrightarrow{\mathbf{x}}[k] &\triangleq \big[-x[k-L_u],x[k-L_u+1],\dots,x[k]\big]^T,\\
&\quad \text{if } \mathbf{x}[k]=\big[x[k-L_u],x[k-L_u+1],\dots,x[k]\big]^T.
\end{aligned} \qquad \text{(A.2)}$$

Correspondingly, for the backward recursion, we define

$$\begin{aligned}
\overleftarrow{\mathbf{s}}[k] &\triangleq \big[x[k-L_u+1],\dots,x[k-1],-x[k]\big]^T,\\
&\quad \text{if } \mathbf{s}[k]=\big[x[k-L_u+1],\dots,x[k-1],x[k]\big]^T,\\
\overleftarrow{\mathbf{x}}[k] &\triangleq \big[x[k-L_u],\dots,x[k-1],-x[k]\big]^T,\\
&\quad \text{if } \mathbf{x}[k]=\big[x[k-L_u],\dots,x[k-1],x[k]\big]^T.
\end{aligned} \qquad \text{(A.3)}$$

(v) For the evaluation of L-values under correct channel coefficients, the relevant state transitions are defined as $\mathbf{x}_r[k]\triangleq[x[k-L],\dots,x[k]]^T$. Accordingly, $\overrightarrow{\mathbf{x}}_r[k]\triangleq[-x[k-L],\dots,x[k]]^T$. The relevant states are defined as $\mathbf{s}_r[k]\triangleq[x[k-L+1],\dots,x[k]]^T$ (for the case $L_u=L+\kappa$) or defined as $\mathbf{s}_r[k]\triangleq[x[k-L],\dots,x[k]]^T$ (for the case $L_u>L+\kappa$). Accordingly, $\overrightarrow{\mathbf{s}}_r[k]\triangleq[-x[k-L+1],\dots,x[k]]^T$ (for the case $L_u=L+\kappa$) and $\overrightarrow{\mathbf{s}}_r[k]\triangleq[-x[k-L],\dots,x[k]]^T$ (for the case $L_u>L+\kappa$).

A state $\mathbf{s}_1[k]=[x_1[k-L_u+1],\dots,x_1[k]]^T$ is *relevant-equivalent* to another state $\mathbf{s}_2[k]=[x_2[k-L_u+1],\dots,x_2[k]]^T$, if $x_1[k-l]=x_2[k-l]$, $0\le l\le L$.

A.2. L-values under shifted channel coefficients

In the following, only the case $L_u=L+\kappa$ is considered, while the extension to $L_u>L+\kappa$ is straightforward.

Theorem 1. *If $\mathbf{s}_1'[k]$ and $\mathbf{s}_2'[k]$ are κ-equivalent states, then $\hat{\alpha}(\mathbf{s}_1'[k])=\hat{\alpha}(\mathbf{s}_2'[k])$.*

Proof. This statement is verified by means of induction as follows.

(1) $k = 0$. For two arbitrary *κ-equivalent* states $\mathbf{s}'_1[0]$ and $\mathbf{s}'_2[0]$, we have

$$\begin{aligned}\hat{\alpha}(\mathbf{s}'_1[0]) &= \max\{\hat{\gamma}(\mathbf{x}'_1[0]), \hat{\gamma}(\vec{\mathbf{x}}'_1[0])\},\\ \hat{\alpha}(\mathbf{s}'_2[0]) &= \max\{\hat{\gamma}(\mathbf{x}'_2[0]), \hat{\gamma}(\vec{\mathbf{x}}'_2[0])\}.\end{aligned} \tag{A.4}$$

Due to the definition of *κ-equivalent* states, $\hat{\gamma}(\mathbf{x}'_2[0]) = \hat{\gamma}(\mathbf{x}'_1[0])$ or $\hat{\gamma}(\vec{\mathbf{x}}'_2[0]) = \hat{\gamma}(\mathbf{x}'_1[0])$, which indicates

$$\hat{\alpha}(\mathbf{s}'_1[0]) = \hat{\alpha}(\mathbf{s}'_2[0]). \tag{A.5}$$

(2) Assuming the following equation is fulfilled for all $k > 0$:

$$\hat{\alpha}(\mathbf{s}'_1[k]) = \hat{\alpha}(\mathbf{s}'_2[k]), \tag{A.6}$$

if $\mathbf{s}'_1[k]$ and $\mathbf{s}'_2[k]$ are *κ-equivalent* states.

(3) For two *κ-equivalent* states at the time index $k+1$, $\mathbf{s}'_1[k+1]$, and $\mathbf{s}'_2[k+1]$, we obtain

$$\begin{aligned}\hat{\alpha}(\mathbf{s}'_1[k+1]) &= \max\{\hat{\alpha}(\mathbf{s}'_1[k]) + \hat{\gamma}(\mathbf{x}'_1[k+1]),\\ &\qquad \hat{\alpha}(\vec{\mathbf{s}}'_1[k]) + \hat{\gamma}(\vec{\mathbf{x}}'_1[k+1])\},\\ \hat{\alpha}(\mathbf{s}'_2[k+1]) &= \max\{\hat{\alpha}(\mathbf{s}'_2[k]) + \hat{\gamma}(\mathbf{x}'_2[k+1]),\\ &\qquad \hat{\alpha}(\vec{\mathbf{s}}'_2[k]) + \hat{\gamma}(\vec{\mathbf{x}}'_2[k+1])\}.\end{aligned} \tag{A.7}$$

Note that $\mathbf{s}'_1[k]$ and $\mathbf{s}'_2[k]$ are *κ-equivalent*, which implies that $\vec{\mathbf{s}}'_1[k]$ and $\vec{\mathbf{s}}'_2[k]$ are also *κ-equivalent* and that $\hat{\alpha}(\vec{\mathbf{s}}'_1[k]) = \hat{\alpha}(\vec{\mathbf{s}}'_2[k])$. Moreover, $\hat{\gamma}(\mathbf{x}'_2[k+1]) = \hat{\gamma}(\mathbf{x}'_1[k+1])$ and $\hat{\gamma}(\vec{\mathbf{x}}'_2[k+1]) = \hat{\gamma}(\vec{\mathbf{x}}'_1[k+1])$ are satisfied. Therefore,

$$\hat{\alpha}(\mathbf{s}'_1[k+1]) = \hat{\alpha}(\mathbf{s}'_2[k+1]). \tag{A.8}$$

□

Theorem 2. *If* $\mathbf{s}'[k]$ *is κ-shift equivalent to* $\mathbf{s}[k] \in \vec{\mathcal{M}}_k(\mathbf{s}[k+\kappa])$,

$$\hat{\alpha}(\mathbf{s}'[k]) = \max\{\alpha(\mathbf{s}[k]) \mid \mathbf{s}[k] \in \vec{\mathcal{M}}_k(\mathbf{s}[k+\kappa])\}. \tag{A.9}$$

Proof. The method of induction is employed again.

(1) $k = 0$:

$$\begin{aligned}\hat{\alpha}(\mathbf{s}'[0]) &= \max\{\hat{\gamma}(\mathbf{x}'[0]), \hat{\gamma}(\vec{\mathbf{x}}'[0])\}\\ &= \max\{\gamma(\mathbf{x}_r[0]),\ \gamma(\vec{\mathbf{x}}_r[0])\},\end{aligned} \tag{A.10}$$

where $\mathbf{x}'[0]$ and $\vec{\mathbf{x}}'[0]$ are *κ-shift equivalent* to $\mathbf{x}_r[0]$ and $\vec{\mathbf{x}}_r[0]$, respectively. On the other hand, we have

$$\begin{aligned}&\max\{\alpha(\mathbf{s}[0]) \mid \mathbf{s}[0] \in \vec{\mathcal{M}}_0(\mathbf{s}[\kappa])\}\\ &\quad = \max\{\gamma(\mathbf{x}_r[0]), \gamma(\vec{\mathbf{x}}_r[0])\},\end{aligned} \tag{A.11}$$

where $\mathbf{s}'[0]$ is *κ-shift equivalent* to $\mathbf{s}[0]$. Note that for two *forward-consistent* states $\mathbf{s}_1[0], \mathbf{s}_2[0] \in \vec{\mathcal{M}}_0(\mathbf{s}[\kappa])$, we have $x_1[-l] = x_2[-l]$, $0 \le l \le L-1$, while $x[-l]$, $0 \le l \le L$, are relevant for the evaluation of $\alpha(\mathbf{s}[0])$. Therefore,

$$\hat{\alpha}(\mathbf{s}'[0]) = \max\{\alpha(\mathbf{s}[0]) \mid \mathbf{s}[0] \in \vec{\mathcal{M}}_0(\mathbf{s}[\kappa])\}. \tag{A.12}$$

(2) Assuming that for all $k > 0$ the following equation is satisfied:

$$\hat{\alpha}(\mathbf{s}'[k]) = \max\{\alpha(\mathbf{s}[k]) \mid \mathbf{s}[k] \in \vec{\mathcal{M}}_k(\mathbf{s}[k+\kappa])\}, \tag{A.13}$$

if $\mathbf{s}'[k]$ is *κ-shift equivalent* to $\mathbf{s}[k]$.

(3) For $k+1$, the $\hat{\alpha}$-term is evaluated as

$$\begin{aligned}&\hat{\alpha}(\mathbf{s}'[k+1])\\ &\quad = \max\{\hat{\alpha}(\mathbf{s}'[k]) + \hat{\gamma}(\mathbf{x}'[k+1]),\\ &\qquad\quad \hat{\alpha}(\vec{\mathbf{s}}'[k]) + \hat{\gamma}(\vec{\mathbf{x}}'[k+1])\}\\ &\quad = \max\{\max\{\alpha(\mathbf{s}_r[k]) + \gamma(\mathbf{x}_r[k+1])\},\\ &\qquad\qquad \max\{\alpha(\vec{\mathbf{s}}_r[k]) + \gamma(\vec{\mathbf{x}}_r[k+1])\}\}\\ &\quad = \max\{\alpha(\mathbf{s}[k+1]) \mid \mathbf{s}[k+1] \in \vec{\mathcal{M}}_{k+1}(\mathbf{s}[k+\kappa+1])\},\end{aligned} \tag{A.14}$$

where $\mathbf{x}'[k+1]$ and $\vec{\mathbf{x}}'[k+1]$ are *κ-shift equivalent* to $\mathbf{x}_r[k+1]$ and $\vec{\mathbf{x}}_r[k+1]$, respectively.

□

From (A.9), the forward recursion with correct channel coefficients can be evaluated as

$$\begin{aligned}&\alpha(\mathbf{s}[k+\kappa])\\ &\quad = \max\{\alpha(\mathbf{s}[k+\kappa-1]) + \gamma(\mathbf{x}[k+\kappa]),\\ &\qquad\quad \alpha(\vec{\mathbf{s}}[k+\kappa-1]) + \gamma(\vec{\mathbf{x}}[k+\kappa])\}\\ &\quad = \max\{\alpha(\mathbf{s}[k+\kappa-1]), \alpha(\vec{\mathbf{s}}[k+\kappa-1])\}\\ &\qquad + \gamma(\mathbf{x}[k+\kappa] \mid \mathbf{x}[k+\kappa] \in \vec{\mathcal{Q}}_{k+\kappa}(\mathbf{s}[k+\kappa]))\\ &\quad = \max\{\alpha(\mathbf{s}[k]) \mid \mathbf{s}[k] \in \vec{\mathcal{M}}_k(\mathbf{s}[k+\kappa])\}\\ &\qquad + \sum_{i=1}^{\kappa} \gamma(\mathbf{x}[k+i] \mid \mathbf{x}[k+i] \in \vec{\mathcal{Q}}_{k+i}(\mathbf{s}[k+\kappa]))\\ &\quad = \hat{\alpha}(\mathbf{s}'[k]) + \sum_{i=1}^{\kappa} \gamma(\mathbf{x}[k+i] \mid \mathbf{x}[k+i] \in \vec{\mathcal{Q}}_{k+i}(\mathbf{s}[k+\kappa])),\end{aligned} \tag{A.15}$$

where all forward-consistent transitions in $\vec{\mathcal{Q}}_{k+i}(\mathbf{s}[k+\kappa])$ will result in the same branch metrics, because the relevant data symbols $x[k+i-l], 0 \le l \le L$, are the same for $\mathbf{x}[k+i] \in \vec{\mathcal{Q}}_{k+i}(\mathbf{s}[k+\kappa])$. Moreover, $\mathbf{s}'[k]$ is κ-shift equivalent to $\mathbf{s}[k]$.

Similarly, for the backward recursion, the following theorems can be verified by means of induction.

Theorem 3. *If* $\mathbf{s}_1[k]$ *and* $\mathbf{s}_2[k]$ *are relevant-equivalent states, then* $\beta(\mathbf{s}_1[k]) = \beta(\mathbf{s}_2[k])$.

Theorem 4. *If* $\mathbf{s}'[k]$ *is* κ*-shift equivalent to* $\mathbf{s}[k]$,

$$\beta(\mathbf{s}[k]) = \max\left\{\widehat{\beta}(\mathbf{s}'[k]) \mid \mathbf{s}'[k] \in \overleftarrow{\mathcal{M}}_k(\mathbf{s}'[k-\kappa])\right\}. \quad \text{(A.16)}$$

Theorem 5. *If* $\mathbf{s}'[k+\kappa]$ *is* κ*-shift equivalent to* $\mathbf{s}[k+\kappa]$,

$$\begin{aligned}\beta(\mathbf{s}[k+\kappa]) &= \widehat{\beta}(\mathbf{s}'[k]) \\ &\quad - \sum_{i=1}^{\kappa} \widehat{\gamma}\left(\mathbf{x}'[k+i] \mid \mathbf{x}'[k+i] \in \overleftarrow{\mathcal{Q}}_{k+i}(\mathbf{s}'[k])\right).\end{aligned} \quad \text{(A.17)}$$

Finally, the estimated L-values under shifted channel coefficients are obtained as

$$\begin{aligned}\widehat{L}(d'[k]) &= \max_{\mathbf{s}'[k]:d'[k]=+1}\left\{\widehat{\alpha}(\mathbf{s}'[k]) + \widehat{\beta}(\mathbf{s}'[k])\right\} \\ &\quad - \max_{\mathbf{s}'[k]:d'[k]=-1}\left\{\widehat{\alpha}(\mathbf{s}'[k]) + \widehat{\beta}(\mathbf{s}'[k])\right\} \\ &= \max_{\mathbf{s}[k+\kappa]:d[k+\kappa]=+1}\left\{\alpha(\mathbf{s}[k+\kappa]) + \beta(\mathbf{s}[k+\kappa])\right\} \\ &\quad - \max_{\mathbf{s}[k+\kappa]:d[k+\kappa]=-1}\left\{\alpha(\mathbf{s}[k+\kappa]) + \beta(\mathbf{s}[k+\kappa])\right\} \\ &= L(d[k+\kappa]).\end{aligned} \quad \text{(A.18)}$$

ACKNOWLEDGMENTS

The authors would like to thank anonymous reviewers for their valuable comments. The work of Xiao-Ming Chen was supported by German Research Foundation (DFG) under Grant no. Ho 2226/1. The material in this paper was presented in part at the 4th International ITG Conference on Source and Channel Coding, Berlin, Germany, January 2002, and at the 6th Baiona Workshop on Signal Processing in Communications, Baiona, Spain, September 2003.

REFERENCES

[1] Y. Sato, "Blind equalization and blind sequence estimation," *IEICE Trans. Commun.*, vol. E77-B, no. 5, pp. 545–556, 1994.

[2] Z. Ding and Y. Li, *Blind Equalization and Identification*, Marcel Dekker, New York, NY, USA, 2001.

[3] Y. Sato, "A method of self-recovering equalization for multilevel amplitude-modulation systems," *IEEE Trans. Commun.*, vol. 23, no. 6, pp. 679–682, 1975.

[4] D. Godard, "Self recovering equalization and carrier tracking in two-dimensional data communication systems," *IEEE Trans. Commun.*, vol. 28, no. 11, pp. 1867–1875, 1980.

[5] O. Shalvi and E. Weinstein, "Super-exponential methods for blind deconvolution," *IEEE Trans. Inform. Theory*, vol. 39, no. 2, pp. 504–519, 1993.

[6] Z. Ding and G. Li, "Single channel blind equalization for GSM cellular systems," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1493–1505, 1998.

[7] B. Jelonnek, D. Boss, and K. D. Kammeyer, "Generalized eigenvector algorithm for blind equalization," *Elsevier Signal Processing*, vol. 61, no. 3, pp. 237–264, 1997.

[8] A. Schmidbauer, W. Specht, and R. Herzog, "A new approach to blind channel identification for mobile radio fading channels," in *Proc. Wireless Communication Conference (WCC '97)*, pp. 69–72, Boulder, Colo, USA, August 1997.

[9] J. B. Anderson and S. Mohan, "Sequential coding algorithms: a survey and cost analysis," *IEEE Trans. Commun.*, vol. 32, no. 2, pp. 169–176, 1984.

[10] M. Feder and J. A. Catipovic, "Algorithm for joint channel estimation and data recovery—application to equalization in underwater communications," *IEEE Journal Oceanic Engineering*, vol. 16, pp. 42–55, January 1991.

[11] G. Kaleh and R. Vallet, "Joint parameter estimation and symbol detection for linear or nonlinear unknown channels," *IEEE Trans. Commun.*, vol. 42, no. 7, pp. 2406–2413, 1994.

[12] N. Seshadri, "Joint data and channel estimation using blind trellis search techniques," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 1000–1011, 1994.

[13] X. M. Chen and P. A. Hoeher, "Blind equalization with iterative joint channel and data estimation for wireless DPSK systems," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '01)*, pp. 274–279, San Antonio, Tex, USA, November 2001.

[14] K. M. Chugg, "Blind acquisition characteristics of PSP-based sequence detectors," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 1518–1529, October 1998.

[15] H. R. Sadjadpour and C. L. Weber, "Pseudo-maximum-likelihood data estimation algorithm and its applications over band-limited channels," *IEEE Trans. Commun.*, vol. 49, no. 1, pp. 120–129, 2001.

[16] S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1996.

[17] R. Raheli, A. Polydoros, and C.-K. Tzou, "Per-survivor processing: a general approach to MLSE in uncertain environments," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 354–364, 1995.

[18] N. Seshadri and C. W. Sundberg, "List Viterbi decoding algorithms with applications," *IEEE Trans. Commun.*, vol. 42, no. 2/3/4, pp. 313–323, 1994.

[19] J. G. Proakis, *Digital Communications*, McGraw-Hill, New York, NY, USA, 4th edition, 2000.

[20] X. Yu and S. Pasupathy, "Innovations-based MLSE for Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 43, no. 2/3/4, pp. 1534–1544, 1995.

[21] L. Davis, I. Collings, and P. Hoeher, "Joint MAP equalization and channel estimation for frequency-selective and frequency-flat fast-fading channels," *IEEE Trans. Commun.*, vol. 49, no. 12, pp. 2106–2114, 2001.

[22] S. A. R. Shah and B.-P. Paris, "Self-adaptive sequence detection via the *M*-algorithm," in *Proc. IEEE International Conference on Communications (ICC '97)*, vol. 3, pp. 1479–1483, Monréal, Québec, Canada, June 1997.

[23] J. Hagenauer, "Source controlled channel decoding," *IEEE Trans. Commun.*, vol. 43, no. 9, pp. 2449–2457, 1995.

[24] K. M. Chugg and A. Polydoros, "MLSE for an unknown channel—Part I: optimality considerations," *IEEE Trans. Commun.*, vol. 44, no. 7, pp. 836–846, 1996.

[25] K. M. Chugg and A. Polydoros, "MLSE for an unknown channel—Part II: tracking performance," *IEEE Trans. Commun.*, vol. 44, no. 8, pp. 949–958, 1996.

[26] M. K. Simon, S. M. Hinedi, and W. C. Lindsey, *Digital Communication Techniques: Signal Design and Detection*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1994.

[27] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: Performance analysis, design and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 909–926, 1998.

[28] F. Gini and G. B. Giannakis, "Generalized differential encoding: a nonlinear signal processing perspective," *IEEE Trans. Signal Processing*, vol. 46, no. 11, pp. 2967–2974, 1998.

[29] M. Ghosh and C. L. Weber, "Maximum likelihood blind equalization," *Optical Engineering*, vol. 31, pp. 1224–1228, June 1992.

[30] K. H. Chang and C. N. Georghiades, "Iterative joint sequence and channel estimation for fast time-varying intersymbol interference channels," in *Proc. IEEE International Conference on Communications (ICC '95)*, pp. 357–361, Seattle, Wash, USA, June 1995.

[31] P. Robertson, P. Hoeher, and E. Villebrun, "Optimal and suboptimal maximum a posteriori algorithms suitable for turbodecoding," *European Transactions on Telecommunications*, vol. 8, no. 2, pp. 119–125, 1997.

[32] Y. Ephraim and N. Merhav, "Hidden Markov processes," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1518–1569, 2002.

[33] C. N. Georghiades and J. C. Han, "Sequence estimation in the presence of random parameters via the EM algorithm," *IEEE Trans. Commun.*, vol. 45, no. 3, pp. 300–308, 1997.

[34] M. J. Omidi, P. G. Gulak, and S. Pasupathy, "Joint data and channel estimation using the per-branch processing method," in *Proc. IEEE Signal Processing Adv. Wireless Commun. (SPAWC '97)*, pp. 384–392, April 1997.

[35] X.-M. Chen and P. A. Hoeher, "Blind turbo equalization for wireless DPSK systems," in *Proc. 4th Int. ITG Conf. on Source and Channel Coding*, pp. 371–378, Berlin, Germany, January 2002.

[36] B. Zervas, J. Proakis, and V. Eyubo, "A quantized channel approach to blind equalization," in *Proc. IEEE International Conference on Communications (ICC '92)*, pp. 1539–1543, Chicago, Ill, USA, June 1992.

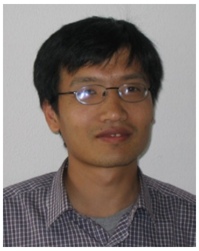
[37] J. Garcia-Frias and J. D. Villasenor, "Combined turbo detection and decoding for unknown ISI channels," *IEEE Trans. Commun.*, vol. 51, no. 1, pp. 79–85, 2003.

[38] P. B. Ha and B. Honary, "Improved blind turbo detector," in *Proc. IEEE Vehicular Technology Conference (VTC '00)*, pp. 1196–1199, Tokyo, Japan, Spring 2000.

[39] K. D. Kammeyer, V. Kuehn, and T. Petermann, "Blind and non-blind turbo estimation for fast fading GSM channels," *IEEE J. Select. Areas Commun.*, vol. 19, no. 9, pp. 1718–1728, 2001.

[40] L. H. C. Lee, "New rate-compatible punctured convolutional codes for Viterbi decoding," *IEEE Trans. Commun.*, vol. 42, no. 12, pp. 3073–3079, 1994.

[41] S. Dolinar and D. Divsalar, "Weight distribution for turbo codes using random and nonrandom permutations," JPL Progress Report 42–122, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, Calif, USA, August 1995.



Xiao-Ming Chen was born in Zhejiang, China, in 1975. He received the B.Sc. and M.Sc. degrees in electrical engineering from Tongji University, Shanghai, China, in 1997 and 2000, respectively. From February 2000 to July 2000 he did his master thesis at the Institute for Communications Engineering, Munich University of Technology, Germany. In October 2000, he joined the Information and Coding Theory Lab, University of Kiel, Germany, as a Ph.D. student and a Teaching Assistant. His research interests include joint data/channel estimation, noncoherent equalization, iterative processing, and signal processing for MIMO systems.

Peter A. Hoeher is a Senior Member of IEEE and a Member of VDE/ITG. He was born in Cologne, Germany, in 1962. He received the Dipl.-Eng. and Dr.-Eng. degrees in electrical engineering from the Technical University of Aachen, Germany, and the University of Kaiserslautern, Germany, in 1986 and 1990, respectively. In October 1998, he joined the University of Kiel, Germany, where he is currently a Professor in electrical engineering. His research interests are in the general area of communication theory with applications in wireless communications and underwater communications. Dr. Hoeher received the Hugo-Denkmeier-Award '90. Since 1999, he serves as an Associated Editor for IEEE Transactions on Communications.