ESA STUDY CONTRACT REPORT

Iterative Joint Detection and Decoding for Communications Under Random Time-varying Carrier Phase

FINAL REPORT

ESA Contract No: 17337/03/NL/LvH	Vol. I/I	

Abstract:

In this report, we describe some iterative joint detection and decoding schemes. We identify two families of algorithms. The first family is based on the Bayesian approach. It makes use of the Bayesian network framework to build a factor graph which takes into ac-count both the code constraint and the statistics of the channel, and build iteratively estimates of the a posteriori probabilities of the transmitted symbols by using the sum-product algorithm.
The second family is based on a non-Bayesian approach, namely, on some iterative low-complexity approximation of the Generalized Likelihood Ratio Test (GLRT) obtained by joining the Expectation-Maximization (EM) algorithm for channel estimation with the sum-product algorithm applied to the code only.
Although both families of algorithms apply to the general case of a coded transmission over a randomly time-varying channel with memory, this work is particularly focused to the case of an Additive White Gaussian Noise channel affected by randomly time-varying carrier phase (phase noise).
The performance of the proposed algorithms is analyzed by computer simulations showing that a time-varying channel phase, with a rate of change typical of the instabilities of the transmitter and receiver oscillators, does not entail significant degradation with respect to the case of a time-invariant (but random and unknown) phase. Our results show that Bayesian algorithms are generally superior to their non-Bayesian counterparts, both in terms of performance and in terms of complexity.

The work described in this report was done under ESA contract. Responsibility for the contents resides in the author or organisation that prepared it.

Names of Authors:
Giulio Colavolpe (Università di Parma)
Giuseppe Caire (Institut Eurecom)

ESA Technical Manager: Alberto Ginesi Div: Payload Systems Dir.: Technical and Operational Support	ESA Budget Heading

1 Introduction

In a digital receiver it is customary to distinguish two main functions: *detection* and *decoding*. By "detection", we mean the function in the receiver that produces symbol-by-symbol observables for the code symbol, by taking into account the channel statistics but not the code constraint. By "decoding", we mean the function in the receiver that consider the sequence of symbol-by-symbol observables produced by the detector and computes a posterior estimate of the code symbols by imposing the code constraint. To fix ideas, consider the standard coherent transmission over a complex baseband equivalent AWGN channel. Let $\mathbf{c} = (c_0, \ldots, c_{K-1})$ be the transmitted codeword, belonging to some code $\mathcal{C} \in \mathbb{C}^K$ over a given complex signal set $\mathcal{X} \in \mathbb{C}$ (e.g., BPSK, QPSK, etc ...). Let $r_k = ac_k + n_k$ denote the discrete-time output of some receiver front-end (sampled matched filter) where a is a deterministic known gain and n_k is distributed as $\mathcal{N}_{\mathbb{C}}(0, N_0)$. Then, an optimal detector produces the sufficient statistics provided by the posterior symbol-by-symbol probabilities

$$\{P_k(x) = P(c_k = x|r_k) \quad : \quad x \in \mathcal{X}\}$$

for all $k = 0, \ldots, K-1$, and the MAP symbol-by-symbol decoder produces the a posteriori probabilities

$$P(c_k = x|\{r_k\}, \mathcal{C}) \doteq \chi(\mathbf{c}) P_k(x) \sum_{c_j \in \mathcal{X}: j \neq k} \prod_{j \neq k} P_j(c_j)$$

where $\chi(\mathbf{c}) = 1\{\mathbf{c} \in \mathcal{C}\}$ is the code constraint function, equal to 1 if $\mathbf{c}$ is a valid codeword (it belongs to the code $\mathcal{C}$) and zero elsewhere, and where we define the following notation (used throughout this report): let $A(x)$ and $B(x)$ be two functions of a variable x. We write $A(x) \doteq B(x)$ to indicate that $A(x) = \kappa B(x) + \zeta$, where $\kappa > 0$ and ζ are constant with respect to the variable of interest x.

Alternatively, instead of a MAP symbol-by-symbol decoder (which minimizes the average symbol error probability) we can consider the joint MAP decoder (erroneously referred to as Maximum Likelihood decoder in most textbooks), that produces the MAP codeword estimate

$$\widehat{\mathbf{c}} = \underset{\mathbf{c} \in \mathcal{C}}{\operatorname{argmax}} \, \chi(\mathbf{c}) \prod_{j=0}^{K-1} P_j(c_j)$$

However, since in this work we deal with iterative decoding of LDPC codes, for which the joint MAP is practically non-feasible because of complexity, we shall restrict to the MAP symbol-by-symbol decoder.

Notice that since the channel is fully known (meaning that a and the noise statistics are perfectly known), the probabilities $P_k(x)$ can be computed exactly. In any case, it is important to remark that if the channel is fully known and the detector produces a sufficient statistics equivalent to the one given above, the separated approach is, obviously, optimal.

The traditional approaches to detection and decoding for randomly time-varying channels consist of: i) using training signals that allow the receiver to estimate the channel and succes-

sively apply techniques for known deterministic channels treating the estimate as if it was exact; ii) using a noncoherent decoding metric insensitive to the channel knowledge. Both approaches are based on the separation between *detection* and *decoding*. However, when the channel is affected by some random parameter unknown a priori to the receiver (in particular, when the channel is randomly time-varying), the separated approach is no longer optimal. In particular, modern coding schemes such as Turbo Codes and LDPC codes allow to work at very low signal-to-noise ratio (SNR), under known deterministic channel. In this condition, the conventional separated approaches fall short to provide satisfactory performance and the presence of a random channel severely limits the overall performance. Therefore, a joint detection and decoding is called for.

The disadvantage of joint detection and decoding is complexity. Hence, iterative techniques where a detector and a decoder are activated in sequence several times and exchange some form of information in order to iteratively approach a global optimum are envisaged. In this report, we describe various techniques for low-complexity iterative joint detection and decoding schemes. We identify two families of algorithms. The first family is based on the Bayesian approach. It makes use of the Bayesian network framework to build a factor graph which takes into account both the code constraint and the statistics of the channel, and build iteratively estimates of the a posteriori probabilities of the transmitted symbols by using the sum-product algorithm. The second family is based on a non-Bayesian approach, namely, on some iterative low-complexity approximation of the Generalized Likelihood Ratio Test (GLRT) obtained by joining the Expectation-Maximization (EM) algorithm for channel estimation with the sum-product algorithm applied to the code only.

Although both families of algorithms apply to the general case of coded transmission over a randomly time-varying channel with memory, this work is particularly focused to the case of an Additive White Gaussian Noise channel affected by randomly time-varying carrier phase (phase noise).

Only preliminary works employing a Bayesian approach may be found in the technical literature. With the exception of [1], in which a general method to embed a parameter description into the code factor graph is described, all these works are related to the noncoherent channel and binary modulations. In [2], a simple noncoherent channel model is considered that tries to capture the phase dynamics. In fact, the unknown carrier phase is considered constant over a block of N symbols and independent from block to block. In [3] an approximate quantized model for the unknown phase is considered and both the code and the receiver are designed based on this model. Finally, in [4], based on the approach in [1], different phase models are considered and approximate solutions are derived. In all these works, the robustness of the proposed schemes is not completely addressed nor is the primary requirement.

A non-Bayesian approach is adopted in [5]–[10]. In [5]–[9] the original concept of *soft-decision-directed* estimation is introduced. The unknown parameters, modeled as deterministic, are estimated by using the EM algorithm [5]–[8] or an ad-hoc procedure [9], and this estimation algorithm is embedded into the iterative decoding process. Finally, in [10] a class of problems is identified for which the optimal, in the sense of the generalized-likelihood ratio test, generation of the symbol a posteriori probabilities can be performed with polynomial complexity. For all these algorithms, when the channel is time-varying, the performance rapidly degrades since the

receiver is not designed exploiting a statistical knowledge of the channel variations.

The performance of the algorithms examined and developed in this report is analyzed by computer simulations showing that a time-varying channel phase, with a rate of change typical of the instabilities of the transmitter and receiver oscillators, does not entail significant degradation with respect to the case of a time-invariant (but random and unknown) phase. Our results show that Bayesian algorithms are generally superior to their non-Bayesian counterparts, both in terms of performance and in terms of complexity.

2 Bayesian algorithms: preview

In recent years, factor graphs (FG) and the sum-product (SP) algorithm [11] have been used to reinterpret a large number of algorithms widely known in the digital communication field, such as the Viterbi algorithm [12], the BCJR algorithm [13], the iterative "turbo" decoding algorithm [14], and the belief propagation algorithm for low-density parity-check (LDPC) codes [15].

In this report, we use these tools to face the problem of iterative detection and decoding of channel codes, described by means of some graphical model, and transmitted over channels with memory. The adopted approach is Bayesian, i.e., the unknown parameters describing the channel, possibly time-varying, are modeled as stochastic processes with a known statistical description. Bayesian methods for joint decoding and channel parameter estimation amount, roughly speaking, to construct a FG modeling the statistical dependency of the transmitted data, of the channel parameters to be estimated, and of the observed signal, and by applying the SP algorithm. The resulting algorithms are naturally iterative, and are well-suited to the decoding of codes such as LDPC and turbo codes, whose decoding algorithms are typically iterative even in the fully coherent setting (all channel parameters known).

Our principal goal is the application to LDPC codes. Since these codes are linear block codes and can be decoded in a fully parallel manner, we are not interested in trellis-based separate detection—see [16]–[21] for the relevant cases of the intersymbol interference (ISI), noncoherent, and fading channels—if we want *simultaneous detection and decoding in a fully parallel manner*. In fact, if separate detection is performed by means of a BCJR algorithm, its serial structure prevents from a fully parallel algorithm implementation. Nevertheless, the proposed algorithms will use soft informations for code symbol probabilities, available in iterative decoding schemes, and for this reason they can be also applied to turbo-TCM codes.

In this report, we describe two methods for building a FG which takes into account the channel model along with the code constraints. In the first one, by means of a factorization of the joint a posteriori probability of the transmitted symbols, we derive a factor graph representing both the code constraints and the channel model but not explicitly the channel parameters. The application of the SP algorithm to this factor graph leads to a scheme for joint detection and decoding. With respect to decoding schemes over a memoryless channel, we have additional factor nodes modeling the channel and performing a marginalization without taking into account the code constraints. This approach is exact in the case of channels with *finite* memory, such as a channel with known ISI, and approximate for channels with *infinite* memory. This latter case includes a noncoherent channel and a flat fading channel, irrespective of the Doppler rate. For

these channels, the factorization is approximate in the sense of a finite dependence assumption only, i.e., only a window of C received samples is considered relevant for the detection of a code symbol. This finite dependence assumption, in general adopted in all practical detection schemes, has also the convenient side-effect of allowing time-varying phase models.

When the SP algorithm is applied to a factor graph, the convergence of its output to the exact marginal probabilities or a good approximation of them is in general determined by the *girth* of the graph.[1] As an example, in designing LDPC codes, cycles of length 4 must be avoided to ensure decoding convergence. The graph derived from the above mentioned factorization has, in general, girth 4. However, we verified by computer simulations that the cycles of length 4, involving factor nodes which model the channel behavior, often do not affect the convergence of the algorithm. If this is not the case, as for transmissions over ISI channels, factor graph transformations can be adopted [22].

By introducing a simple approximation, for equal energy signals, a modified version of the described factor graphs for noncoherent and flat fading channels may be devised. The application of the SP algorithm to these modified graphs has a complexity *linear* in C, allowing a low-complexity receiver implementation for all practical values of C. The possibility of an implementation of the receiver for any value of C is a key point since this parameter has a fundamental role in determining the receiver performance. In fact, the optimal value of C must be chosen as a function of the channel rate of change—the faster the channel, the lower the optimal value of C.

In a second approach, suggested by [1], variable nodes representing the channel parameters are explicitly introduced in the FG. The marginalization of the joint distribution of symbols and channel parameters, a priori performed with respect to the channel parameters in the previous approach, is now performed by the SP algorithm. The problem with this approach is that, while the SP algorithm is well-suited to handle probability mass functions (i.e., discrete random variables), the channel parameters are typically continuous random variables. There are two classical solutions to this problem. One is based on the use of *canonical distributions*, i.e., on pdfs that are efficiently parameterized. Hence, the SP has just to forward the parameters of the distribution. The other solution is based on the quantization of the parameter space. Obviously, this second approach becomes "optimal" (in the sense that it approaches the performance of the exact SP algorithm) for a sufficiently large number of quantization levels, at the expenses of an increased computational complexity. In the case of the use of canonical distributions, we derive an algorithm with complexity linear in the number of transmitted symbols, which is optimal excepting for a truncation of the Fourier series used to represent some messages in the graphs. Simplified low-complexity algorithms are also derived.

We focus on LDPC codes, properly mapped on multilevel constellations and transmitted on the noncoherent satellite channel. In this case, by using the proposed framework we design detection schemes robust to oscillator instabilities (generating phase noise) and time-varying frequency offsets (due to possible Doppler shifts). As an example, these schemes may be employed in the next generation digital video broadcasting satellite transmission systems (DVB-S2) where

[1] A *cycle* is a closed path in the graph and its *length* is defined as the corresponding number of path edges. The length of the smallest cycle is the *girth* of the graph.

these problems of phase and frequency instabilities are of particular relevance due to the use of Q- and V-bands [23].

The proposed algorithms exhibit a high robustness in the presence of a time-varying channel phase. This high robustness allows one to reduce the rate of insertion of the pilot symbols added to the transmitted codeword with the aim of supporting the phase estimation or tracking in more "classical" receivers.

3 System Model

In the considered transmission system, a sequence of M-ary code symbols $\{c_k\}$ is transmitted from epoch 0 to epoch $K-1$. These code symbols are obtained from the encoding of a sequence of information bits[2] and a proper mapping on a multilevel constellation. In addition, to avoid phase ambiguity problems, pilot symbols or differential encoding may be also inserted in the sequence $\{c_k\}$. A sequence of code symbols is denoted in vector notation as

$$\mathbf{c}_{k_1}^{k_2} = (c_{k_1}, c_{k_1+1}, \ldots, c_{k_2}) \,,\; k_2 > k_1 \,. \tag{1}$$

For brevity, the entire sequence is denoted by $\mathbf{c} = \mathbf{c}_0^{K-1}$. This sequence is then modulated and transmitted over a channel which is modeled as a noiseless filter (possibly stochastic) rendered noisy by additive white Gaussian noise (AWGN) with two-sided power spectral density $N_0/2$.

At the receiver side, by means of a discretization process, the received signal $r(t)$ is converted into a time-discrete sequence $\mathbf{r}$ [24]. We assume that a single sample r_k is used for each code symbol, which is practically sufficient in many cases. In the case of oversampling, the extension is straightforward—the observation r_k must be considered as a vector whose dimensionality is given by the number of samples per code symbol. With a notation similar to that used for code symbols, the sequence of observations $\{r_k\}$ is denoted by $\mathbf{r} = \mathbf{r}_0^{K-1}$.

We also assume that the channel is *causal*, that is the observation sequence $\mathbf{r}_0^k$ up to epoch k depends on the code sequence up to epoch k only. This condition may be formulated in terms of the following statistical dependence:

$$p(\mathbf{r}_0^k|\mathbf{c}) = p(\mathbf{r}_0^k|\mathbf{c}_0^k) \,. \tag{2}$$

This condition characterizes the very general channel model which is assumed in this report corresponding to the following expression of the observation r_k:

$$r_k = g(c_{k-L}^k, \boldsymbol{\theta}_0^k) + n_k \tag{3}$$

where $g(.)$ is a suitable function, $\boldsymbol{\theta}_k = (\theta_k^{(1)}, \ldots, \theta_k^{(\beta)})$ is a vector of β channel parameters, possibly unknown and stochastic, corresponding to epoch k, L is a suitable integer parameter, and n_k is a discrete-time complex AWGN noise sample with each component of variance σ^2. In this model the cases of a noncoherent channel, a flat or a frequency-selective fading channel,

[2] In the numerical results, LDPC codes are considered.

and a channel with known and time-invariant ISI are included, by considering both linear or continuous phase modulations (CPM).

As an example, for a noncoherent channel, which is characterized by an unknown stochastic and possibly time-varying phase θ_k, considering a linear modulation at the transmitter side and assuming that one sample per code symbol is adequate (as in the absence of strong phase variations), if transmit and receive filters are such that there is absence of ISI, we have

$$r_k = c_k e^{j\theta_k} + n_k \,. \tag{4}$$

Hence, in this case $L = 0$, $\beta = 1$. This noncoherent channel will be considered in detail in the numerical results.

4 Factor Graphs and Sum-Product Algorithm

A FG is a *bipartite* graph which expresses the factorization of a complicated *global* function of many variables, usually a joint probability density function (pdf) or a joint probability mass function (pmf), as a product of *local* functions, each of which depends on a subset of the variables. The SP algorithm works on a factor graph and computes—either exactly or approximately—the marginal functions derived from the global function. A wide variety of algorithms developed in artificial intelligence, signal processing and digital communications can be derived as specific instances of the SP algorithm [11].

Suppose that the global function $g(x_1, x_2, \ldots, x_n)$ factors into a product of several local functions, each having some subset of $\{x_1, x_2, \ldots, x_n\}$ as arguments:

$$g(x_1, x_2, \ldots, x_n) = \prod_{j \in J} f_j(X_j)$$

where J is a discrete index set, X_j is a subset of $\{x_1, x_2, \ldots, x_n\}$, and $f_j(X_j)$ is a function having the elements of X_j as arguments. A FG is a bipartite graph that expresses the structure of this factorization. It has a *variable node* for each variable x_i, a *factor node* for each local function f_j, and an *edge* connecting variable node x_i to function node f_j if and only if x_i is an argument of f_j.

Example. Let g be a function of five variables, and suppose that it can be expressed as a product

$$g(x_1, x_2, x_3, x_4, x_5) = f_A(x_1) f_B(x_2) f_C(x_1, x_2, x_3) f_D(x_3, x_4) f_E(x_3, x_5) \,.$$

In this case, $J = \{A, B, C, D, E\}$, $X_A = \{x_1\}$, $X_B = \{x_2\}$, $X_C = \{x_1, x_2, x_3\}$, $X_D = \{x_3, x_4\}$, and $X_E = \{x_3, x_5\}$. The corresponding factor graph is shown in Fig. 1. □

Let us assume that variables $\{x_i\}$ represent discrete random variables and we want to perform the marginalization of their joint pmf. When the corresponding FG has cycles, the application of the SP algorithm to this graph leads to a suboptimal iterative detection process which often converges to a very good approximation of the marginal pmfs. The exchange of messages, representing the marginal pmfs of variables x_i and hence assuming M values, fulfills the following

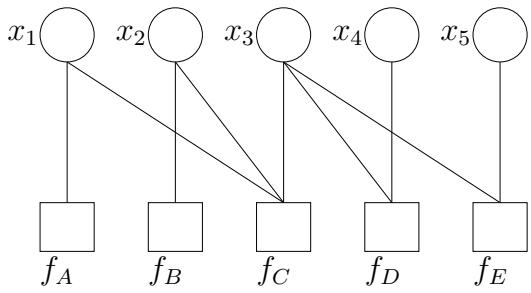

Figure 1: An example of factor graph.

rule. Denoting by $\mu_{x\to f}(x)$ a message sent from variable node x to function node f, by $\mu_{f\to x}(x)$ a message in the opposite direction, and by $n(v)$ the set of neighbors of a given node v, the message computations performed at variable and factor nodes are, respectively [11]

$$\mu_{x\to f}(x) = \prod_{h\in n(x)\setminus\{f\}} \mu_{h\to x}(x) \tag{5}$$

$$\mu_{f\to x}(x) = \sum_{\sim\{x\}} \left[f(X) \prod_{y\in n(f)\setminus\{x\}} \mu_{y\to f}(y) \right] \tag{6}$$

where $\sum_{\sim\{x\}}$ is a *summary* operator, i.e., a sum over all variables excluding x.[3] It may be observed that the message sent to an edge does not depend on the message previously received on the same edge, i.e., only *extrinsic information* is exchanged.

The messages in eqn. (5) and (6) may be also computed in the logarithmic domain. Defining $\overline{\mu}_{f\to x}(x) = \ln \mu_{f\to x}(x)$ and $\overline{\mu}_{x\to f}(x) = \ln \mu_{x\to f}(x)$, the message computations performed at variable and factor nodes become

$$\overline{\mu}_{x\to f}(x) = \sum_{h\in n(x)\setminus\{f\}} \overline{\mu}_{h\to x}(x) \tag{7}$$

$$\overline{\mu}_{f\to x}(x) = \ln \left\{ \sum_{\sim\{x\}} \exp \left[\ln f(X) + \sum_{y\in n(f)\setminus\{x\}} \overline{\mu}_{y\to f}(y) \right] \right\} . \tag{8}$$

The implementation of this latter rule does not require multiplications but only additions and the evaluation of a nonlinear function. In fact, by using the Jacobian logarithm [25]–[27], it is well known that, if x_1 and x_2 are real numbers

$$\ln(e^{x_1} + e^{x_2}) = \max(x_1, x_2) + \ln(1 + e^{-|x_2 - x_1|}) \tag{9}$$

[3]When variables x represent continuous random variables, the global function represents their joint pdf and messages on the graph edges represent their marginal pdfs. In this case, the summary operator represents an integral over all variables excluding x.

and $\ln(1+e^{-|x_2-x_1|})$ is the nonlinear function whose evaluation requires a look-up table. In our case, the evaluation of $\ln(e^{x_1}+e^{x_2}+\cdots+e^{x_n})$ can be done recursively [27]. A further simplification of the updating rule (8) may be obtained by using the so-called *max-log* approximation [27]:

$$\overline{\mu}_{f\to x}(x) = \max_{\sim\{x\}} \left[\ln f(X) + \sum_{y\in n(f)\setminus\{x\}} \overline{\mu}_{y\to f}(y) \right] . \tag{10}$$

The most demanding computation is that performed at factor nodes. For this reason, in order to compare the complexity of the described detection algorithms, we will define a *cost per code symbol per iteration* $\mathcal{C}_T$. This cost will only take into account the number of evaluations of the above mentioned nonlinear function that factor nodes, describing the channel behavior, have to compute to update the messages related to code symbols. Since we will consider factor graphs having different structures, the definition of the cost $\mathcal{C}_T$ will be better specified for each algorithm. Note that it is possible to use as a complexity measure the cost per code symbol per iteration since the complexity of all the described algorithms is linear with respect to the number of transmitted symbols.

A *message-passing schedule* in a factor graph is the specification of the order in which messages are updated. In general, the so-called *flooding schedule* is adopted [28]: in each iteration, all variable nodes and subsequently all factor nodes, pass new messages to their neighbors. As can be easily understood, this schedule is well suited for a fully parallel implementation of the detector/decoder. Other schedules may be adopted, serial or mixed serial-parallel, according to the specific implementation requirements.

5 First Approach: A Priori Average over Channel Parameters

The application of the SP algorithm [11] to a factor graph representing the joint a posteriori probability (APP) of the transmitted code sequence $\mathbf{c}$ conditioned to a given observation sequence $\mathbf{r}$, allows the exact or approximate computation of the single marginal APPs $P(c_k|\mathbf{r})$ [11]. Therefore, this algorithm may be used to implement the MAP symbol detection algorithm. This code sequence APP may be expressed as

$$P(\mathbf{c}|\mathbf{r}) \doteq P(\mathbf{c})p(\mathbf{r}|\mathbf{c}) = P(\mathbf{c}) \prod_{k=0}^{K-1} p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}_0^k) \doteq \chi(\mathbf{c}) \prod_{k=0}^{K-1} p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}_0^k) \tag{11}$$

having used the causality condition (2), assumed that the a priori distribution of the transmitted codewords is uniform, and denoted by $\chi(\mathbf{c})$ the code characteristic function ($\chi(\mathbf{c}) = 1$ for all codewords in the codebook, and zero elsewhere).

If the pdf $p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}_0^k)$, which appears in (11), satisfies the condition

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}_0^k) = p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}_{k-C}^k) \tag{12}$$

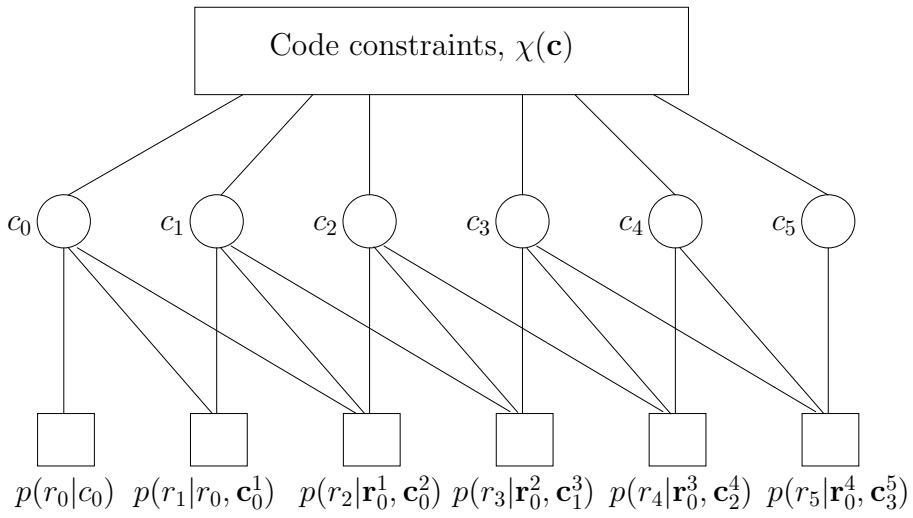

Figure 2: Overall factor graph for $C = 2$.

where C is a suitable parameter, the channel has *finite memory* [29,30]. For this reason, parameter C will be nicknamed *finite memory parameter*. Substituting (12) into (11), the code sequence APP may be expressed as

$$P(\mathbf{c}|\mathbf{r}) \doteq \chi(\mathbf{c}) \prod_{k=0}^{K-1} p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}_{k-C}^k) \tag{13}$$

The corresponding factor graph is shown in Fig. 2 for $C = 2$, and represents both the code constraints (described by $\chi(\mathbf{c})$) and the channel behavior. With respect to SP-based decoding schemes for linear block codes (e.g., LDPC codes) over a memoryless channel, *additional factor nodes* must be added at the bottom of the graph, as shown in Fig. 2. These additional factor nodes perform a marginalization, based on the channel model, without taking into account the code constraints. The complexity of this marginalization is, in general, exponential in C.

The finite memory condition (12) is exactly verified in the case of channels with known ISI. In fact, in this case it is

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}_0^k) = p(r_k|\mathbf{c}_{k-L}^k) \tag{14}$$

where L is the length of the discrete-time channel impulse response. This case is not further considered in this report since analyzed in depth in [22]. If the finite memory condition (12) is not verified in an exact sense as for a noncoherent or a fading channel (channels with *infinite* memory), a factor graph may be devised but the complexity of the message computation at the factor node $p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}_0^k)$ modeling the channel grows exponentially with k and thus becomes impractical. For this reason, considering the channel model (3), an approximation is introduced assuming that r_k depends on the R most recent observations and the most recent $C = R + L$

code symbols only. This *finite dependence property* may be expressed as

$$p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}_0^k) \simeq p(r_k|\mathbf{r}_{k-R}^{k-1}, \mathbf{c}_{k-C}^k)\,. \tag{15}$$

This property, in general adopted in all practical detection schemes, is intuitive in the case of time-varying channels. In fact, in this case the conditional observations are asymptotically independent for an increasing index difference. The resulting (approximate) code sequence APP becomes

$$P(\mathbf{c}|\mathbf{r}) \simeq \chi(\mathbf{c}) \prod_{k=0}^{K-1} p(r_k|\mathbf{r}_{k-R}^{k-1}, \mathbf{c}_{k-C}^k) \tag{16}$$

Considering the channel model (3), the pdf $p(r_k|\mathbf{r}_{k-R}^{k-1}, \mathbf{c}_{k-C}^k)$ which appears in eqn. (16) may be computed in a closed form as

$$p(r_k|\mathbf{r}_{k-R}^{k-1}, \mathbf{c}_{k-C}^k) = \frac{E_{\boldsymbol{\theta}_0^k}\{p(\mathbf{r}_{k-R}^k|\mathbf{c}_{k-C}^k, \boldsymbol{\theta}_0^k)\}}{E_{\boldsymbol{\theta}_0^{k-1}}\{p(\mathbf{r}_{k-R}^{k-1}|\mathbf{c}_{k-C}^{k-1}, \boldsymbol{\theta}_0^{k-1})\}}\,. \tag{17}$$

The quality of the convergence of the SP algorithm to the exact marginal probabilities is in general determined by the *girth* of the graph. As an example, in designing LDPC codes, cycles of length 4 must be avoided to ensure decoding convergence. The graph derived from the proposed factorization has, in general, girth 4. However, we verified by computer simulations that these length-4 cycles involving two factor nodes which model the channel behavior often do not affect the convergence of the algorithm.[4] If this is not the case, as for transmissions over ISI channels, factor graph transformations can be adopted [22].

For the SP algorithm working on the described factor graphs, the most demanding computation is that performed at factor nodes modeling the channel. In fact, the marginalization performed by these nodes has in general a complexity which increases exponentially with C. This complexity may be reduced following a technique similar to reduced-state sequence estimation (RSSE) used for maximum likelihood sequence detection. In fact, by choosing an integer $Q < C$, we may compute the marginalization at factor nodes on the Q symbols with lowest reliabilities while the $C - Q$ symbols with highest reliabilities are hard-quantized on the basis of the messages on the graph. In this way, the complexity becomes exponential in Q.

For equal energy signals, a modified version of the described factor graphs for noncoherent and flat fading channels may be devised. In fact, it can be shown that for fading channels the function $p(r_k|\mathbf{r}_0^{k-1}, \mathbf{c}_{k-C}^k)$ exactly factors into the product of C functions of two code symbols. For noncoherent channels this factorization is not exact but involves a simple approximation. The SP algorithm on these modified graphs has a complexity *linear* in C, allowing a low-complexity receiver implementation for all practical values of C. For a noncoherent channel details are given in Section 5.1.

[4] A length-4 cycle involving a factor node modeling the channel and a factor node representing a code constraint may be easily removed by interleaving the code symbols before transmission.

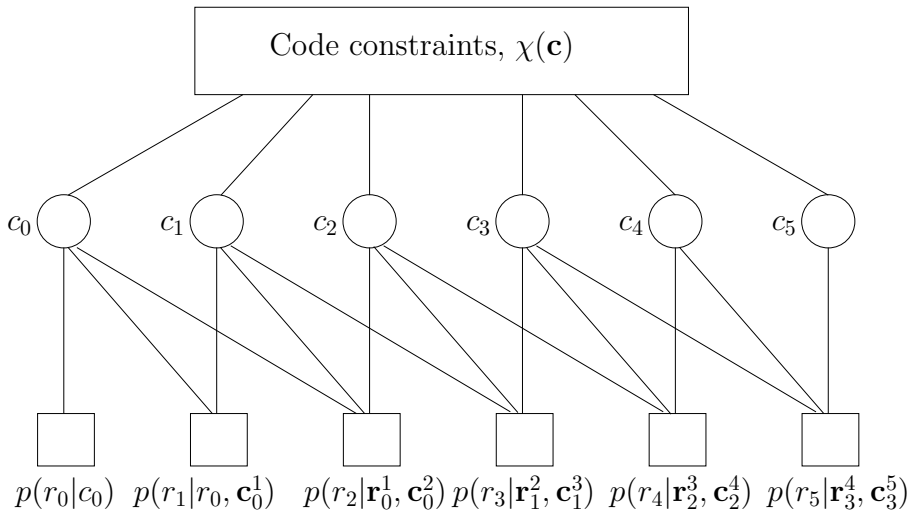

Figure 3: Overall factor graph for the noncoherent channel and $C = 2$.

5.1 Noncoherent channel

In this section, we consider the application of the detection approach described in the previous Section to the case of a noncoherent channel. The system model is given by eqn. (4).

First algorithm. For the time being, we model the channel phase as a time-invariant random variable θ with uniform distribution in $[0, 2\pi)$. However, the finite dependence property (15) will lead to a detection algorithm that can be used for slowly-varying channels also. In this case, it is $R = C$ and the pdf $p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k})$ which appears in eqn. (16) may be expressed, based on eqn. (17), as

$$p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k}) \doteq \frac{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=0}^{C} r_{k-i}c_{k-i}^*\right|\right)}{\mathrm{I}_0\left(\frac{1}{\sigma^2}\left|\sum_{i=1}^{C} r_{k-i}c_{k-i}^*\right|\right)} e^{-\frac{|c_k|^2}{2\sigma^2}} \tag{18}$$

where $\mathrm{I}_0(x)$ is the zeroth order modified Bessel function of the first kind. The corresponding factor graph is shown in Fig. 3 for $C = 2$. Based on the SP algorithm, these factor nodes, perform a marginalization whose computational burden grows exponentially with C. The cost per code symbol per iteration is $\mathcal{C}_T = (C+1)M^{C+1}$, being $(C+1)$ the number of edges entering a variable node representing a code symbol, M the number of values to be computed for each message, and M^C the number of evaluations of the nonlinear function in (9) for each summary operation performed by a factor node representing the channel. This cost can be reduced to $\mathcal{C}_T = (C+1)M^{Q+1}$ if the described reduced complexity technique is adopted. In the following, this algorithm will be referred to as *exponential-complexity algorithm.*

Second algorithm. We now introduce some approximations with the aim of further factorizing the probability density function $p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k})$. As a first step, approximating $\mathrm{I}_0(x) \simeq e^x$,

which holds for large values of the argument $x \in \mathbb{R}^+$, we have

$$p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k}) \simeq \exp\left\{\frac{1}{\sigma^2}\left[\left|\sum_{i=0}^{C} r_{k-i}c_{k-i}^*\right| - \left|\sum_{i=1}^{C} r_{k-i}c_{k-i}^*\right| - \frac{|c_k|^2}{2}\right]\right\}. \tag{19}$$

For a complex number z, we may express $|z| = \mathrm{Re}\{ze^{-j\mathrm{arg}\{z\}}\}$, and denoting

$$\begin{aligned}\hat{\theta}_k^{(C+1)} &= \arg\left\{\sum_{i=0}^{C} r_{k-i}c_{k-i}^*\right\} \\ \hat{\theta}_{k-1}^{(C)} &= \arg\left\{\sum_{i=1}^{C} r_{k-i}c_{k-i}^*\right\}\end{aligned} \tag{20}$$

with the further approximation $\hat{\theta}_k^{(C+1)} \simeq \hat{\theta}_{k-1}^{(C)}$, we have

$$\begin{aligned}p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k}) &\simeq \exp\left\{\frac{1}{\sigma^2}\mathrm{Re}\left[\sum_{i=0}^{C} r_{k-i}c_{k-i}^* e^{-j\hat{\theta}_{k-1}^{(C)}} - \sum_{i=1}^{C} r_{k-i}c_{k-i}^* e^{-j\hat{\theta}_{k-1}^{(C)}} - \frac{|c_k|^2}{2}\right]\right\} \\ &= \exp\left\{\frac{1}{\sigma^2}\mathrm{Re}\left[r_k c_k^* e^{-j\hat{\theta}_{k-1}^{(C)}} - \frac{|c_k|^2}{2}\right]\right\}.\end{aligned} \tag{21}$$

Since

$$e^{j\hat{\theta}_{k-1}^{(C)}} = \frac{\sum_{i=1}^{C} r_{k-i}c_{k-i}^*}{\left|\sum_{i=1}^{C} r_{k-i}c_{k-i}^*\right|} \tag{22}$$

we may express

$$p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k}) \simeq \exp\left\{\frac{1}{\sigma^2}\mathrm{Re}\left[r_k c_k^* \frac{\sum_{i=1}^{C} r_{k-i}^* c_{k-i}}{\left|\sum_{i=1}^{C} r_{k-i}c_{k-i}^*\right|} - \frac{|c_k|^2}{2}\right]\right\}. \tag{23}$$

We now consider phase-shift keying (PSK) signals. In this case, $|c_k| = 1$ and we also approximate $\left|\sum_{i=1}^{C} r_{k-i}c_{k-i}^*\right| \simeq C$ which is true for the transmitted sequence at a sufficiently high signal-to-noise ratio (SNR). Therefore, eqn. (23) becomes

$$\begin{aligned}p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k}) &\simeq \exp\left\{\frac{1}{\sigma^2 C}\mathrm{Re}\left[r_k c_k^* \sum_{i=1}^{C} r_{k-i}^* c_{k-i}\right]\right\} \\ &= \prod_{i=1}^{C} \exp\left\{\frac{1}{\sigma^2 C}\mathrm{Re}\left[r_k c_k^* r_{k-i}^* c_{k-i}\right]\right\} \\ &= \prod_{i=1}^{C} g_{i,k}(c_{k-i}, c_k)\end{aligned} \tag{24}$$

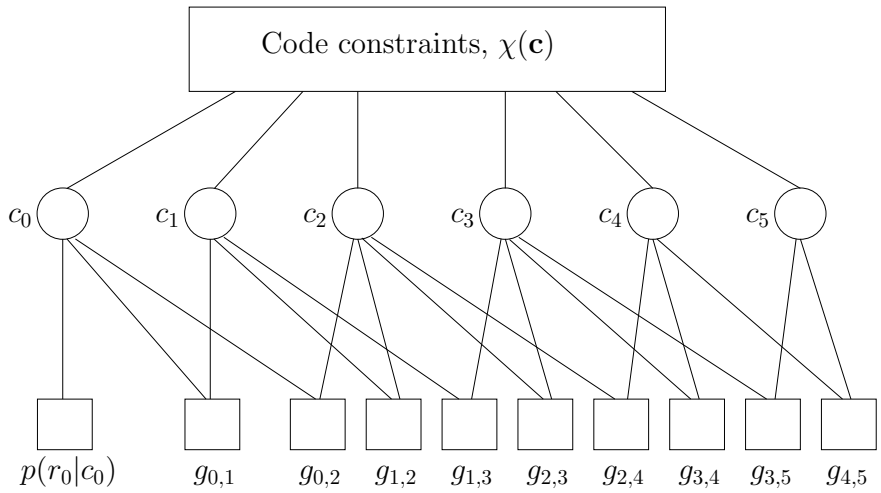

Figure 4: Simplified overall factor graph for PSK signals and $C = 2$.

having defined

$$g_{k-i,k}(c_{k-i}, c_k) = \exp\left\{\frac{1}{\sigma^2 C}\mathrm{Re}\left[r_k c_k^* r_{k-i}^* c_{k-i}\right]\right\}. \tag{25}$$

This further factorization as an immediate impact on the graph structure. In fact, each factor node can be decomposed in C simpler degree-2 factor nodes. As an example, for $C = 2$, the factor graph in Fig. 3 becomes that in Fig. 4 (for brevity, the argument of functions $g_{i,k}(c_{k-i}, c_k)$ are omitted). Hence, for increasing values of C, the number of factor nodes increases linearly but the computational burden at each factor node remains the same. The cost per code symbol per iteration is now $\mathcal{C}_T = 2M^2C$, being $2C$ the number of edges entering a variable node representing a code symbol, M the number of values to be computed for each message, and M the number of evaluations of the nonlinear function in (9) for each summary operation performed by a factor node representing the channel. Note that in this modified factor graph there are no cycles of length 4. In the following, this algorithm will be referred to as *linear-complexity algorithm.*

Third algorithm. For a general time-varying phase process θ_k, assumed stationary, zero-mean and described by a given autocorrelation sequence of the phasor process $h_k = e^{j\theta_k}$, denoted by $R_h(n) = E\{e^{j\theta_{n+k}} e^{-j\theta_k}\}$, the approximate linear predictive approach described in [31] for Viterbi-based maximum-likelihood sequence estimation receivers may be adopted. In this case, omitting irrelevant constant terms, the pdf $p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k})$ may be approximated as (omitting irrelevant constant terms)

$$p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k}) \simeq \exp\left\{-\frac{1}{\sigma_e^2}\left|r_k - c_k \frac{\sum_{i=1}^{C} p_i \frac{r_{k-i}}{c_{k-i}}}{|\sum_{i=1}^{C} p_i \frac{r_{k-i}}{c_{k-i}}|}\right|^2\right\} \tag{26}$$

where, in this case, C assumes the meaning of *prediction order*, $\{p_i\}_{i=1}^{C}$ are the prediction coefficients and σ_e^2 is the mean square prediction error. The prediction coefficients $\{p_i\}_{i=1}^{C}$ in (26) can be computed by solving a Yule-Walker linear system $\mathbf{R}\,\mathbf{p} = \mathbf{b}$, where $\mathbf{R}$ is a square $C \times C$ matrix whose elements have the following expression

$$[\mathbf{R}]_{\ell,m} = \begin{cases} R_h(|\ell - m|) & \text{if} \quad \ell \neq m \\ 1 + \frac{2\sigma^2}{|c_{k-\ell}|^2} & \text{if} \quad \ell = m \end{cases} \tag{27}$$

$\mathbf{p} \stackrel{\Delta}{=} [p_1 \cdots p_C]^T$ is the unknown vector, and $\mathbf{b} = [R_\theta(1), R_\theta(2), \cdots, R_\theta(C)]^T$. The mean square prediction error may be expressed as [31]

$$\sigma_e^2 = 1 + \frac{2\sigma^2}{|c_k|^2} - \sum_{i=1}^{C} p_i R_h(i)\,. \tag{28}$$

For PSK signals, the prediction coefficients and the mean square prediction error become independent of the considered sequence. In addition, approximating $|\sum_{i=1}^{C} p_i \frac{r_{k-i}}{c_{k-i}}| \simeq |\sum_{i=1}^{C} p_i|$, and taking into account that $|c_k| = 1$, we have

$$\begin{aligned} p(r_k|\mathbf{r}_{k-C}^{k-1}, \mathbf{c}_{k-C}^{k}) &\doteq \exp\left\{ \frac{2}{\sigma_e^2 |\sum_{i=1}^{C} p_i|} \mathrm{Re}\left[r_k c_k^* \sum_{i=1}^{C} p_i \frac{r_{k-i}^*}{c_{k-i}^*} \right] \right\} \\ &= \prod_{i=1}^{C} \exp\left\{ \frac{2}{\sigma_e^2 |\sum_{i=1}^{C} p_i|} \mathrm{Re}\left[r_k c_k^* p_i \frac{r_{k-i}^*}{c_{k-i}^*} \right] \right\} \\ &= \prod_{i=1}^{C} \exp\left\{ \frac{2}{\sigma_e^2 |\sum_{i=1}^{C} p_i|} \mathrm{Re}\left[p_i r_k c_k^* r_{k-i}^* c_{k-i} \right] \right\} \end{aligned} \tag{29}$$

which leads to a graph with structure as in Fig. 4 and a complexity of the SP algorithm linear in C. Even in this case, in fact, the cost per code symbol per iteration is $\mathcal{C}_T = 2M^2C$. Note that, when the phase is time-invariant, it is $p_i = 1/C$ and $\sigma_e^2 = 2\sigma^2$, and (29) reduces to (24). In the following, this algorithm will be referred to as *prediction algorithm.*

6 Second Approach: Numerical Average over Channel Parameters

In the previous Section, we built the factor graph of the joint APP $P(\mathbf{c}|\mathbf{r})$. This APP was obtained by an average over the channel parameters (see (17)). In the case of the noncoherent channel, the result of this average can be given in a closed form if the channel phase is time-invariant only. For a time-varying phase, we resorted to an approximate solution based on linear prediction.

Another approach to solve the problem may be adopted [1, 3, 4]. By considering the joint distribution of symbols and unknown parameters $P(\mathbf{c}|\boldsymbol{\theta}, \mathbf{r})p(\boldsymbol{\theta}|\mathbf{r})$ and the corresponding factor

graph, the application of the SP algorithm to this graph will allow us the computation of the desired marginal APPs of code symbols by also marginalizing out over channel parameters. Therefore, the average over the channel parameters is performed by the SP algorithm. A major difference with respect to the previous approach is that we now have in the factor graph explicit variable nodes representing the channel parameters. In addition, since the channel parameters are often continuous random variables, messages sent or received from nodes representing the channel parameters are arbitrary pdfs. Hence, the implementation complexity of the exact SP algorithm becomes impractical. Possible solutions for this problem can be the following.

1. Use of canonical distributions. We represent the pdfs, which are the messages sent or received from nodes representing the channel parameters, with given canonical pdfs, described by some parameters. This representation can be exact or, more often, involve some approximate assumptions. As an example, we could assume that these messages are Gaussian pdfs which can be completely specified by their mean and variance, as in the case of Kalman filtering [11].[5] Hence, the computation of these messages simply reduces to the computation of the describing parameters.

2. Quantization of the channel parameters. If we quantize the channel parameters, we may consider the factor graph representing the joint APP $P(\mathbf{c}, \boldsymbol{\theta}|\mathbf{r})$. The application of the SP algorithm to this factor graph allows us to compute the desired marginal APPs $P(c_k|\mathbf{r})$ and the collateral ones $P(\theta_k|\mathbf{r})$. From a point of view of the involved approximations, if the approach described in Section 5 is approximated for the finite dependence only, in this case we have the quantization of in general continuous channel parameters and their statistics. Obviously, this second approach becomes "optimal" (in the sense that it approaches the performance of the exact SP algorithm) for a sufficiently large number of quantization levels, at the expenses of an increased computational complexity.

6.1 Noncoherent channel: canonical distributions

First algorithm. Modeling the channel phase as a time-invariant random variable θ with uniform distribution in $[0, 2\pi)$, we have

$$\begin{aligned} P(\mathbf{c}|\theta, \mathbf{r})p(\theta|\mathbf{r}) &\doteq p(\mathbf{r}|\mathbf{c}, \theta)\chi(\mathbf{c})p(\theta) = \frac{1}{2\pi}\chi(\mathbf{c}) \prod_{k=0}^{K-1} p(r_k|c_k, \theta) \\ &\doteq \chi(\mathbf{c}) \prod_{k=0}^{K-1} f_k(c_k, \theta) \end{aligned} \tag{30}$$

where

$$f_k(c_k, \theta) = p(r_k|c_k, \theta) \doteq \exp\left\{-\frac{1}{2\sigma^2}|r_k - c_k e^{j\theta}|^2\right\} . \tag{31}$$

The corresponding factor graph is shown in Fig. 5.

[5] This choice must be determined from the observation of the true messages propagating along the graph edges.

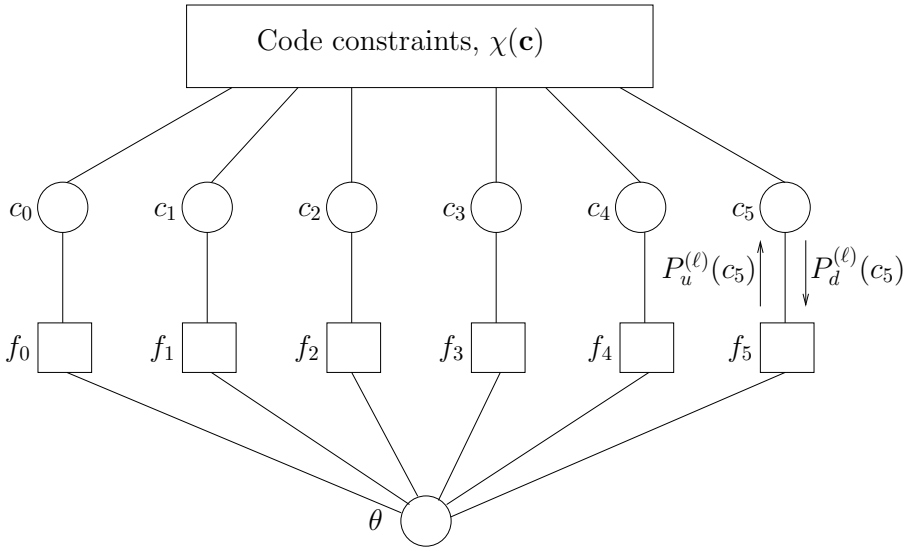

Figure 5: Factor graph corresponding to eqn. (30).

Let us assume that the SP algorithm works in the natural domain. Hence, messages corresponding to variable nodes assume the meaning of pmfs or pdfs, if variables represent discrete or continuous random variables, respectively. Denoting by $P_d^{(n)}(c_k)$ the message sent at the n-th iteration from variable node c_k to factor node f_k, and by $P_u^{(n)}(c_k)$ the message in the opposite direction, following the rules of the SP algorithm (5) and (6) we may write

$$P_u^{(n)}(c_k) = \int_0^{2\pi} \left[\prod_{i \neq k} \left(\sum_{c_i} f_i(c_i, \theta) P_d^{(n)}(c_i) \right) \right] f_k(c_k, \theta)\, d\theta\,. \tag{32}$$

This integral can be computed in a closed form as

$$\begin{aligned} P_u^{(n)}(c_k) &= \sum_{c_0} \cdots \sum_{c_{k-1}} \sum_{c_{k+1}} \cdots \sum_{c_{K-1}} P_d^{(n)}(c_0) \ldots P_d^{(n)}(c_{k-1}) P_d^{(n)}(c_{k+1}) \ldots P_d^{(n)}(c_{K-1}) \cdot \\ &\quad \cdot \int_0^{2\pi} \prod_i f_i(c_i, \theta)\, d\theta \\ &= \sum_{c_0} \cdots \sum_{c_{k-1}} \sum_{c_{k+1}} \cdots \sum_{c_{K-1}} P_d^{(n)}(c_0) \ldots P_d^{(n)}(c_{k-1}) P_d^{(n)}(c_{k+1}) \ldots P_d^{(n)}(c_{K-1}) \cdot \\ &\quad \cdot \mathrm{I}_0 \left(\frac{1}{\sigma^2} \left| \sum_i r_i c_i^* \right| \right) \prod_i \exp\left\{ -\frac{|c_i|^2}{2\sigma^2} \right\} \end{aligned} \tag{33}$$

whose computational complexity grows exponentially with K. A further simplification, that can be useful when the channel is time-varying, can be the computation of the message (33) relative to symbol c_k taking into account the C nearest symbols only. However, in this case we obtain

an algorithm whose complexity is exponential in C and performance worse than that of the first algorithm described in Section 5.1. For this reason, we resort to an alternative approximate approach. The quantity in square brackets in (32) may be expressed as

$$\begin{aligned}\prod_{i\neq k}\left(\sum_{c_i} f_i(c_i,\theta)P_d^{(n)}(c_i)\right) &\doteq \prod_{i\neq k}\left(\sum_{c_i} P_d^{(n)}(c_i)\exp\left\{\frac{1}{\sigma^2}\mathrm{Re}\left[r_i c_i^* e^{-j\theta}\right]-\frac{|c_i|^2}{2\sigma^2}\right\}\right)\\ &\doteq \exp\left\{\sum_{i\neq k}\ln\left(\sum_{c_i} P_d^{(n)}(c_i)\exp\left\{\frac{1}{\sigma^2}\mathrm{Re}\left[r_i c_i^* e^{-j\theta}\right]-\frac{|c_i|^2}{2\sigma^2}\right\}\right)\right\}\end{aligned} \tag{34}$$

where the identity $e^{\ln x}=x$ has been used. We now adopt the approximations $e^x\simeq 1+x$ and $\ln(1+x)\simeq x$ already used in [5, 6] obtaining

$$\prod_{i\neq k}\left(\sum_{c_i} f(c_i,\theta)P_d^{(n)}(c_i)\right)\simeq\exp\left\{\frac{1}{\sigma^2}\sum_{i\neq k}\left[\mathrm{Re}\left(r_i^*\alpha_i^{(n)}e^{j\theta}\right)-\frac{\beta_i^{(n)}}{2}\right]\right\} \tag{35}$$

having defined, as in [5, 6],

$$\alpha_i^{(n)}=\sum_{c_i} c_i P_d^{(n)}(c_i) \tag{36}$$

and also

$$\beta_i^{(n)}=\sum_{c_i}|c_i|^2 P_d^{(n)}(c_i)\,. \tag{37}$$

Note that the introduced approximations correspond to the following one

$$\sum_i P_i e^{x_i}\simeq e^{\sum_i P_i x_i}\,. \tag{38}$$

This is a good approximation when there exists an index j such that $P_j\gg P_i$, $\forall i\neq j$. In this case, the following approximation can be also adopted:

$$\sum_i P_i e^{x_i}\simeq P_j e^{x_j} \tag{39}$$

leading to an algorithm with the same complexity.

As expressed by (35), we approximate the generic message at the output of variable node θ, with a distribution described by some parameters ($\alpha_i^{(n)}$ and $\beta_i^{(n)}$) which are function of other

messages. Substituting (35) in (32) and solving the integral, we obtain

$$\begin{aligned} P_u^{(n)}(c_k) &\doteq \exp\left\{-\frac{1}{2\sigma^2}\left[\sum_{i\neq k}\beta_i^{(n)}+|c_k|^2\right]\right\}\mathrm{I}_0\left(\frac{|\sum_{i\neq k} r_i^*\alpha_i^{(n)}+r_k^*c_k|}{\sigma^2}\right) \\ &\doteq \exp\left\{-\frac{|c_k|^2}{2\sigma^2}\right\}\mathrm{I}_0\left(\frac{|\sum_{i\neq k} r_i^*\alpha_i^{(n)}+r_k^*c_k|}{\sigma^2}\right) \end{aligned} \tag{40}$$

having discarded a multiplicative factor independent of c_k. This algorithm can be also used when the channel phase is time-varying. In this case, when message $P_u^{(n)}(c_k)$ is computed, it is more convenient to take into account the quantities $\alpha_i^{(n)}$ corresponding to the nearest C symbols c_i. The optimal value of C can be chosen by computer simulations. The cost per code symbol per iteration is in fact $\mathcal{C}_T = M$, being M the number of evaluations of the nonlinear function in (9) to compute a single $\alpha_i^{(n)}$.

The derived algorithm is based on the approximation (38). Better approximations are not further pursuit here since the case of a constant phase may be seen as a particular case of the model considered for the derivation of the following second and third algorithms.

Remark. Instead of (35), we may approximate the pdf representing the generic message at the output of variable node θ as $\delta(\theta-\hat{\theta}_k^{(n)})$ where[6]

$$\hat{\theta}_k^{(n)} = \underset{\theta}{\mathrm{argmax}}\left\{\exp\left\{\frac{1}{\sigma^2}\sum_{i\neq k}\left[\mathrm{Re}\left(r_i^*\alpha_i^{(n)}e^{j\theta}\right)-\frac{\beta_i^{(n)}}{2}\right]\right\}\right\} = \arg\{\sum_{i\neq k} r_i\alpha_i^{(n)*}\}\,. \tag{41}$$

Moreover, if we approximate all values $\hat{\theta}_k^{(n)}$ with a value $\hat{\theta}^{(n)} = \arg\{\sum_i r_i\alpha_i^{(n)*}\}$ independent of k, that is not only the extrinsic information is employed but the overall APP, we obtain the non-Bayesian approach described in [5,6] which can be shown to be equivalent to the application of the EM algorithm [7,8].[7] When the channel is time-varying, the only possibility is to compute a different phase estimate for each symbol taking into account the quantities $\alpha_i^{(n)}$ corresponding to the nearest C symbols c_i. The optimal value of C can be chosen by computer simulations. In the following, this algorithm will be referred to as *windowed Luise algorithm* (see more details in Section 9). □

Second algorithm. For a time-varying channel phase, it is also possible to adopt a more refined channel model. We assume that the observation r_k is described by (4). A realistic model of phase noise is based on a discrete-time Wiener process $\theta_k = \theta_{k-1}+\Delta_k$, characterized by i.i.d. Gaussian increments Δ_k with zero mean and standard deviation σ_Δ, descriptive of the phase noise intensity.

[6] $\delta(x)$ is the Dirac delta function.

[7] An approach similar to that in [5,6] but involving different approximations is described in [9].

Hence[8]

$$\begin{aligned} p(\theta_k|\theta_{k-1},\theta_{k-2},\ldots,\theta_0) &= p(\theta_k|\theta_{k-1}) = p_\Delta(\theta_k-\theta_{k-1}) \\ &= \frac{1}{\sqrt{2\pi\sigma_\Delta^2}}\exp\left\{-\frac{(\theta_k-\theta_{k-1})^2}{2\sigma_\Delta^2}\right\}\ ,\ k=1,\ldots,K-1 \quad (42) \\ p(\theta_0) &= \frac{1}{2\pi}\ ,\ \theta_0\in[0,2\pi)\,. \quad (43) \end{aligned}$$

In this case, we have

$$\begin{aligned} P(\mathbf{c}|\boldsymbol{\theta},\mathbf{r})p(\boldsymbol{\theta}|\mathbf{r}) &\doteq p(\mathbf{r}|\mathbf{c},\boldsymbol{\theta})\chi(\mathbf{c})p(\boldsymbol{\theta}) = \chi(\mathbf{c})p(\theta_0)\prod_{k=0}^{K-1}p(r_k|c_k,\theta_k)\prod_{k=1}^{K-1}p(\theta_k|\theta_{k-1}) \\ &= \chi(\mathbf{c})h_0(\theta_0)\prod_{k=0}^{K-1}f_k(c_k,\theta_k)\prod_{k=1}^{K-1}h_{k-1,k}(\theta_k,\theta_{k-1}) \quad (44) \end{aligned}$$

having defined

$$\begin{aligned} f_k(c_k,\theta_k) &= p(r_k|c_k,\theta_k)\doteq\exp\left\{-\frac{1}{2\sigma^2}|r_k-c_ke^{j\theta_k}|^2\right\} \\ &\doteq \exp\left\{\frac{1}{\sigma^2}\mathrm{Re}[r_kc_k^*e^{-j\theta_k}]-\frac{|c_k|^2}{2\sigma^2}\right\} \\ h_{k-1,k}(\theta_k,\theta_{k-1}) &= p(\theta_k|\theta_{k-1}) \\ h_0(\theta_0) &= p(\theta_0)\,. \quad (45) \end{aligned}$$

The corresponding factor graph is shown in Fig. 6.

Omitting the explicit reference to the current iteration and assuming that the SP algorithm works in the natural domain, as in the previous case let us denote by $P_d(c_k)$ the message from variable node c_k to factor node f_k, and by $P_u(c_k)$ the message in the opposite direction (see Fig. 7). The message $p_d(\theta_k)$ from factor node f_k to variable node θ_k can be expressed as

$$p_d(\theta_k)=\sum_{c_k}P_d(c_k)f_k(c_k,\theta_k)\,. \quad (46)$$

We also assume that in the lower part of the factor graph, describing the noncoherent channel, a forward-backward schedule is adopted. Hence, messages $p_f(\theta_k)$ from factor node $h_{k-1,k}(\theta_{k-1},\theta_k)$ to variable node θ_k, and $p_b(\theta_k)$ from factor node $h_{k,k+1}(\theta_k,\theta_{k+1})$ to variable node θ_k, may be re-

[8]Note that, since the channel phase is defined modulus 2π, the pdf $p_\Delta(\Delta_k)$ can be approximated as Gaussian only if $\sigma_\Delta \ll 2\pi$.

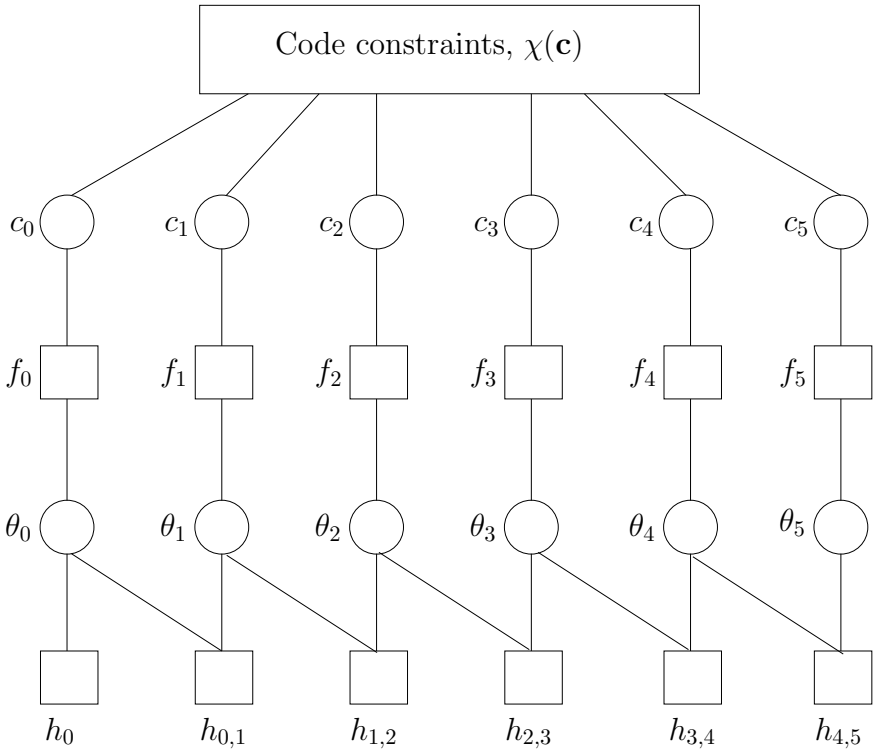

Figure 6: Factor graph corresponding to eqn. (44).

cursively computed as follows:

$$p_f(\theta_k) = \int_0^{2\pi} p_d(\theta_{k-1}) p_f(\theta_{k-1}) h_{k-1,k}(\theta_{k-1}, \theta_k)\, d\theta_{k-1} \tag{47}$$

$$p_b(\theta_k) = \int_0^{2\pi} p_d(\theta_{k+1}) p_b(\theta_{k+1}) h_{k,k+1}(\theta_k, \theta_{k+1})\, d\theta_{k+1} \tag{48}$$

with the following starting conditions:

$$p_f(\theta_0) = h_0(\theta_0) \tag{49}$$

$$p_b(\theta_{K-1}) = \frac{1}{2\pi}\ ,\ \theta_{K-1} \in [0, 2\pi)\,. \tag{50}$$

The probability $P_u(c_k)$ can be finally computed as

$$P_u(c_k) = \int_0^{2\pi} p_f(\theta_k) p_b(\theta_k) f_k(c_k, \theta_k)\, d\theta_k\,. \tag{51}$$

We now show a method for the computation of the probability $P_u(c_k)$ in the form of a series expansion. Note that, differently to [4], we do not approximate the involved pdfs of variables θ_k,

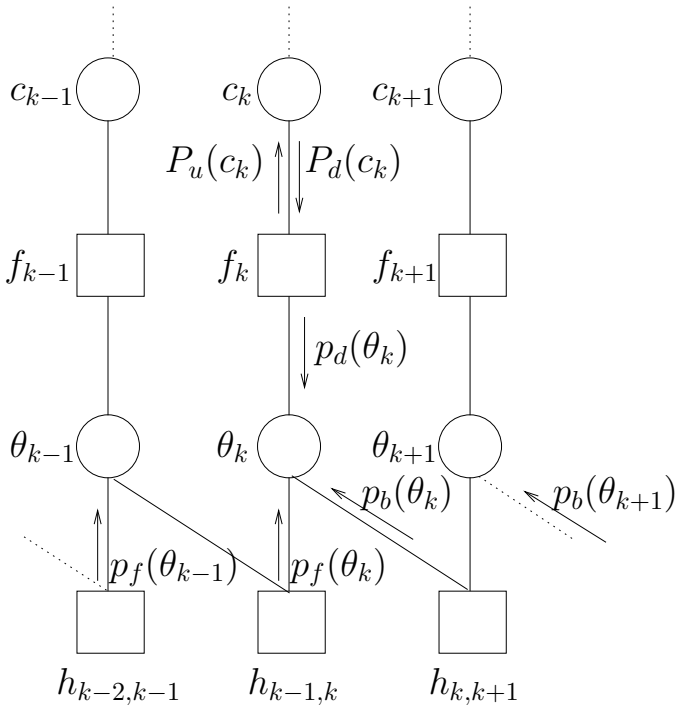

Figure 7: Part of the factor graph in Fig. 6 with explicit reference to the messages sent on the edges.

but *exactly* compute their expression.

The function $f_k(c_k, \theta_k)$ is periodic in θ_k. Hence, it can be expanded in Fourier series. We use the following known result [32, eqn. (9.6.34)]:

$$e^{x\cos\theta} = \mathrm{I}_0(x) + 2\sum_{\ell=1}^{\infty} \mathrm{I}_\ell(x)\cos(\ell\theta) \tag{52}$$

where $\mathrm{I}_\ell(x)$ is the modified Bessel function of the first kind of order ℓ defined as [32, eqn. (9.6.19)]

$$\mathrm{I}_\ell(x) = \frac{1}{\pi}\int_0^\pi e^{x\cos\theta}\cos(\ell\theta)\,d\theta\,. \tag{53}$$

These functions may be recursively computed by observing that $\mathrm{I}_{-\ell}(x) = \mathrm{I}_\ell(x)$ and that

$$\mathrm{I}_{\ell+1}(x) = -\frac{2\ell}{x}\mathrm{I}_\ell(x) + \mathrm{I}_{\ell-1}(x)\,. \tag{54}$$

Defining, for a complex number z, $\phi(z) = \arg(z)$, after some straightforward manipulations we

obtain

$$f_k(c_k,\theta_k) = e^{-\frac{|c_k|^2}{2\sigma^2}} \sum_{\ell=-\infty}^{\infty} \mathrm{I}_\ell\left(\frac{|r_k||c_k|}{\sigma^2}\right) e^{-j\ell\phi(r_k c_k^*)} e^{j\ell\theta_k}\,. \tag{55}$$

Substituting (55) into eqn. (46), we may express

$$p_d(\theta_k) = \sum_{\ell=-\infty}^{\infty} A_k^{(\ell)} e^{j\ell\theta_k} \tag{56}$$

having defined

$$\begin{aligned} A_k^{(\ell)} &= \sum_{c_k} P_d(c_k) e^{-\frac{|c_k|^2}{2\sigma^2}} \mathrm{I}_\ell\left(\frac{|r_k||c_k|}{\sigma^2}\right) e^{-j\ell\phi(r_k c_k^*)} \\ &= e^{-j\ell\phi(r_k)} \sum_{c_k} P_d(c_k) e^{-\frac{|c_k|^2}{2\sigma^2}} \mathrm{I}_\ell\left(\frac{|r_k||c_k|}{\sigma^2}\right) e^{j\ell\phi(c_k)} \\ &= e^{-j\ell\phi(r_k)} \sum_{c_k} P_d(c_k) e^{-\frac{|c_k|^2}{2\sigma^2}} \mathrm{I}_\ell\left(\frac{|r_k||c_k|}{\sigma^2}\right) \frac{c_k^\ell}{|c_k|^\ell}\,. \end{aligned} \tag{57}$$

Note that for M-PSK signals, the expression of coefficients $A_k^{(\ell)}$, neglecting irrelevant terms, simplifies to

$$A_k^{(\ell)} = e^{-j\ell\phi(r_k)} \mathrm{I}_\ell\left(\frac{|r_k|}{\sigma^2}\right) \sum_{c_k} P_d(c_k) c_k^\ell\,. \tag{58}$$

In this case, at the first iteration, when the probabilities of symbols $P(c_k)$ are all equal to $1/M$ (excepting for pilot symbols), these coefficients are zero for $\ell \neq 0, \pm M, \pm 2M, \pm 3M, \ldots$. In general, a reduced number N of coefficients must be taken into account due to the fact that, for a given x, functions $\mathrm{I}_\ell(x)$ are monotonically decreasing for increasing values of ℓ as shown in Fig. 8. When this truncation is performed, it is suitable to apply a window to the truncated coefficients. By means of computer simulations, we found that the Kaiser window with an optimized parameter β [33] assures the best performance.

Pdfs $p_f(\theta_k)$ and $p_b(\theta_k)$ will be of the same form, i.e., they will be periodic and can be expanded in Fourier series:

$$p_f(\theta_k) = \sum_{\ell=-\infty}^{\infty} B_{f,k}^{(\ell)} e^{j\ell\theta_k} \tag{59}$$

$$p_b(\theta_k) = \sum_{\ell=-\infty}^{\infty} B_{b,k}^{(\ell)} e^{j\ell\theta_k}\,. \tag{60}$$

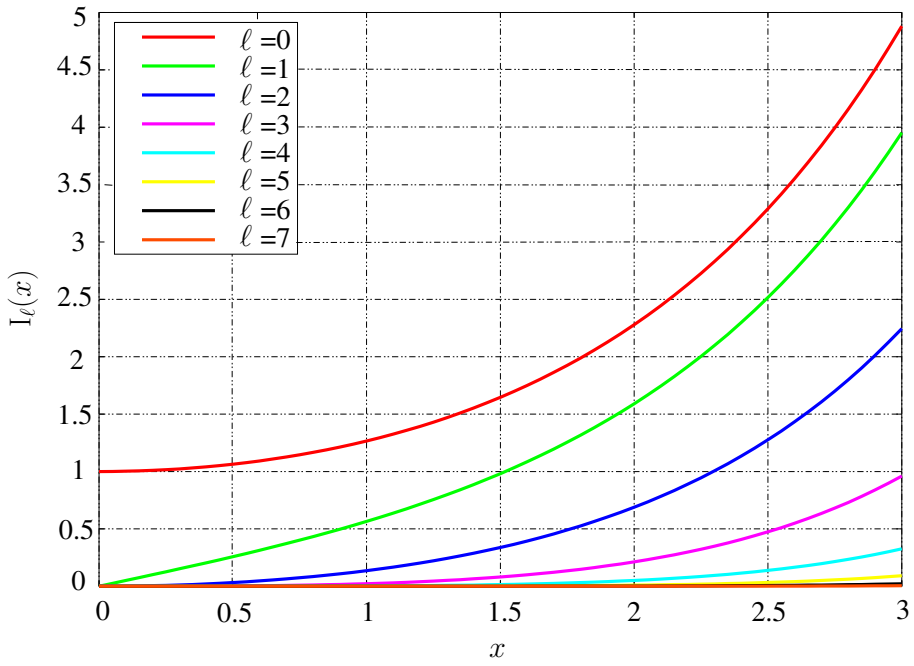

Figure 8: Modified Bessel functions of the first kind.

Substituting (56) and (59) into eqn. (47), we obtain

$$\begin{aligned}\sum_{\ell=-\infty}^{\infty} B_{f,k}^{(\ell)} e^{j\ell\theta_k} &= \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} A_{k-1}^{(m)} B_{f,k-1}^{(n)} \int_0^{2\pi} e^{j(m+n)\theta_{k-1}} p(\theta_k|\theta_{k-1})\, d\theta_{k-1} \\ &= \sum_{\ell=-\infty}^{\infty}\sum_{m=-\infty}^{\infty} A_{k-1}^{(m)} B_{f,k-1}^{(\ell-m)} \int_0^{2\pi} e^{j\ell\theta_{k-1}} p(\theta_k|\theta_{k-1})\, d\theta_{k-1}\,. \end{aligned} \tag{61}$$

For practical values of σ_Δ, the pdf $p(\theta_k|\theta_{k-1})$ given by (42) is strictly limited to an interval of duration less than 2π. Hence we may write

$$\begin{aligned}\int_0^{2\pi} e^{j\ell\theta_{k-1}} p(\theta_k|\theta_{k-1})\, d\theta_{k-1} &\simeq \frac{1}{\sqrt{2\pi\sigma_\Delta^2}} \int_{-\infty}^{\infty} e^{j\ell\theta_{k-1}} e^{-\frac{(\theta_k-\theta_{k-1})^2}{2\sigma_\Delta^2}}\, d\theta_{k-1} \\ &= e^{j\ell\theta_k} \frac{1}{\sqrt{2\pi\sigma_\Delta^2}} \int_{-\infty}^{\infty} e^{j\ell x} e^{-x^2/2\sigma_\Delta^2}\, dx \\ &= D_\ell(\sigma_\Delta) e^{j\ell\theta_k} \end{aligned} \tag{62}$$

where we defined

$$D_\ell(\sigma_\Delta) = \frac{1}{\sqrt{2\pi\sigma_\Delta^2}} \int_{-\infty}^{\infty} e^{j\ell x} e^{-x^2/2\sigma_\Delta^2}\, dx = e^{-\frac{\sigma_\Delta^2 \ell^2}{2}} \tag{63}$$

and, for the last equality, a result in [34, p. 1185] has been used. Hence

$$\sum_{\ell=-\infty}^{\infty} B_{f,k}^{(\ell)} e^{j\ell\theta_k} = \sum_{\ell=-\infty}^{\infty} \left[e^{-\frac{\sigma_\Delta^2 \ell^2}{2}} \sum_{m=-\infty}^{\infty} A_{k-1}^{(m)} B_{f,k-1}^{(\ell-m)} \right] e^{j\ell\theta_k} \tag{64}$$

obtaining a recursive equation for the computation of the coefficients $B_{f,k}^{(\ell)}$:

$$B_{f,k}^{(\ell)} = e^{-\frac{\sigma_\Delta^2 \ell^2}{2}} \sum_{m=-\infty}^{\infty} A_{k-1}^{(m)} B_{f,k-1}^{(\ell-m)} = e^{-\frac{\sigma_\Delta^2 \ell^2}{2}} [A_{k-1}^{(\ell)} \otimes B_{f,k-1}^{(\ell)}] \tag{65}$$

where $\otimes$ denotes convolution between sequences. From condition (49), we derive the following starting condition:

$$B_{f,0}^{(\ell)} = \delta(\ell) \tag{66}$$

where $\delta(\ell)$ denotes the Kronecker delta. Similarly, to compute coefficients $\{B_{b,k}^{(\ell)}\}$, we have the following backward recursion:

$$B_{b,k}^{(\ell)} = e^{-\frac{\sigma_\Delta^2 \ell^2}{2}} [A_{k+1}^{(\ell)} \otimes B_{b,k+1}^{(\ell)}] \tag{67}$$

with starting condition

$$B_{b,K-1}^{(\ell)} = \delta(\ell)\,. \tag{68}$$

Note that the computation of these coefficients can be simplified taking into account that $A_k^{(-\ell)} = A_k^{(\ell)*}$, $B_{f,k}^{(-\ell)} = B_{f,k}^{(\ell)*}$, and $B_{b,k}^{(-\ell)} = B_{b,k}^{(\ell)*}$. Finally, substituting (55), (59), and (60) into eqn. (51) and defining

$$E_k^{(\ell)} = e^{-\frac{|c_k|^2}{2\sigma^2}} \left\{ B_{f,k}^{(\ell)} \otimes B_{b,k}^{(\ell)} \otimes \left[\mathrm{I}_\ell \left(\frac{|r_k||c_k|}{\sigma^2} \right) e^{-j\ell\phi(r_k c_k^*)} \right] \right\} \tag{69}$$

we have

$$\begin{aligned} P_u(c_k) &= \sum_{\ell=-\infty}^{\infty} E_k^{(\ell)} \int_0^{2\pi} e^{j\ell\theta_k}\, d\theta_k = E_k^{(0)} \\ &= e^{-\frac{|c_k|^2}{2\sigma^2}} \left\{ \sum_{m=-\infty}^{\infty} B_{f,k}^{(m)} \sum_{n=-\infty}^{\infty} B_{b,k}^{(n-m)} \mathrm{I}_n \left(\frac{|r_k||c_k|}{\sigma^2} \right) e^{jn\phi(r_k c_k^*)} \right\}. \end{aligned} \tag{70}$$

We now restate the described algorithm in the logarithmic domain. Defining for a complex number z

$$\overline{z} = \ln z = \ln |z| + j\phi(z) \tag{71}$$

from the knowledge of $\ln P_d(c_k)$, we first compute

$$\overline{A}_k^{(\ell)} = -j\ell\phi(r_k) + \ln \left[\sum_{c_k} \exp \left(\ln P_d(c_k) - \frac{|c_k|^2}{2\sigma^2} + \ln \mathrm{I}_\ell \left(\frac{|r_k||c_k|}{\sigma^2} \right) + j\ell\phi(c_k) \right) \right]. \tag{72}$$

In this case the extension of the Jacobian logarithm to complex numbers is necessary: if $z_1 = x_1 + jy_1$ and $z_2 = x_2 + jy_2$ are complex numbers with $x_1 > x_2$

$$\ln(e^{z_1} + e^{z_2}) = z_1 + \ln[1 + e^{-|x_2 - x_1|} e^{j(y_2 - y_1)}] \tag{73}$$

The forward and backward recursions (65) and (67), in the logarithmic domain become

$$\overline{B}_{f,k}^{(\ell)} = -\frac{\sigma_\Delta^2 \ell^2}{2} + \ln\left[\sum_{m=-\infty}^{\infty} \exp\{\overline{A}_{k-1}^{(m)} + \overline{B}_{f,k-1}^{(\ell-m)}\}\right] \tag{74}$$

$$\overline{B}_{b,k}^{(\ell)} = -\frac{\sigma_\Delta^2 \ell^2}{2} + \ln\left[\sum_{m=-\infty}^{\infty} \exp\{\overline{A}_{k+1}^{(m)} + \overline{B}_{b,k+1}^{(\ell-m)}\}\right] \tag{75}$$

and finally, from (70)

$$\ln P_u(c_k) = -\frac{|c_k|^2}{2\sigma^2} + \ln\left\{\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \exp\left[\overline{B}_{f,k}^{(m)} + \overline{B}_{b,k}^{(n-m)} + \ln \mathrm{I}_n\left(\frac{|r_k||c_k|}{\sigma^2}\right) + jn\phi(r_k c_k^*)\right]\right\}. \tag{76}$$

Note that we supposed to employ the serial schedule. However, if a parallel implementation is required, the flooding schedule can be adopted. The extension of the proposed algorithm is straightforward. In fact, in the forward and backward recursions (65) and (67), the previous coefficients will be now related to the previous iteration.

As already mentioned, it is sufficient to consider a reduced number N (supposed odd) of coefficients of the Fourier series. In this case the cost per code symbol per iteration is $\mathcal{C}_T = M(N+1)/2 + 3N(N+1)/2 + MN$. In fact, exploiting the Hermitian symmetry of the Fourier coefficients, for a given k, $M(N+1)/2$ is the cost necessary for the computation of coefficients $A_k^{(\ell)}$ for different values of ℓ , $N(N+1)/2$ is the cost for the computation of coefficients $B_{f,k}^{(\ell)}$ ($B_{b,k}^{(\ell)}$), $N(N+1)/2$ the cost for the computation of $B_{f,k}^{(\ell)} \otimes B_{b,k}^{(\ell)}$, N the cost associated to the computation of $P_u(c_k) = E_k^{(0)}$, and finally M the number of values assumed by each message computed for each code symbol. In the following, this algorithm will be referred to as *Fourier algorithm.*

Remark. Note that, excepting for the truncation of the Fourier coefficients, the algorithm described in this Section, whose complexity is linear in the number of transmitted symbols, is optimal, even for a time-invariant channel phase (that is for $\sigma_\Delta \to 0$). This algorithm can be considered as the Bayesian counterpart of the polynomial complexity algorithm in [10]. In addition, provided that the channel phase process is Markovian, i.e.,

$$p(\theta_k|\theta_{k-1}, \theta_{k-2}, \ldots, \theta_0) = p(\theta_k|\theta_{k-1})$$

irrespective of the distribution of $p(\theta_k|\theta_{k-1})$, the structure of the algorithm described in this Section does not change—the only modification is the expression of coefficients D_ℓ in (63). □

Third algorithm. Let us consider eqn. (46). If the messages $P_d(c_k)$ were the exact a posteriori probabilities of the code symbols, it would be

$$p_d(\theta_k) = \sum_{c_k} P_d(c_k) f_k(c_k, \theta_k) = p(r_k|c_k)\,. \tag{77}$$

The pdf $p(r_k|\theta_k)$, which is the linear combination of Gaussian functions, can be approximated as Gaussian with the same mean and variance (first and second moment matching). Being

$$E\{r_k|c_k,\theta_k\} = c_k e^{j\theta_k} \tag{78}$$
$$E\{|r_k|^2|c_k,\theta_k\} = 2\sigma^2 + |c_k|^2 \tag{79}$$

from direct calculation we have

$$E\{r_k|\theta_k\} = \sum_{c_k} E\{r_k|c_k,\theta_k\} P_d(c_k) = e^{j\theta_k} \sum_{c_k} c_k P_d(c_k) = \alpha_k e^{j\theta_k} \tag{80}$$
$$E\{|r_k|^2|\theta_k\} = \sum_{c_k} E\{|r_k||c_k,\theta_k\} P_d(c_k) = 2\sigma^2 + \sum_{c_k} |c_k|^2 P_d(c_k) = 2\sigma^2 + \beta_k \tag{81}$$

where α_k and β_k are given in (36) and (37), respectively. Hence

$$var\{r_k|\theta_k\} = E\{|r_k|^2|\theta_k\} - |E\{r_k|\theta_k\}|^2 = 2\sigma^2 + \beta_k - |\alpha_k|^2 \tag{82}$$

and

$$p_d(\theta_k) = p(r_k|\theta_k) \simeq \exp\left\{-\frac{|r_k - \alpha_k e^{j\theta_k}|^2}{2\sigma^2 + \beta_k - |\alpha_k|^2}\right\} \doteq \exp\left\{2\frac{\mathrm{Re}[r_k \alpha_k^* e^{-j\theta_k}]}{2\sigma^2 + \beta_k - |\alpha_k|^2}\right\}\,. \tag{83}$$

having discarded a multiplicative factor independent of θ_k. Substituting (83) in the recursive integral equation (47), we obtain

$$p_f(\theta_k) \simeq \int_0^{2\pi} \exp\left\{2\frac{\mathrm{Re}[r_{k-1}\alpha_{k-1}^* e^{-j\theta_{k-1}}]}{2\sigma^2 + \beta_{k-1} - |\alpha_{k-1}|^2}\right\} p_f(\theta_{k-1}) h_{k-1,k}(\theta_{k-1},\theta_k)\, d\theta_{k-1}\,. \tag{84}$$

When the channel phase is slowly-varying, i.e., for $\sigma_\Delta \to 0$, we have $h_{k-1,k}(\theta_{k-1},\theta_k) = p(\theta_k|\theta_{k-1}) \simeq \delta(\theta_k - \theta_{k-1})$. In this case, the solution of the integral eqn. (84) is a $p_f(\theta_k)$ with Tikhonov distribution, i.e., of the form

$$p_f(\theta_k) = \exp\left\{\mathrm{Re}[a_{f,k} e^{-j\theta_k}]\right\} \tag{85}$$

and $a_{f,k}$ can be recursively computed as

$$a_{f,k} = a_{f,k-1} + 2\frac{r_{k-1}\alpha_{k-1}^*}{2\sigma^2 + \beta_{k-1} - |\alpha_{k-1}|^2} \tag{86}$$

with the starting condition $a_{f,0} = 0$. Similarly, the solution for the recursive integral equation (48) is a $p_b(\theta_k)$ of the form

$$p_b(\theta_k) = \exp\left\{\mathrm{Re}[a_{b,k}e^{-j\theta_k}]\right\} \tag{87}$$

and $a_{b,k}$ can be recursively computed as

$$a_{b,k} \;=\; a_{b,k+1} + 2\frac{r_{k+1}\alpha_{k+1}^*}{2\sigma^2 + \beta_{k+1} - |\alpha_{k+1}|^2} \tag{88}$$

with the starting condition $a_{b,K-1} = 0$. Hence,

$$P_u(c_k) \sim \exp\left\{-\frac{|c_k|^2}{2\sigma^2}\right\} \mathrm{I}_0\left(\left|a_{f,k} + a_{b,k} + \frac{r_k c_k^*}{\sigma^2}\right|\right) . \tag{89}$$

When the phase is rapidly-varying, the approximation $h_{k-1,k}(\theta_{k-1}, \theta_k) \simeq \delta(\theta_k - \theta_{k-1})$ does not hold. However, we found that good approximations of functions $p_f(\theta_k)$ and $p_b(\theta_k)$ are still of the form (85) and (87) but coefficients $a_{f,k}$ and $a_{b,k}$ must be updated as

$$a_{f,k} \;=\; \left[a_{f,k-1} + 2\frac{r_{k-1}\alpha_{k-1}^*}{2\sigma^2 + \beta_{k-1} - |\alpha_{k-1}|^2}\right] \cdot \gamma\left(\sigma_\Delta^2, \left|a_{f,k-1} + 2\frac{r_{k-1}\alpha_{k-1}^*}{2\sigma^2 + \beta_{k-1} - |\alpha_{k-1}|^2}\right|\right) \tag{90}$$

$$a_{b,k} \;=\; \left[a_{b,k+1} + 2\frac{r_{k+1}\alpha_{k+1}^*}{2\sigma^2 + \beta_{k+1} - |\alpha_{k+1}|^2}\right] \cdot \gamma\left(\sigma_\Delta^2, \left|a_{f,k+1} + 2\frac{r_{k+1}\alpha_{k+1}^*}{2\sigma^2 + \beta_{k+1} - |\alpha_{k+1}|^2}\right|\right) \tag{91}$$

where the real function $\gamma(x_1, x_2)$, of real arguments x_1 and x_2 can be numerically computed and stored in advance. This choice derives from the observation that the result of the integral

$$\frac{1}{\sqrt{2\pi\sigma_\Delta^2}} \int_0^{2\pi} e^{\mathrm{Re}[ze^{-jx}]} e^{-\frac{(x-y)^2}{2\sigma_\Delta^2}}\, dx$$

where z is a complex number and x and y are real numbers, excepting for an irrelevant amplitude factor, looks like the function $e^{\gamma \mathrm{Re}[ze^{-jy}]}$, where γ is a real number which depends on $|z|$ and σ_Δ^2, since the integral does not change the position of the maxima and minima of the function $e^{\mathrm{Re}[ze^{-jy}]}$. For some values of σ_Δ^2, a plot of $\gamma(\sigma_\Delta^2, |z|)$ versus $|z|$ is shown in Fig. 9. The details on the computation of $\gamma(\sigma_\Delta^2, |z|)$, along with its heuristic closed-form expression, are given in the Appendix A. For this algorithm, that in the following will be referred to as *recursive algorithm*, the cost per code symbol per iteration is $\mathcal{C}_T = 2M$, being $2M$ the cost associated to the computation of a single couple (α_k, β_k).

Modification in the case of long pilot fields. When the pilot symbols are arranged in long fields separated by long blocks of code symbols, as in the case of the DVB-S2 system, it is necessary to slightly modify the algorithm in order to speed-up the convergence process and to avoid the risk of a phase ambiguity. In fact, after the first iteration, in the recursive integral equations (47) and (48), if the product $p_d(\theta_k)p_f(\theta_k)$ (considering the exact expression of $p_d(\theta_k)$) is characterized by

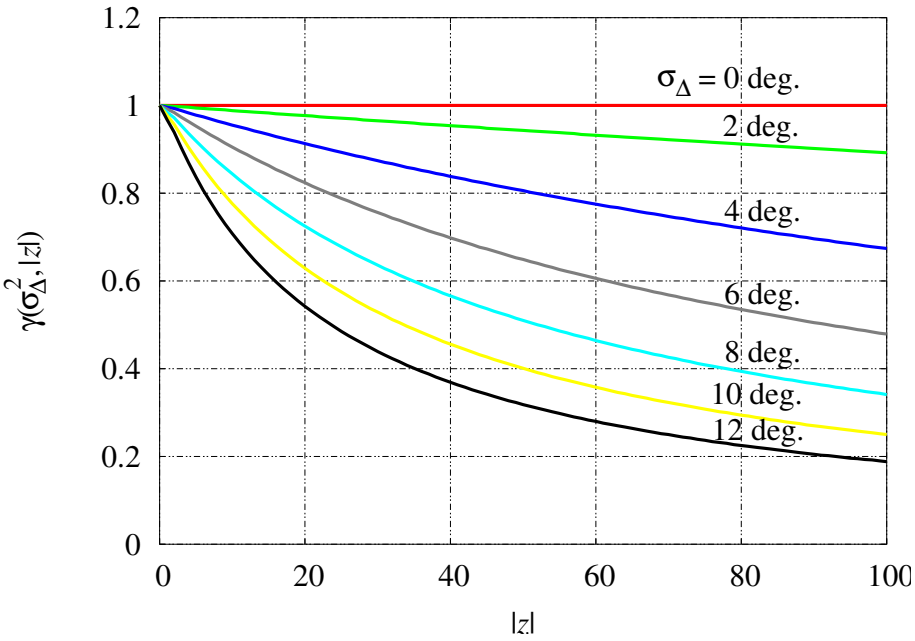

Figure 9: Coefficient $\gamma(\sigma, |z|)$.

a dominant exponential term, i.e., if there exists a value $\overline{c}_k$ such that

$$\ln P_d(\overline{c}_k) + \left| a_{f,k} + \frac{r_k \overline{c}_k^*}{\sigma^2} \right| > \delta + \ln P_d(c_k) + \left| a_{f,k} + \frac{r_k c_k^*}{\sigma^2} \right| , \forall c_k \neq \overline{c}_k \tag{92}$$

where δ is a real parameter to be optimized by computer simulation for the particular pilot symbol distribution and phase noise intensity, it is preferable to choose $\alpha_k = \overline{c}_k$ and $\beta_k = |\overline{c}_k|^2$. Otherwise, we choose α_k and β_k as in (36) and (37), i.e., $\alpha_k = \sum_{c_k} P_d(c_k) c_k$ and $\beta_k = \sum_{c_k} P_d(c_k)|c_k|^2$. In the numerical results related to the DVB-S2 system settings, parameter δ has been chosen equal to 1.5.

Summary of the algorithm. As shown in the numerical results (see Section 8), the recursive algorithm represents the best solution from both the complexity and performance point of view. For this reason, we now summarize this algorithm.

1. Given the messages $P_d(c_k), k = 0, 1, \ldots, K-1, c_k \in \mathcal{X}$, sent by the decoder to the detector at the n-th iteration, for $k = 0, 1, \ldots, K-1$ compute

$$\alpha_k = \begin{cases} \overline{c}_k & \text{from the 2nd iteration on, if there exists a value } \overline{c}_k \text{ such that (92) holds} \\ \sum_{c_k} P_d(c_k) c_k & \text{otherwise} \end{cases}$$

and

$$\beta_k = \begin{cases} |\overline{c}_k|^2 & \text{from the 2nd iteration on, if there exists a value } \overline{c}_k \text{ such that (92) holds} \\ \sum_{c_k} P_d(c_k)|c_k|^2 & \text{otherwise.} \end{cases}$$

2. Forward recursion. Let $a_{f,0} = 0$. For all $k = 1, 2, \ldots, K-1$ compute

$$a_{f,k} = \left[a_{f,k-1} + 2\frac{r_{k-1}\alpha_{k-1}^*}{2\sigma^2 + \beta_{k-1} - |\alpha_{k-1}|^2} \right] \cdot \gamma \left(\sigma_\Delta^2, \left| a_{f,k-1} + 2\frac{r_{k-1}\alpha_{k-1}^*}{2\sigma^2 + \beta_{k-1} - |\alpha_{k-1}|^2} \right| \right)$$

where the function γ is defined in the appendix.

3. Backward recursion. Let $a_{b,K-1} = 0$. For all $k = K-2, \ldots, 1, 0$ compute

$$a_{b,k} = \left[a_{b,k+1} + 2\frac{r_{k+1}\alpha_{k+1}^*}{2\sigma^2 + \beta_{k+1} - |\alpha_{k+1}|^2} \right] \cdot \gamma \left(\sigma_\Delta^2, \left| a_{f,k+1} + 2\frac{r_{k+1}\alpha_{k+1}^*}{2\sigma^2 + \beta_{k+1} - |\alpha_{k+1}|^2} \right| \right) .$$

4. The messages to be sent from detector to the decoder for a new decoder iteration will be, for all $k = 0, 1, \ldots, K-1, c_k \in \mathcal{X}$

$$P_u(c_k) \sim \exp\left\{ -\frac{|c_k|^2}{2\sigma^2} \right\} \mathrm{I}_0 \left(\left| a_{f,k} + a_{b,k} + \frac{r_k c_k^*}{\sigma^2} \right| \right) .$$

6.2 Noncoherent channel: quantization of channel parameters

First algorithm. Considering a time-invariant noncoherent channel, we assume, as in [35], that the channel phase θ may take on the following L values: $\Theta = \{0, 2\pi/L, \ldots, 2\pi(L-1)/L\}$.[9] In this case, a factorization as in (30) still holds, the only difference is that pdfs $p(\theta|\mathbf{r})$ and $p(\theta)$ are now pmfs. As a consequence, omitting the explicit dependence on the current iteration and defining

$$\eta_k(\ell) = \sum_{c_k} f_k(c_k, 2\pi\ell/L) P_d(c_k) \tag{93}$$

the integral which appears in (32) becomes

$$P_u(c_k) \simeq \sum_{\ell=0}^{L-1} f_k(c_k, 2\pi\ell/L) \prod_{i \neq k} \eta_i(\ell) \tag{94}$$

that can be easily numerically computed. As already mentioned, when the channel phase is time-varying, it is more convenient to compute the probabilities $P_u(c_k)$ by taking into account the contribution of the nearest C symbols c_i only. In this case, expressing (93) and (94) in the

[9] In [35], the authors state that for M-PSK signals, $L = 8M$ values are sufficient to have no performance loss.

logarithmic domain, we have (assuming C even)

$$\overline{\eta}_k(\ell) = \ln \eta_k(\ell) = \ln \sum_{c_k} \exp\{\ln f_k(c_k, 2\pi\ell/L) + \ln P_d(c_k)\} \tag{95}$$

$$\ln P_u(c_k) \simeq \ln \sum_{\ell=0}^{L-1} \exp \left\{ f_k(c_k, 2\pi\ell/L) + \sum_{\substack{i=k-C/2 \\ i\neq k}}^{k+C/2} \overline{\eta}_i(\ell) \right\} . \tag{96}$$

The cost per code symbol per iteration is $\mathcal{C}_T = 2LM$, being, for a given k, LM the cost associated to the computation of $\overline{\eta}_k(\ell)$ for all values of ℓ, L the cost associated to the computation of each value of $\ln P_u(c_k)$, and finally M the number of values assumed by each message computed for each code symbol.

Second algorithm. We now apply the quantization approach to a more refined time-varying noncoherent channel model (channel model given by eqn. (4)). Even in this case, we assume that the channel phase θ_k may take on the following L values: $\{0, 2\pi/L, \ldots, 2\pi(L-1)/L\}$. As already mentioned, a realistic model of phase noise is based on a discrete-time Wiener process. As a consequence, we may compute the recursive equations (47) and (48) by approximating the integrals with a sum on the considered L values. However, we adopt another approach. Since we need a quantized phase model, we approximate the real continuous distribution with a discrete random walk with phase differences of $\pm 2\pi/L$ occurring with probability P_Δ.[10] Hence, the pmf of the random variable Δ_k is

$$P_\Delta(\Delta_k) = \begin{cases} 1 - P_\Delta & \text{for } \Delta_k = 0 \\ P_\Delta/2 & \text{for } \Delta_k = \pm\frac{2\pi}{L} \end{cases} \tag{97}$$

and it is also

$$P(\theta_k|\theta_{k-1}, \theta_{k-2}, \ldots, \theta_0) = P(\theta_k|\theta_{k-1}) = P_\Delta(\theta_k - \theta_{k-1}) . \tag{98}$$

The value of P_Δ can be chosen in such a way the variance of the discrete random walk increments equals the variance σ_Δ^2 of the increments in the Wiener model, i.e.,

$$P_\Delta = \sigma_\Delta^2 \left(\frac{L}{2\pi} \right)^2 . \tag{99}$$

With this phase model, the joint APP $P(\mathbf{c}, \boldsymbol{\theta}|\mathbf{r})$ may be expressed as (assuming that the

[10] This model is sufficient in most cases. Otherwise, a more refined model can be adopted if large values of L are necessary, due to the use of a dense constellation, or large values of σ_Δ occur.

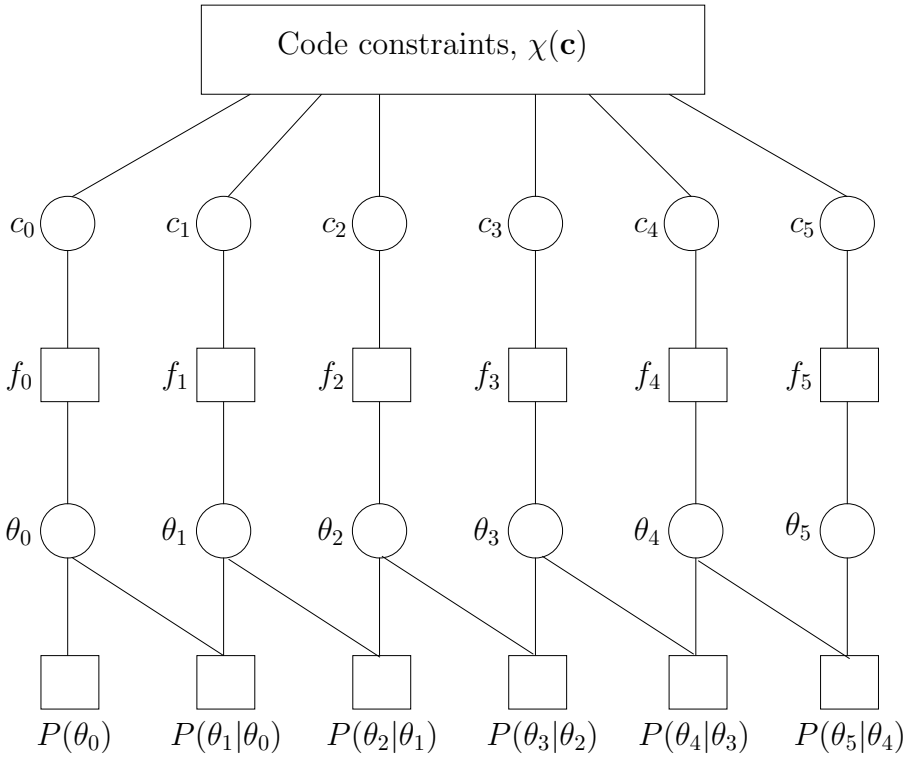

Figure 10: Factor graph corresponding to eqn. (100).

transmitted symbols and the channel parameters are independent)

$$\begin{aligned} P(\mathbf{c},\boldsymbol{\theta}|\mathbf{r}) &\doteq \chi(\mathbf{c})P(\boldsymbol{\theta})p(\mathbf{r}|\mathbf{c},\boldsymbol{\theta}) \\ &\doteq \chi(\mathbf{c})P(\theta_0)\prod_{k=1}^{K-1}P(\theta_k|\theta_{k-1})\prod_{k=0}^{K-1}f_k(c_k,\theta_k) \end{aligned} \tag{100}$$

where $f_k(c_k,\theta_k)$ is defined as in (45). The random variable θ_0 can be assumed to take on all discrete values with the same probability. Hence, its pmf is $P(\theta_0)=1/L$, $\theta_0=0,2\pi/L,\ldots,2\pi(L-1)/L$. The factor graph corresponding to eqn. (100) is shown in Fig. 10. The complexity of the SP algorithm applied to this factor graph depends exclusively on the number of phase quantization levels L.

Omitting the explicit reference to the current iteration and assuming that the SP algorithm works in the natural domain, as in the previous case let us denote by $P_d(c_k)$ the message from variable node c_k to factor node f_k, and by $P_u(c_k)$ the message in the opposite direction (see Fig. 11). The message $\eta_k(\ell)$ from factor node f_k to variable node θ_k can be expressed as in (93). Messages $P_{f,k}(\ell)$ from factor node $P(\theta_k|\theta_{k-1})$ to variable node θ_k, and $P_{b,k}(\ell)$ from factor node

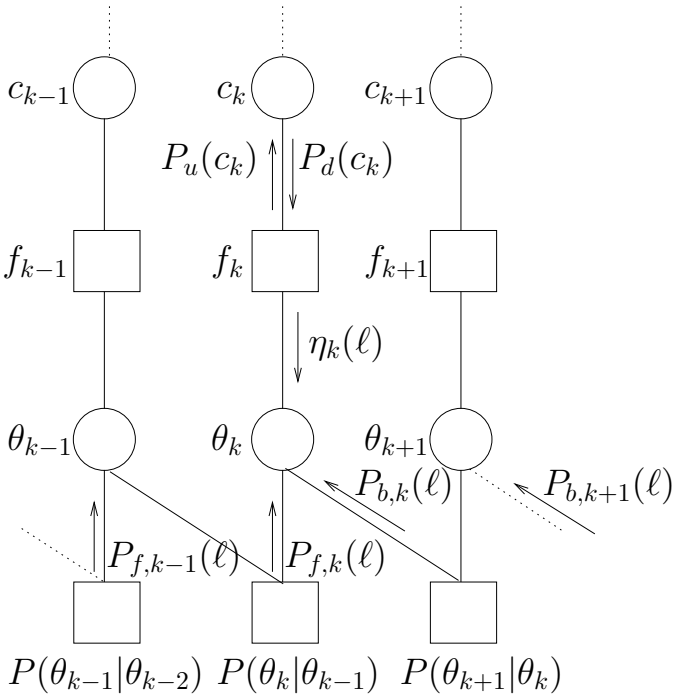

Figure 11: Part of the factor graph in Fig. 10 with explicit reference to the messages sent on the edges.

$P(\theta_{k+1}|\theta_k)$ to variable node θ_k, may be recursively computed as follows:

$$\begin{aligned} P_{f,k}(\ell) &= (1-P_\Delta)P_{f,k-1}(\ell)\eta_{k-1}(\ell) \\ &\quad + \frac{P_\Delta}{2}P_{f,k-1}(\ell-1)\eta_{k-1}(\ell-1) \\ &\quad + \frac{P_\Delta}{2}P_{f,k-1}(\ell+1)\eta_{k-1}(\ell+1) \qquad (101) \\ P_{b,k}(\ell) &= (1-P_\Delta)P_{b,k+1}(\ell)\eta_{k+1}(\ell) \\ &\quad + \frac{P_\Delta}{2}P_{b,k+1}(\ell-1)\eta_{k+1}(\ell-1) \\ &\quad + \frac{P_\Delta}{2}P_{b,k+1}(\ell+1)\eta_{k+1}(\ell+1)\,. \qquad (102) \end{aligned}$$

Finally

$$P_u(c_k) = \sum_{\ell=0}^{L-1} P_{f,k}(\ell)P_{b,k}(\ell)f_k(c_k, 2\pi\ell/L)\,. \qquad (103)$$

Algorithm	$\mathcal{C}_T$
Sect. 5.1, 1st (exponential-complexity algorithm)	$(C+1)M^{Q+1}$
Sect. 5.1, 2nd (linear-complexity algorithm)	$2M^2C$
Sect. 5.1, 3rd (prediction algorithm)	$2M^2C$
Sect. 6.1, 1st	M
Sect. 6.1, 2nd (Fourier algorithm)	$M(N+1)/2+3N(N+1)/2+MN$
Sect. 6.1, 3rd (recursive algorithm)	$2M$
Sect. 6.2, 1st	$2LM$
Sect. 6.2, 2nd (quantization-based algorithm)	$2LM+6L$

Table 1: Cost per code symbol per iteration for the proposed algorithms.

In the logarithmic domain

$$\begin{aligned}
\ln P_{f,k}(\ell) &= \ln\left[e^{\ln(1-P_\Delta)+\ln P_{f,k-1}(\ell)+\overline{\eta}_{k-1}(\ell)}\right. \\
&\quad +e^{\ln(P_\Delta/2)+\ln P_{f,k-1}(\ell-1)+\overline{\eta}_{k-1}(\ell-1)} \\
&\quad \left.+e^{\ln(P_\Delta/2)+\ln P_{f,k-1}(\ell+1)+\overline{\eta}_{k-1}(\ell+1)}\right] \qquad (104)\\
\ln P_{b,k}(\ell) &= \ln\left[e^{\ln(1-P_\Delta)+\ln P_{b,k+1}(\ell)+\overline{\eta}_{k+1}(\ell)}\right. \\
&\quad +e^{\ln(P_\Delta/2)+\ln P_{b,k+1}(\ell-1)+\overline{\eta}_{k+1}(\ell-1)} \\
&\quad \left.+e^{\ln(P_\Delta/2)+\ln P_{b,k+1}(\ell+1)+\overline{\eta}_{k+1}(\ell+1)}\right] \qquad (105)\\
\ln P_u(c_k) &= \ln\sum_{\ell=0}^{L-1} e^{\ln P_{f,k}(\ell)+\ln P_{b,k}(\ell)+\ln f_k(c_k,2\pi\ell/L)} \qquad (106)
\end{aligned}$$

where $\overline{\eta}_k(\ell)$ is given by (95). The cost per code symbol per iteration is $\mathcal{C}_T = 2LM+6L$, being, for a given k, LM the cost associated to the computation of $\overline{\eta}_k(\ell)$ for all values of ℓ, $3L$ the cost associated to the computation of $P_{f,k}(\ell)$ ($P_{b,k}(\ell)$) for all values of ℓ, and LM the cost associated to the computation of all values of $\ln P_u(c_k)$. In the following, this algorithm will be referred to as *quantization-based algorithm.*

7 Complexity

In Table 1, the cost per code symbol per iteration of the described algorithms is reported. Considering the numerical results shown in the next Section, the algorithm exhibiting the best trade-off between complexity and performance is the recursive algorithm.

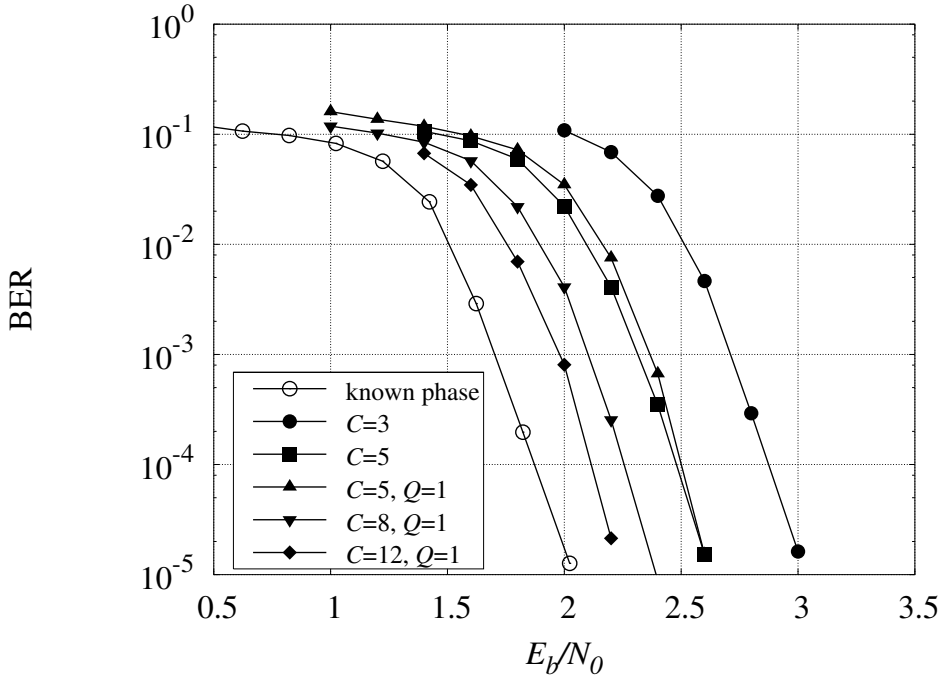

Figure 12: Performance of the exponential-complexity algorithm in the case of a noncoherent time-invariant channel.

8 Numerical Results

In this Section, the performance of the described detection schemes is assessed by computer simulations in terms of bit error rate (BER) versus E_b/N_0, E_b being the received signal energy per information bit and $N_0/2$ the two-sided noise power spectral density. The considered code is a (3,6)-regular LDPC code with codewords of length 4000. The BPSK modulation is used and a maximum of 200 iterations of the SP algorithm on the overall graph is allowed. A pilot symbol every 19 code bits is added in order to make the iterative decoding algorithms bootstrap. This corresponds to a decrease in the effective transmission rate, resulting in an increase in the required signal-to-noise ratio of about 0.223 dB which has been introduced artificially in the curve labeled "known phase" for the sake of comparison. Hence, the gap between the "known phase" curve and the others is uniquely due to the need for phase estimation/compensation, and not to the rate decrease due to pilot symbols insertion.

8.1 A priori average over channel parameters

In Fig. 12 and 13, the performance of the exponential- and linear-complexity algorithms described in Section 5.1 is shown for different values of C in the case of a noncoherent time-invariant channel. In Fig. 12, the exponential-complexity algorithm is considered and compared with an ideal coherent receiver (curve labeled "known phase"). Complexity reduction is also

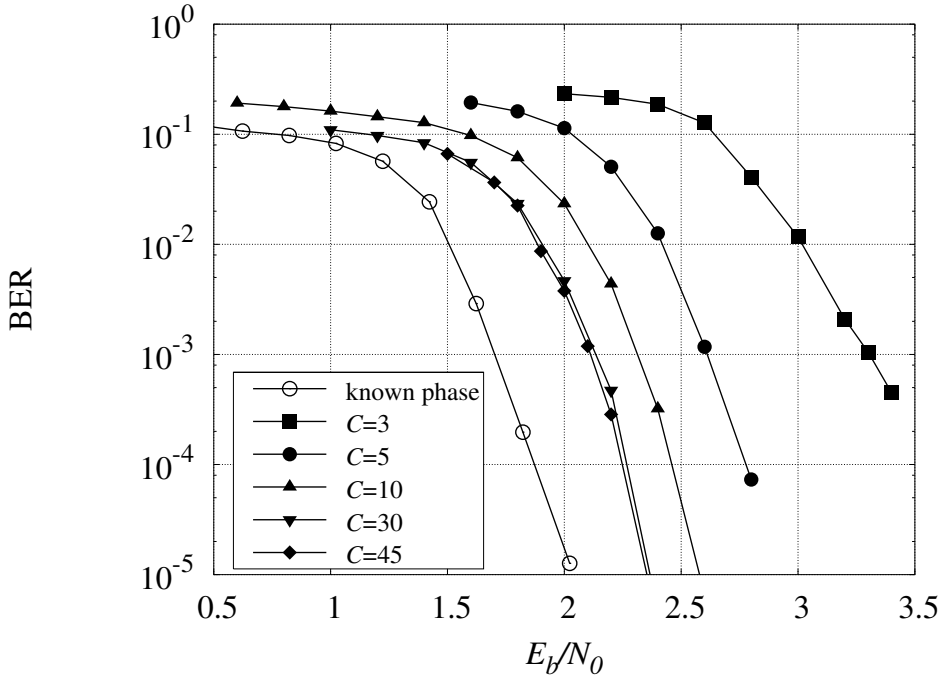

Figure 13: Performance of the linear-complexity algorithm in the case of a noncoherent time-invariant channel.

considered to increase the phase memory C without an increase in complexity. As intuitively expected, the performance of the ideal coherent receiver is approached with a very limited complexity. In fact a value of $Q = 1$ is in practice sufficient to attain the performance of the full-complexity receiver. For the linear-complexity algorithm similar considerations do not hold. In fact, it can be observed that larger values of C are required with respect to the exponential-complexity algorithm to obtain a given performance and, in addition, from Fig. 13 it seems that this algorithm is not able to reach the optimal "known phase" performance.

We also considered (see Fig. 14) a time-varying noncoherent channel. As already mentioned, the phase noise is modeled as a discrete-time Wiener process with incremental variance over a signaling interval equal to σ_Δ^2. We considered the linear-complexity algorithm with $C = 10$. Note that an optimization of the phase memory parameter C was not performed for each considered channel rate of change—we simply assessed the performance of the linear-complexity algorithm for different values of σ_Δ. From Fig. 14 we may observe that a phase noise standard deviation up to 10 degrees (per signaling interval) does not significantly degrade the receiver performance demonstrating the high robustness of the proposed detection schemes. For high values of σ_Δ, a performance improvement may be also obtained by using the prediction algorithm.

For a phase noise standard deviation of 6 degrees we optimized, by means of computer simulations, the value of C for the exponential- and linear-complexity algorithms. In Fig. 15 we show the BER versus C of the exponential-complexity algorithm ($Q = 1$) for $E_b/N_0 = 2$ dB, and that of the linear-complexity algorithm for $E_b/N_0 = 2.5$ dB. In both cases, we may ob-

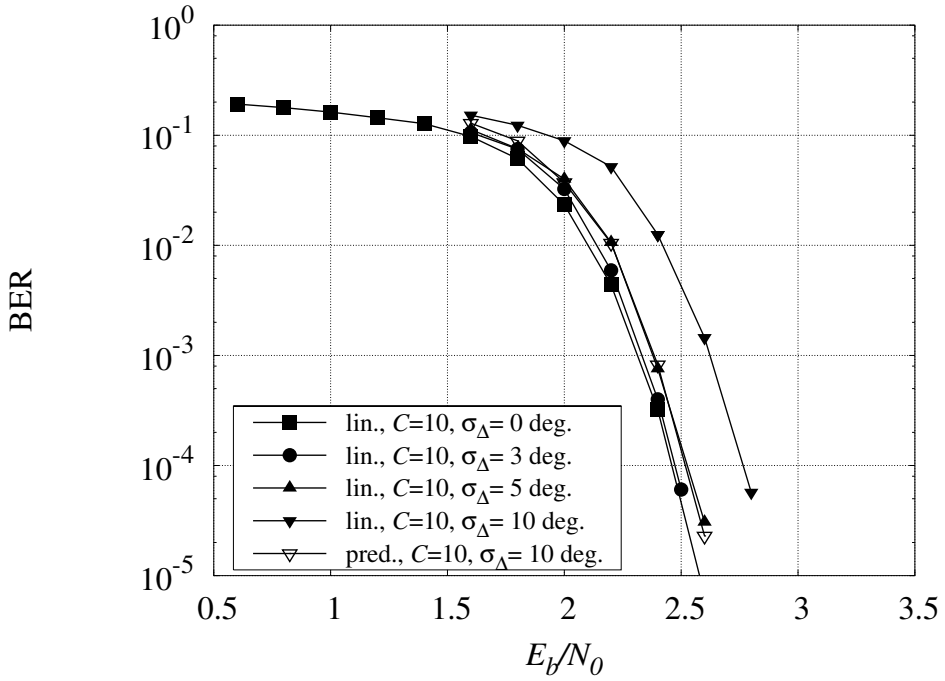

Figure 14: Performance of the linear-complexity and prediction algorithms in the case of a time-varying channel phase.

serve that for this channel rate of change, the optimal value of C is around $C = 20$. For this reason, in Fig. 16 we reported, for $\sigma_\Delta = 6$ degrees, the performance of the algorithms for the optimized value of C. Despite the fact that, being $Q = 1$, the complexity is approximately the same, the exponential-complexity algorithm has a better performance of about 0.2 dB. As already mentioned, an improved may be obtained by using the prediction-based algorithm. In Fig. 17 we compare the performance of the algorithm with exponential complexity with that of the prediction-based algorithm. Different values of σ_Δ have been considered and the value of C is optimized in each case by computer simulation.

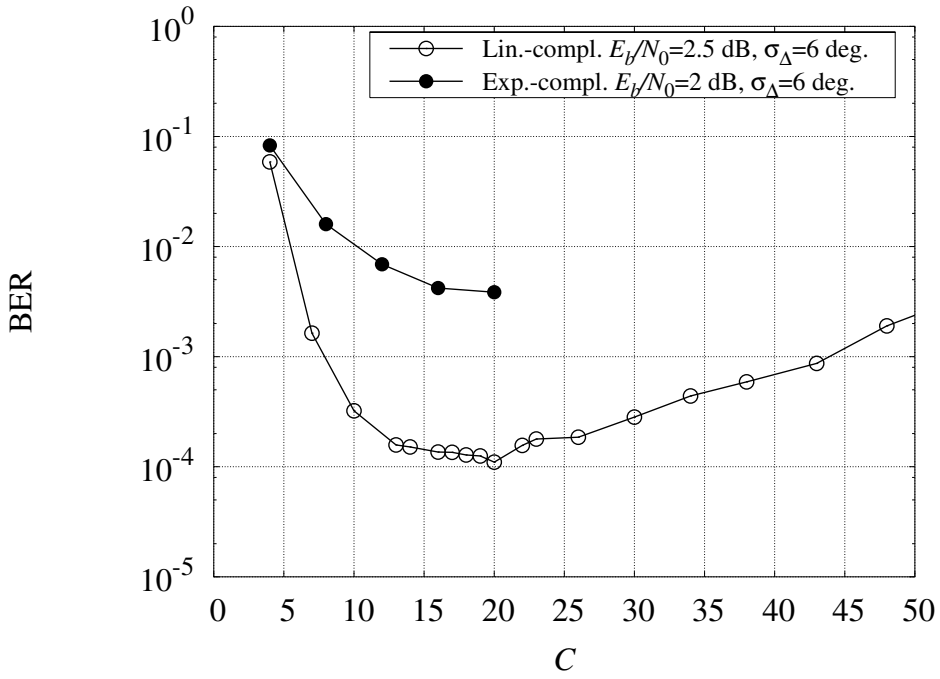

Figure 15: Performance of the exponential- and linear-complexity algorithms versus C for $\sigma_\Delta = 6$ degrees and a given value of E_b/N_0.

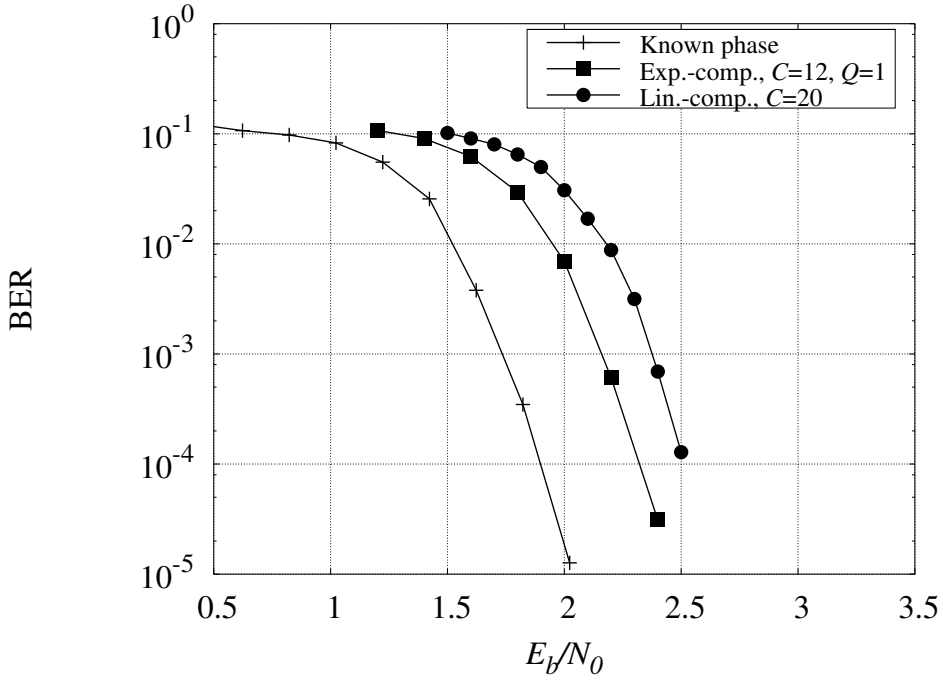

Figure 16: Performance of the exponential- and linear-complexity algorithms for the optimized value of C and $\sigma_\Delta = 6$ degrees.

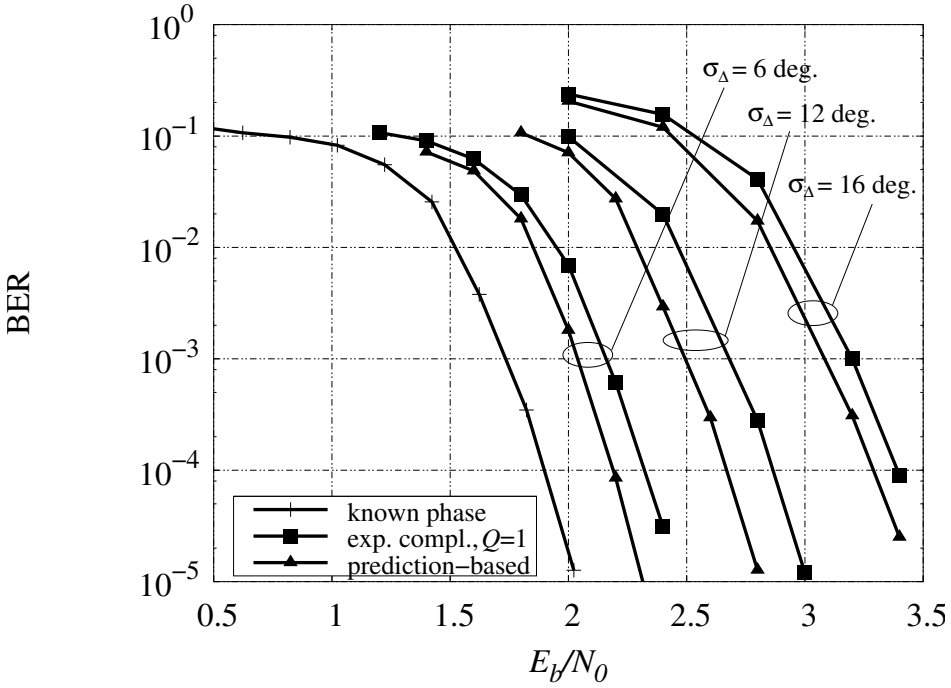

Figure 17: Performance of the exponential-complexity and prediction-based algorithms for different values of σ_Δ. The values of C have been optimized by means of computer simulations.

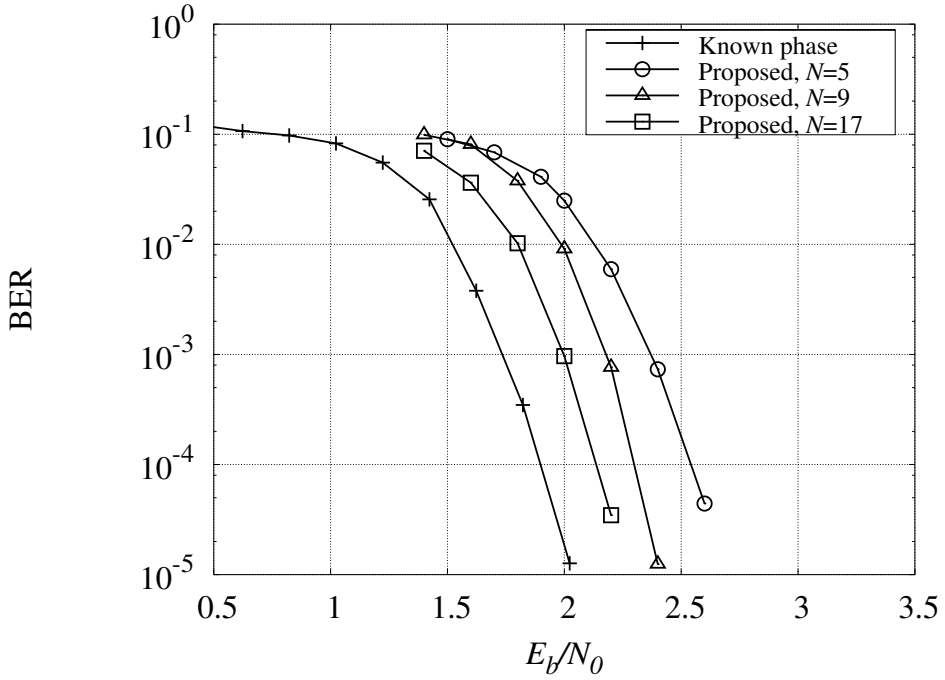

Figure 18: Performance of the Fourier algorithm for $\sigma_\Delta = 6$ degrees and different values of the number N of considered Fourier coefficients.

8.2 Canonical distributions

We now consider the algorithms described in Section 6.1.

In Fig. 18, the performance of the Fourier algorithm is shown for $\sigma_\Delta = 6$ degrees and different values of the number N of considered Fourier coefficients. Values of $N > 17$ are not considered since they do not produce any performance improvement. Therefore, the value of $N = 17$ can be considered as optimal and the relevant performance shown in the figure can be considered as the optimal performance for $\sigma_\Delta = 6$ degrees. The gap of about 0.2 dB with respect to the curve labeled "known phase" is only due to the loss in channel capacity for a time-varying channel phase.

In Fig. 19, the performance of the recursive algorithm is shown for different values of σ_Δ. As it can be observed from the figure, despite its very low complexity, this algorithm performs as well as the ideal "known phase" algorithm when the channel is time invariant. For a time-varying channel phase with $\sigma_\Delta = 6$ degrees, this algorithm has practically the same performance of the Fourier algorithm. Hence this algorithm has the best performance-complexity trade-off. It is also almost insensitive to the distributions of the pilot symbols as can be observed from Fig. 20. In this figure, in fact, two different pilot symbols distributions have been considered. In the first one, we have one pilot symbol in each block of 20 consecutive bits. In the second one we have a block of 20 pilot symbols in each block of 400 consecutive bits (hence the effective transmission rate is the same). Note that not all the considered algorithms have the same insensitivity to the

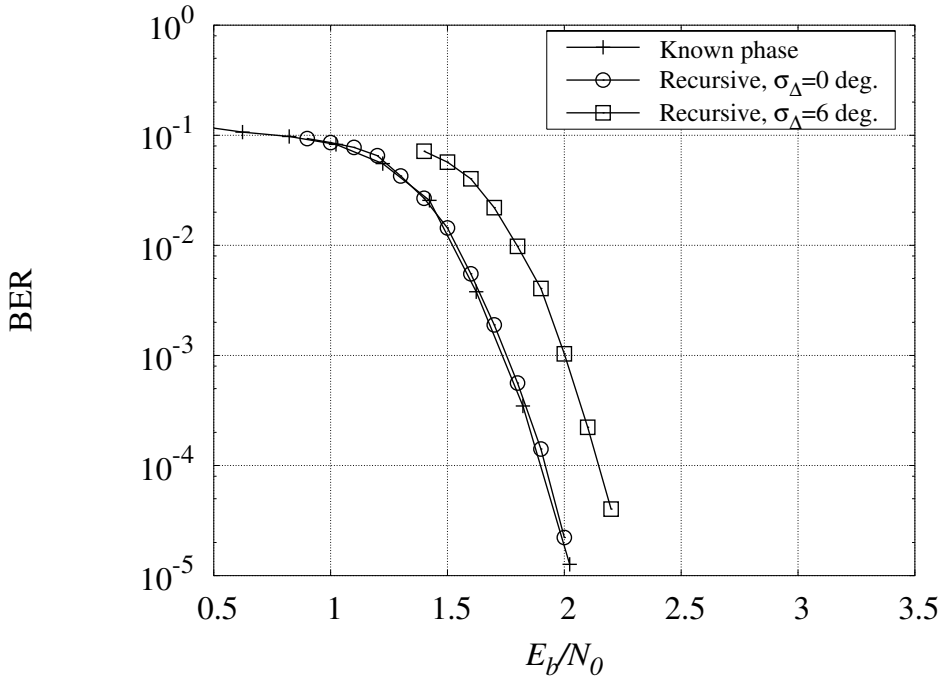

Figure 19: Performance of the recursive algorithm for different values of σ_Δ.

pilot symbols distribution and in general, the best distribution has to be designed *after* the choice of the detection algorithm.

The performance of the recursive algorithm has been also assessed for different phase models. In fact, in Fig. 21 the DVB-S2 compliant ESA model [36] has been also considered. This model is not Markovian and therefore the exact FG is not that in Fig. 6. Nevertheless, the recursive algorithm, which has been designed for the Wiener model, works well. From a practical point of view, in the case of the ESA model, in the forward and backward computations (90) and (91), parameter σ_Δ has been chosen equal to $\sigma_\Delta = 0.3$ degrees. In other words, the ESA model has been considered as it were a Wiener phase model with $\sigma_\Delta = 0.3$ degrees.

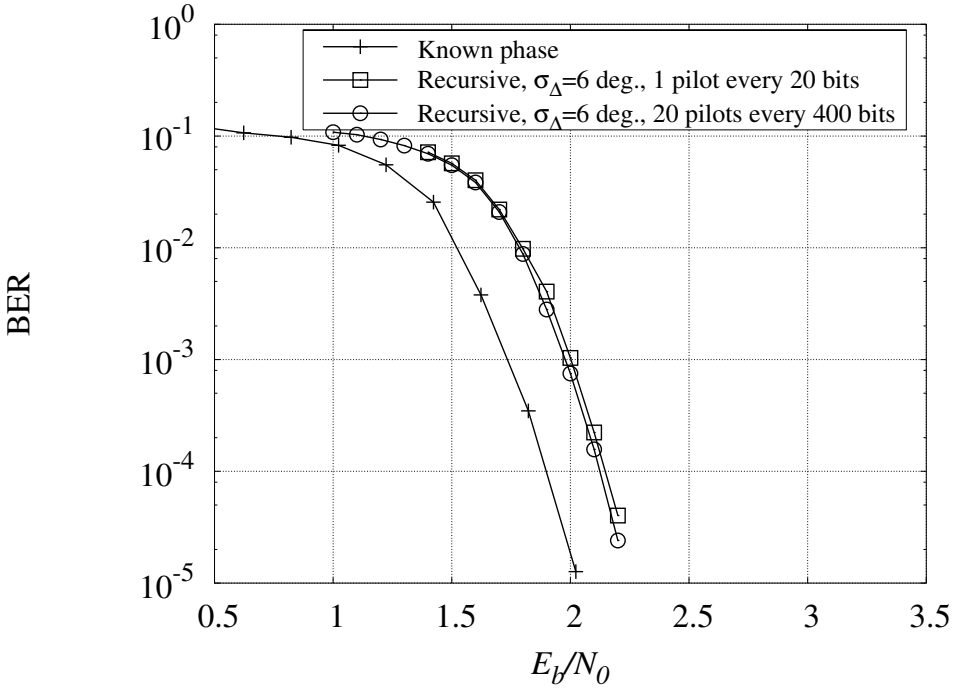

Figure 20: Performance of the recursive algorithm for $\sigma_\Delta = 6$ degrees and different distributions of the pilot symbols.

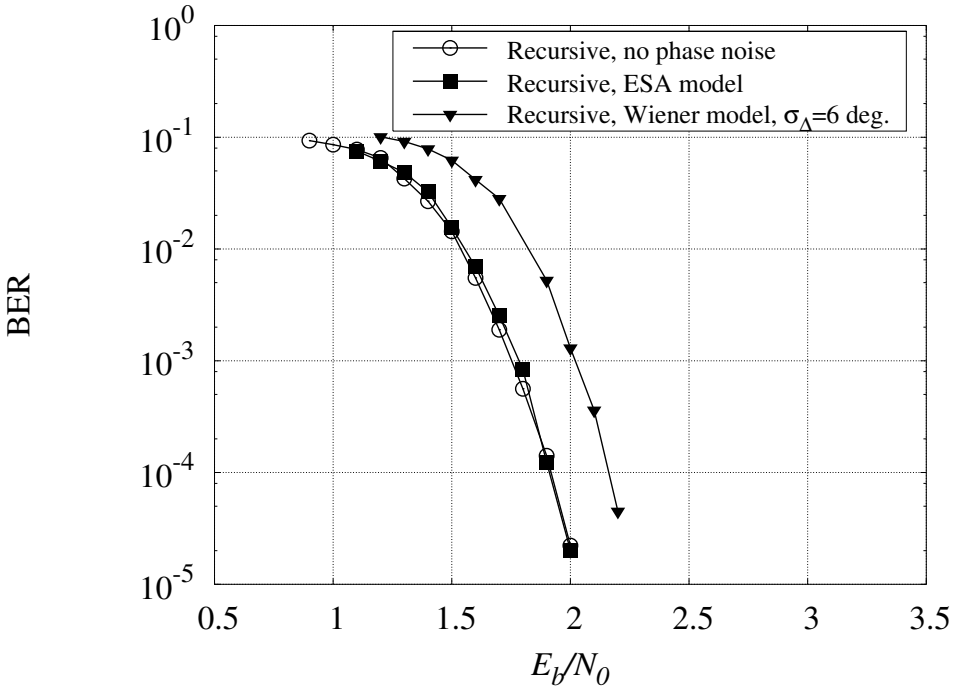

Figure 21: Performance of the recursive algorithm for different phase noise models.

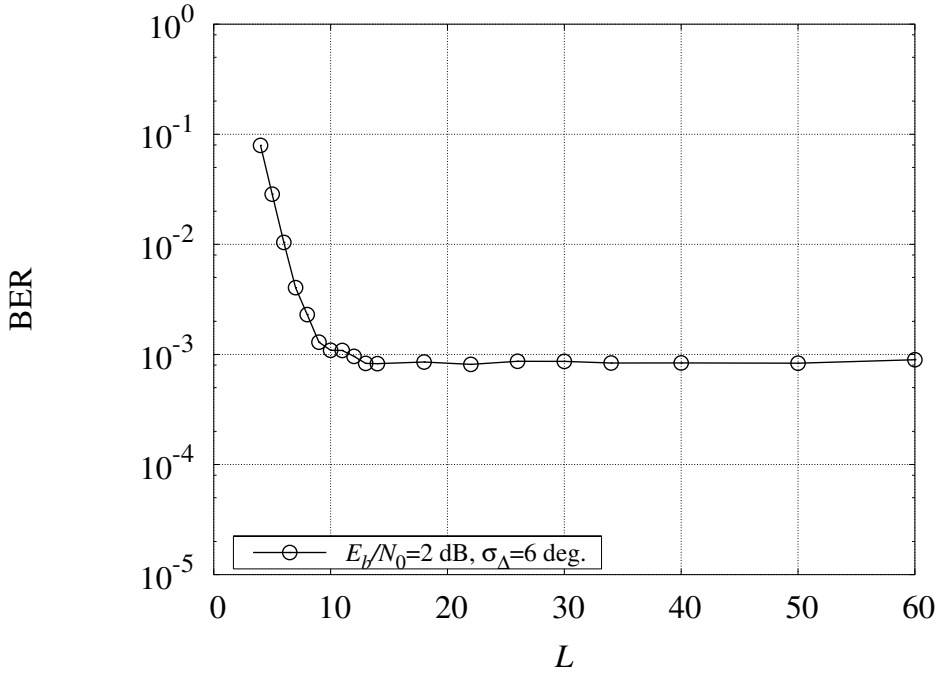

Figure 22: Performance versus L of the quantization-based algorithm for $\sigma_\Delta = 6$ degrees and $E_b/N_0 = 2$ dB.

8.3 Quantization of the channel parameters

Finally, we consider the performance of the quantization-based algorithm described in Section 6.2. In Fig. 22 we show the performance of the algorithm for different values of L, $\sigma_\Delta = 6$ degrees, and $E_b/N_0 = 2$ dB. It can be notice that a number of quantization levels $L = 16$ is practically optimal, i.e., no improvement can be observed for increasing values of L. The performance versus E_b/N_0 of this algorithm that can be regarded as a "practically optimal" benchmark, is shown in Fig. 23.

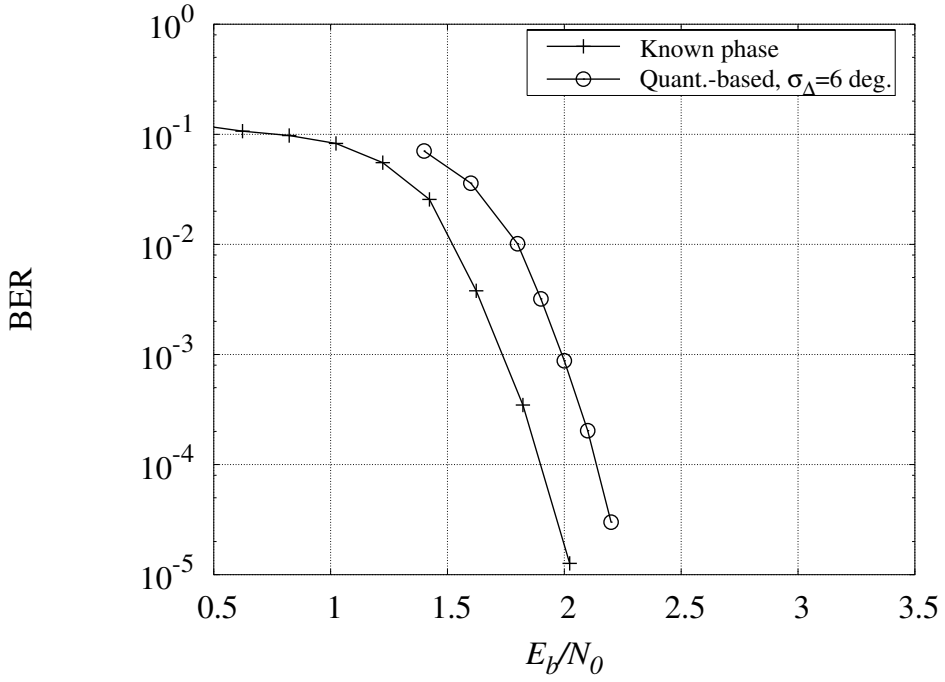

Figure 23: Performance of the quantization-based algorithm for $L = 16$ and $\sigma_\Delta = 6$ degrees.

8.4 EXIT Charts

Let the ensemble of LDPC codes be determined by the left and right degree sequences $\lambda(x) = \sum_i \lambda_i x^{i-1}$, $\rho(x) = \sum_i \rho_i x^{i-1}$. For later use, we also define the bitnode-wise degree sequence

$$\Lambda(x) = \sum_i \Lambda_i x^i = \frac{\int_0^x \lambda(z)dz}{\int_0^1 \lambda(z)dz} \tag{107}$$

such that $\Lambda_i = \frac{\lambda_i/i}{\sum_j \lambda_j/j}$ is the fraction of bitnodes of degree i.

Let X, Y be jointly distributed random variables with $X \in \{0, 1\}$, and let

$$\mathcal{L} = \log \frac{P(X=0|Y)}{P(X=1|Y)} \tag{108}$$

denote the posterior log-likelihood ratio for estimating X from Y. The mutual information $I(X;\mathcal{L})$, assuming that $\mathcal{L}$ is conditionally distributed as $\mathcal{N}(\mu, 2\mu)$ given $X = 0$, is given by the function

$$J(\mu) = 1 - \frac{1}{\sqrt{\pi}} \int_{-\infty}^{+\infty} e^{-z^2} \log_2 \left(1 + e^{-2\sqrt{\mu}z - \mu}\right) dz\,. \tag{109}$$

We are interested in approximating the *Extrinsic Information Transfer* (EXIT) function [37] of the LDPC decoder. We assume that the input messages (from the channel) are Gaussian log-likelihood ratios, distributed as $\mathcal{N}(\mu_0, 2\mu_0)$, where $\mu_0 = J^{-1}(I_{\rm in})$, and $I_{\rm in}$ is the mutual information of the channel (input to the decoder). We let the decoder run for an infinite number of iterations, until it reaches an equilibrium point. Then, the extrinsic output message for each bitnode of degree i is given by the sum of the i messages it receives from the i adjacent checknodes.

We assume that all messages exchanged over all edges of the graph are Gaussian distributed, and we denote by I_L the mutual information between a leftbound message (from checknode to bitnode) and the binary symbol associated to the destination bitnode, and by I_R the mutual information between a rightbound message (from bitnode to checknode) and the binary symbol associated to the departure bitnode.

Since the message output by a bitnode over an edge is the sum of all messages incoming over the other incident edges, from the Gaussian assumption we obtain

$$I_L = \sum_i \lambda_i J\left((i-1)J^{-1}(I_R) + J^{-1}(I_{\rm in})\right) \triangleq F_\lambda(I_L, I_{\rm in}) \tag{110}$$

The input-output relationship of the checknodes is much more complicated. However, it has been observed that the following *approximate duality* holds for a great variety of message distributions, including the Gaussian distribution [38]. A checknode with incident mutual information I_R and output mutual information I_L can be replaced by a bitnode with incident mutual information $1 - I_R$ and output mutual information $1 - I_L$. Eventually, we obtain the approximate

checknode transfer as

$$I_L = 1 - \sum_i \rho_i J\left((i-1)J^{-1}(1-I_R)\right) = 1 - F_\rho(1-I_R, 0) \tag{111}$$

Putting together (110) and (111), we obtain the fixed-point equation

$$I_R = F_\lambda(1 - F_\rho(1-I_R, 0), I_{\rm in}) \tag{112}$$

where I_R plays the role of the state of a dynamical system, and $I_{\rm in}$ is the external excitation.

Let $I_R^\star$ denote the unique solution of (112) in the interval $(0, 1)$. The output mutual information at the equilibrium (fixed-point of the state equation) is given by

$$I_{\rm out} = \sum_i \Lambda_i J(iJ^{-1}(I_R^\star)) = F_\Lambda(I_R^\star, 0) \tag{113}$$

This is the mutual information between the code symbols and their extrinsic messages after an infinite number of iterations. The LDPC decoder EXIT function is given implicitly by the fixed-point state equation (112) and by the output equation (113) and expresses the relationship between $I_{\rm in}$ and $I_{\rm out}$.

When the LDPC decoder work jointly with a channel parameter estimator, we can obtain the EXIT function of the channel estimator by Monte Carlo simulation and study the fixed points of the iterative joint decoding and channel estimation algorithm. For the channel parameter estimator, we model the distribution of the messages that the LDPC decoder sends to the parameter estimator about the code symbols (in both the FG-based Bayesian and the EM non-Bayesian approaches given above) as conditionally Gaussian distributed as $\mathcal{N}(\mu, 2\mu)$, where $\mu = J^{-1}(I_{\rm out})$. By letting $I_{\rm out}$ varying in $[0, 1]$, we can measure by Monte Carlo simulation the mutual information between the code symbols and the output of the virtual channel induced by a specific channel parameter estimator algorithm. This mutual information yields $I_{\rm in}$ at the LDPC decoder input.

By studing the intersections of the LDPC and the channel estimator EXIT functions, we obtain some (approximated) information about the SNR required by the overall system to converge to small bit-error probability, in the limit of large block length. We observed that the EXIT function approximation is able to predict accurately the threshold SNR. A specific example for the new recursive Bayesian algorithm is given in the following.

Fig. 24 shows the EXIT functions for the recursive algorithm and the $(3, 6)$ LDPC ensemble (the inverse EXIT function of the LDPC ensemble is shown, plotting $I_{\rm in}$ as function of $I_{\rm out}$) in the absence of pilot symbols. The phase noise has $\sigma_\Delta = 6$ degrees. We notice that in the absence of pilot symbols the algorithm cannot bootstrap, since all curves intersect at $I_{\rm out} = 0$. This clearly identifies the role of pilot symbols in iterative joint decoding and phase estimation: they serve as "doping" symbols in order to remove the zero fixed point. In principle, a very small number of pilot symbols is sufficient in order to make the algorithm start (see Fig. 25 which shows the case of 1 pilot every 100 bits). The EXIT curves of the recursive algorithm are shown for E_b/N_0 ranging from 1.0 dB to 2.4 dB. We observe that for E_b/N_0 roughly above 1.5 dB, the curves

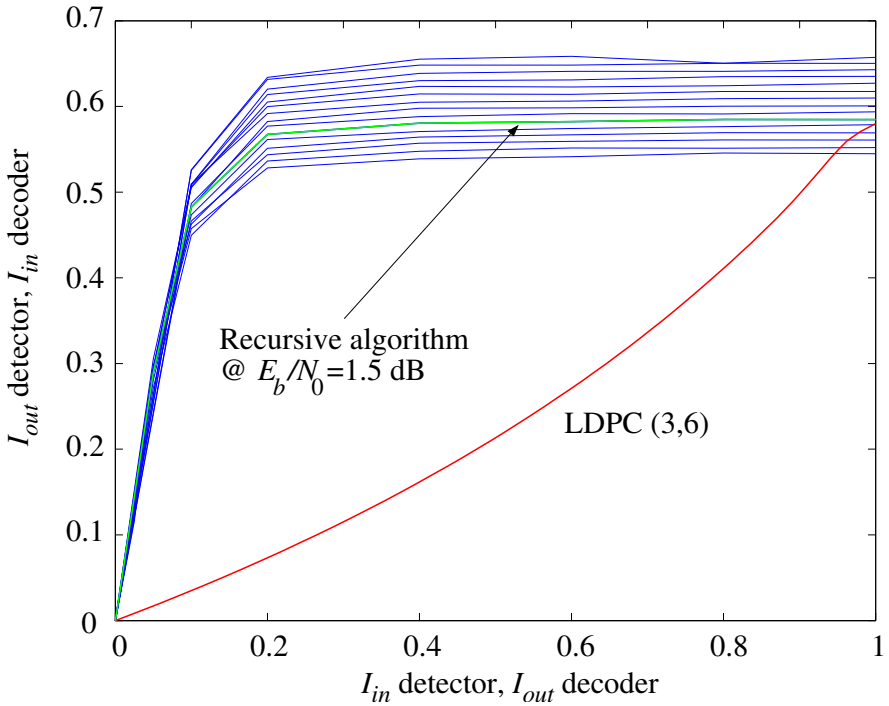

Figure 24: Exit chart for the recursive algorithm considering the $(3, 6)$ LDPC ensemble and $\sigma_\Delta = 6$ degrees. No pilot symbols.

do not intersect the LDPC curve any longer. Hence, the iterative decoding threshold for this noncoherent detector/decoder is around 1.5 dB. We notice that the iterative decoding threshold of the $(3, 6)$ ensemble over the coherent binary-input AWGN channel is ≈ 1.1 dB. This shows that an inherent degradation of 0.4 dB (plus the degradation due to the insertion of pilot symbols) is to be expected because of the presence of phase noise. This prediction is in full agreement with the best simulated performance of the algorithms in Fig. 19. In principle, this degradation could be lower if the LDPC code is designed jointly with the used detection algorithm.

9 Non-Bayesian algorithms: preview

In the non-Bayesian framework, we treat the channel parameters as deterministically unknown, i.e., no a priori probability distribution is assumed on the parameters. Generally speaking, decoding in the presence of unknown channel parameters falls in the class of composite Hypothesis testing problems [39]. A popular heuristics consists of the so-called Generalized Likelihood Ratio Test (GLRT) [39]. For each codeword $\mathbf{c} \in \mathcal{C}$ we compute the Maximum-Likelihood (ML) parameter estimate

$$\widehat{\boldsymbol{\theta}}(\mathbf{c}, \mathbf{r}) = \arg \max_{\boldsymbol{\theta}} P(\mathbf{c}) p(\mathbf{r}|\mathbf{c}, \boldsymbol{\theta}) \tag{114}$$

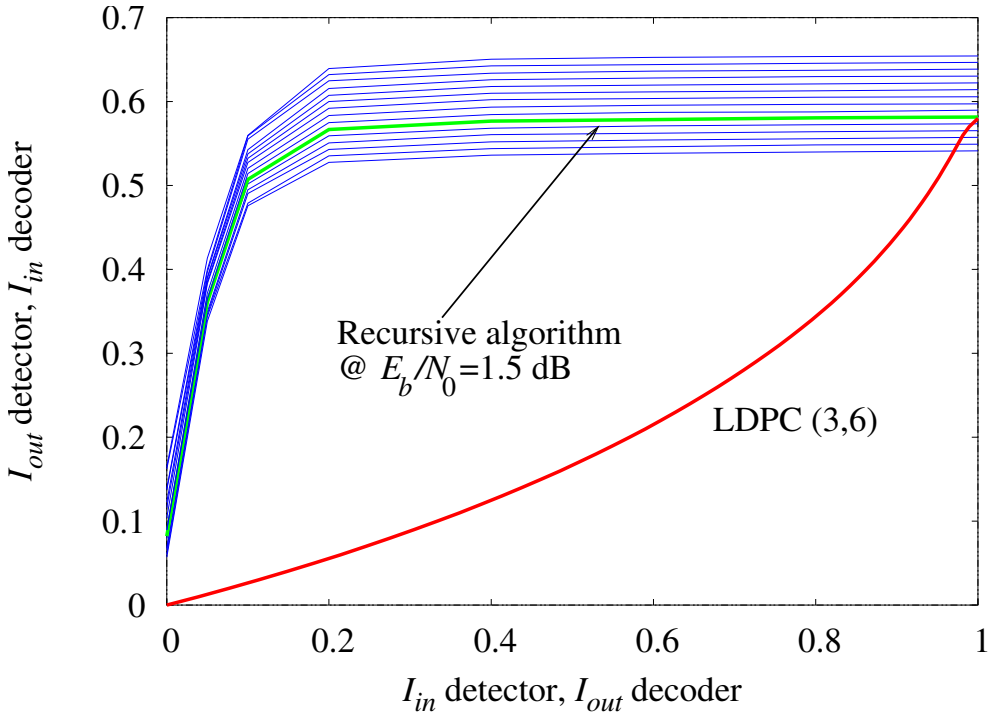

Figure 25: Exit chart for the recursive algorithm considering the $(3, 6)$ LDPC ensemble and $\sigma_\Delta = 6$ degrees. 1 pilot every 100 bits.

Then, we find the MAP codeword as

$$\widehat{\mathbf{c}} = \arg \max_{\mathbf{c}\in\mathcal{C}} P(\mathbf{c}) p(\mathbf{r}|\mathbf{c}, \widehat{\boldsymbol{\theta}}(\mathbf{c}, \mathbf{r})) \,. \tag{115}$$

The GLRT has not a general theoretical explanation, beyond the fact that it works well in many situations and that its asymptotics can be analyzed in some cases).

The GLRT can be seen as the solution of the augmented ML/MAP detection problem

$$(\widehat{\mathbf{c}}, \widehat{\boldsymbol{\theta}}) = \arg \max_{\mathbf{c}\in\mathcal{C},\boldsymbol{\theta}} \{P(\mathbf{c}) p(\mathbf{r}|\mathbf{c}, \boldsymbol{\theta})\} \tag{116}$$

where the parameter and the codeword are estimated/detected jointly, according to the ML principle. Notice that the explicit estimate of $\boldsymbol{\theta}$ is a by-product of the approach, not an ultimate goal.

A low-complexity approach to approximate the GLRT consists of replacing (114) by the simpler ML estimation

$$\widehat{\boldsymbol{\theta}}(\mathbf{r}) = \arg \max_{\boldsymbol{\theta}} p(\mathbf{r}|\boldsymbol{\theta}) \tag{117}$$

obtained by averaging out the code data, i.e., by letting

$$p(\mathbf{r}|\boldsymbol{\theta}) = \sum_{\mathbf{c}\in\mathcal{C}} P(\mathbf{c}) p(\mathbf{r}|\mathbf{c},\boldsymbol{\theta})$$

and then let

$$\widehat{\mathbf{c}} = \arg \max_{\mathbf{c}\in\mathcal{C}} P(\mathbf{c}) p(\mathbf{r}|\mathbf{c},\widehat{\boldsymbol{\theta}}(\mathbf{r}))\,. \tag{118}$$

Let us focus on the ML estimation problem (117). The likelihood equation for $\boldsymbol{\theta}$ is given by

$$\frac{\partial}{\partial\boldsymbol{\theta}} \log p(\mathbf{r}|\boldsymbol{\theta}) = 0\,. \tag{119}$$

We have

$$\begin{aligned}
\frac{\partial}{\partial\boldsymbol{\theta}} \log p(\mathbf{r}|\boldsymbol{\theta}) &= \frac{1}{p(\mathbf{r}|\boldsymbol{\theta})} \sum_{\mathbf{c}\in\mathcal{C}} P(\mathbf{c}) \frac{\partial}{\partial\boldsymbol{\theta}} p(\mathbf{r}|\mathbf{c},\boldsymbol{\theta}) \\
&= \frac{1}{p(\mathbf{r}|\boldsymbol{\theta})} \sum_{\mathbf{c}\in\mathcal{C}} P(\mathbf{c}) p(\mathbf{r}|\mathbf{c},\boldsymbol{\theta}) \frac{\partial}{\partial\boldsymbol{\theta}} \log p(\mathbf{r}|\mathbf{c},\boldsymbol{\theta}) \\
&= \sum_{\mathbf{c}\in\mathcal{C}} P(\mathbf{c}|\mathbf{r},\boldsymbol{\theta}) \frac{\partial}{\partial\boldsymbol{\theta}} \log p(\mathbf{r}|\mathbf{c},\boldsymbol{\theta})
\end{aligned} \tag{120}$$

where we have defined the codeword a posteriori probability (APP)

$$P(\mathbf{c}|\mathbf{r},\boldsymbol{\theta}) = \frac{P(\mathbf{c}) p(\mathbf{r}|\mathbf{c},\boldsymbol{\theta})}{p(\mathbf{r}|\boldsymbol{\theta})}\,. \tag{121}$$

The difficulty of solving (119) consists of the fact that in the last line of (120) the argument $\boldsymbol{\theta}$ appears in both the APP $P(\mathbf{c}|\mathbf{r},\boldsymbol{\theta})$ and in the conditional log-likelihood function $\log p(\mathbf{r}|\mathbf{c},\boldsymbol{\theta})$.

Instead of solving (117) directly, we can approximate the solution iteratively following the Expectation-Maximization (EM) approach [40]. For some initial value $\boldsymbol{\theta}^{(0)}$, for $\ell = 0, 1, 2, \ldots$ we define the E-step:

$$Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(\ell)}\right) = \sum_{\mathbf{c}\in\mathcal{C}} P\left(\mathbf{c}|\mathbf{r},\boldsymbol{\theta}^{(\ell)}\right) \log p(\mathbf{r}|\mathbf{c},\boldsymbol{\theta}) \tag{122}$$

and the M-step

$$\boldsymbol{\theta}^{(\ell+1)} = \arg \max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(\ell)}\right)\,. \tag{123}$$

General results (see [40] and references therein) state that if the EM recursion converges to a finite value, this is the ML solution of (117). In order to compute $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(\ell)}\right)$ in (122), we need the posterior probability distribution (121) for $\boldsymbol{\theta} = \boldsymbol{\theta}^{(\ell)}$. This can be provided by a MAP decoder for the code $\mathcal{C}$ with observation $\mathbf{r}$ and assuming the parameter value $\boldsymbol{\theta}^{(\ell)}$.

The above EM approach is particularly suited when it is possible to implement easily a

symbol-by-symbol MAP decoder for the code $\mathcal{C}$. In this case, the joint APP $P(\mathbf{c}|\mathbf{r}, \boldsymbol{\theta}^{(\ell)})$ needed in the E-step can be approximated as the product of the symbol-by-symbol (marginal) APPs $P(c_k|\mathbf{r}, \boldsymbol{\theta}^{(\ell)})$. Moreover, in several cases of interest the log-likelihood function $\log p(\mathbf{r}|\mathbf{c}, \boldsymbol{\theta})$ is a quadratic form in $\mathbf{c}$. Hence, only the first and second moments of the posterior distribution are needed at each step. If the quadratic form does not contain cross-terms (i.e., products $c_k c_j^*$ for $k \neq j$), then the a posteriori marginals are sufficient to evaluate exactly $Q\left(\boldsymbol{\theta}, \boldsymbol{\theta}^{(\ell)}\right)$ exactly, via the a posteriori mean and the a posteriori second moment of the code symbols, defined as

$$\begin{aligned} \alpha_k^{(\ell)} &= E[c_k|\mathbf{r}, \boldsymbol{\theta}^{(\ell)}] \\ \beta_k^{(\ell)} &= E[|c_k|^2|\mathbf{r}, \boldsymbol{\theta}^{(\ell)}] \end{aligned} \tag{124}$$

Another important remark is the following: the iterative EM algorithm given by (122) and (123) requires the re-evaluation of the joint APP $P(\mathbf{c}|\mathbf{r}, \boldsymbol{\theta}^{(\ell)})$ (or of its marginals $P(c_k|\mathbf{r}, \boldsymbol{\theta}^{(\ell)})$) at each iteration ℓ. Since $P(\mathbf{c}|\mathbf{r}, \boldsymbol{\theta}) \doteq P(\mathbf{c})p(\mathbf{r}|\mathbf{c}, \boldsymbol{\theta})$, the MAP decision (118) is then given by

$$\widehat{\mathbf{c}} = \underset{\mathbf{c}\in\mathcal{C}}{\operatorname{argmax}}\, P\left(\mathbf{c}|\mathbf{r}, \widehat{\boldsymbol{\theta}}(\mathbf{r})\right) . \tag{125}$$

Alternatively, if the decoder is symbol-by-symbol MAP, this can be replaced by the symbol-by-symbol decision

$$\widehat{c}_k = \arg \max_{c_k\in\mathcal{X}} P\left(c_k|\mathbf{r}, \widehat{\boldsymbol{\theta}}(\mathbf{r})\right) \tag{126}$$

where in both cases $\widehat{\boldsymbol{\theta}}(\mathbf{r})$ is the parameter estimate after a suitable number of EM iterations. This makes the EM approach particularly suited to the case where $\mathcal{C}$ is a Turbo or an LDPC code, since for this class of codes the symbol-by-symbol APPs necessary to the EM recursion and for the final MAP decision are computed easily, along the iterations of the EM detector.

10 Noncoherent channel: the standard EM algorithm

The iterative EM approach has been used in [41] in the context of phase estimation in multiuser CDMA and in [5] for phase estimation in turbo-coded transmission, and it is illustrated in general in the tutorial work [7] (see also the references in [7, 41]).

For the case of unknown time-invariant phase ($\theta_k = \theta$), the EM approach has proved to provide excellent performance with LDPC and turbo codes [5]. In brief, we have,

$$\log p(\mathbf{r}|\mathbf{c}, \theta) \doteq 2\mathrm{Re}\left\{e^{-j\theta}\mathbf{c}^H\mathbf{r}\right\} - |\mathbf{c}|^2 . \tag{127}$$

By taking the conditional expectation of the above log-likelihood function with respect to $\mathbf{c}$,

given $\mathbf{r}$ and $\theta = \theta^{(\ell)}$, we obtain

$$\begin{aligned} Q\left(\theta, \theta^{(\ell)}\right) &\doteq 2\mathrm{Re}\left\{e^{-j\theta}\sum_k (\alpha_k^{(\ell)})^* r_k\right\} - \sum_{k=0}^{K-1}\beta_k^{(\ell)} \\ &= 2\cos\left(\arg\left\{\sum_k (\alpha_k^{(\ell)})^* r_k\right\} - \theta\right)\left|\sum_k (\alpha_k^{(\ell)})^* r_k\right| - \sum_{k=0}^{K-1}\beta_k^{(\ell)} \end{aligned} \tag{128}$$

where we have defined the conditional mean and second moments of the data symbols in (124) explicitly by

$$\alpha_k^{\ell)} = E[c_k|\mathbf{r}, \theta^{(\ell)}] = \sum_{a\in\mathcal{X}} aP(c_k = a|\mathbf{r}, \theta^{(\ell)}) \tag{129}$$

and by

$$\beta_k^{(\ell)} = E[|c_k|^2|\mathbf{r}, \theta^{(\ell)}] = \sum_{a\in\mathcal{X}} |a|^2 P(c_k = a|\mathbf{r}, \theta^{(\ell)})\,. \tag{130}$$

The M-step yields

$$\theta^{(\ell+1)} = \arg\left\{\sum_k (\alpha_k^{(\ell)})^* r_k\right\}\,. \tag{131}$$

In order to compute the symbol posterior probabilities $P(c_k|\mathbf{r}, \boldsymbol{\theta}^{(\ell)})$ at each iteration, the detector produces the symbol-by-symbol observables

$$\begin{aligned} P(c_k|r_k, \boldsymbol{\theta}^{(\ell)}) &\doteq E\left[p(r_k|c_k, \theta)|\,\theta = \theta^{(\ell)}\right] \\ &\doteq \int_{-\pi}^{\pi} \exp\left(-\frac{1}{\sigma^2}\left|r_k - e^{j\theta}c_k\right|^2\right) p(\theta|\theta^{\ell)})d\theta \end{aligned} \tag{132}$$

The problem with the above decoding metric is that we have to guess the conditional distribution of $p(\theta|\theta^{\ell)})$. The EM approach [5] assumes $p(\theta|\theta^{\ell)}) = \delta(\theta - \theta^{(\ell)})$, i.e.,

$$\begin{aligned} P(c_k|r_k, \boldsymbol{\theta}^{(\ell)}) &\doteq \exp\left(-\frac{1}{\sigma^2}\left|r_k - e^{j\theta^{(\ell)}}c_k\right|^2\right) \\ &\doteq \exp\left(\frac{1}{\sigma^2}\left(2\mathrm{Re}\left\{r_k e^{-j\theta^{(\ell)}}c_k^*\right\} - |c_k|^2\right)\right) \end{aligned} \tag{133}$$

11 Extensions to time-varying phase

Generally speaking, non-Bayesian estimation methods are suited to parameter estimation, but not to random signal estimation. Unfortunately, the case of a random time-varying phase (phase-noise) falls into the class of signal estimation. Hence, we have to find an efficient parameterization of the phase noise process such that an arbitrary realization of the phase $\{\theta_k\}$ can be expressed through a small number of deterministically unknown variables (parameters). This parameterization must be driven by some a priori knowledge on the "typical behavior" of the

process we want to estimate.

The already introduced *windowed Luise algorithm* (denoted here as Sliding-Window EM (SW-EM)) is based on the a priori knowledge that the phase noise process is lowpass, i.e., phase variations are slow in comparison to the symbol rate. Hence, it assumes that the phase is constant over a window of C symbols, and estimates the phase at time k by applying (128) and (131) over a window of size C centered around time k.

In the reminder of this section we outline a new non-Bayesian EM-based algorithm that takes into account some more accurate a priori knowledge of the phase time variations. We seek a useful parameterization of the phase noise signal $\boldsymbol{\theta} = (\theta_0, \ldots, \theta_{K-1})^T$. To this purpose, we define the complex phasor process $h_k = e^{j\theta_k}$. Assuming the discrete-time Wiener process model introduced before, the autocorrelation function of h_k is given by

$$R_h(k) = E[h_{i+k} h_i^*] = \exp(-|k|\sigma_\Delta^2/2)\,. \tag{134}$$

This yields an exponentially correlated (non-Gaussian) process with parameter $\rho = \exp(-\sigma_\Delta^2/2)$ and power spectral density

$$S_h(f) = \frac{1-\rho^2}{1 - 2\rho\cos(2\pi f) + \rho^2}\,.$$

Knowing the autocorrelation function of $\{h_k\}$, we can parameterize $\mathbf{h} \stackrel{\Delta}{=} (h_0, \ldots, h_{K-1})^T$ by using the Karhunen-Loeve expansion $\mathbf{h} = \mathbf{U}\boldsymbol{\Lambda}^{1/2}\mathbf{w}$, where we define the covariance matrix $\mathbf{R}_h = E[\mathbf{h}\mathbf{h}^H] = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^H$, where $\mathbf{U}$ is unitary and $\boldsymbol{\Lambda}$ is diagonal with non-negative non-increasing diagonal elements λ_i.

In the non-Bayesian framework we are not concerned with the probability distribution of $\mathbf{w}$. On the contrary, we treat $\mathbf{w}$ as a deterministically unknown vector of parameters. The advantage of the K-L parameterization is that if the phase noise rate of variation (roughly speaking, the bandwidth of the spectrum $S_h(f)$) is much less than the symbol rate, we can estimate a small number $m \ll K$ of parameters $\mathbf{w}$ and then use the K-L basis (the columns of $\mathbf{U}$) to interpolate the time-varying phase.

We define $\mathbf{w}_m$ as the first (most significant) m K-L coefficients, i.e., we *assume* the model

$$\mathbf{h} = \mathbf{U}_m \boldsymbol{\Lambda}_m^{1/2} \mathbf{w}_m$$

where $\mathbf{U}_m$ is a $K \times m$ matrix formed by the first m columns of $\mathbf{U}$ and $\boldsymbol{\Lambda}_m$ as is the $m \times m$ left-upper submatrix of $\boldsymbol{\Lambda}$. With this parameterization, we write the channel model as

$$\mathbf{r} = \mathbf{C}\mathbf{U}_m \boldsymbol{\Lambda}_m^{1/2} \mathbf{w}_m + \mathbf{n} \tag{135}$$

where $\mathbf{C} = \mathrm{diag}(c_0, \ldots, c_{K-1})$. The joint log-likelihood function is given by

$$\begin{aligned} \log p(\mathbf{r}|\mathbf{c}, \boldsymbol{\theta}) &\doteq -\left|\mathbf{r} - \mathbf{C}\mathbf{U}_m \boldsymbol{\Lambda}_m^{1/2} \mathbf{w}_m\right|^2 \\ &\doteq 2\mathrm{Re}\left\{\mathbf{w}_m^H \boldsymbol{\Lambda}_m^{1/2} \mathbf{U}_m^H \mathbf{C}^H \mathbf{r}\right\} - \mathbf{w}_m^H \boldsymbol{\Lambda}_m^{1/2} \mathbf{U}_m^H \mathbf{C}^H \mathbf{C} \mathbf{U}_m \boldsymbol{\Lambda}_m^{1/2} \mathbf{w}_m\,. \end{aligned} \tag{136}$$

By taking the conditional expectation of the above log-likelihood function with respect to $\mathbf{c}$, given $\mathbf{r}$ and $\mathbf{w}_m = \mathbf{w}_m^{(\ell)}$, we obtain

$$Q\left(\mathbf{w}_m, \mathbf{w}_m^{(\ell)}\right) \doteq 2\mathrm{Re}\left\{\mathbf{w}_m^H \mathbf{\Lambda}_m^{1/2} \mathbf{U}_m^H (\boldsymbol{\alpha}^{(\ell)})^H \mathbf{r}\right\} - \mathbf{w}_m^H \mathbf{\Lambda}_m^{1/2} \mathbf{U}_m^H \boldsymbol{\beta}^{(\ell)} \mathbf{U}_m \mathbf{\Lambda}_m^{1/2} \mathbf{w}_m \quad (137)$$

where we have defined $\boldsymbol{\alpha}^{(\ell)} = \mathrm{diag}(\alpha_0^{(\ell)}, \ldots, \alpha_{K-1}^{(\ell)})$ and $\boldsymbol{\beta}^{(\ell)} = \mathrm{diag}(\beta_0^{(\ell)}, \ldots, \beta_{K-1}^{(\ell)})$.

The M-step consists of the maximization of the quadratic form (137) with respect to $\mathbf{w}_m$. The solution is easily obtained as

$$\mathbf{w}_m^{(\ell+1)} = \mathbf{\Lambda}_m^{-1/2} \left[\mathbf{U}_m^H \boldsymbol{\beta}^{(\ell)} \mathbf{U}_m\right]^{-1} \mathbf{U}_m^H (\boldsymbol{\alpha}^{(\ell)})^H \mathbf{r}\,. \quad (138)$$

In the important case of PSK modulation, $\boldsymbol{\beta}^{(\ell)} = \mathbf{I}_K$ and since $\mathbf{U}_m^H \mathbf{U}_m = \mathbf{I}_m$ we obtain directly the estimate of $\mathbf{h}$ at the EM step $\ell + 1$

$$\mathbf{h}^{(\ell+1)} = \mathbf{U}_m \mathbf{U}_m^H (\boldsymbol{\alpha}^{(\ell)})^H \mathbf{r}\,. \quad (139)$$

This estimator has the following nice interpretation: first, the effect of the modulation symbols is removed from the received signal by componentwise premultiplication by α_k^*; then, the resulting modulation-free signal is orthogonally projected on the space spanned by the phase noise process via the orthogonal projector of rank m given by $\mathbf{U}_m \mathbf{U}_m^H$.

Remark: choice of the KL model order. In the above estimator we have the degree of freedom of choosing the order m of the KL model. Assuming that all symbols are perfectly known, i.e., $\alpha_k^{(\ell)} = c_k$, it is not difficult to show that the overall estimation mean square error of the phasor process $\mathbf{h}$ is given by

$$\mathrm{MSE} = E[|\mathbf{h} - \widehat{\mathbf{h}}|^2] = m\sigma^2 + \sum_{i=m+1}^{K} \lambda_i \quad (140)$$

The first term in (140) is due to the noise, and increases linearly with m. The second term in (140) is due to the truncation, i.e., to the fact that we have used only the first most significant m KL eigenfunctions to represent the phasor process. The above MSE is minimized by choosing

$$m = \max\left\{1 \le i \le K \;\; : \;\; \lambda_i > \sigma^2\right\}$$

In other words, we explicitly estimate only the KL eigenmodes whose energy λ_i is above the noise level σ^2. This choice, of course, is optimal only if all symbols are perfectly known, i.e., in the convergence condition of the overall EM-KL iterative detector/decoder to small symbol error probability. At the beginning, for small iteration index ℓ or small SNR, a lower value of m might be beneficial.

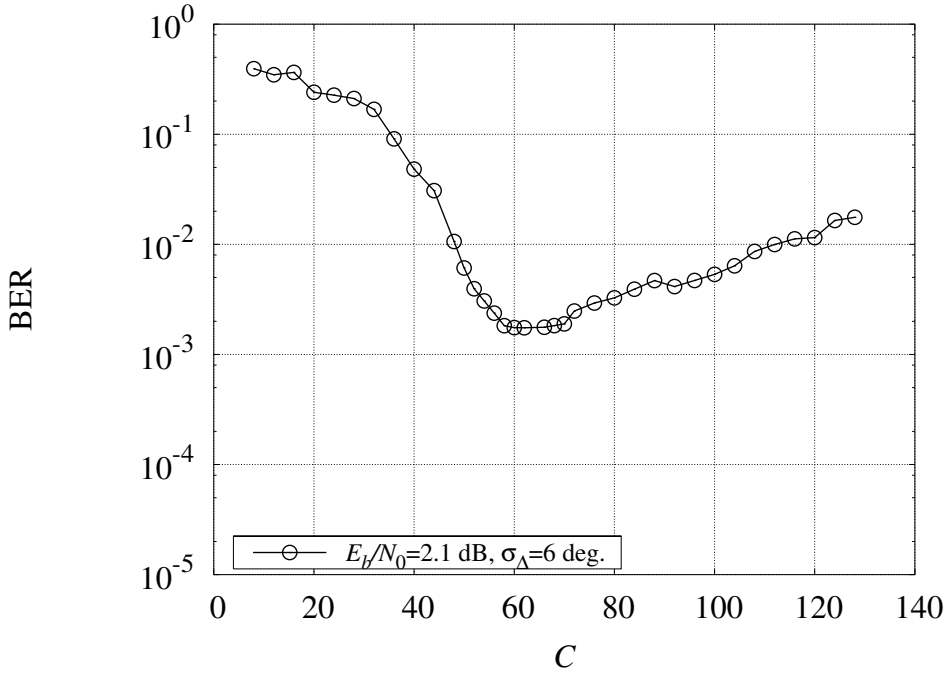

Figure 26: Windowed Luise algorithm versus C for $\sigma_\Delta = 6$ degrees and $E_b/N_0 = 2.1$ dB.

11.1 Performance comparison

We now compare all the described detection algorithms for a phase noise standard deviation of $\sigma_\Delta = 6$ degrees. The performance of the algorithm described in [10] (Anastasopoulos algorithm) and that of the windowed-Luise algorithm, is also shown for comparison. In the latter case, the value of C has been optimized by means of computer simulations. As shown in Fig. 26, for $\sigma_\Delta = 6$ degrees and $E_b/N_0 = 2.1$ dB, the optimal value is $C = 60$.

Quantized-based, Fourier and recursive algorithms exhibit a practically optimal performance. Among them, the recursive algorithm, with its very low complexity, represents the best candidate for this detection scenario. The KL-EM algorithm outperforms generally the windowed Luise algorithm (EM-SW) especially for high SNR (low BER). In our simulations, the KL-EM algorithm was simulated with $m = 5$ and applied over windows of 100 symbols, shifted by 50 symbols in order to maintain phase continuity. The complexity of this algorithm is roughly equivalent to the windowed Luise algorithm. In any case, it is apparent that both in terms of complexity and performance the Bayesian recursive algorithm is preferable.

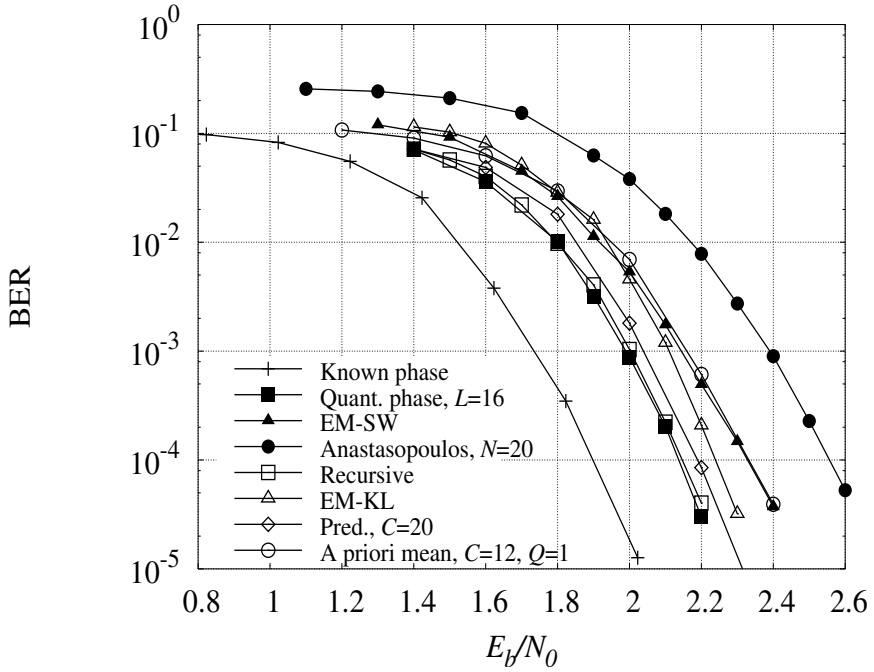

Figure 27: Performance comparison among all the considered algorithm for $\sigma_\Delta = 6$ degrees.

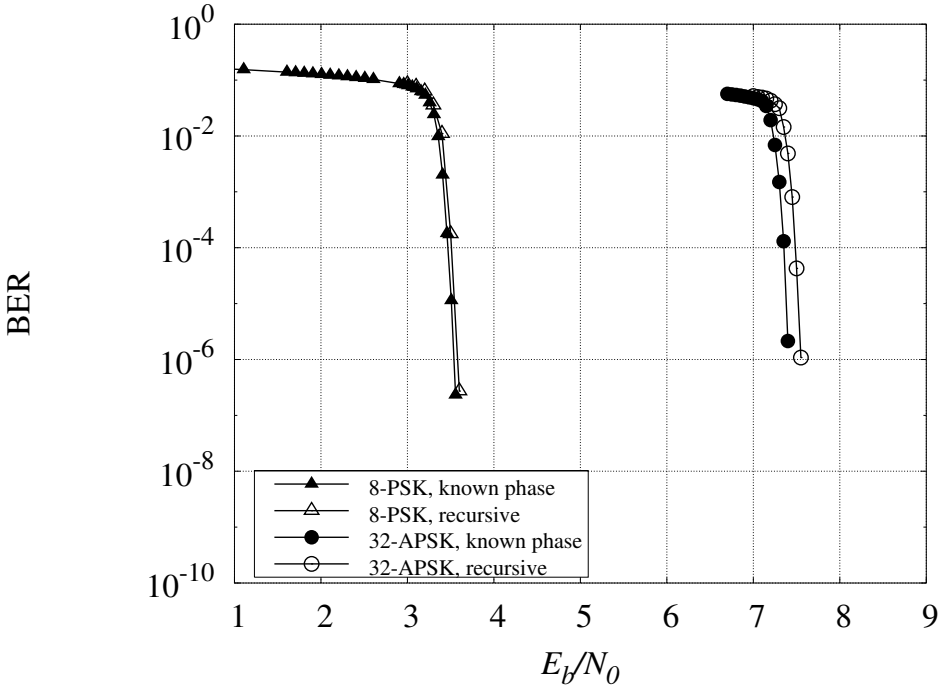

Figure 28: Performance of the recursive algorithm in the case of 8-PSK and 32-APSK modulations.

12 Numerical results for the DVB-S2 system

We now consider the application of the recursive algorithm in the DVB-S2 system. We consider two standardized LDPC codes with codewords of length 64800 [42]. The first one has rate 2/3 and is mapped on an 8-PSK modulation. The second one has rate 4/5 and is mapped on a 32-APSK modulation. A maximum number of 50 iterations is considered and 36 pilot symbols every 1476 symbols are included as described in the standard [42]. The above mentioned phase noise ESA model is considered. In Fig. 28 the relevant performance is shown. The curves labeled "known phase" have been artificially translated to take into account the effective transmission rate due to the insertion of pilots. The loss due to the presence of phase noise is less than 0.1 dB in both cases as can be also observed from Fig. 29 and 30. In Fig. 29, for the 8-PSK modulation we also show the performance in the case of absence of joint detection/decoding when a carrier recovery algorithm is used based on a data-aided ML phase estimate performed on each pilot field, and a linear interpolation of these estimates. Notice that a further improvement in performance may be obtained if the maximum number of iterations is not limited to 50. In Fig. 28, we also show the performance that can be obtained in the case of 32-APSK when a different mapping is considered. This mapping is obtained from the standardized mapping by a simple reversal of the label bits. A performance improvement of 0.2 dB could be obtained (obviously not only for the known phase curve but also for the recursive algorithm) and this fact should be considered in the final implementation.

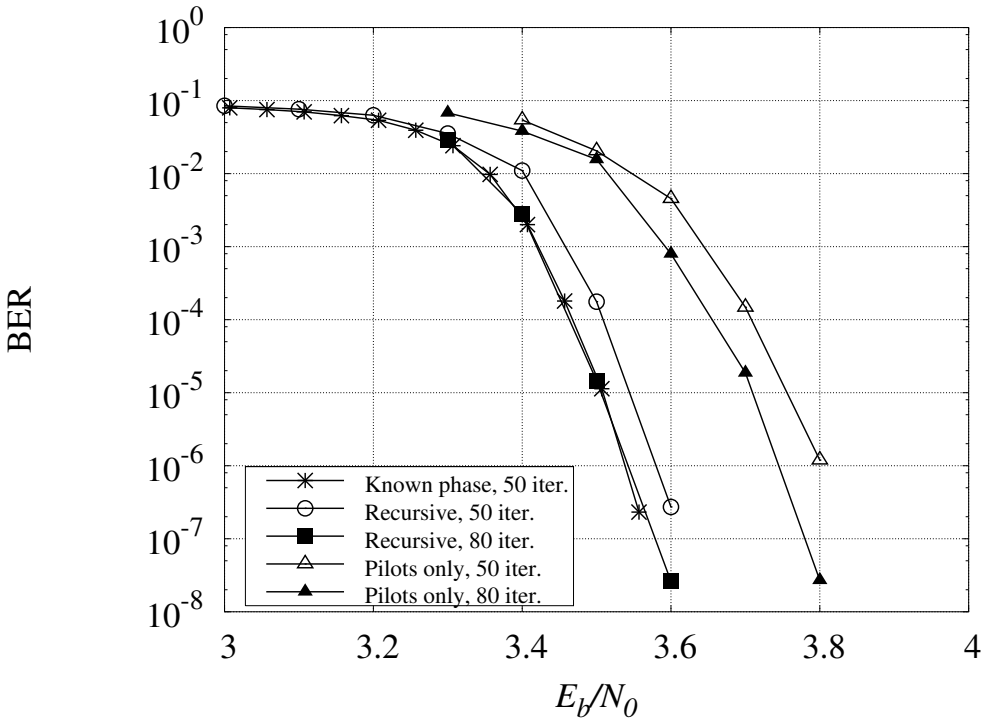

Figure 29: Performance of the recursive algorithm in the case of an 8-PSK modulation.

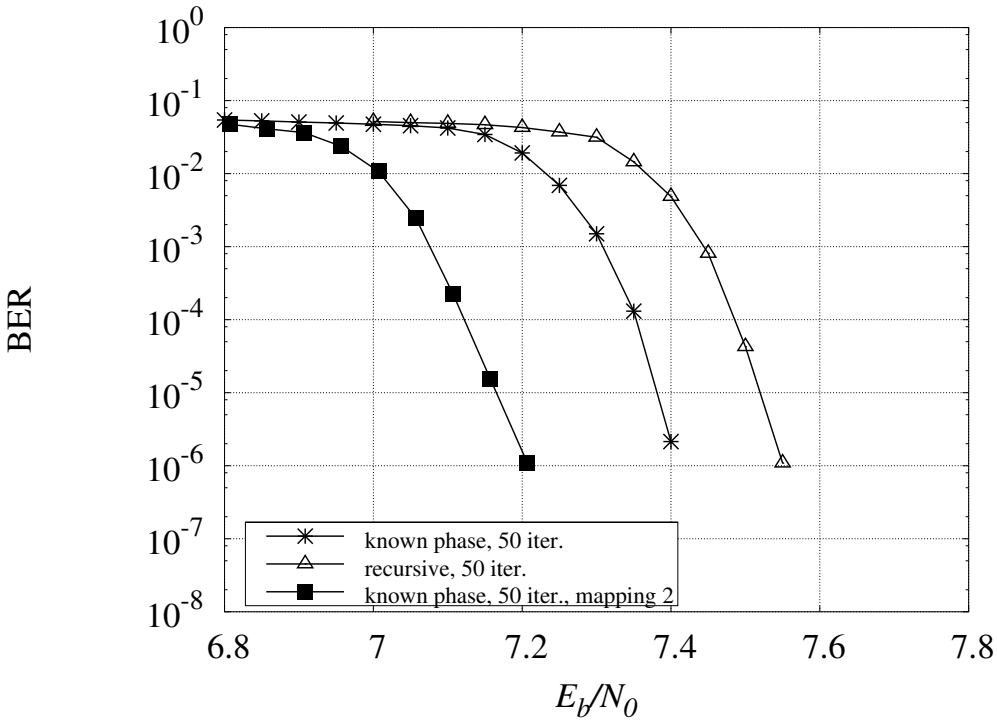

Figure 30: Performance of the recursive algorithm in the case of a 32-APSK modulation.

13 Conclusions

In this report, the problem of joint detection and decoding of LDPC codes transmitted over an AWGN channel affected by phase noise has been considered. Bayesian and non-Bayesian approaches have been considered. The Bayesian approaches are essentially based on various applications of the factor graph and of the sum-product algorithm. Non-Bayesian approaches are essentially based on the approximation of the GLRT by the EM method.

In a first Bayesian approach, the expectation over channel parameters is a priori performed. In this case, for time-invariant noncoherent channels two algorithms have been derived. The first one has a complexity growing exponentially with the phase memory C. Hence, complexity reduction is necessary for large values of phase memory which allow to approach the performance of an ideal coherent receiver. On the contrary, the second one has a complexity which grows linearly with C. These proposed algorithms exhibit a high robustness in the presence of a time-varying channel phase. This robustness can be increased by using a linear predictive approach in which the statistic description of the phase process is taken into account.

In a second Bayesian approach, the average over channel parameters is performed by the sum-product algorithm since channel parameters are explicitly represented in the overall factor graph. To overcome the problem of an exchange of messages in the graph representing the probability density functions of continuous random variables, we considered two methods, the quantization of channel parameters and the use of canonical distributions. In the latter case, the above mentioned probability density functions are represented by means of a finite number of parameters which become the messages to exchange.

The non-Bayesian EM methods have all the problem of "guessing" an efficient parameterization of the phase noise process, since non-Bayesian parameter estimation techniques are generally not suited to track time-varying processes and need ad-hoc modifications. We have reviewed the sliding-window EM approach and introduced a new parameterization based on an orthogonal basis expansion given by the KL decomposition. The KL estimator generally outperform the sliding-window estimator, but it is generally inferior to the best Bayesian methods.

In conclusions, we recommend the simple Bayesian recursive algorithm due to its practically optimal performance and extremely low complexity.

Appendix A

In this appendix, we will show that the function

$$f(y) = \frac{1}{\sqrt{2\pi\sigma_\Delta^2}} \int_0^{2\pi} e^{\mathrm{Re}[ze^{-jx}]} e^{-\frac{(x-y)^2}{2\sigma_\Delta^2}}\, dx = \frac{1}{\sqrt{2\pi\sigma_\Delta^2}} \int_{-\pi}^{\pi} e^{\mathrm{Re}[ze^{-j(x+y)}]} e^{-\frac{(x)^2}{2\sigma_\Delta^2}}\, dx$$

where z is a complex number and x and y are real numbers, can be well approximated, discarding irrelevant multiplicative factors, by the function $g(y) = e^{\gamma(\sigma_\Delta^2, |z|)\mathrm{Re}[ze^{-jy}]}$, where $\gamma(\sigma_\Delta^2, |z|)$ is a real function of $|z|$ and σ_Δ^2.

For $|\sigma_\Delta| \ll 1$, since both functions are periodic of period 2π, for given values of $|z|$ and

σ_Δ^2, we may compute in a period the mean square error $\epsilon(\gamma, |z|, \sigma_\Delta)$ between these functions, normalized to unit area. We chose the real function $\gamma(\sigma_\Delta^2, |z|)$ as the value of γ which minimizes the mean square error and is represented in Fig. 9 for different values of σ_Δ.

A good closed-form approximation is represented by the expression

$$\gamma(\sigma_\Delta^2, |z|) = \frac{1}{1+\sigma_\Delta^2|z|}\,.$$

This result can be derived by using the following approximation which holds for large values of $x \in \mathbb{R}^+$ (in practice $x > 5$)

$$\frac{e^{a\cos(x-y)}}{2\pi \mathrm{I}_0(a)} \simeq \frac{1}{\sqrt{2\pi/a}} e^{-\frac{a}{2}(x-y)^2} = g(x, y, \frac{1}{a}) \tag{141}$$

having denoted by $g(x, y, \frac{1}{a})$ a Gaussian pdf with mean x and variance $\frac{1}{a}$. In fact, for sufficiently large values of x, the Tikhonov pdf $\frac{e^{a\cos(x-y)}}{2\pi \mathrm{I}_0(a)}$ has its support in a small interval around y. Hence, by using a second-order Taylor expansion, we have $\cos(x-y) \simeq 1 - \frac{(x-y)^2}{2}$. A normalization constant has been further added to obtain a pdf. Parameter γ can be derived by using the approximation (141). In fact, we may express

$$\begin{aligned}
f(y) &= \frac{1}{\sqrt{2\pi\sigma_\Delta^2}} \int_0^{2\pi} e^{\mathrm{Re}[ze^{-jx}]} e^{-\frac{(x-y)^2}{2\sigma_\Delta^2}}\, dx \\
&\overset{(a)}{\simeq} \int_{-\infty}^{\infty} e^{\mathrm{Re}[ze^{-jx}]} g(y, x, \sigma_\Delta^2)\, dx \\
&\overset{(b)}{\simeq} 2\pi \mathrm{I}_0(|z|) \int_{-\infty}^{\infty} g(x, \phi(z), \frac{1}{|z|}) g(y, x, \sigma_\Delta^2)\, dx \\
&\overset{(c)}{\doteq} g(y, \phi(z), \frac{1}{|z|} + \sigma_\Delta^2) \\
&\overset{(d)}{\simeq} \frac{1}{2\pi \mathrm{I}_0(\frac{|z|}{1+\sigma_\Delta^2|z|})} \exp\left\{ \frac{1}{1+\sigma_\Delta^2|z|} \mathrm{Re}[ze^{-jy}] \right\}.
\end{aligned} \tag{142}$$

Step (a) derives from the observation that for $\sigma_\Delta \ll 1$ the function $e^{-\frac{(x-y)^2}{2\sigma_\Delta^2}}$ has its support in a small interval around y. Steps (b) and (d) derive from the approximation (141). Finally, step (c) derives from the equality

$$\int_{-\infty}^{\infty} g(x, \eta_1, \sigma_1^2) g(y, \eta_2 x, \sigma_2^2)\, dx = g(y, \eta_1\eta_2, \eta_2^2\sigma_1^2 + \sigma_2^2)\,.$$

Hence

$$\gamma(\sigma_\Delta^2, |z|) = \frac{1}{1+\sigma_\Delta^2|z|}\,.$$

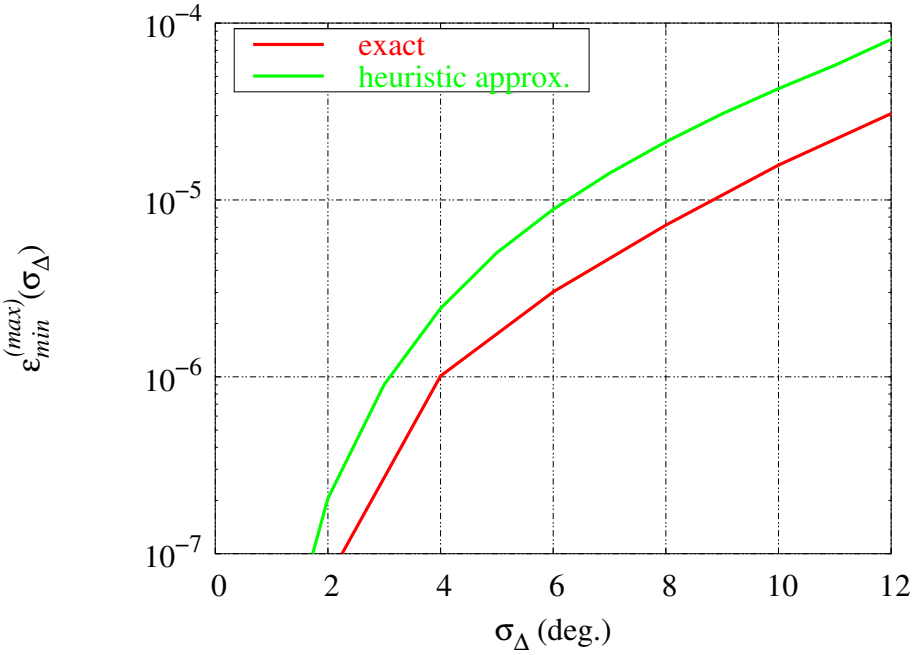

Figure 31: Function $\epsilon_{min}^{(max)}(\sigma_\Delta)$.

We now consider the minimum mean square error $\epsilon_{min}(|z|, \sigma_\Delta)$ defined as

$$\epsilon_{min}(|z|, \sigma_\Delta) = \epsilon(\gamma(\sigma_\Delta^2, |z|), |z|, \sigma_\Delta)\,.$$

Assuming a maximum values of $|z|$ equal to 200, we consider the function

$$\epsilon_{min}^{(max)}(\sigma_\Delta) = \max_{|z| \le 200} \epsilon_{min}(|z|, \sigma_\Delta)\,.$$

This function is plotted in Fig. 31 along with the maximum mean square error corresponding to the described heuristic closed form approximation. As it can be observed, this error is negligible for all values of σ_Δ of practical significance.

References

[1] A. P. Worthen and W. E Stark, "Unified design of iterative receivers using factor graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 843–849, Feb. 2001.

[2] R. Nuriyev and A. Anastasopoulos, "Pilot-symbol-assisted coded transmission over the block-noncoherent AWGN channel," *IEEE Trans. on Commun.*, vol. 51, pp. 953–963, June 2003.

[3] H. Jin and T. Richardson, "Design of low-density parity-check codes for noncoherent MPSK communication," in *Proc. IEEE symposium on information theory*, Lausanne, Switzerland, June-July 2002, p. 169.

[4] J. Dauwels and H.-A. Loeliger, "Joint decoding and phase estimation: an exercise in factor graphs," in *Proc. IEEE Symposium on Information Theory*, Yokohama, Japan, July 2003, p. 231.

[5] V. Lottici and M. Luise, "Carrier phase recovery for turbo-coded linear modulations," in *Proc. IEEE International Conf. Commun.*, Apr. 2002, pp. 1541–1545.

[6] V. Lottici and M. Luise, "Embedding carrier phase recovery into iterative decoding of turbo-coded linear modulations," submitted to IEEE Trans. on Commun., 2002.

[7] N. Noels, C. Herzet, A. Dejonghe, V. Lottici, H. Steendam, M. Moeneclaey, M. Luise, and L. Vandendorpe, "Turbo synchronization: an EM algorithm interpretation," in *Proc. IEEE International Conf. Commun.*, Anchorage, AK, U.S.A., June 2003, pp. 2933–2937.

[8] N. Noels, V. Lottici, A. Dejonghe, H. Steendam, M. Moeneclaey, M. Luise, and L. Vandendorpe, "A theoretical framework for soft information based synchronization in iterative (turbo) receivers," submitted to IEEE Trans. on Commun., 2003.

[9] L. Zhang and A. Burr, "Application of turbo principle to carrier phase recovery in turbo encoded bit-interleaved coded modulation system," in *Proc. Intern. Symp. on Turbo Codes & Relat. Topics*, Brest, France, Sept. 2003, pp. 87–90.

[10] I. Motedayen-Aval and A. Anastasopoulos, "Polynomial-complexity noncoherent symbol-by-symbol detection with application to adaptive iterative decoding of turbo-like codes," *IEEE Trans. on Commun.*, vol. 51, pp. 197–207, Feb. 2003.

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

[12] G. D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, pp. 268–278, Mar. 1973.

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

[14] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *IEEE Trans. on Commun.*, vol. 44, no. 10, pp. 1261–1271, October 1996.

[15] R. G. Gallager, *Low-Density Parity-Check Codes*, MIT Press, Cambridge, MA, 1963.

[16] C. Douillard, M Jezequel, C. Berrou, A. Picart, P. Didier, and A. Glavieux, "Iterative correction of intersymbol interference: turbo-equalization," *European Trans. Telecommun.*, vol. 6, no. 5, pp. 507–511, September/October 1995.

[17] I. D. Marsland and P. T. Mathiopoulos, "On the performance of iterative noncoherent detection of coded M-PSK signals," *IEEE Trans. on Commun.*, vol. 48, no. 4, pp. 588–596, Apr. 2000.

[18] G. Colavolpe, G. Ferrari, and R. Raheli, "Noncoherent iterative (turbo) detection," *IEEE Trans. on Commun.*, vol. 48, no. 9, pp. 1488–1498, Sept. 2000.

[19] G. Ferrari, G. Colavolpe, and R. Raheli, "Noncoherent iterative decoding of spectrally efficient coded modulations," *Annals of Telecommun.*, vol. 56, pp. 409–421, July/August 2001.

[20] A. Anastasopoulos and K. M. Chugg, "Adaptive iterative detection for phase tracking in turbo coded systems," *IEEE Trans. on Commun.*, vol. 49, Dec. 2001.

[21] P. Hoeher and J. Lodge, ""Turbo DPSK": Iterative differential PSK demodulation and channel decoding," *IEEE Trans. on Commun.*, vol. 47, no. 6, pp. 837–843, June 1999.

[22] G. Colavolpe and G. Germi, "Simple iterative detection schemes for ISI channels," in *Proc. Intern. Symp. on Turbo Codes & Relat. Topics*, Brest, France, Sept. 2003, pp. 283–286.

[23] Digital Video Broadcasting Project 2002, "DVB-S," DVB Website: `http://www.dvb.org`.

[24] H. Meyr, M. Oerder, and A. Polydoros, "On sampling rate, analog prefiltering, and sufficient statistics for digital receivers," *IEEE Trans. on Commun.*, vol. 42, pp. 3208–3214, Dec. 1994.

[25] J. Erfanian, S. Pasupathy, and G. Gulak, "Reduced complexity symbol detectors with parallel structures for ISI channels," *IEEE Trans. on Commun.*, vol. 42, no. 2/3/4, pp. 1661–1671, Feb/Mar/Apr. 1994.

[26] W. Koch and A. Baier, "Optimum and sub-optimum detection of coded data disturbed by time-varying intersymbol interference," in *Proc. IEEE Global Telecommun. Conf.*, San Diego, CA, U.S.A., Dec. 1990, pp. 807.5.1–5.

[27] P. Roberston, E. Villebrun, and P. Hoeher, "Optimal and sub-optimal maximum a posteriori algorithms suitable for turbo decoding," *European Trans. Telecommun.*, vol. 8, no. 2, pp. 119–125, March/April 1997.

[28] F.R. Kschischang and B.J. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE J. Select. Areas Commun.*, pp. 219–231, February 1998.

[29] G. Ferrari, G. Colavolpe, and R. Raheli, "On trellis-based truncated-memory detection," in *Proc. IEEE Global Telecommun. Conf.*, San Francisco, CA, U.S.A., Nov. 2003, pp. 2218–2222.

[30] G. Ferrari, G. Colavolpe, and R. Raheli, "Finite memory: optimality and reality," in *Proc. Intern. Symp. on Turbo Codes & Relat. Topics*, Brest, France, Sept. 2003, pp. 1–8.

[31] G. Ferrari, G. Colavolpe, and R. Raheli, "Linear predictive receivers for phase uncertain channels," in *Intern. Symp. on Signal Processing and its Applications, (ISSPA'03)*, Paris, France, July 2003, pp. 3493–3496.

[32] M. Abramowitz and I. A. Stegun, Eds., *Handbook of Mathematical Functions*, Dover, 1972.

[33] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[34] I. S. Gradshtein and I. M. Ryzhik, *Table of Integrals, Series and Products*, Academic Press, New York, 1980.

[35] M. Peleg, S. Shamai (Shitz), and S. Galán, "Iterative decoding for coded noncoherent MPSK communications over phase-noisy AWGN channel," *IEE Proc. Commun.*, vol. 147, pp. 87–95, Apr. 2000.

[36] A. Ginesi, D. Fittipaldi, A. Bigi, and R. De Gaudenzi, "Pilot-aided carrier synchronization techniques for broadband satellite transmissions," Tech. Rep., ESA-ESTEC, Sept. 2003.

[37] S. ten Brink, "Designing iterative decoding schemes with the extrinsic information transfer chart," *AEU Int. J. Electronic. Commun.*, vol. 54, no. 6, pp. 389–398, Dec. 2000.

[38] S. ten Brink and G. Kramer, "Turbo processing for scalar and vector channels," in *Int. Symp. on Turbo-Codes*, Brest, France, September 2003.

[39] S. M. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory*, Prentice-Hall, Upper Saddle River, NJ, 1993.

[40] T. Moon, "The expectation maximization algorithm," *IEEE Sig. Proc. Mag.*, vol. 13, pp. 47–60, Nov. 1996.

[41] M. Kobayashi, J. Boutros, and G. Caire, "Successive interference cancellation with SISO decoding and EM channel estimation," *IEEE J. Select. Areas Commun.*, vol. 19, no. 8, pp. 1450–1460, Aug. 2001.

[42] ETSI, "ETSI - DVBS2 74r13, Digital Video Broadcasting (DVB): Second generation framing structure, channel coding and modulation systems for Braoadcasting, Interactive Services, News Gathering and other broadband satellite applications," 2003.