Code-Aided Frame Synchronization and Phase Ambiguity Resolution

Henk Wymeersch, *Member, IEEE*, Heidi Steendam, *Senior Member, IEEE*, Herwig Bruneel, and Marc Moeneclaey, *Fellow, IEEE*

***Abstract*—This contribution deals with two hypothesis testing problems for digital receivers: frame synchronization and phase ambiguity resolution. As current receivers use powerful error-correcting codes and operate at low signal-to-noise ratio (SNR), these problems have become increasingly challenging: one is forced either to waste a part of the bandwidth on training symbols or to consider novel hypothesis testing techniques. We will consider five algorithms for hypothesis testing that exploit properties of the underlying channel code: a re-encoding (REEN) technique, an algorithm we previously derived from the expectation-maximization (EM) algorithm, two recently proposed algorithms known as mode separation (MS) and pseudo-ML (PML), and a technique where all hypotheses are tested *simultaneously* by applying the sum–product algorithm (SPA) to the overall factor graph of the system. These techniques will be compared in terms of their computational complexity, the class of problems to which they can be applied and their error rate performance. Through computer simulations we show that the EM-based and the PML algorithms have excellent performance. The MS, PML, REEN, and EM-based algorithms all have similar complexity, but the latter algorithm is suitable for a much wider range of applications. The SPA has the lowest computational complexity, but might yield poor performance.**

***Index Terms*—Expectation-maximization (EM) algorithm, factor graphs, frame synchronization, turbo synchronization.**

I. Introduction

THE advent of capacity approaching error-correcting codes (such as turbo- and LDPC codes [1], [2]) has directed a lot of research effort toward a whole new class of iterative algorithms. These algorithms are used not only in channel (de)coding, but also source coding [3], [4], equalization [5], demodulation [6], multiuser detection [7], and, more recently, synchronization and estimation [8]–[11]. This contribution will focus on synchronization, more specifically frame synchronization (FS) [12] and phase ambiguity resolution (PAR) [13], both of which are hypothesis testing problems.

Manuscript received November 16, 2004; revised August 9, 2005. This work has been supported by the Interuniversity Attraction Poles (IAP) Program P5/11, Belgian Science Policy, Federal Office for Scientific, Technical and Cultural Affairs, Belgium, and by EU 6FP IST Network of Excellence in Wireless Communications (NEWCOM) funded by the European Commission. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiaodong Wang.

H. Wymeersch is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: hwymeers@mit.edu).

H. Steendam and M. Moeneclaey are with the DIGCOM Research Group, TELIN Department, Ghent University, 9000 Gent, Belgium (e-mail: hs@telin.ugent.be; mm@telin.ugent.be).

H. Bruneel is with the SMACS Research Group, TELIN Department, Ghent University, 9000 Gent, Belgium (e-mail: hb@telin.ugent.be).

Digital Object Identifier 10.1109/TSP.2006.874844

Conventional hypothesis testing algorithms make use of fixed symbols in the burst (often referred to as pilot symbols or training symbols). This results in data-aided (DA) hypothesis testing algorithms. As powerful error-correcting codes achieve low bit-error rates (BERs) at low signal-to-noise ratios (SNRs), synchronization has become an increasingly difficult task: this low-SNR low-BER combination requires DA algorithms with many pilot symbols in the burst, thus effectively reducing both the bandwidth efficiency and power efficiency for data detection. For instance, when 10% of a frame consists of pilot symbols, only 90% of the received power is available for data detection, which constitutes a power loss of almost 0.5 dB.

These problems have motivated many research groups to look into so-called "code-aided" hypothesis testing algorithms. In the context of FS and PAR, we mention the following technical papers. Code-aided frame synchronization was discussed in [11] and [14]–[17]: [14] uses a list-based synchronizer; the pilot sequence is part of the codeword, so the coder is forced into a sequence of known states. The decoder verifies this sequence to determine if frame synchronization is achieved. In [15], the so-called *path surface metric*, based on the forward and backward metrics in the BCJR algorithm [18], is used for frame synchronization. The properties of this metric change when the decoder is not synchronized. In [16], termination symbols in the convolutional codes are taken into account in deriving the maximum-likelihood (ML) frame position. Yet another approach is mentioned in [11], where it was observed that a frame synchronization failure reduces the amplitude of the so-called extrinsic log-likelihood ratios (LLRs) as compared with a synchronized decoder. This idea was reconsidered very recently: a powerful frame synchronizer was proposed in [17], based on mode separation (MS): the extrinsic LLRs computed by the decoder have a bimodal distribution. The distance between these modes is maximal when the packet is perfectly synchronized. On the other hand, code-aided phase ambiguity resolution was investigated in [19], where it was observed that the statistics of the branch metrics in the Viterbi decoder change depending on the phase shift. We should mention that many code-aided FS algorithms can be applied to PAR with only minor modifications. Indeed, we have proposed an algorithm based on the expectation-maximization (EM) algorithm [20] suitable for both FS and PAR in [21]–[23].

Very recently, algorithms that perform *joint* estimation and detection have begun to surface [24]–[27]. In most cases, they operate on a graphical representation (a factor graph) of the overall system, including the unknown synchronization parameters. By applying the well-known sum–product algorithm (SPA)

[28] to this factor graph, one can perform near-optimum iterative data detection. These algorithms may be computationally quite demanding and commonly cannot be used with off-the-shelf components [29].

In this contribution, we will examine the following different practical code-aided hypothesis testing techniques:

- a re-encoding (REEN) technique;
- an EM-based algorithm (with hard and soft decisions);
- the powerful MS algorithm from [17];
- a technique known as pseudo-ML (PML) from [23];
- the SPA for joint decoding and hypothesis testing.

To our knowledge, applying the sum–product algorithm in this way has never been investigated.[1] The hard EM and the REEN technique exploit *hard* information from the decoder (i.e., operating on hard decisions of symbols), while the other algorithms exploit *soft* information from the decoder. We will compare these algorithms in terms of computational complexity and performance.

This paper is organized as follows. In Section II, the system model is presented. As soft decoding and hypothesis testing can be cast in the framework of factor graphs, we give a short overview of factor graphs and the sum-product algorithm in Section III. In Section IV, we will deal with the problem at hand from two points of view: codeword detection and hypothesis testing. Numerical results are presented in Section V, both for a "toy"code that can be decoded optimally and a powerful turbo code. Finally, conclusions are presented in Section VI. Our main conclusion is that although the SPA has some very attractive properties, is it systematically outperformed by the EM-based algorithm. Indeed, for the cases we investigated, the EM-based algorithm achieves virtually optimal performance. Furthermore, it has some attractive properties that are not present in other code-aided hypothesis testing algorithms.

II. System Model

The transmitter sends a block ($\mathbf{b}$) of N_b information bits to the receiver. The N_b bits are first protected by a rate N_b/N_c error-correcting code, resulting in a binary sequence $\mathbf{c} \doteq \chi(\mathbf{b})$. The coded bits are mapped to an M-point signaling constellation Ω using a mapping function $\varphi : \{0,1\}^{N_c} \rightarrow \Omega^{N_s}$, with $N_s = \lceil N_c/\log_2(M) \rceil$; here, $\lceil x \rceil$ represents the smallest integer greater than or equal to x. We do not put any constraints on the kind of code nor on the way how bits are mapped to the constellation. Hence, the algorithms we will consider can be applied to a wide range of scenarios (including uncoded transmission, bit-interleaved coded modulation, trellis-coded modulation, multidimensional modulation, multiantenna systems, etc.). These operations result in a sequence $\mathbf{a} = [a_0, a_1, \ldots, a_{N_s-1}]$ of N_s symbols, with $a_k \in \Omega$ and $E\left[|a_k|^2\right] = 1$. The transmitted complex baseband signal is then given by

$$s(t) = \sqrt{E_s} \sum_{k=0}^{N_s-1} a_k p(t - kT) \tag{1}$$

[1]Except for [23], where hypothesis testing was performed on the factor graph using an ad hoc criterion.

where E_s denotes the energy per transmitted symbol, $1/T$ is the symbol rate and $p(t)$ is a unit-energy square-root Nyquist transmit pulse. E_s is related to the energy per bit (E_b) through $E_s = (\log_2 M) N_b E_b / N_c$.

The signal $s(t)$ propagates through an AWGN channel and is affected by an unknown propagation delay τ and an unknown carrier phase θ. The signal is further corrupted by thermal noise. Hence, the signal $r(t)$ at the input of the receiver is given by

$$r(t) = s(t - \tau) e^{j\theta} + n(t) \tag{2}$$

where $n(t)$ is a complex additive white Gaussian noise (AWGN) process with spectral density N_0. We further break up τ and θ as follows:

$$\begin{aligned} \tau &= k_\tau T + \epsilon_\tau - \frac{T}{2} \leq \epsilon_\tau < \frac{T}{2} \\ \theta &= k_\theta \psi + \epsilon_\theta - \frac{\psi}{2} \leq \epsilon_\theta < \frac{\psi}{2} \end{aligned} \tag{3}$$

where k_τ and k_θ are integers and ψ denotes the smallest angle of rotational symmetry of the constellation Ω. Hence, for M-PSK signaling, $\psi = 2\pi/M$, while for M-QAM, we have $\psi = \pi/4$ [30]. Generally, the receiver roughly determines the arrival of the burst, e.g., through some form of energy detection. However, because of the presence of noise and signal disturbances, the starting point of the burst can only be determined within an uncertainty range of M_τ symbol durations. In this case, the uncertainty range of k_τ and k_θ is given by $(k_\tau, k_\theta) \in \{0, \ldots, M_\tau - 1\} \times \{0, \ldots, M_\theta - 1\}$, with $M_\theta = 2\pi/\psi$. In many cases ϵ_τ can be estimated very accurately by means of an algorithm that exploits the cyclostationarity of the signal (e.g., see [31]). Similarly, ϵ_θ can be estimated based on statistical properties of the data symbols (e.g., see [32]). Precisely these same properties (i.e., cyclostationarity and statistics of the data symbols) make it very difficult to determine k_τ and k_θ, respectively. For the remainder of this paper, we assume ϵ_τ and ϵ_θ have been accurately estimated by the receiver's symbol timing and carrier phase synchronizer, so that only k_τ and k_θ still need to be determined. We apply the received complex baseband signal to a matched filter, sample the output at time instants $t = kT + \varepsilon_\tau$ and multiply the resulting samples with $\exp(-j\varepsilon_\theta)$. We assume that the interval between bursts exceeds $M_\tau T$ and contains only noise. This permits successive frames to be treated independently. It can be shown that the resulting sufficient statistics can be written as a row-vector of length $K = N_s + M_\tau - 1$

$$\mathbf{r} = [\mathbf{0}_{k_\tau}\, \mathbf{a}\, \mathbf{0}_{M_\tau - 1 - k_\tau}]\, e^{jk_\theta\psi} + \mathbf{n} \tag{4}$$

$$\doteq D^{k_\tau}(\mathbf{a})\, e^{jk_\theta\psi} + \mathbf{n} \tag{5}$$

where $\mathbf{0}_m$ is an array consisting of m zeros, $\mathbf{a}$ is the unknown data sequence, and $\mathbf{n}$ a complex AWGN vector of length K with independent real and imaginary parts and variance $\sigma^2 = N_0/(2E_s)$. We have also defined an invertible function $D^k : \Omega^{N_s} \rightarrow \{\Omega \cup \{0\}\}^K$, for $k \in \{0, \ldots, M_\tau - 1\}$ that prefixes to its argument k zeros and postfixes $M_\tau - 1 - k$ zeros. The inverse of this function will be denoted by $D^{-k}(.)$, so that $D^{-k}\left(D^k(\mathbf{a})\right) = \mathbf{a}$. The prefix and suffix zeros in (4) reflect the uncertainty with regard to the frame starting point.

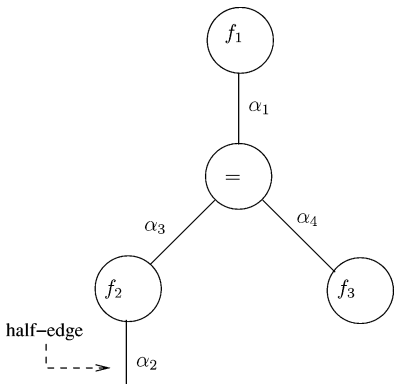


Fig. 1. Simple factor graph of the function $f(\alpha_1,\alpha_2,\alpha_3,\alpha_4) = f_1(\alpha_1)\, f_2(\alpha_2,\alpha_3)\, f_3(\alpha_4) I[\alpha_1{=}\alpha_3{=}\alpha_4]$.

The goal of the receiver is to recover the information bits $\mathbf{b}$. In order to do this, the receiver requires to take into account the unknown parameters k_θ and k_τ. Before we deal with this problem, we introduce some concepts related to factor graphs and the sum–product algorithm, which will be useful later on.

III. Factor Graphs and the Sum–Product Algorithm

A. Factor Graphs

We will use the factor graphs that were introduced in [33]. A factor graph is a diagram that represents the factorization of a function of several variables

$$f(\alpha_1,\alpha_2,\ldots,\alpha_N) = \prod_j f_j(A_j) \tag{6}$$

where A_j is a subset of $\boldsymbol{\alpha} = \{\alpha_1,\alpha_2,\ldots,\alpha_N\}$. A factor graph consists of nodes, edges and half-edges (the latter are connected to only one node). The factor graph is related to the function $f(.)$ as follows: There is a node for every factor $f_j(.)$ and one (half-) edge for every variable α_k. Node f_j is connected to variable $\alpha_k \iff \alpha_k \in A_j$. Finally, edges (respectively, half-edges) are connected to exactly two nodes (respectively, one node). We also introduce the notion of the so-called *indicator function*, $I[P]$: for a predicate P, $I[P] = 0$ if P is false and $I[P] = 1$ if P is true. A simple factor graph is shown in Fig. 1, representing the function $f(\alpha_1,\alpha_2,\alpha_3,\alpha_4) = f_1(\alpha_1)\, f_2(\alpha_2,\alpha_3)\, f_3(\alpha_4) I[\alpha_1{=}\alpha_3{=}\alpha_4]$.

B. Sum–Product Algorithm

The *marginal* corresponding to a variable α_i is a function defined as

$$g_i(\alpha_i) = \sum_{\sim\{\alpha_i\}} f(\alpha_1,\alpha_2,\ldots,\alpha_N) \tag{7}$$

where $\sim\{\alpha_i\}$ represents the not-sum: the summation over all variables except α_i. The SPA is a technique that computes all the marginals of a function in an efficient manner, based on the factor graph representation of that function. The SPA operates by passing *messages* over the edges of the factor graph of $f(\alpha_1,\alpha_2,\ldots,\alpha_N)$. A message over an edge is a function of the variable corresponding to that edge. We denote the message from function (node) f_k to variable (edge) α_i by $\mu_{f_k\to\alpha_i}(.)$. Similarly, we denote the message from variable (edge) α_i to function (node) f_l by $\mu_{\alpha_i\to f_l}(.)$. When the factor graph is a tree (i.e., it contains no cycles), the SPA computes the marginals as follows (see [28] for an excellent introduction on factor graphs):

1) **Initialization**: Half-edges transmit messages equal to 1, while degree-one nodes (corresponding to a function, say $f_k(.)$) transmit the message[2] $f_k(.)$.
2) **Message update**: Once a degree-d node has received messages on $d-1$ edges, it can compute an outgoing message over the remaining edge (evaluated in α_i)

$$\mu_{f_k\to\alpha_i}(\alpha_i) = \sum_{\sim\{\alpha_i\}} f_k(\alpha_i,\alpha_{j\neq i}) \prod_{j\neq i} \mu_{\alpha_j\to f_k}(\alpha_j). \tag{8}$$

3) **Termination**: once all messages have been computed, the marginal of a variable, say α_i, is given by (evaluated in $\alpha_i = a$)

$$g_i(\alpha_i = a) = \mu_{\alpha_i\to f_k}(a) \times \mu_{f_k\to\alpha_i}(a) \tag{9}$$

where f_k is an arbitrary function such that $\alpha_i \in A_k$.

In the case of the factor graph from Fig. 1, the reader can easily verify that SPA yields the correct marginals: $g_1(\alpha_1) = \sum_{\alpha_2} f_1(\alpha_1)\, f_3(\alpha_1)\, f_2(\alpha_1,\alpha_2)$ and $g_2(\alpha_2) = \sum_{x_1} f_1(\alpha_1)\, f_3(\alpha_1)\, f_2(\alpha_1,\alpha_2)$. Note that $g_1(\alpha_1) = g_3(\alpha_1) = g_4(\alpha_1)$.

Computational Complexity and Cycles: From (8), it can be seen that the computational complexity of the SPA depends on the number of elements in the domain of $\sim\{\alpha_i\}$ and the nature of the functions $f_k(\alpha_i,\alpha_{j\neq i})$. To minimize the computational complexity, we would like the variables α_j to be defined over small domains (i.e., so that α_j can take on few values) and the functions $f_k(\alpha_i,\alpha_{j\neq i})$ to depend on only few variables α_j. This goal can be achieved by decomposing functions into less complex functions and introducing additional variables. This process of graph-transformation has one important drawback: the creation of cycles. Cycles in a factor graph lead to an iterative version of the SPA, which now no longer computes the exact marginals, but rather approximations of them. In this iterative SPA, when messages are not available (due to cyclic dependencies), they are simply replaced with uniform messages, equal to 1 over the corresponding domain. As long as cycles are sufficiently long, the performance loss may be acceptable.

C. Factor Graphs in Communications

A common application in communications research is the factorization of an *a posteriori* distribution. Suppose we wish to estimate a parameter (-vector) $\mathbf{d}_e$ in the presence of a nuisance parameter $\mathbf{d}_n$ from an observation $\mathbf{r}$. Assume that $\mathbf{d}_e$ and $\mathbf{d}_n$ are independent. Then, the joint *a posteriori* distribution is given by

$$p(\mathbf{d}_e,\mathbf{d}_n|\,\mathbf{r}) \propto p(\mathbf{r}|\,\mathbf{d}_e,\mathbf{d}_n)\, p(\mathbf{d}_e)\, p(\mathbf{d}_n). \tag{10}$$

[2]Remember that messages are functions.

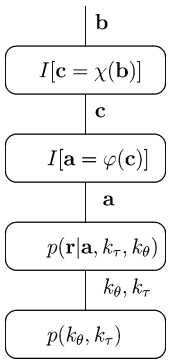


Fig. 2. Optimal sequence detection in the presence of unknown delay shifts and phase ambiguities. This overall factor graph corresponds to a function proportional to $p(\mathbf{a},\mathbf{b},\mathbf{c},k_\tau,k_\theta|\mathbf{r})$. The observation $\mathbf{r}$ should be treated as a parameter in this graph (rather than a variable).

If we create a factor graph of this factorization of $p(\mathbf{d}_e,\mathbf{d}_n|\mathbf{r})$ and apply the SPA, we can obtain (up to an irrelevant constant) the marginals $p(\mathbf{d}_e|\mathbf{r})$ and $p(\mathbf{d}_n|\mathbf{r})$. Based on these marginals, a decision with regard to $\mathbf{d}_e$ can be taken, as follows:

$$\hat{\mathbf{d}}_e = \arg\max_{\mathbf{d}_e} p(\mathbf{d}_e|\mathbf{r}). \tag{11}$$

When $\mathbf{d}_e$ or $\mathbf{d}_n$ can take on many values (which is commonly the case) the SPA may be too complex to implement. Fortunately, $p(\mathbf{r}|\mathbf{d}_e,\mathbf{d}_n)$, $p(\mathbf{d}_e)$ and $p(\mathbf{d}_n)$ tend to be highly structured in function of their components $d_{e,k}$ and $d_{n,k}$, so that we can factorize these three functions. This will result in the introduction of new variables and functions and possibly create cycles in the transformed factor graph. Applying the SPA then leads to (approximate) marginals $p(d_{e,k}|\mathbf{r})$ from which (near-)optimal decisions with regard to the components $d_{e,k}$ can be taken.

Many state-of-the-art algorithms are based on this principle, including decoding algorithms for turbo codes, LDPC codes, RA codes, bit-interleaved modulation with iterative decoding, ML detection for multiantenna systems, etc.

IV. Data Detection and Hypothesis Testing

Let us now return to the problem at hand: frame synchronization and phase ambiguity resolution. Our ultimate goal is to recover the transmitted data sequence $\mathbf{b}$. In principle, this can be achieved by estimating the transmitted codeword, irrespective of the delay shift and phase ambiguity (e.g., maximizing $p(\mathbf{b}|\mathbf{r})$ with regard to $\mathbf{b}$). Such algorithms operate on a factor graph where the synchronization parameters k_θ and k_τ appear as variables (i.e., as edges). We will name the corresponding factor graph the *overall factor graph* (an example of which is shown in Fig. 2). Related algorithms will be described in Section IV-A.

Alternatively, we can first try to determine the delay shift and the phase ambiguity and then perform data detection (e.g., maximizing $p\left(\mathbf{b}\middle|\mathbf{r},\hat{k}_\theta,\hat{k}_\tau\right)$ with regard to $\mathbf{b}$). Such algorithms operate on a factor graph which assumes perfect knowledge of both k_θ and k_τ. This implies that k_θ and k_τ do not appear as variables in the factor graph. We will name the corresponding factor graph the *synchronized factor graph*. For a given estimate $\left(\hat{k}_\theta,\hat{k}_\tau\right)$,

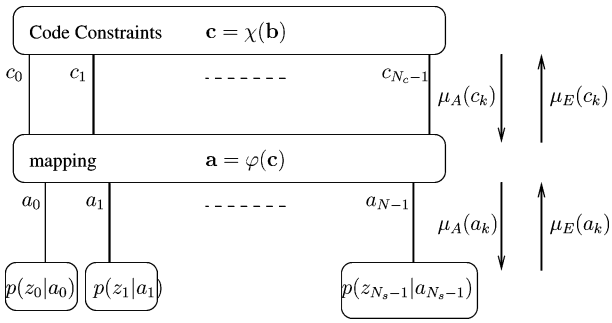


Fig. 3. Synchronized factor graph, corresponding to the function $p(\mathbf{a},\mathbf{b},\mathbf{c}|\mathbf{z})$.

this graph accepts a vector $\mathbf{z} = D^{-\hat{k}_\tau}(\mathbf{r})\, e^{-j\hat{k}_\theta\psi}$ of length N_s. We then have

$$p\left(\mathbf{c},\mathbf{b},\mathbf{a}\middle|\mathbf{r},\hat{k}_\theta,\hat{k}_\tau\right) \propto p\left(\mathbf{r}\middle|\mathbf{a},\hat{k}_\theta,\hat{k}_\tau\right) p(\mathbf{a},\mathbf{b},\mathbf{c}) \tag{12}$$

$$\propto p(\mathbf{z}|\mathbf{a})\, p(\mathbf{a}|\mathbf{c})\, p(\mathbf{c}|\mathbf{b})\, p(\mathbf{b}) \tag{13}$$

$$\propto \prod_{k=0}^{N_s-1} p(z_k|a_k)\, I[\mathbf{a}=\varphi(\mathbf{c})]\, I[\mathbf{c}=\chi(\mathbf{b})]. \tag{14}$$

The resulting synchronized factor graph is shown in Fig. 3. Applying the SPA on this graph yields the a posteriori probabilities (APPs) $\left\{p\left(b_k\middle|\mathbf{r},\hat{k}_\theta,\hat{k}_\tau\right)\right\}$, $\left\{p\left(c_k\middle|\mathbf{r},\hat{k}_\theta,\hat{k}_\tau\right)\right\}$ and $\left\{p\left(a_k\middle|\mathbf{r},\hat{k}_\theta,\hat{k}_\tau\right)\right\}$. Related hypothesis testing algorithms are described in Sections IV-B–IV-C.

For each of the algorithms we discuss, we will determine the total computational complexity, i.e., the number of operations required to decode the packet. As a point of reference, the SPA on the synchronized factor graph will generally be iterative, requiring, say, I_c iterations to decode a packet, so that the computational cost of the synchronized factor graph is $\mathcal{O}(I_c C_c N_b)$, where C_c is a constant that depends on the type of code (e.g., C_c is proportional to 2^ν for convolutional and turbo codes, with ν denoting the constraint length).

A. Data Detection

1) Optimal Sequence Detection: The frame error rate $P\left[\hat{\mathbf{b}}\neq\mathbf{b}\right]$ is minimized when applying MAP sequence detection

$$\hat{\mathbf{b}}_{\mathrm{MAP}} = \arg\max_{\mathbf{b}} p(\mathbf{b}|\mathbf{r}) \tag{15}$$

$$= \arg\max_{\mathbf{b}} \left\{ \sum_{k_\tau,k_\theta} p(\mathbf{r}|\mathbf{b},k_\theta,k_\tau) \right\} \tag{16}$$

$$= \arg\max_{\mathbf{b}} \left\{ \sum_{k_\tau,k_\theta} \exp\left(-\frac{1}{2\sigma^2}\left\|\mathbf{r}-D^{k_\tau}\left(\mathbf{a}e^{jk_\theta\psi}\right)\right\|^2\right) \right\} \tag{17}$$

with $\mathbf{a} = \varphi(\chi(\mathbf{b}))$. The *a posteriori* distribution $p(\mathbf{a},\mathbf{b},\mathbf{c},k_\tau,k_\theta|\mathbf{r})$ can be written as

$$p(\mathbf{a},\mathbf{b},\mathbf{c},k_\tau,k_\theta|\mathbf{r}) \propto p(\mathbf{r}|\mathbf{a},k_\tau,k_\theta)\, p(k_\tau,k_\theta)\, I[\mathbf{a}=\varphi(\mathbf{c})]\, I[\mathbf{c}=\chi(\mathbf{b})]. \tag{18}$$

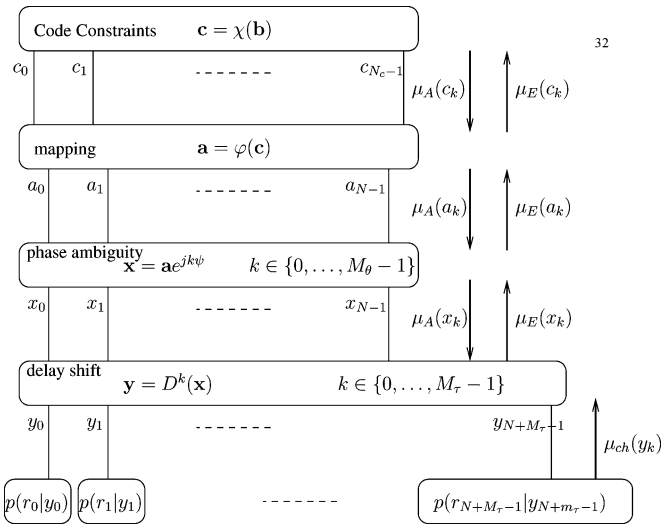


Fig. 4. Overall factor graph of a coded system with phase ambiguity and delay shift, corresponding to the function $p(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{x}, \mathbf{y} | \mathbf{r})$.

The overall factor graph of this factorization is depicted in Fig. 2. In principle, applying the SPA on this overall factor graph yields the exact marginal $p(\mathbf{b}|\mathbf{r})$.

Computational Complexity: Unfortunately, solving this problem requires a maximization over all 2^{N_b} possible codewords. This is generally an intractable problem. The performance of this detection algorithm will serve as a benchmark for other, practical algorithms. This technique has a computational complexity of the order $\mathcal{O}\left(M_\tau M_\theta 2^{N_b}\right)$.

2) Optimal Bit Detection: The BER $P\left[\hat{b}_k \neq b_k\right]$ is minimized when applying MAP detection on each bit individually, as follows:

$$\hat{b}_k = \arg \max_{b \in \{0,1\}} p(b_k | \mathbf{r}). \tag{19}$$

The required *a posteriori* probabilities $p(b_k|\mathbf{r})$ can be determined by applying the SPA on an overall factor graph, where where the variables are now the *scalar components* of $[\mathbf{b}, \mathbf{c}, \mathbf{a}, \mathbf{x}, \mathbf{y}]$, where $\mathbf{x} = \mathbf{a}e^{jk_1\psi}$ for some $k_1 \in \{0, \ldots, M_\theta - 1\}$ and $\mathbf{y} = D^{k_2}(\mathbf{x})$ for some $k_2 \in \{0, \ldots, M_\tau - 1\}$. The factor graph of the *a posteriori* distribution $p(\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{x}, \mathbf{y}, k_\tau, k_\theta | \mathbf{r})$ is depicted in Fig. 4. This factor graph consists of the following nodes:

- the code constraints node, representing $I[\mathbf{c} = \chi(\mathbf{b})]$ (it forces the vector $\mathbf{c} \in \{0,1\}^{N_c}$ to satisfy the code-constraints, and in most cases, this function can be itself factorized and represented by a factor graph);
- the mapper node representing $I[\varphi(\mathbf{c}) = \mathbf{a}]$ (it forces the vector $\mathbf{a} \in \Omega^N$ to be a mapped word (not necessarily a codeword!); generally, this node would consist of many disconnected mapper nodes, each mapping $\log_2 M$ coded bits to a single constellation point;
- the phase ambiguity constraint node[3] $I[\mathbf{x} = \mathbf{a}e^{jk\psi},\ k = 0 \vee \cdots \vee, k = M_\theta - 1]$; only vectors $\mathbf{x} \in \Omega^N$ corresponding to rotated $\mathbf{a}$'s are valid (a detailed view of this node is shown in Fig. 5);

[3]Here, $\vee$ represents the logical OR.

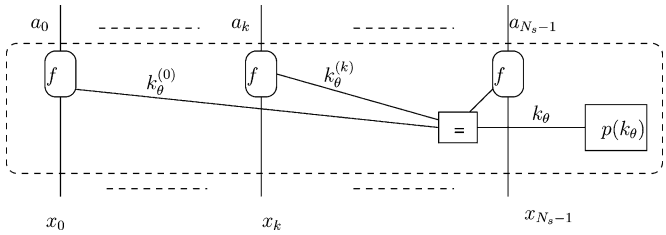


Fig. 5. Phase ambiguity node: Detailed factor graph view. The function f enforces the following constraint: $f\left(a_k, x_k, k_\theta^{(k)}\right) = I\left[x_k = a_k e^{jk_\theta^{(k)}\psi}\right]$.

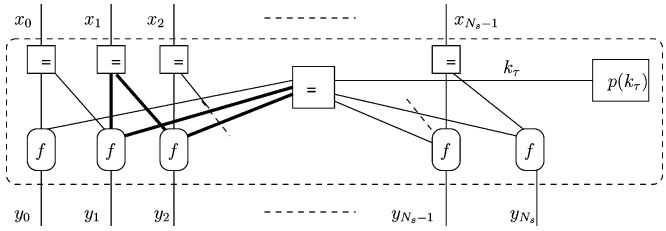


Fig. 6. Delay shift node: Detailed factor graph view for $M_\tau = 2$. The function f enforces the following constraint: $f(x_k, x_{k-1}, y_k, k_\tau) = I[(x_k = y_k \wedge k_\tau = 0) \vee (x_{k-1} = y_k \wedge k_\tau = 1)]$. A cycle of length four is shown in bold.

- the delay shift node $I[\mathbf{y} = D^k(\mathbf{x}),\ k = 0 \vee \cdots \vee k = M_\tau - 1]$: the vector of noise-free matched filter outputs $\mathbf{y} \in (\Omega \cup \{0\})^K$ can only be a delay-shifted version of $\mathbf{x}$ (a detailed view of this node is shown in Fig. 6);
- observation nodes: these correspond to the factorization of the likelihood function $p(\mathbf{r}|\mathbf{y}, k_\tau, k_\theta)$, and depend on the statistics of the noise.

Applying the SPA will yield (approximations of) $p(b_k|\mathbf{r})$, from which near-optimal decisions with regard to b_k can be taken. To lighten the notation, we will introduce a different labeling of the messages (seen in Fig. 4): upward messages will be subscripted[4] by "E", while downward messages are subscripted by "A". Messages from the channel-observation nodes are subscripted with "ch". Additional information regarding the exact computation of these messages on this particular graph can be found in [23], [26].

For phase ambiguity resolution, assuming equiprobable ambiguities, it is readily verified that the first time the upward messages $\mu_E(a_k)$ are computed, they are given by (see Fig. 5)

$$\mu_E(a_k = \omega) = \frac{1}{M_\theta} \sum_{k_\theta^{(k)}} \mu_E\left(\omega e^{jk_\theta^{(k)}\psi}\right). \tag{20}$$

Due to the rotational symmetries in the constellation, the messages $\mu_E(a_k)$ will not convey much useful information. In fact, for many practical codes and mapping schemes, $\mu_E(a_k)$ will be uniform, and will result in uniform $\mu_A(a_k)$, so that SPA converges at the first iteration but is not able to provide the receiver with any useful information with regard to $\mathbf{b}$. Hence, PAR using the SP algorithm can not reliably be achieved.

As far as frame synchronization is concerned, from Fig. 6 it is clear that many cycles of length 4 are present. It is well known that the approximations of the marginal probability mass

[4]In keeping with turbo principle parlance, E and A stand for "extrinsic" and "a priori," respectively.

functions (pmfs) become less accurate as the girth[5] of the factor graph decreases. In fact, avoiding short cycles is one of the main criteria in LDPC code design [28], [34].

Computational Complexity: This technique leads to an iterative SPA (with, say I_{bd} iterations) and a computational complexity that is linear[6] in N_b: $\mathcal{O}(I_{bd}C_cN_b)$, where $I_{bd} \geq I_c$.

B. Hypothesis Testing

In many cases, it is preferred to first determine an estimate of k_θ and k_τ, then correct $\mathbf{r}$ accordingly and perform decoding using the synchronized detector. This boils down to treating the coded symbols as nuisance parameters in trying to determine the phase ambiguity and the delay shift.

1) Optimal Hypothesis Testing: The error probability $P\left[\left[\hat{k}_\tau, \hat{k}_\theta\right] \neq [k_\tau, k_\theta]\right]$ is minimized by applying the MAP criterion

$$\left[\hat{k}_\tau, \hat{k}_\theta\right] = \arg\max_{k_\tau, k_\theta} p(k_\tau, k_\theta \,|\mathbf{r}) \tag{21}$$

$$= \arg\max_{k_\tau, k_\theta} \left\{ \sum_{\mathbf{b}} p(\mathbf{r} \,|\mathbf{b}, k_\theta, k_\tau) \right\}. \tag{22}$$

The distribution $p(k_\tau, k_\theta \,|\mathbf{r})$ can be interpreted as a marginal of $p(\mathbf{a}, \mathbf{b}, \mathbf{c}, k_\tau, k_\theta \,|\mathbf{r})$, so that it can be computed by applying the SPA on the graph from Fig. 2.

Computational Complexity: As the summation in (22) goes over 2^{N_b} terms and needs to be performed for all hypotheses, this technique has a computational complexity of the order $\mathcal{O}\left(M_\tau M_\theta 2^{N_b}\right)$.

2) Data-Aided Hypothesis Testing: When $\mathbf{a}$ consists partly of pilot symbols, one can obtain a low-complexity DA algorithm from (22) by keeping only one term in the summation. After some straightforward manipulations, we obtain

$$\left[\hat{k}_{\tau,DA}, \hat{k}_{\theta,DA}\right] = \arg\max_{k_\tau, k_\theta} \left\{ \Re\left\{ \mathbf{r}^H D^{k_\tau}\left(\mathbf{a}_p e^{jk_\theta\psi}\right)\right\}\right\} \tag{23}$$

where $\mathbf{a}_p$ is obtained by replacing in $\mathbf{a}$ all nonpilot symbols by zeros. As we mentioned in the introduction, adding pilot symbols reduces the spectral efficiency and power efficiency of the system.

Computational Complexity: This technique has very low complexity, so the total complexity goes into the SPA on the synchronized factor graph, which results in total computational cost of the order $\mathcal{O}(I_cC_cN_b)$.

C. Practical Code-Aided Hypothesis Testing

As the optimal technique described in the previous section cannot be implemented in practice, we will consider four tractable code-aided hypothesis testing algorithms. These algorithms can be interpreted as breaking the factor graph from Fig. 4 by testing the different hypothesis sequentially (rather than in parallel). Different algorithms are then obtained depending on how the exploit the messages in the corresponding sum–product algorithms, in order to find the "best" estimate of the delay shift and phase ambiguity.

[5]Girth = length of the shortest cycle.

[6]We assume a vertical scheduling. Per node only a single iteration is performed.

The available information computed by the synchronized factor graph (for given trial values k_θ and k_τ) are $\{\mu_E(a_k)\}$, $\{\mu_E(c_k)\}$, $\{\mu_A(a_k)\}$, $\{\mu_A(c_k)\}$, as well as (approximations of) the APPs $\{p(b_k \,|\mathbf{r}, k_\theta, k_\tau)\}$, $\{p(c_k \,|\mathbf{r}, k_\theta, k_\tau)\}$ and $\{p(a_k \,|\mathbf{r}, k_\theta, k_\tau)\}$.

1) Re-Encoding: A first technique is fairly standard and operates as follows: the *joint* ML data detection and hypothesis testing algorithm is obtained by maximizing the likelihood function $p(\mathbf{r} \,|\mathbf{b}, k_\theta, k_\tau)$ [35]

$$\left[\hat{\mathbf{b}}, \hat{k}_\theta, \hat{k}_\tau\right] = \arg\max_{\mathbf{b}, k_\theta, k_\tau} p(\mathbf{r} \,|\mathbf{b}, k_\theta, k_\tau). \tag{24}$$

This leads to the following algorithm. We first define, for given each hypothesis (k_θ, k_τ):

$$\hat{\mathbf{b}}(k_\theta, k_\tau) = \arg\max_{\mathbf{b}} p(\mathbf{r} \,|\mathbf{b}, k_\theta, k_\tau). \tag{25}$$

Finally, the estimate of the delay shift and phase ambiguity are given by

$$\left[\hat{k}_\theta, \hat{k}_\tau\right] = \arg\max_{k_\theta, k_\tau} p\left(\mathbf{r} \,\middle|\, \hat{\mathbf{b}}(k_\theta, k_\tau), k_\theta, k_\tau\right). \tag{26}$$

In practice, $\hat{\mathbf{b}}(k_\theta, k_\tau)$ is not available. The synchronized factor graph gives us

$$\hat{b}_k(k_\theta, k_\tau) = \arg\max_{b_k} p(b_k \,|\mathbf{r}, k_\theta, k_\tau),\ k = 0, \ldots, N_b - 1. \tag{27}$$

Replacing $\hat{\mathbf{b}}(k_\theta, k_\tau)$ with $\left[\hat{b}_0(k_\theta, k_\tau), \ldots, \hat{b}_{N_b-1}(k_\theta, k_\tau)\right]$ in (26) results in the *re-encoding technique* (since $\hat{\mathbf{b}}(k_\theta, k_\tau)$ must be re-encoded and remapped to constellation symbols, in order to evaluate the likelihood function $p\left(\mathbf{r} \,\middle|\, \hat{\mathbf{b}}(k_\theta, k_\tau), k_\theta, k_\tau\right)$).

Alternatively, the re-encoding step can be avoided by instead computing

$$\left[\hat{k}_\theta, \hat{k}_\tau\right] = \arg\max_{k_\theta, k_\tau} p(\mathbf{r} \,|\hat{\mathbf{a}}(k_\theta, k_\tau), k_\theta, k_\tau) \tag{28}$$

where the components of $\hat{\mathbf{a}}(k_\theta, k_\tau)$ are given by

$$\hat{a}_k(k_\theta, k_\tau) = \arg\max_{a_k} p(a_k \,|\mathbf{r}, k_\theta, k_\tau),\ k = 0, \ldots, N_s - 1. \tag{29}$$

We will refer to these techniques as REEN (26) and hard-decision-directed (HDD) (28). Note that both REEN and HDD exploit hard information from the decoder.

2) Mode Separation: A second technique was recently proposed in [17] and also decodes the packet for each of the possible hypotheses. It has been observed that the distribution of the LLRs of the messages $\mu_A(c_k)$ is bimodal when the receiver is synchronized. Based on this observation, we determine the correct hypothesis through MS, defined as the distance between the modes of the LLRs. We then select the hypothesis corresponding to the maximum MS. Mathematically, this translates in to determining $\lambda_k \doteq \log(\mu_A(c_k = 1)) - \log(\mu_A(c_k = 0))$, for $k = 0, \ldots, N_c - 1$, computing the modes, $M_+ = E[\lambda | \lambda > 0]$, $M_- = E[\lambda | \lambda < 0]$, and the mode separation $MS(k_\tau, k_\theta) = M_+ - M_-$, so that

$$\left[\hat{k}_\tau, \hat{k}_\theta\right] = \arg\max_{k_\tau, k_\theta} MS(k_\tau, k_\theta). \tag{30}$$

One of the main drawbacks of this algorithm is that MS cannot exploit the presence of pilot symbols or pilot bits (as they will

have the same $\mu_A(c_k)$ for all hypotheses). This means that, in order to combine MS with pilot symbols, one must resort to suboptimal techniques such as list-synchronizers [14].

3) Pseudo-ML: The third technique was recently proposed in [23] and is closely related to the sum–product algorithm. For each hypothesis (k_θ, k_τ), we decode the packet and compute a pseudomarginal by multiplying the messages over the a_k-edges. We define

$$\mathrm{PML}(k_\tau, k_\theta) = \sum_{k=0}^{N_s-1} \log\left(\sum_{\omega\in\Omega} \mu_E(a_k=\omega)\times\mu_A(a_k=\omega)\right). \tag{31}$$

Hypothesis testing is then performed as follows:

$$\left[\hat{k}_\tau, \hat{k}_\theta\right] = \arg\max_{k_\tau,k_\theta} \mathrm{PML}(k_\tau, k_\theta). \tag{32}$$

We note that in [23], the pseudo-ML technique was combined (in an ad hoc fashion) with phase estimation.

4) Discrete EM Algorithm: The final algorithm is based on the EM algorithm [20]. The EM algorithm a method that iteratively solves the problem of ML estimation of a parameter $\mathbf{d}_e$ in the presence of a nuisance parameter $\mathbf{d}_n$. However, the EM algorithm suffers from convergence problems in the case of discrete parameter estimation. To avoid these problems, we define the discrete EM algorithm as (see our related work in [22], [36], and [37] for additional details):

$$\hat{\mathbf{d}}_e = \arg\max_{\mathbf{d}_e\in S} Q(\mathbf{d}_e) \tag{33}$$

where S is the discrete and finite set over which $\mathbf{d}_e$ is defined and $Q(\mathbf{d}_e)$ is given by

$$Q(\mathbf{d}_e) = \sum_{\mathbf{d}_n} p(\mathbf{d}_n|\,\mathbf{r}, \mathbf{d}_e)\log p(\mathbf{r}, \mathbf{d}_n|\,\mathbf{d}_e). \tag{34}$$

In our case, with $\mathbf{d}_n = \mathbf{a}$ and $\mathbf{d}_e = [k_\theta, k_\tau]$, we readily obtain

$$Q(k_\theta, k_\tau) = \Re\left\{\mathbf{r}^H D^{k_\tau}\left(\tilde{\mathbf{a}}e^{jk_\theta\psi}\right)\right\} - \frac{1}{2}\sum_{k=0}^{N-1}\widetilde{|a_k|^2} \tag{35}$$

where $\tilde{\mathbf{a}} = [\tilde{a}_0, \ldots, \tilde{a}_{N_s-1}]$ is a vector of so-called soft symbol decisions (SSD) and is computed using the APPs of the coded symbols. Each component $\tilde{a}_k$ is a weighted average of all possible symbols: $\tilde{a}_k = \sum_{\omega\in\Omega}\omega \times p(a_k = \omega\,|\mathbf{r}, k_\tau, k_\theta)$. Similarly, $\widetilde{|a_k|^2} = \sum_{\omega\in\Omega}|\omega|^2 \times p(a_k = \omega\,|\mathbf{r}, k_\tau, k_\theta)$. Note that for M-PSK mapping, $|\omega|^2 = 1, \forall\omega\in\Omega$, in which case the last term in (35) may be dropped.

In discrete EM algorithm, pilot symbols are exploited in a natural way: simply observe the similarities between (23) and (35). Even pilot bits and uncoded bits can be exploited through the *a posteriori* probabilities. The discrete EM algorithm accepts all available information, both from the decoder and from the channel. Contrary to the previous three hypothesis testing algorithms, the EM-based technique can easily be modified to include code-aided estimation of continuous parameters [36], [38], such as the fractional parts ε_τ and ε_θ of the time delay and the carrier phase (see (3)).

Finally, we note that when we replace the SSD in the discrete EM algorithm with hard decisions, we obtain the HDD hypothesis testing algorithm from Section IV-C-1).

5) Computational Complexity: These four techniques have roughly the same computational complexity: We first perform (for each of the possible hypotheses) I_H iterations in the synchronized factor graph to compute the required metrics (messages and approximations of APP's). Based on these metrics, a decision with regard to k_θ and k_τ is made (according to (26), (30), (32), or (33)). The remaining $I_c - I_H$ decoding iterations are then devoted to complete the SPA only for the estimated values of k_θ and k_τ. This results in a computational load of $\mathcal{O}((M_\tau M_\theta I_H + (I_c - I_H))C_c N_b)$. In the case of the re-encoding technique, we should add the cost of $M_\tau M_\theta$ encoding operations.

V. Numerical Results

We have carried out computer simulations to evaluate the performance of the code-aided hypothesis testing algorithms. In order to compare practical algorithms with an optimal one, we will first consider a random code with few codewords so that ML sequence detection is possible. We then move on to a more practical scenario. As a point of reference we also included the performance of the perfectly synchronized system. All degradations will be measured against this performance. We note that we do not expect any of the algorithms to approach this bound: Some codes are inherently difficult to synchronize.[7]

Short Random Code: We first consider a random code with 16 codewords, each consisting of $N_s = 30$ 16-PSK symbols ($N_c = 120$ coded bits). The mapper is a conventional one, in a sense that it maps groups of four bits to a single constellation point using a Gray mapping. To have a fair basis of comparison, we will express the performance of the different algorithms in terms of their bit error rate (BER) and not their hypothesis testing error rate: the algorithms from Section IV-A consider the ambiguities as nuisance parameters, and hence do not provide an estimate of the ambiguities. We have implemented the following algorithms:

- the true ML sequence detection algorithm (15) from Section IV-A-1);
- the approximate ML bit detection algorithm from Section IV-A-2) by applying the SPA (from Section III) on the overall factor graph;
- the MS algorithm from Section IV-C-2);
- the discrete EM algorithm from Section IV-C-4);
- the pseudo-ML algorithm from Section IV-C-3).

Clearly, the number of phase ambiguities is $M_\theta = 16$. We have set $M_\tau = 3$. This means the delay is either zero, one, or two symbol durations. The sum–product algorithm is allowed to run for a maximum of 20 iterations. When the messages $\mu_A(a_k)$ do not change (noticeably) from one iteration to the next, $\forall k$, the SP algorithm is halted.

[7]For instance, a regular LDPC code where all check nodes have an even degree in combination with BPSK mapping cannot be used for PAR since both the all-zero and the all-one words are codewords.

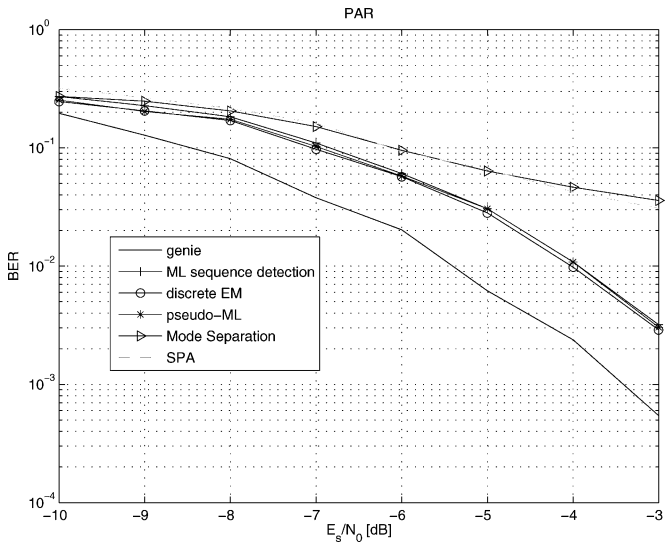


Fig. 7. Short random code. Phase ambiguity resolution: BER performance.

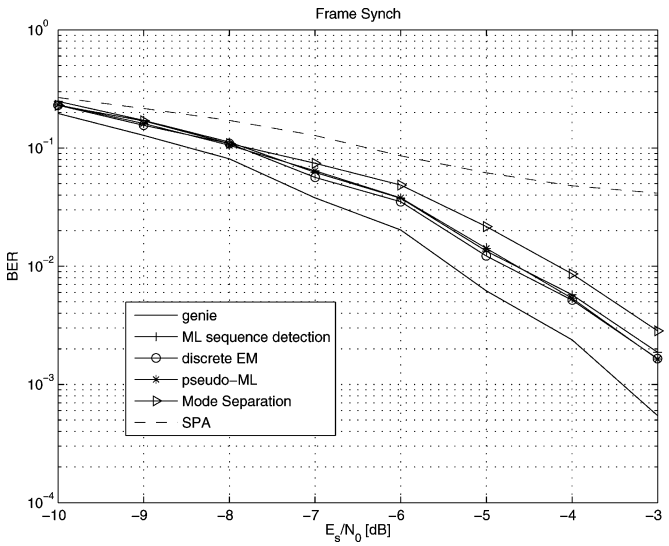


Fig. 8. Short random code. Frame synchronization: BER performance.

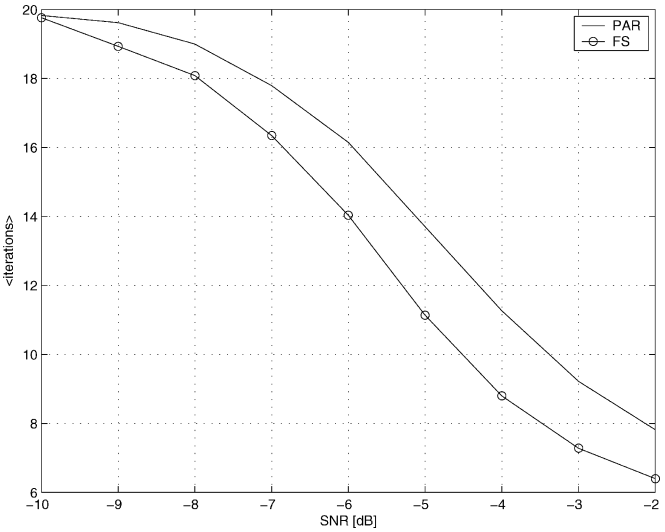


Fig. 9. Sum–product algorithm: Average number of iterations until convergence.

We first consider phase ambiguity resolution (assuming perfect frame synchronization). In Fig. 7, we show BER performance as a function of the SNR (defined as $E_s/N_0 = 1/\left(2\sigma^2\right)$ in decibels). We observe that both the MS and the SP algorithm give rise to severe degradations, especially with increasing SNR. The discrete EM algorithm and the PML technique fare better: they are able to reduce the degradation to around 1 dB, independent of the SNR. Finally, the ML algorithm has almost exactly the same performance of the discrete EM and PML algorithms. Hence, the latter two algorithms have achieved virtually optimal performance at a fraction of the complexity of the ML algorithm. Note that the gap between the ML performance and the performance under perfect synchronization can never be closed.

In Fig. 8 similar results are shown for frame synchronization (assuming perfect phase ambiguity resolution). Again, the SP algorithms results in significant degradations at higher SNR. The MS technique reduces this degradation to around 1 dB for all SNR. As in the previous case, discrete EM, PML, and ML performance are almost equal with an overall (irreducible) degradation of around 0.5 dB.

Although the SP algorithm on the overall factor graph has fairly poor performance for the code we have just considered, it does have one significant benefit: whereas MS and discrete EM always require $M_\theta \times M_\tau$ decoding operations, the SP algorithm does not. Because all possible hypotheses are allowed to exist simultaneously, the SP algorithm can get away with fewer decoding operations. To illustrate this point, we have plotted the average number of iterations versus SNR for the scenarios mentioned above (see Fig. 9). As the SNR increases, the number of iterations decreases. For instance, at an SNR of -2 dB, the SP algorithm requires only around 40% of the computational effort of MS or discrete EM.

Turbo Code: We now investigate a more practical system, consisting of a turbo code with bit-interleaved coded modulation. The constituent convolutional codes of the turbo code are systematic and recursive with rate 1/2, generator polynomials $(21, 37)_8$ and constraint length $\nu = 5$. The turbo code consists of the parallel concatenation of two unpunctured constituent encoders, which yields an overall code rate of 1/3. We consider Gray mapped 4-PSK transmission with a frame length of 360 symbols. We carry out $I_c = 10$ decoding iterations for decoding. We set $M_\theta = 4$ and $M_\tau = 3$. Considering the remarks from Section IV-A-2) and the results for the short random code, application of the SP algorithm will cause severe degradations. Hence, we have only implemented the following algorithms:

- the REEN algorithm and the HDD algorithm from Section IV-C-1);
- the MS algorithm from Section IV-C-2);
- the discrete EM algorithm from Section IV-C-4;
- the PML algorithm from Section IV-C-3).

Simulation results are presented in Figs. 10–12 in the form of BER versus SNR (defined as $E_b/N_0 = 1/\left(2R\log_2 M\sigma^2\right)$ in dB, with R denoting the code rate). In order to reduce the computational complexity to an acceptable level, we perform hypothesis testing after a single ($I_H = 1$) decoding iteration. The

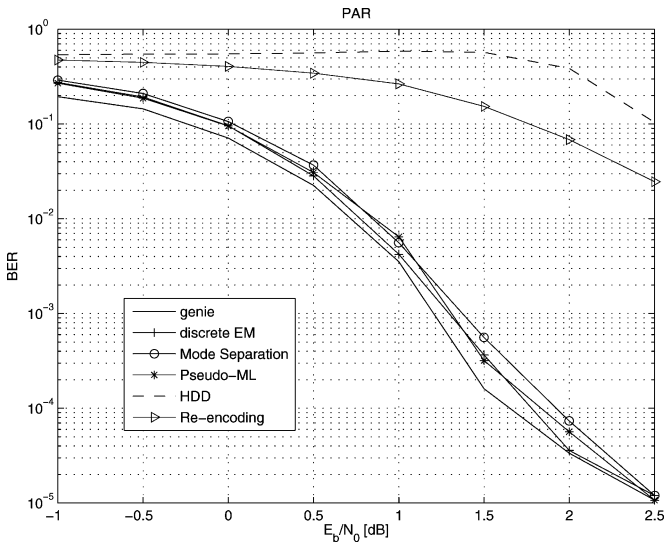


Fig. 10. Turbo code. Phase ambiguity resolution: BER performance.

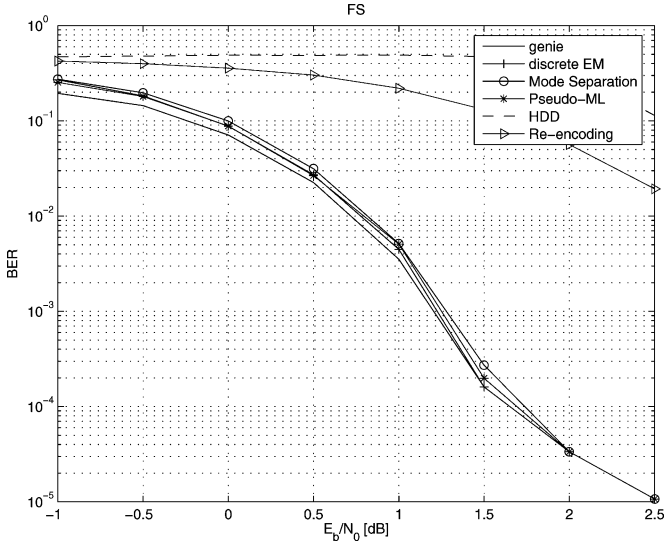


Fig. 11. Turbo code. Frame synchronization: BER performance.

remaining $I_c - I_H = 9$ decoding iterations are devoted to decoding according to the selected hypothesis.

Fig. 10 shows PAR results. The discrete EM, PML, and MS algorithms lead to similar performances: for BER below 10^{-2}, a BER degradation (compared to the case of perfect synchronization) of around less than 0.2 dB is visible. The REEN and the HDD algorithms lead to significant performance degradations. Due to the hard decisions that are made in both algorithms, they can only function at (very) high SNR. Similar conclusions can be drawn for frame synchronization (see Fig. 11). Finally, the BER performance for joint FS and PAR is shown in Fig. 12: a small degradation (around 0.2 dB for BERs below 10^{-2}) is visible for the discrete EM and PML algorithms. The MS technique leads to slightly larger degradations. Both the REEN and HDD technique cause very large degradations, for all considered SNRs. As a point of reference, in Fig. 12 we also show the

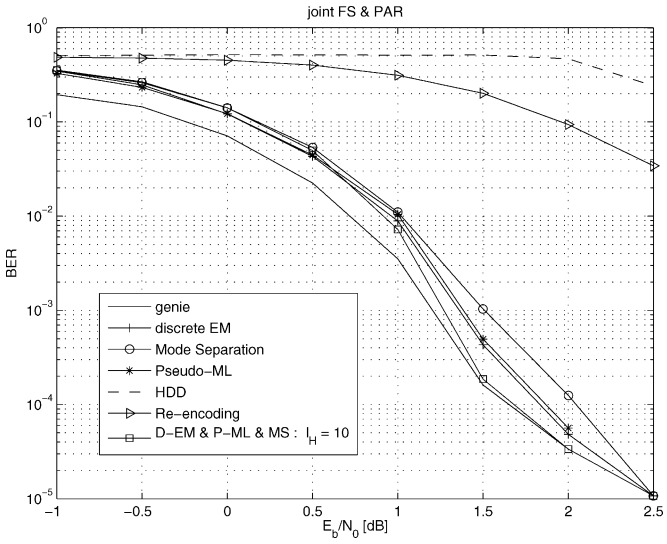


Fig. 12. Turbo code. Joint phase ambiguity resolution and frame synchronization: BER performance.

BER performance of the discrete EM, pseudo-ML and MS algorithms with $I_H = 10$ (i.e., full decoding is performed for each of the hypotheses before a decision is made). The three corresponding curves essentially coincide and are plotted as a single line.

Finally, we note that as far as Frame Error Rate performance is concerned, the discrete EM, PML, and MS algorithms all lead to negligible performance degradations (less than 0.05 dB) even for $I_H = 1$ (results not shown).

VI. Conclusion

In this contribution, we have considered the problem of hypothesis testing in coded digital communications. Such problems occur naturally in many scenarios. We have focused on phase ambiguity resolution and frame synchronization and have compared the performance of different algorithms: the true ML sequence detector, the SPA operating on an overall factor graph of the entire system (including the unknown synchronization parameters), a re-encoding technique, an algorithm based on the EM algorithm, two state-of-the-art algorithms (MS and PML). It turns out that the SPA has the poorest performance. The REEN technique only works reliably at high SNR. The MS algorithm achieves better performance, but is in some cases still far away from the ML performance. From our results, the discrete EM and PML algorithms achieve near ML performance at only a fraction of the computational cost of ML detection. Moreover, taking into account the mathematical elegance of the discrete EM algorithm, it is to be preferred over the other techniques: it can easily be extended to more challenging scenarios and combined with code-aided algorithms that estimate continuous parameters. A comparison of the different techniques is shown in Table I.

We have assumed a binary code. However, the different hypothesis testing algorithms can be applied to nonbinary codes defined over fields or rings with no modification, except for the MS algorithms, which would require some ad hoc modifications since a model of the extrinsic LLRs would be needed.

TABLE I
COMPARISON OF CODE-AIDED HYPOTHESIS TESTING ALGORITHMS WITH $M_\tau M_\theta$ HYPOTHESES. "PILOT" AND "UNCODED" REFER TO THE ABILITY TO EXPLOIT PILOT SYMBOLS (RESPECTIVELY, UNCODED SYMBOLS) IN THE HYPOTHESIS TESTING ALGORITHM. "ESTIMATION" REFERS TO THE POSSIBILITY OF ESTIMATING CONTINUOUS PARAMETERS

	complexity	performance	pilot	uncoded	estimation
optimal	exp. in N_b	optimal	yes	yes	yes
SPA	linear in N_b	poor	yes	yes	yes
Discrete EM	linear in N_b	near-optimal	yes	yes	yes
Mode Sepration	linear in N_b	good	no	no	no
pseudo-ML	linear in N_b	good	yes	yes	no
re-encoding	linear in N_b	good @ high SNR	yes	yes	no
HDD	linear in N_b	poor	yes	yes	no

The SPA algorithm has the lowest computational complexity. Its performance can be improved by removing the assumption of equiprobable hypotheses, nudging the SPA in the right direction. For instance, using a distribution obtained through some other means (such as a list synchronizer) we could initialize $p(k_\theta)$ and $p(k_\tau)$. This remains a topic for future research.

Finally, we mention that other hypothesis testing problems can be tackled in the same fashion. For instance, the EM-based hypothesis testing technique was applied to rate detection for multirate communication in [39].

REFERENCES

[1] C. Berrou, A. Glavieux, and P. Thitimajsima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," in *Proc. IEEE Int. Conf. Communications (ICC)*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Trans. Inf. Theory*, vol. 45, no. 2, pp. 399–431, Mar. 1999.

[3] J. Garcia-Frias and Y. Zhao, "Compression of binary memoryless sources using punctured turbo codes," *IEEE Commun. Lett.*, pp. 394–396, Sep. 2002.

[4] G. Caire, S. Shamai, and S. Verdu, "A new data compression algorithm for sources with memory based on error correcting codes," in *Proc. IEEE Information Theory Workshop (ITW)*, Paris, France, Apr. 2003, pp. 291–295.

[5] A. Glavieux, C. Laot, and J. Labat, "Turbo equalization over a frequency selective channel," in *Proc. Int. Symp. Turbo Codes*, Brest, France, Sep. 1997, pp. 96–102.

[6] X. Li and J. A. Ritcey, "Trellis-coded modulation with bit interleaving and iterative decoding," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 4, pp. 715–724, Apr. 1999.

[7] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, Jul. 1999.

[8] N. Noels, C. Herzet, A. Dejonghe, V. Lottici, H. Steendam, M. Moeneclaey, M. Luise, and L. Vandendorpe, "Turbo-synchronization: An EM algorithm interpretation," presented at the IEEE Int. Conf. Communications (ICC), Anchorage, AL, May 2003.

[9] V. Lottici and M. Luise, "Carrier phase recovery for turbo-coded linear modulations," presented at the Int. Conf. Communications (ICC), New York, Apr. 2002.

[10] A. R. Nayak, J. R. Barry, and S. W. McLaughlin, "Joint timing recovery and turbo equalization for coded partial response channels," *IEEE Trans. Magn.*, vol. 38, no. 5, pp. 2295–2297, Sep. 2002.

[11] B. Mielczarek and A. Svensson, "Timing error recovery in turbo coded systems on AWGN channels," *IEEE Trans. Commun.*, vol. 50, no. 10, pp. 1584–1592, Oct. 2002.

[12] J. L. Massey, "Optimum frame synchronization," *IEEE Trans. Commun.*, vol. COM-20, no. 2, pp. 115–119, Apr. 1972.

[13] E. Cacciamani and C. Wolejsza, "Phase-ambiguity resolution in a four-phase PSK communications system," *IEEE Trans. Commun.*, vol. COM-19, no. 6, pp. 1200–1210, Dec. 1971.

[14] M. K. Howlader and B. D. Woerner, "Decoder-assisted frame synchronization for packet transmission," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 12, pp. 2331–2345, Dec. 2001.

[15] J. Sodha, "Turbo code frame synchronization," *Signal Process. J., Elsevier*, vol. 82, pp. 803–809, 2002.

[16] P. Robertson, "Improving frame synchronization when using convolutional codes," in *Proc. IEEE GLOBECOM*, Houston, TX, 1993, pp. 1606–1611.

[17] T. M. Cassaro and C. N. Georghiades, "Frame synchronization for coded systems over AWGN channel," *IEEE Trans. Commun.*, vol. 52, no. 3, pp. 484–489, Mar. 2004.

[18] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, pp. 284–287, Mar. 1974.

[19] U. Mengali, A. Sandri, and A. Spalvieri, "Phase ambiguity resolution in trellis-coded modulations," *IEEE Trans. Commun.*, vol. 38, no. 12, pp. 2087–2088, Dec. 1990.

[20] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, ser. B, vol. 39, no. 1, pp. 1–38, 1977.

[21] H. Wymeersch and M. Moeneclaey, "Code-aided phase and timing ambiguity resolution for AWGN channels," presented at the IASTED Int. Conf. Acoustics, Signal, Image Processing (SIP03), Honolulu, HI, Aug. 2003.

[22] ——, "Code-aided frame synchronizers for AWGN channels," presented at the Int. Symp. Turbo Codes Related Topics, Brest, France, Sep. 2003.

[23] J. Dauwels, H. Wymeersch, H.-A Loeliger, and M. Moeneclaey, *Phase Estimation and Phase Ambiguity Resolution by Message Passing*, ser. Lecture Notes in Computer Science. New York: Springer-Verlag, 2004.

[24] I. Sutskover, S. Shamai, and J. Ziv, "A novel approach to iterative joint decoding and phase estimation," in *Proc. 3rd Int. Symp. Turbo Codes Related Topics*, Brest, France, Sep. 2003, pp. 87–90.

[25] J. Dauwels and H.-A Loeliger, "Joint decoding and phase estimation: An exercise in factor graphs," presented at the Int. Symp. Information Theory, Yokohama, Japan, Jul. 2003.

[26] ——, "Phase estimation by message passing," in *Proc. IEEE Int. Conf. Communications (ICC)*, Paris, France, Jun. 2004, pp. 523–527.

[27] G. Colavolpe, A. Barbieri, G. Caire, and N. Bonneau, "Bayesian and non-Bayesian methods for joint decoding and detection in the presense of phase noise," presented at the IEEE Int. Symp. Information Theory (ISIT), Chicago, IL, Jul. 2004.

[28] F. Kschischang, B. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[29] H.-A. Loeliger, J. Dauwels, V. M. Koch, and S. Korl, "Signal processing with factor graphs: Examples," presented at the Int. Symp. Control, Communications Signal Processing (ISCCSP), Hammamet, Tunisia, Mar. 2004.

[30] E. Serpedin, P. Ciblat, G. B. Giannakis, and P. Loubaton, "Performance analysis of blind carrier phase estimators for general QAM constellations," *IEEE Trans. Signal Process.*, vol. 49, no. 8, pp. 1816–1823, Aug. 2001.

[31] M. Oerder and H. Meyr, "Digital filter and square timing recovery," *IEEE Trans. Commun.*, vol. 36, pp. 605–612, May 1988.

[32] A. J. Viterbi and A. M. Viterbi, "Nonlinear estimation of PSK-modulated carrier phase with application to burst digital transmission," *IEEE Trans. Inf. Theory*, vol. IT-29, pp. 543–551, Jul. 1983.

[33] G. D. Forney, "Codes on graphs: Normal realizations," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 520–545, Feb. 2001.

[34] J. Campello and D. S. Modha, "Extended bit-filling and LDPC code design," presented at the IEEE GLOBECOM, San Antonio, TX, Nov. 2001.

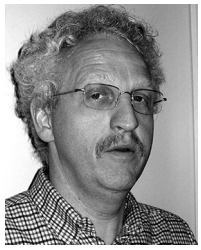
[35] H. Meyr, M. Moeneclaey, and S. A. Fechtel, *Synchronization, Channel Estimation, and Signal Processing*. New York: Wiley, 1997, vol. 2, Digital Communication Receivers.
[36] H. Wymeersch and M. Moeneclaey, "Iterative code-aided ML phase estimation and phase ambiguity resolution," *EURASIP J. Appl. Signal Process. (Special Issue on Turbo Process)*, vol. 2005, no. 6, May 2004.
[37] ——, *Code-Aided ML Joint Delay Estimation and Frame Synchronization*. New York: Springer-Verlag, 2005, vol. 27, Signal Processing for Telecommunications and Multimedia, Multimedia Systems and Applications, ch. 8, pp. 97–110.
[38] H. Wymeersch, F. Simoens, and M. Moeneclaey, "Code-aided joint channel estimation and frame synchronization for MIMO systems," presented at the IEEE Workshop Signal Processing Advances Wireless Communications (SPAWC), Lisbon, Portugal, Jul. 2004.
[39] H. Wymeersch and M. Moeneclaey, "ML-based blind symbol rate detection for multi-rate receivers," presented at the IEEE Int. Conf. Communications (ICC), Seoul, Korea, May 2005.

Henk Wymeersch (S'00–M'05) received the diploma of computer science engineer and the Ph.D. degree in applied sciences from Ghent University, Gent, Belgium, in 2001 and 2005, respectively.

Between 2001 and 2005, he was with the Department of Telecommunications and Information Processing at Ghent University. Currently, he is a Postdoctoral Fellow of the Belgian–American Educational Foundation and is affiliated with the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology (MIT), Cambridge. His research interests include channel coding, synchronization, iterative processing, and sensor networks.

Heidi Steendam (M'01–SM'05) received the M.Sc. degree in electrical engineering and the Ph.D. degree in applied sciences from Ghent University, Gent, Belgium, in 1995 and 2000, respectively.

Since October 2002, she has been a full-time Professor with the Digital Communications (DIGCOM) Research Group, Department of Telecommunications and Information Processing (TELIN), at Ghent University. Her main research interests are in statistical communication theory, carrier and symbol synchronization, bandwidth-efficient modulation and coding, spread-spectrum (multicarrier spread spectrum), and satellite and mobile communication. She is the author of more than 70 scientific papers in international journals and conference proceedings.



Herwig Bruneel was born in Zottegem, Belgium, in 1954. He received the M.S. degree in electrical engineering, the degree of Licentiate in Computer Science, and the Ph.D. degree in computer science all from Ghent University, Gent, Belgium, in 1978, 1979, and 1984, respectively.

From October 2001 to September 2003, he served as the Academic Director for Research Affairs at Ghent University, where he is currently a full Professor in the Faculty of Engineering and head of the Department of Telecommunications and Information Processing. He also leads the SMACS Research Group within this department. His main personal research interests include stochastic modeling and analysis of communication systems, discrete-time queueing theory, and the study of ARQ protocols. He has published more than 250 papers on these subjects and is coauthor (with B. G. Kim) of the book *Discrete-Time Models for Communication Systems Including ATM* (Boston, MA: Kluwer, 1993).

Marc Moeneclaey (M'93–SM'99–F'02) received the Diploma and the Ph.D. degrees, both in electrical engineering, from Ghent University, Gent, Belgium, in 1978 and 1983, respectively.

He is currently a Professor in the Department of Telecommunications and Information Processing at Ghent University. His main research interest are in statistical communication theory, carrier and symbol synchronization, bandwidth-efficient modulation and coding, spread spectrum, and satellite and mobile communication. He is the author of about 250 scientific papers in international journals and conference proceedings. He is coauthor (with H. Meyr (RWTH Aachen) and S. Fechtel (Siemens AG) of the book *Digital Communication Receivers—Synchronization, Channel estimation, and Signal Processing* (New York: Wiley, 1998).