A Low-Complexity Iterative Phase Noise Tracker for Bit-Interleaved Coded CPM Signals in AWGN

Nele Noels, *Member, IEEE*, Marc Moeneclaey, *Fellow, IEEE*, Frederik Simoens, and Daniel Delaruelle, *Member, IEEE*

***Abstract*—This paper considers iterative detection of bit-interleaved coded continuous phase modulation in the presence of both phase noise (PN) and additive white Gaussian noise (AWGN). The proposed receiver iterates between a detection module and an estimation module. The detection module operates according to the sum-product algorithm and the factor graph framework in order to perform coherent maximum *a posteriori* bit detection in AWGN, using a PN estimate provided by the estimation module. The latter module, which results from the expectation-maximization algorithm for maximum likelihood estimation of the unknown PN samples, is implemented as a smoothing phase-locked loop that uses soft decisions provided by the detector. The separation between the detection and the estimation modules allows the use of an off-the-shelf (coherent) bit detector. The technique is further characterized by a very low computational complexity, a small error performance degradation and a small number of overhead symbols.**

***Index Terms*—Belief propagation, bit-interleaved coded modulation, continuous-phase modulation (CPM), factor graph, iterative estimation and detection, phase noise tracking, sum-product algorithm, turbo carrier synchronization.**

I. Introduction

THIS paper is concerned with the derivation of a practical receiver for bandpass communication of bit interleaved coded continuous phase modulated (BIC-CPM) signals over a typical satellite channel and affected by phase noise.

In bandpass communication systems, imperfections in the oscillators, used for up- and downconversion at transmitter and receiver, cause random perturbations in the phase of the received base-band signal. These random perturbations are referred to as *phase noise* (PN). The PN process typically has a low-pass spectrum. A description of the characteristics of oscillator PN is given in [1] and [2]. The presence of PN can considerably degrade the overall system performance when left unaccounted for. For this reason it must be dealt with, either by a processing unit external to the detector, or by the detector itself.

BIC-CPM is a combination of continuous phase modulation (CPM) and bit interleaved coded modulation (BICM). CPM is a modulation method commonly used in wireless modems [3]. The transmitted CPM waveform has a constant envelope, and its phase is a continuous function of time that changes according to the digital information to be transmitted. CPM is attractive because of its high power and spectral efficiency, and because of its robustness to nonlinearities. CPM is also well suited for use in BICM schemes. BICM is a pragmatic yet powerful coding technique to improve system performance [4]. Well-known binary channel encoders are combined with basic off-the-shelf modulators, with a bit interleaver connecting these two entities. Although the optimal (coherent) symbol-by-symbol detection [5] of a BICM signal is prohibitively complex, there exist approximate iterative detectors with reasonable complexity that yield a very good performance. Such practical detectors can be derived from the sum-product algorithm and the factor graph framework [6].

During the last decade, several advanced techniques have been proposed for iterative bit detection of coded signals affected by PN ([7]–[16], and references therein). These techniques make use of 'turbo' estimation [17]: using pilot information, an initial estimate of the PN is obtained; this initial PN estimate is iteratively improved by also exploiting decisions on the data symbols, obtained from the decoder that uses the PN estimates from a previous iteration. The resulting algorithms strongly depend on the specific modulation format used. In [7]–[9], novel PN estimation and compensation techniques were developed for multicarrier systems using orthogonal frequency division multiplexing (OFDM) as the modulation type. These techniques go beyond the estimation/compensation of only the common phase error by also considering the time variation of the PN within an OFDM symbol interval, in order to also mitigate the PN-induced intercarrier interference. Their operation is based on the estimation of (a small number of) coefficients of the discrete Fourier-series expansion of the time-varying phasor over the OFDM symbol interval. Phase-noise compensation is achieved by performing a convolution in the frequency domain. Iterative estimation and detection for single-carrier systems affected by PN is considered in, e.g., [10]–[16]. Considering a single-carrier system and a multicarrier system operating at the same baudrate, the former has a constellation symbol interval which is much smaller than the symbol interval of the latter. Hence, in single-carrier systems, one PN estimate per symbol interval is generated, and the time-variation of the PN is tracked

Manuscript received November 05, 2010; revised March 04, 2011 and May 16, 2011; accepted May 18, 2011. Date of publication June 13, 2011; date of current version August 10, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ta-Sung Lee. The work of N. Noels was supported by the Research Foundation-Flanders (FWO Vlaanderen). This work was supported by Newtec Cy N.V. and by the European Space Agency, ESA-ESTEC, Noordwijk, The Netherlands, under Contract 22397/09/NL/AD.

N. Noels and M. Moeneclaey are with the TELIN Department, Ghent University, Gent, Belgium (e-mail: nnoels@telin.UGent.be; mm@telin.UGent.be).

F. Simoens and D. Delaruelle are with the Newtec NV, Sint Niklaas, Belgium (e-mail: Frederik.Simoens@newtec.eu; Daniel.Delaruelle@newtec.eu).

Digital Object Identifier 10.1109/TSP.2011.2159498

from one symbol to the next. PN compensation is achieved by multiplication in the time-domain. References [15] and [16] consider adaptive PN tracking algorithms for generic coded single-carrier systems. References [10] and [12] deal with coded single-carrier systems using linear modulation. In [12], the PN tracking is accomplished by means of a decision-aided discrete-time phase-locked loop (PLL). In [10], the PN is represented as a truncated expansion of discrete-cosine transform basis functions over the duration of the data burst, and the coefficients of these basis functions are iteratively estimated. The work presented in [11], [13], [14], and [18] focuses explicitly on CPM, which is also the topic of the present paper. In [11], [13], and [14], three different receiver structures have been proposed which exhibit a very good performance-complexity tradeoff. However, all these schemes require a modification of the detection operation compared to transmission without PN. From an implementation perspective, it is preferable that the PN is dealt with by an *external* processing unit, such that an off-the-shelf*conventional* coherent BIC-CPM bit detector can be used (as in [12] and [18]). The present paper provides a novel solution in that sense. It proposes a receiver that iterates between a conventional detection module and an estimation module. The latter module is implemented as a low-complexity first-order soft-decision-directed smoothing PLL [19]. The estimation is based on the expectation-maximization algorithm [20] and extends to CPM the method for linear modulation derived in [12].

The proposed receiver has a very small computational overhead compared to a coherent detector designed for transmission without phase noise. Its complexity is comparable to that of the simplified version of the algorithm from [15] (denoted A-SISO-sing); it is significantly less complex than the BIC-CPM receivers presented in [13], [14], and [18]. Numerical results further show that, at a packet error rate of 10^{-4}, our approach performs within 1 dB of the optimum coherent detector with perfect phase information [11], which serves as a suitable benchmark. For the CPM parameters yielding high spectral efficiency, a significant error performance improvement with respect to the A-SISO-sing approach is also observed.

II. System Model

The following system model is considered. The transmitter encodes the N_b-element information bit vector $\mathbf{b} = \{b_k\}$ into a vector of N_c coded bits $\mathbf{c} = \{c_l\}$. This vector of coded bits is subsequently interleaved and mapped to an N element symbol vector $\mathbf{a} = (a_1, \ldots, a_N)$, with a_n taking values from the M-ary alphabet $\Omega_M = \{\pm 1, \pm 3, \ldots, \pm(M-1)\}$, with $M = 2^m$. The resulting symbol $\mathbf{a}$ consists of $N_d = N_c/m$ data symbols and a limited amount of overhead symbols ($(N - N_d) \lll N$, see also Section III-F). The resulting symbol vector is then used to generate the complex envelope $s(t)$ of the CPM signal

$$s(t) = e^{j\psi(t;\mathbf{a})} \tag{1}$$

$$\psi(t;\mathbf{a}) = 2\pi h \sum_i a_i q(t - iT). \tag{2}$$

Here, T is the symbol period and $h = K/P$ is the modulation index (K and P are relatively prime integers). The function $q(t)$ is the phase-smoothing response, which is related to the frequency pulse $f(t)$ by the relationship $q(t) = \int_0^t f(u)du$. The pulse $f(t)$ is time-limited to the interval $[0, LT]$ and satisfies the conditions $f(t) = f(LT - t)$ and $\int_0^{LT} f(u)du = 0.5$. It follows from the boundary conditions on $f(t)$ that $q(t) = 0$ for $t \leq 0$ and $q(t) = 0.5$ for $t \geq LT$. Using this, we can rewrite (2) for $nT \leq t \leq (n+1)T$ as

$$\psi(t;\mathbf{a}) = \Psi(t - nT; \mathbf{S}_n), \quad nT \leq t \leq (n+1)T \tag{3}$$

where $n = 1, \ldots, N$. The quantity $\mathbf{S}_n$ in (3) describes the CPM state transition during the nth symbol interval

$$\mathbf{S}_n = (\boldsymbol{\sigma}_n, a_n) \tag{4}$$

with $\boldsymbol{\sigma}_n$ an L element vector $(\sigma_{n,0}, \sigma_{n,1}, \ldots, \sigma_{n,L-1})$ denoting the CPM state at time instant n and

$$\Psi(u;\mathbf{S}_n) = 2\pi h \sum_{i=1}^{L-1} \sigma_{n,i} q(u - (L-i)T) + 2\pi h a_n q(u) + \sigma_{n,0}, \quad 0 \leq u < T. \tag{5}$$

The first element $\sigma_{n,0}$ of the CPM state vector $\boldsymbol{\sigma}_n$ is referred to as the phase state. When K is even, $\sigma_{n,0} \in \frac{2\pi}{P} \cdot \{0, 1, \ldots, P-1\}$. When K is odd, $\sigma_{n,0}$ belongs alternately to $\frac{2\pi}{P} \cdot \{0, 1, \ldots, P-1\}$ and $\frac{\pi}{P} \cdot \{1, 3, \ldots, 2P-1\}$. For $i > 0$, $\sigma_{n,i}$ take values from Ω_M. At each time instant n the CPM transmission scheme has $N_{\boldsymbol{\sigma}} = P \cdot M^{L-1}$ possible states. Given the symbol vector $\mathbf{a}$ and starting from a given initial CPM state $\boldsymbol{\sigma}_1$, the vectors $\boldsymbol{\sigma}_n$, with $n = 2, 3, \ldots, N+1$, can be computed recursively according to the following equations:

$$\sigma_{n,0} = [\sigma_{n-1,0} + \pi h \sigma_{n-1,1}]_{2\pi} \tag{6}$$

$$\sigma_{n,i} = \sigma_{n-1,i+1}, \quad 1 \leq i \leq L-2 \tag{7}$$

$$\sigma_{n,L-1} = a_{n-1} \tag{8}$$

where $[x]_{2\pi}$ denotes modulo-2 reduction of x to the interval $[0, 2\pi[$.

We consider the transmission of the signal $s(t)$ over an additive white Gaussian noise (AWGN) channel, which is typical for satellite communications. It is assumed that $s(t)$ is band-limited—although this is not strictly true in the case of CPM signals whose spectrum has an infinite support- with bandwidth lower than $R_s/2T$, where R_s is a proper integer. The received baseband signal $r(t)$ is low-pass filtered and sampled at R_s samples per symbol interval. It is further assumed that the spacing between the carrier frequencies that are assigned to different users is sufficiently large, such that the leakage of the neighboring signal energy into the desired frequency band can be safely ignored[1]. The observed samples $r_l = r(lT_s)$, with $T_s = T/R_s$, can then be modeled as

$$r_l = s(l'T_s)e^{j\theta(l'T_s)} + n_l, \quad l = 1, 2, \ldots, NR_s \tag{9}$$

[1]An important CPM system design parameter is the product of the spectral channel spacing Δf_c and the symbol duration T. A higher value of $(\Delta f_c \cdot T)$ decreases the inter-channel interference, but comes at the cost of a reduced spectral efficiency; i.e., the number of transmitted information bits per second per Hertz, or, $N_b/(N \cdot \Delta f_c \cdot T)$.

where $l' = R_s - 1 + l$, $s(t)$ is given by (1), $\theta(t)$ represents the PN disturbance and n_l are zero-mean complex AWGN samples with variance equal to $N_0 R_s/E_s$; the first term in (9) has unit power. It is the task of the receiver to recover from $\{r_l\}$ the value of the information bits $\{b_k\}$.

In the next section a new receiver algorithm is derived which assumes that the PN $\theta(t)$ is a slowly varying random process that can be considered constant over the duration of a symbol interval [2]. We define the PN vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_N)$ with θ_n denoting the value of $\theta(t)$ during the nth symbol interval ($nT \leq t \leq (n+1)T$). This corresponds to the assumption that, in (9), $\theta(l'T_s) = \theta_n$ for $l' = nR_s, nR_s + 1, \ldots, (n+1)R_s - 1$. The assumption is valid when most of the PN power is located in an interval $[-B, B]$, with $BT \lll 1$. Moreover, it should be noted that in Section VI the performance of this algorithm will be investigated for actual PN that is allowed to vary within a symbol interval (samples $\theta(l'T_s)$ are generated at a rate $1/T_s = R_s/T$), whereas the receiver estimates only one PN sample per symbol interval.

III. Proposed Receiver

A. Maximum a Posteriori Detection

If an information bit is detected erroneously at the receiver, a bit error occurs. Optimal detection, which minimizes the bit error probability is achieved by *maximum a posteriori* (MAP) bit decision of the individual information bits b_k. The corresponding bit-by-bit MAP detector maximizes the *a posteriori* probability (APP) $p(b_k|\mathbf{r})$ with respect to b_k [5]

$$\hat{b}_k = \arg \max_{b \in \{0,1\}} p(b_k = b|\mathbf{r}). \tag{10}$$

B. Coherent Detection

When the phase vector $\boldsymbol{\theta}$ is known at the receiver, the APPs $p(b_k \,|\, \mathbf{r})$ involved in (10) can be computed efficiently by applying the sum-product (SP) algorithm to a factor graph (FG) representing a suitable factorization of the joint APP $p(\mathbf{b} \,|\, \mathbf{r}; \boldsymbol{\theta})$ [6]; here, the vector $\mathbf{r}$ contains the samples r_l from (9) and $\boldsymbol{\theta}$ is to be considered as a parameter (rather than an argument) of the joint probability. This factorization results from the CPM state (6)–(8) and is given by

$$\begin{aligned} &p(\mathbf{b} \,|\, \mathbf{r}; \boldsymbol{\theta}) \\ &\quad = p(\mathbf{b}, \mathbf{c}, \mathbf{a}, \boldsymbol{\sigma}, \mathbf{S} \,|\, \mathbf{r}; \boldsymbol{\theta}) \\ &\quad \propto p(\mathbf{r} \,|\, \mathbf{b}; \boldsymbol{\theta}) p(\mathbf{b}) \\ &\quad \propto p(\mathbf{c} \,|\, \mathbf{b}) p(\mathbf{a} \,|\, \mathbf{c}) p(\boldsymbol{\sigma}_1) \\ &\qquad \times \prod_{n=1}^{N} J_n(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\sigma}_n, a_n, \mathbf{S}_n; \theta_n) \prod_k p(b_k) \end{aligned} \tag{11}$$

where

$$\begin{aligned} &J_n(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\sigma}_n, a_n, \mathbf{S}_n; \theta_n) \\ &\quad = G_n(\mathbf{S}_n; \theta_n) I(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\sigma}_n, a_n, \mathbf{S}_n) \end{aligned} \tag{12}$$

$$\begin{aligned} &G_n(\mathbf{S}_n; \theta_n) \\ &\quad = \exp\left\{ \frac{2E_s}{N_0} \Re\{ z_n(\mathbf{S}_n) e^{-j\theta_n} \} \right\} \end{aligned} \tag{13}$$

$$\begin{aligned} &z_n(\mathbf{S}_n) \\ &\quad = \frac{1}{R_s} \sum_{l=(n-1)R_s+1}^{nR_s} r_l e^{-j\Psi((l-1)T_s - (n-1)T; \mathbf{S}_n)} \end{aligned} \tag{14}$$

and $I(\boldsymbol{\sigma}_{n+1}, \boldsymbol{\sigma}_n, a_n, \mathbf{S}_n)$ equals one when its arguments satisfy (4) and (6)–(8), and zero otherwise.

The FG representing the factorization (11) is shown in Fig. 1. It consists of a set of function nodes (one for each factor in (11)) that are connected by variable edges (one for each variable in (11)). An approximation of the marginals of $p(\mathbf{b}, \mathbf{c}, \mathbf{a}, \boldsymbol{\sigma}, \mathbf{S} \,|\, \mathbf{r}; \boldsymbol{\theta})$ can be obtained efficiently by passing messages along the edges of the FG from Fig. 1, from one node to an other. Each variable edge in Fig. 1 carries two messages, one in each direction; both these messages are a function of the associated variable. Fig. 2 zooms in on the messages along the lower edges of the FG of $p(\mathbf{b}, \mathbf{c}, \mathbf{a}, \boldsymbol{\sigma}, \mathbf{S} \,|\, \mathbf{r}; \boldsymbol{\theta})$ and introduces the short-hand notations $f_n(\boldsymbol{\sigma}_n)$, $b_n(\boldsymbol{\sigma}_n)$, $u_n(a_n)$, $d_n(a_n)$, and $s_n(\mathbf{S}_n)$ for the nontrivial messages on the edges $\boldsymbol{\sigma}_n$, a_n and $\mathbf{S}_n$. Messages entering the FG on half-edges (e.g., edge $\mathbf{S}_n$) are identically equal to 1. The other messages are computed according to the SP algorithm [6] in the originating function nodes (e.g., f_{n+1} is computed in node J_n); this computation requires the knowledge of the incoming messages on all other edges connected to that same node (e.g., computing f_{n+1} requires knowledge of f_n and d_n). Multiplying the two messages on a given variable edge yields (an approximation of) the marginal APP of this variable, within a normalization constant. For example, on edge $\mathbf{S}_n$

$$\hat{p}(\mathbf{S}_n \,|\, \mathbf{r}; \boldsymbol{\theta}) \propto s_n(\mathbf{S}_n) \cdot 1. \tag{15}$$

Message passing on a FG that corresponds to a tree is straightforward and yields the exact marginals. When a FG contains cycles (i.e., paths from a node to itself via other nodes; e.g., between CPM constraint nodes and code constraint node in Fig. 1) the message passing is an iterative process that, even after convergence, yields only an approximation of the marginals. In the latter case, one also must decide on a particular initialization and scheduling strategy. The strategy that is commonly used by the coherent detector derived from Fig. 1 is outlined in Algorithm 1. From a practical point of view, message passing on a FG boils down to exchanging soft-information (messages) between several soft-input soft-output (SISO) receiver modules. Conventionally, the CPM constraint nodes $J_n, n = 1, 2, \ldots, N$, are grouped into a single SISO module that is efficiently implemented by using the BCJR algorithm [21], as indicated in step COH-2 of Algorithm 1. The coherent detector can schematically be represented as depicted in Fig. 3(a), where CPDEM stands for continuous phase demodulation and DEC/M refers to a joint decoding and demapping unit.

C. Noncoherent Detection

When the phase vector $\boldsymbol{\theta}$ is unknown at the receiver, the APPs $p(b_k \,|\, \mathbf{r})$ involved in (10) can, in principle, be obtained as marginals of the joint APP $p(\mathbf{b}, \boldsymbol{\theta} \,|\, \mathbf{r})$. However, in this case the framework of the sum-product algorithm and factor graphs

[2] A similar assumption was made in [12]–[15], [18].

Algorithm 1 Coherent detector.

COH-1 The downward messages $d_n(a_n)$, as well as the downward messages along the edges $\{a_n\}$ and $\{c_l\}$ are initialized with uniform distributions.

For $i = 1, 2, \ldots N_{it}$:

COH-2 The CPM-related SISO module successively performs
- a forward recursion from n = 1 to $n = N+1$ to compute $f_n(\boldsymbol{\sigma}_n)$,
- a backward recursion from $n = N+1$ to n = 1 to compute $b_n(\boldsymbol{\sigma}_n)$,
- a computation of $u_n(a_n)$, $n = 1, 2, \ldots, N$.

COH-3 The upward messages along the edges $\{c_l\}$ are computed.

COH-4 An appropriate SISO module associated with the code computes the downward messages along the edges $\{c_l\}$.

COH-5 The downward messages $d_n(a_n)$ are computed.

□

COH-6 The upward messages along the edges $\{b_k\}$ are computed; after normalization this yields an estimate $\hat{p}(b_k|\mathbf{r};\boldsymbol{\theta})$ of the marginal information bit APPs. The resulting APPs are used to perform MAP bit detection.

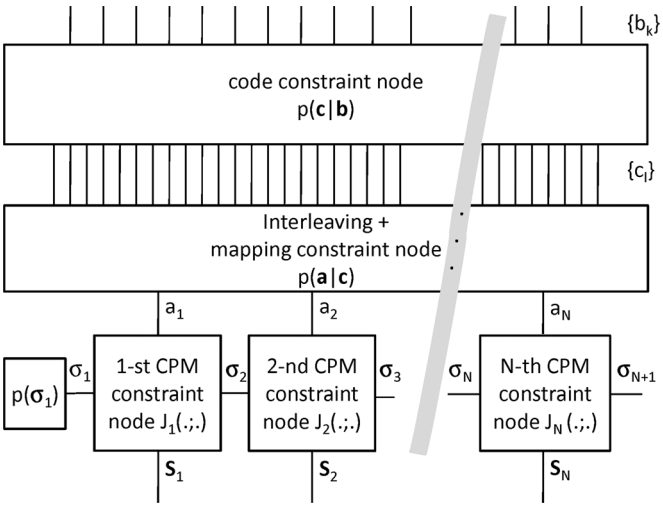

Fig. 1. FG of $p(\mathbf{b}\,|\,\mathbf{r};\boldsymbol{\theta}) = p(\mathbf{b},\mathbf{c},\mathbf{a},\boldsymbol{\sigma},\mathbf{S}\,|\,\mathbf{r};\boldsymbol{\theta})$.

is not the appropriate way to compute these marginals because $\boldsymbol{\theta}$ is continuous-valued. While discrete variables can usually be handled very efficiently by the sum-product algorithm, continuous variables lead to intractable integrals. On top of that, the messages over the corresponding edges are probability density functions rather than probability mass functions (the latter have convenient vector representations). Therefore, the question arises how the messages that correspond to continuous variables should be represented.

A straightforward solution is obtained by discretizing the phase noise variables θ_n. Provided that the number of discretization levels is taken large enough, the performance of the resulting receiver approaches that of the true optimal noncoherent MAP bit detector. The latter approach will be referred

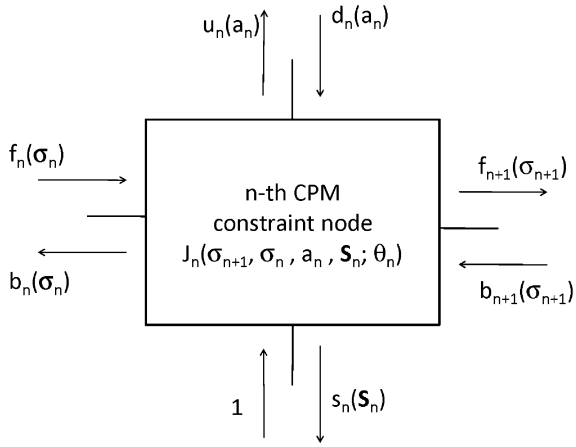

Fig. 2. Messages along lower edges of the FG of $p(\mathbf{b},\mathbf{c},\mathbf{a},\boldsymbol{\sigma},\mathbf{S}\,|\,\mathbf{r};\boldsymbol{\theta})$.

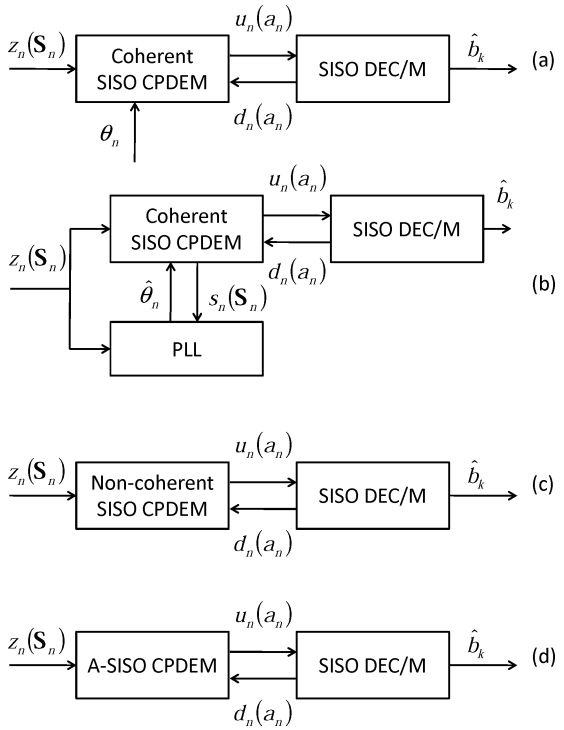

Fig. 3. FG-based receiver schemes.

to as the dp-algorithm. The corresponding receiver scheme is shown in Fig. 3(e). The *coherent* SISO CPDEM unit in the scheme of the coherent receiver from Fig. 3(a) is replaced with a *noncoherent* SISO CPDEM block that performs noncoherent soft continuous phase demodulation by exploiting the phase noise statistics.

D. Proposed Receiver

Instead of noncoherent detection, a more *ad-hoc*, yet simpler, solution is to use a two-stage receiver structure. First, an estimate $\hat{\boldsymbol{\theta}}$ of the continuous-valued PN vector $\boldsymbol{\theta}$ is derived from the observation $\mathbf{r}$. Then, the information bits b_k are coherently detected assuming ideal knowledge of the phase vector, i.e., assuming that $\boldsymbol{\theta} \equiv \hat{\boldsymbol{\theta}}$.

The two-stage receiver structure requires a PN estimator to provide an estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ to the coherent detector. Applica-

tion of the expectation-maximization (EM) algorithm [20] to the maximum likelihood estimation of $\boldsymbol{\theta}$ would yield the following iterative estimation process:

$$\hat{\boldsymbol{\theta}}^{(i+1)} = \arg\max_{\tilde{\boldsymbol{\theta}}} E\left[\ln p(\mathbf{r}\,|\,\mathbf{S},\tilde{\boldsymbol{\theta}})\,|\,\mathbf{r},\hat{\boldsymbol{\theta}}^{(i)}\right] \tag{16}$$

where

$$E\left[\ln p(\mathbf{r}\,|\,\mathbf{S},\tilde{\boldsymbol{\theta}})\,|\,\mathbf{r},\hat{\boldsymbol{\theta}}^{(i)}\right] = \frac{2E_s}{N_0}\sum_{n=1}^{N}\Re\left\{e^{-j\tilde{\theta}_n}\hat{z}_n^{(i)}\right\} \tag{17}$$

$$\hat{z}_n^{(i)}\left(\hat{\boldsymbol{\theta}}^{(i)}\right) = \sum_{\mathbf{S}_n} z_n(\mathbf{S}_n)p\left(\mathbf{S}_n\,|\,\mathbf{r};\hat{\boldsymbol{\theta}}^{(i)}\right) \tag{18}$$

and $z_n(\mathbf{S}_n)$ given by (14). An approximation of the APPs $p(\mathbf{S}_n\,|\,\mathbf{r};\hat{\boldsymbol{\theta}}^{(i)})$ easily results from the coherent detector that operates according to the SP algorithm on the FG representing the factorization (11) and that considers $\hat{\boldsymbol{\theta}}^{(i)}$ as the true PN vector. It suffices to also evaluate, after convergence of the iterative process in Algorithm 1, the messages $s_n(\mathbf{S}_n)$. According to the SP algorithm, we have, with $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}^{(i)}$

$$\hat{p}\left(\mathbf{S}_n\,|\,\mathbf{r};\hat{\boldsymbol{\theta}}^{(i)}\right) \propto s_n(\mathbf{S}_n) = f_n(\boldsymbol{\sigma}_n)b_{n+1}(\boldsymbol{\sigma}_{n+1})d_n(a_n)G_n\left(\mathbf{S}_n;\hat{\theta}_n^{(i)}\right) \tag{19}$$

where $(\mathbf{S}_n,\boldsymbol{\sigma}_n,\boldsymbol{\sigma}_{n+1},a_n)$ satisfies (4) and (6)–(8).

Straightforward application of the EM algorithm thus results in a receiver that iterates between a coherent detector and a PN estimator. During the ith iteration, the PN estimator provides an estimate $\hat{\boldsymbol{\theta}}^{(i)}$ of the PN vector $\boldsymbol{\theta}$ to the coherent detector. The coherent detector assumes that $\hat{\boldsymbol{\theta}}^{(i)}$ equals the true PN vector; it applies carrier phase correction based on $\hat{\boldsymbol{\theta}}^{(i)}$, and computes APPs of the transitions $\mathbf{S}_n$ by applying the SP algorithm on a FG. The PN estimator uses these APPs in the next iteration to compute the PN vector estimate $\hat{\boldsymbol{\theta}}^{(i+1)}$, which is fed to the coherent detector for carrier phase correction and APP computation. When these iterations have converged, the coherent detector evaluates the information bit APPs using the FG framework and outputs information bit decisions that maximize these APPs.

In principle, the coherent detector, which itself involves an iterative process (of demapping and decoding), has to converge for each estimator iteration. The computational complexity of the receiver reduces significantly when the estimator iterations and the detector iterations are intertwined [12], [18]. The ith iteration then consists of an estimator iteration (resulting in $\hat{\boldsymbol{\theta}}^{(i)}$) followed by only one detector iteration (resulting in $\hat{p}(\mathbf{S}_n\,|\,\mathbf{r};\hat{\boldsymbol{\theta}}^{(i)})$), without resetting the messages along the edges of the FG. The resulting receiver operation is outlined in Algorithm 2.

To avoid the multidimensional maximization in (16) that results from the EM algorithm, we use a first-order smoothing phase-locked loop (PLL) to compute $\hat{\boldsymbol{\theta}}^{(i)}$ from

Algorithm 2 Proposed receiver.

PROP-1 An initial phase vector estimate is computed.

PROP-2 The FG messages are initialized as in step COH-1 of Algorithm 1.

For $i = 1, 2, \ldots N_{it}$:

PROP-3 One iteration of the coherent detector is performed: steps COH-2 to COH-5 of Algorithm 1.

PROP-4 The messages $s_n(\mathbf{S}_n)$ are computed; after normalization this yields an estimate $\hat{p}\left(\mathbf{S}_n\,\middle|\,\mathbf{r};\hat{\boldsymbol{\theta}}^{(i-1)}\right)$ of the marginal CPM state transition APPs. The resulting APPs are used to compute a new phase vector estimate $\hat{\boldsymbol{\theta}}^{(i)}$.

□

PROP-5 The upward messages along the edges $\{b_k\}$ are computed and MAP bit detection is performed.

$\{\hat{z}_n^{(i-1)}(\hat{\boldsymbol{\theta}}^{(i-1)}), n = 1,\ldots,N\}$. During the ith iteration, this PLL makes use of the phase error detector function $\Im\{e^{-j\tilde{\theta}_n}\hat{z}_n^{(i-1)}(\hat{\boldsymbol{\theta}}^{(i-1)})\}$, which is proportional to the derivative of $E[\ln p(\mathbf{r}\,|\,\mathbf{S},\tilde{\boldsymbol{\theta}})\,|\,\mathbf{r},\hat{\boldsymbol{\theta}}^{(i-1)}]$ with respect to $\tilde{\theta}_n$. Given an initial estimate $\hat{\theta}_1^{(F,i)}$, a forward PLL recursion computes the estimates $\hat{\theta}_n^{(F,i)}$ for $n = 2,\ldots,N+1$ according to

$$\hat{\theta}_n^{(F,i)} = \hat{\theta}_{n-1}^{(F,i)} + \lambda\cdot\Im\left\{e^{-j\hat{\theta}_{n-1}^{(F,i)}}\hat{z}_{n-1}^{(i-1)}\left(\hat{\boldsymbol{\theta}}^{(i-1)}\right)\right\} \tag{20}$$

where $\hat{z}_{n-1}^{(i-1)}(\hat{\boldsymbol{\theta}}^{(i-1)})$ depends on the PN vector estimate $\hat{\boldsymbol{\theta}}^{(i-1)}$ from the previous iteration [see (18)]. Next, given an initial estimate $\hat{\theta}_N^{(B,i)}$, a backward PLL recursion computes the estimates $\hat{\theta}_n^{(B,i)}$ for $n = 0,\ldots,N-1$ according to

$$\hat{\theta}_n^{(B,i)} = \hat{\theta}_{n+1}^{(B,i)} + \lambda\cdot\Im\left\{e^{-j\hat{\theta}_{n+1}^{(B,i)}}\hat{z}_{n+1}^{(i-1)}\left(\hat{\boldsymbol{\theta}}^{(i-1)}\right)\right\}. \tag{21}$$

Finally, the nth element $\hat{\theta}_n^{(i)}$ of the PN estimate $\hat{\boldsymbol{\theta}}^{(i)} = (\hat{\theta}_1^{(i)},\hat{\theta}_2^{(i)},\ldots,\hat{\theta}_N^{(i)})$ delivered to the coherent detector is the arithmetical average of $\hat{\theta}_n^{(F,i)}$ and $\hat{\theta}_{n-1}^{(B,i)}$

$$\hat{\theta}_n^{(i)} = \frac{\hat{\theta}_n^{(F,i)} + \hat{\theta}_{n-1}^{(B,i)}}{2}, \quad n = 1,2,\ldots,N. \tag{22}$$

The latter operation is referred to as smoothing. The parameter λ in (20)–(21) determines the bandwidth of the PLL: increasing λ improves the PN tracking capability, but at the same time increases the sensitivity to AWGN. Note from (20)–(21) that $\hat{\theta}_n^{(F,i)}$ is based on $\{\hat{z}_{\tilde{n}}^{(i-1)}(\hat{\boldsymbol{\theta}}^{(i-1)}),\tilde{n} = 1,\ldots,n-1\}$, whereas $\hat{\theta}_{n-1}^{(B,i)}$ is based on $\{\hat{z}_{\tilde{n}}^{(i-1)}(\hat{\boldsymbol{\theta}}^{(i-1)}),\tilde{n} = n,\ldots,N\}$, such that $\hat{\theta}_n^{(i)}$ depends on all past, present and future observations. The resulting receiver algorithm, which combines conventional coherent SISO CPM demodulation with an external bidirectional smoothing PLL, is denoted SISO+PLL BiS.

Further simplification can be taken by removing the backward phase estimator, i.e., by setting

$$\hat{\theta}_n^{(i)} = \hat{\theta}_n^{(F,i)}, \quad n = 1,2,\ldots,N.$$

In this case, the smoothing PLL reduces to a conventional PLL that performs forward-only (FO) phase tracking. This approach will be referred to as SISO+PLL FO.

A schematical view of the proposed receiver is presented in Fig. 3(b). The coherent receiver from Fig. 3(a) is extended with a separate subsystem that performs PLL-based phase tracking.

E. Cycle Slipping

If the code structure is disregarded, the set of CPM signal waveforms (5) is invariant under a rotation over an angle $2\pi/P$. As a result, the employed PLLs cannot distinguish between angles θ and $\theta + 2k\pi/P$ with $k = \pm 1, \pm 2, \ldots$. In other words, the PLL has infinitely many stable operation points, which are spaced by $2\pi/P$. Although the rotational invariance is affected by the code structure, the usual $2\pi/P$ estimation ambiguity due to the symmetry of the CPM scheme remains apparent [12]. In steady-state operation, the carrier phase estimate exhibits small random fluctuations about a stable operating point. Occasionally, noise or other disturbances push the estimate away from the current stable operating point, into the domain of attraction of a neighboring stable operating point. This phenomenon is called a cycle slip. After this, the estimate remains for some time in the close vicinity of the new operating point, until the next slip occurs.

In the case of a conventional PLL (PLL FO), cycle slips manifest themselves as occasional phase shifts over $2\pi/P$. When a smoothing PLL (PLL BiS) is employed, a cycle slip results in a phase shift of π/P (instead of $2\pi/P$) because the arithmetical average of forward and backward estimates is computed. It is known that CPM detectors can tolerate phase shifts of even multiples of π/P, but not of odd multiple of π/P. A cycle slip event is therefore likely to prevent error-free decoding when a smoothing PLL is employed, while it hardly has any effect on the detection process in the case of FO phase tracking. Similar observations were made in [13] and [14]. The probability of a cycle slip increases exponentially with the mean square phase error (MSPE) $E[(\theta_n - \hat{\theta}_n^{(i)})]$ and linearly with the block length N. For BIC-CPM systems (like BIC-CPM 0 from Table III) with large N that operate at low values of E_s/N_0, cycle slipping may be a major performance limiting factor of the SISO+PLL BiS receiver.

F. Overhead Symbol Exploitation

In many practical scenarios, phase forcing sequences (i.e., short data dependent overhead symbols that force the CPM signal to pass through a known phase state $\sigma_{n,0}$) and known pilot symbols (PS) are multiplexed with the transmitted data symbols. They are meant for frequency and timing synchronization purposes at the receiver, or to improve the performance of the SISO CPM demodulator by increasing the *a priori* information about the final CPM state. The phase tracking process can of coarse only benefit from also exploiting the available *a priori* information brought by these overhead symbols. Assuming that a_n is a PS with true value $\tilde{a}_n$, we distinguish between the following two situations:

1) The value of the CPM state σ_n at the start of the nth symbol interval is not known at the receiver. The following downward messages are used in the CPM SISO module, during all receiver iterations: $d_n(a_n) = \delta(a_n - \tilde{a}_n)$.
2) The true value $\tilde{\sigma}_n$ of σ_n is known at the receiver, such that the waveform transmitted in $[nT, (n+1)T]$ is perfectly known. Hence, in (20), (21), (24), and (26) we set $\hat{z}_n^{(i)} = \hat{z}_n^{(F,i)} = \hat{z}_n^{(B,i)} = z_n(\tilde{\mathbf{S}}_n)$, where $\tilde{\mathbf{S}}_n = (\tilde{\sigma}_n, \tilde{a}_n)$.

Only a marginal effect can be expected from this exploitation because, for efficiency reasons, the number of overhead symbols needs to be kept small as compared to N.

G. Initialization of PLL Recursions

The forward and backward recursions (20)–(21) of the PLL must be initialized with estimates $\hat{\theta}_1^{(F,i)}$ and $\hat{\theta}_N^{(B,i)}$, respectively. We take $\hat{\theta}_N^{(B,i)} = \hat{\theta}_{N+1}^{(F,i)}$, indicating that the phase estimate at the end of the forward recursion is used as initial value for the backward recursion. We further take $\hat{\theta}_1^{(F,i)} = \hat{\theta}_0$, with

1) $\hat{\theta}_0$ denoting the phase estimate resulting from a short sequence of known preamble symbols (that precedes the actual data symbols), over which the phase is assumed to be constant. Hence

$$\hat{\theta}_0 = \angle \left\{ \sum_{n=1}^{L_{pre}} z_n \left(\tilde{\mathbf{S}}_n \right) \right\}. \tag{23}$$

The summation in (23) is over all preamble symbol intervals and $\tilde{\mathbf{S}}_n$ denotes the true value of $\mathbf{S}_n$, fully determined by the value of the pilot symbols and by the value of the initial CPM state.

2) $\hat{\theta}_0$ equal to zero, i.e., $\hat{\theta}_0 = 0$, in the absence of a preamble. The disadvantage of this approach is the occurrence of a transient at the start of the burst, during which period the phase error $(\hat{\theta}_n^{(F,i)} - \theta_n)$ can take large values.

H. Initialization of Estimator Iterations

According to Algorithm 2, an initial vector estimate $\hat{\boldsymbol{\theta}}^{(0)} = (\hat{\theta}_1^{(0)}, \hat{\theta}_2^{(0)}, \ldots, \hat{\theta}_N^{(0)})$ is needed (step PROP-1) to start the iterations of the proposed receiver. A practical way to circumvent this requirement is to replace the forward and backward phase update rules (20) and (21) by

$$\hat{\theta}_n^{(F,i)} = \hat{\theta}_{n-1}^{(F,i)} + \lambda \cdot \Im \left\{ e^{-j\hat{\theta}_{n-1}^{(F,i)}} \hat{z}_{n-1}^{(F,i)} \left(\hat{\boldsymbol{\theta}}_{n-1}^{(F,i)} \right) \right\} \tag{24}$$

with

$$\hat{z}_n^{(F,i)} \left(\hat{\boldsymbol{\theta}}_n^{(F,i)} \right) = \sum_{\mathbf{S}_n} z_n(\mathbf{S}_n) p \left(\mathbf{S}_n \,|\, \mathbf{r}_n^{(F)}; \hat{\boldsymbol{\theta}}_n^{(F,i)} \right) \tag{25}$$

and

$$\hat{\theta}_n^{(B,i)} = \hat{\theta}_{n+1}^{(B,i)} + \lambda \cdot \Im \left\{ e^{-j\hat{\theta}_{n+1}^{(B,i)}} \hat{z}_{n+1}^{(B,i)} \left(\hat{\boldsymbol{\theta}}_{n+1}^{(B,i)} \right) \right\} \tag{26}$$

with

$$\hat{z}_n^{(B,i)} \left(\hat{\boldsymbol{\theta}}_n^{(B,i)} \right) = \sum_{\mathbf{S}_n} z_n \left(\mathbf{S}_n \right) p \left(\mathbf{S}_n \,|\, \mathbf{r}_n^{(B)}; \hat{\boldsymbol{\theta}}_n^{(B,i)} \right). \tag{27}$$

Starting from $\hat{\theta}_1^{(F,i)} = \hat{\theta}_P$, forward PN estimates $\hat{\theta}_n^{(F,i)}$ are computed for $n = 2, 3, \ldots, N+1$ according to (24)–(25). The quantities $p(\mathbf{S}_n \,|\, \mathbf{r}_n^{(F)}; \hat{\boldsymbol{\theta}}_n^{(F,i)})$ in (25) are *forward-only* CPM state transition APPs, with $\hat{\boldsymbol{\theta}}_n^{(F,i)}$ collecting the PN sample es-

timates $\hat{\theta}_k^{(F,i)}$ for $k = 1, 2, \ldots, n$ and $\mathbf{r}_n^{(F)}$ a vector that contains the first nR_s samples r_l from (9). Then, starting from $\hat{\theta}_N^{(B,i)} = \hat{\theta}_{N+1}^{(F,i)}$, backward PN estimates $\hat{\theta}_n^{(B,i)}$ are computed for $n = N-1, N-2, \ldots, 0$ according to (26)–(27). The quantities $p(\mathbf{S}_n \,|\, \mathbf{r}_n^{(B)}; \hat{\boldsymbol{\theta}}_n^{(B,i)})$ in (27) are *backward-only* CPM state transition APPs, with $\hat{\boldsymbol{\theta}}_n^{(B,i)}$ collecting the PN sample estimates $\hat{\theta}_k^{(B,i)}$ for $k = n, n+1, \ldots, N+1$ and $\mathbf{r}_n^{(B)}$ a vector that contains the last $(N-n+1)R_s$ samples r_l from (9). A hard-decision-directed version of (24)–(27) is employed in one of the receivers proposed in [15] (this will be further discussed in Section IV). Unfortunately, in general, the forward-only and backward-only CPM state transition APPs required by (26)–(27) do not result as easily from the conventional coherent detector as the correct CPM state transition APPs used in (18), (20)–(21). For a BIC-CPM receiver to produce estimates of $p(\mathbf{S}_n \,|\, \mathbf{r}_n^{(F)}; \hat{\boldsymbol{\theta}}_n^{(F,i)})$ and $p(\mathbf{S}_n \,|\, \mathbf{r}_n^{(B)}; \hat{\boldsymbol{\theta}}_n^{(B,i)})$, it requires the modification of the detector operation, which in turn prevents the use of off-the-shelf coherent bit detectors.

In order to avoid this, we will use (24)–(27) only to initialize the receiver described by Algorithm 2. With $i = 0$, one forward and one backward PLL recursion is performed, while the data symbols $a_1, a_2, \ldots, a_N$ are assumed to be statistically independent. The corresponding FG consists of the lower part of Fig. 1 (i.e., the code constraint node and the interleaving + mapping constraint node are removed), and, therefore, is cycle-free. In this case, the forward-only and backward-only CPM state transition APPs can be efficiently computed according to the BCJR algorithm [21]. The resulting forward recursion computes $\hat{p}(\mathbf{S}_n \,|\, \mathbf{r}_n^{(F)}; \hat{\boldsymbol{\theta}}_n^{(F,0)})$ for $n = 2, 3, \ldots, N$ according to

$$\hat{p}\left(\mathbf{S}_n \,|\, \mathbf{r}_n^{(F)}; \hat{\boldsymbol{\theta}}_n^{(F,0)}\right) \propto G_n\left(\mathbf{S}_n; \hat{\boldsymbol{\theta}}_n^{(F,0)}\right) \sum_{\mathbf{S}_{n-1}:\mathbf{S}_n} \hat{p}\left(\mathbf{S}_{n-1} \,|\, \mathbf{r}_{n-1}^{(F)}; \hat{\boldsymbol{\theta}}_{n-1}^{(F,0)}\right) \quad (28)$$

where the sum is over all CPM state transitions $\mathbf{S}_{n-1}$ that can precede the CPM state transition $\mathbf{S}_n$ [according to (4) and (6)–(8)]. For $n = 1$, we have

$$\hat{p}\left(\mathbf{S}_1 \,|\, \mathbf{r}_1^{(F)}; \hat{\boldsymbol{\theta}}_1^{(F,0)}\right) = G_1\left(\mathbf{S}_1; \hat{\boldsymbol{\theta}}_1^{(F,0)}\right) I((\mathbf{S}_1)_1 = \boldsymbol{\sigma}_1)$$

where $\hat{\theta}_1^{(F,0)} = \hat{\theta}_P$ and $I((\mathbf{S}_1)_1 = \boldsymbol{\sigma}_1)$ equals 1 when the CPM state transition $\mathbf{S}_1$ departs from the given initial CPM state $\boldsymbol{\sigma}_1$, and zero otherwise. Similarly, a backward recursion computes $\hat{p}(\mathbf{S}_n \,|\, \mathbf{r}_n^{(F)}; \hat{\boldsymbol{\theta}}_n^{(B,0)})$ for $n = N-1, N-2, \ldots, 1$ according to

$$\hat{p}\left(\mathbf{S}_n \,|\, \mathbf{r}_n^{(B)}; \hat{\boldsymbol{\theta}}_n^{(B,0)}\right) \propto G_n\left(\mathbf{S}_n; \hat{\boldsymbol{\theta}}_n^{(B,0)}\right) \times \sum_{\mathbf{S}_{n+1}:\mathbf{S}_n} \hat{p}\left(\mathbf{S}_{n+1} \,|\, \mathbf{r}_{n+1}^{(B)}; \hat{\boldsymbol{\theta}}_{n+1}^{(B,0)}\right) \quad (29)$$

where the sum is over all CPM state transitions $\mathbf{S}_{n+1}$ that can succeed the CPM state transition $\mathbf{S}_n$ [according to (4) and (6)–(8)]. For n=N, we have

$$\hat{p}\left(\mathbf{S}_N \,|\, \mathbf{r}_N^{(B)}; \hat{\boldsymbol{\theta}}_N^{(B,0)}\right) = G_N\left(\mathbf{S}_N; \hat{\boldsymbol{\theta}}_N^{(B,0)}\right)$$

with $\hat{\theta}_N^{(B,0)} = \hat{\theta}_{N+1}^{(F,0)}$. We note that the above recursions are perfectly compatible with the phase update recursions (24) and (26). Combining (28) with (24)–(25), [(29) with (26)–(27)] allows the joint computation of the forward (backward) PN estimates and the required forward-only (backward-only) CPM state transition APPs at discrete time instant n. Finally, the arithmetical average of the resulting estimates $\hat{\theta}_n^{(F,0)}$ and $\hat{\theta}_{n-1}^{(B,0)}$ is computed and delivered as the initial estimate $\hat{\boldsymbol{\theta}}^{(0)}$ to the receiver from Algorithm 2. The computational complexity of this initialization process (which has to be executed only once at the start of the iterations) is proportional to the number of CPM states and to the length of the received signal.

IV. Comparison With Existing Algorithms

The receiver from Algorithm 2 and Fig. 3(b) employs a conventional coherent BIC-CPM symbol detector. To cope with the presence of PN in the received signal it is equipped with an external module for PN estimation that introduces very little computational and symbol overhead. By iterating between the detection and the PN estimation module, the receiver implements an approximate implementation of the EM algorithm.

During the last decade, several other techniques [17], [16] have been proposed for iterative symbol-by-symbol detection of BICM signals affected by both AWGN and PN.The linear modulation counterpart of the proposed receiver was presented in [12]. Considering CPM leads to a more complicated receiver, as the messages $d_n(a_n)$ coming from the soft decoder and demapper are not readily suited to be fed directly to the PLL (as is the case in [12]). In order to benefit from the smoothing gain addressed in [19], the conventional PLL employed in [12] is replaced with its smoothing variant [14], [19].

The BIC-CPM receiver from [18] for transmission over fast flat Rayleigh fading channels can also be used for a PN affected signal. We can indeed interpret the values $v_n = e^{j\theta_n}$, for $n = 1, 2, \ldots, N$, of the phasor associated with the PN distortion θ_n during the nth symbol interval as unknown (complex-valued) channel coefficients. The operation of the receiver in [18] also results from the EM algorithm; it uses feed-forward (FF) linear minimum mean square error (LMMSE) estimation (rather than PLLs) to generate an estimate of the channel coefficient vector at each detector iteration. Straightforward application of the algorithm from [18] to the problem of estimating the phasor vector $\mathbf{v}$ from the PN and AWGN affected received BIC-CPM signal $r(t)$ yields

$$\hat{\mathbf{v}}^{(i)} = \hat{\mathbf{z}}^{(i-1)}\left(\hat{\mathbf{v}}^{(i-1)}\right)\left\{\mathbf{I} + \frac{N_0}{E_s}\mathbf{R}_v^{-1}\right\}^{-1} \quad (30)$$

with $\hat{\mathbf{z}}^{(i)} = (\hat{z}_1^{(i)} \hat{z}_2^{(i)} \cdots \hat{z}_N^{(i)})$ and $\hat{z}_n^{(i)}$ defined as

$$\hat{z}_n^{(i)}\left(\hat{\mathbf{v}}^{(i)}\right) = \sum_{\mathbf{S}_n} z_n(\mathbf{S}_n)\, p\left(\mathbf{S}_n \,|\, \mathbf{r}; \hat{\mathbf{v}}^{(i)}\right) \quad (31)$$

where $z_n(\mathbf{S}_n)$ is given by (14) and the APPs $p(\mathbf{S}_n \,|\, \mathbf{r}; \hat{\mathbf{v}}^{(i)})$ result from a coherent detector that operates according to the SP algorithm on a FG representing a suitable factorization $p(\mathbf{b}, \mathbf{c}, \mathbf{a}, \boldsymbol{\sigma}, \mathbf{S} \,|\, \mathbf{r}; \mathbf{v})$ and that considers $\hat{\mathbf{v}}^{(i)}$ as the true PN-related phasor vector. In (30), $\mathbf{R}_v$ is the autocorrelation matrix of

the vector $\mathbf{v} = (v_1, v_2, \ldots, v_N)$. Assuming that $\boldsymbol{\theta}$ is Gaussian with autocorrelation matrix $\mathbf{R}_\theta$, one obtains

$$\mathbf{R}_v(k,l) = e^{\{\mathbf{R}_\theta(k,l) - \frac{1}{2}(\mathbf{R}_\theta(k,k) + \mathbf{R}_\theta(l,l))\}}.$$

To reduce the computational complexity, the estimator and detector iterations can be intertwined. The matrix inversion in (30) does not need to be recomputed at each iteration, but the computational load of the vector-matrix multiplication in (30) is nevertheless higher than that of the bi-directional recursive smoothing process from (20)–(22) ($2N$ versus only 3 real-valued multiplications (and additions) per transmitted symbol). This receiver scheme will be referred to as SISO+FF LMMSE; it looks exactly like that of Fig. 3(b), but with the PLL phase tracker replaced with a unit that performs the feed-forward phasor estimation process from (30). In [18], the initial PN estimate is provided by the estimation algorithm presented in [22]. This approach relies on the periodical insertion (e.g., once every $L_{\text{block}} = 10$ or 20 symbols) of phase forcing sequences. For an arbitrary CPM format, the required length $L_{\text{phase forcing}}$ of such a sequence is

$$L_{\text{phase forcing}} = \text{ceil}\left(\frac{P-1}{M-1}\right)$$

where $\text{ceil}(x)$ returns the smallest integer larger than or equal to x. The disadvantages of this technique are the strict requirements imposed on the burst structure and the symbol overhead which (especially for CPM formats with $P > M$) quickly results in an adverse effect on the system's power and bandwidth efficiency. Let us consider the BIC-CPM 2 signal example: $M = 4$ and $P = 5$ (see Table III), then $L_{\text{phase forcing}} = 2$ and $L_{\text{block}} = 10$, which yields a symbol overhead of already 20%.

Receivers that (in contrast to the proposed one) modify the detection operation itself (in order to make it robust against PN) are presented in [22], [14], [15], [13]. Given $r(t)$, the receiver in [13] produces an estimate $\hat{b}_k$ of the information bits b_k according to the MAP bit detection rule (10). An estimate of the APPs $p(b_k \,|\, \mathbf{r})$ required to implement (10) are derived from the FG framework. Using Bayes' rule, $p(\mathbf{b}, \boldsymbol{\theta} \,|\, \mathbf{r})$ can be factorized as follows:

$$p(\mathbf{b}, \boldsymbol{\theta} \,|\, \mathbf{r}) \propto p(\mathbf{r} \,|\, \mathbf{b}, \boldsymbol{\theta}) p(\mathbf{b}) p(\boldsymbol{\theta}) \tag{32}$$

where $p(\boldsymbol{\theta})$ is the a *priori* probability density function of the PN vector $\boldsymbol{\theta}$ and $p(\mathbf{r} \,|\, \mathbf{b}, \boldsymbol{\theta})$ can be further factorized as in (11). In (32), $\boldsymbol{\theta}$ is considered as a random variable (rather than a parameter as in (11)). The practical implementation of the receiver in [13] results from the further factorization of $p(\boldsymbol{\theta})$ that is based on the assumption of Wiener PN, i.e.

$$p(\boldsymbol{\theta}) = p(\theta_1) \prod_{i=1}^{N-1} p(\theta_{i+1} \,|\, \theta_i).$$

Furthermore, to circumvent the complexity increase that typically results from the straightforward application of the SP algorithm to a FG containing continuous-valued variables (such as θ_n), [13] adopts an approximation involving canonical Tikhonov distributions (Tikh-SISO algorithm). The paper also considers a more straightforward solution which is obtained by discretizing the phase noise variable (dp-SISO algorithm). The receiver scheme from Fig. 3(c) applies. In order to further reduce the complexity of the noncoherent CPM-related SISO unit, the factorization of $p(\mathbf{b} \,|\, \mathbf{r}, \boldsymbol{\theta})$ is based on a *truncated* Laurent decomposition of the transmitted CPM signal [23]. Only the so-called principle components of the Laurent decomposition are taken into account. This approximation reduces the number of trellis states from $N_{\boldsymbol{\sigma}} = pM^{L-1}$ to p.

Another type of receivers that modify the detection operation is presented in [14], [15]. In these contributions per survivor PN estimates are used inside one of the inner SISO modulesof an iterative detector in order to cope with PN uncertainty; the considered module is said to execute an *adaptive* SISO (A-SISO) algorithm. Applying the approach from [14], [15] to the CPM-related SISO module of the conventional BIC-CPM detector involves running $N_{\boldsymbol{\sigma}}$ forward and as much backward PLL recursions, for each detector iteration. Hence, the complexity overhead introduced by these per-survivor-processing-based PN-tracking implementations (as compared to conventional coherent detection) is about $N_{\boldsymbol{\sigma}}$ times as high as in the case of the proposed receiver that uses only one PLL. However, [16], [14], [15], [13], and [18] also mentions a simplified version of its main algorithm that uses only one (rather than $N_{\boldsymbol{\sigma}}$) PN tracking PLL. During iteration i, this low-complexity *A-SISO-sing* receiver executes the following adaptive SISO module that replaces the set of CPM constraint nodes in the FG of Fig. 1.

1) Forward messages $f_{n+1}^{(A)}(\boldsymbol{\sigma}_{n+1})$, approximate forward-only CPM state transition APPs $\hat{p}(\mathbf{S}_n \,|\, \mathbf{r}_n^{(F)}; \hat{\boldsymbol{\theta}}_n^{(i,F)})$ and forward phase estimates $\hat{\theta}_{n+1}^{(F,i)}$ and are computed by means of a forward recursion, for $n = 1, 2, \ldots, N$. The rule for computing $f_{n+1}^{(A)}(\boldsymbol{\sigma}_{n+1})$ results from applying to the CPM trellis a forward BCJR recursion that considers the forward phase estimates $\hat{\theta}_n^{(F,i)}$ as the true phase samples and the messages $d_n(a_n)$ (see Fig. 2) available from the interleaving and mapping constraint node of the FG as (unnormalized) data symbol *a priori* probabilities. This yields

$$f_{n+1}^{(A)}(\boldsymbol{\sigma}_{n+1}) = \sum_{(a_n, \boldsymbol{\sigma}_n): \boldsymbol{\sigma}_{n+1}} f_n^{(A)}(\boldsymbol{\sigma}_n) d_n(a_n) G_n\left(\mathbf{S}_n; \hat{\boldsymbol{\theta}}_n^{(F,i)}\right)$$

where the summation is with respect to all $(a_n, \boldsymbol{\sigma}_{n-1})$ for which $(a_n, \boldsymbol{\sigma}_n, \boldsymbol{\sigma}_{n+1})$ satisfies (4) and (6)–(8). Similarly, the approximate APPs $\hat{p}(\mathbf{S}_n \,|\, \mathbf{r}_n^{(F)}; \hat{\boldsymbol{\theta}}_n^{(i,F)})$ are computed as

$$\hat{p}\left(\mathbf{S}_n \,|\, \mathbf{r}_n^{(F)}; \hat{\boldsymbol{\theta}}_n^{(i,F)}\right) \propto f_n^{(A)}(\boldsymbol{\sigma}_n) G_n\left(\mathbf{S}_n; \hat{\boldsymbol{\theta}}_n^{(F,i)}\right) d_n(a_n).$$

The forward phase estimates are updated according to (24), but with hard rather than soft CPM state transition decisions, i.e.

$$\hat{z}_n^{(i,F)} = z_n\left(\arg\max_{\mathbf{S}_n} \hat{p}\left(\mathbf{S}_n \,|\, \mathbf{r}_n^{(F)}; \hat{\boldsymbol{\theta}}_n^{(i,F)}\right)\right). \tag{33}$$

Given $s_n(\mathbf{S}_n)$, evaluating (33) requires 5 times less operations (real comparisons, real additions or real multiplications) than evaluating (18).

2) Backward messages $b_n^{(A)}(\boldsymbol{\sigma}_n)$, approximate backward-only CPM state transition APPs $\hat{p}(\mathbf{S}_n \,|\, \mathbf{r}_n^{(B)}; \hat{\boldsymbol{\theta}}_n^{(i,B)})$ and backward phase estimates $\hat{\theta}_{n-1}^{(B,i)}$ are computed by means of a backward recursion, for $n = N, N-2, \ldots, 1$. The update rule for $b_n^{(A)}(\boldsymbol{\sigma}_n)$ results from applying a forward BCJR recursion that considers the forward phase estimates $\hat{\theta}_n^{(B,i)}$ as the true phase samples and the messages $d_n(a_n)$ (see Fig. 2) computed in the interleaving and mapping constraint node of the FG as (unnormalized) data symbol *a priori* probabilities to the CPM system trellis. This yields

$$b_n^{(A)}(\boldsymbol{\sigma}_n) = \sum_{(a_n, \boldsymbol{\sigma}_{n+1}):\boldsymbol{\sigma}_n} b_{n+1}^{(A)}(\boldsymbol{\sigma}_{n+1}) d_n(a_n) G_n\left(\mathbf{S}_n; \hat{\boldsymbol{\theta}}_n^{(B,i)}\right)$$

where the summation is with respect to all $(a_n, \boldsymbol{\sigma}_{n+1})$ for which $(a_n, \boldsymbol{\sigma}_{n+1}, \boldsymbol{\sigma}_n)$ satisfies (4) and (6)–(8). Similarly, the approximate APPs $\hat{p}(\mathbf{S}_n \,|\, \mathbf{r}_n^{(B)}; \hat{\boldsymbol{\theta}}_n^{(i,B)})$ are computed as

$$\hat{p}\left(\mathbf{S}_n \,|\, \mathbf{r}_n^{(B)}; \hat{\boldsymbol{\theta}}_n^{(i,B)}\right) \propto b_{n+1}^{(A)}(\boldsymbol{\sigma}_{n+1}) G_n\left(\mathbf{S}_n; \hat{\boldsymbol{\theta}}_n^{(B,i)}\right) d_n(a_n).$$

The backward phase estimates are updated according to (26), but with hard rather than soft CPM state transition decisions $\hat{z}_n^{(i,B)}$, given by (33) but with $F = B$.

3) For $n = 1, 2, \ldots, N$, upward messages $u_n^{(A)}(a_n)$ are computed according to

$$u_n^{(A)}(a_n) = \sum_{(\boldsymbol{\sigma}_n, \boldsymbol{\sigma}_{n+1}, \mathbf{S}_n):a_n} f_n^{(A)}(\boldsymbol{\sigma}_n) b_{n+1}^{(A)}(\boldsymbol{\sigma}_{n+1}) G_n\left(\mathbf{S}_n; \hat{\boldsymbol{\theta}}_n^{(F,i)}\right) \beta_n(\mathbf{S}_n)$$

with β_n a so-called binding factor that quantifies the agreement between $\hat{\theta}_n^{(F,i)}$ and $\hat{\theta}_n^{(B,i)}$, given by

$$\beta_n(\mathbf{S}_n) = e^{\{-\frac{1-\lambda}{\lambda(2-\lambda)} \frac{E_s}{N_0} |e^{j\tilde{\theta}_n^{(F,i)}(\mathbf{S}_n)} - e^{j\hat{\theta}_n^{(B,i)}}|^2\}}$$
$$\tilde{\theta}_n^{(F,i)}(\mathbf{S}_n) = \hat{\theta}_n^{(F,i)} + \lambda \cdot \Im\left\{e^{-j\hat{\theta}_n^{(F,i)}} z_n(\mathbf{S}_n)\right\}$$

where λ is the PLL parameter in (24) and (26). According to, this procedure is an approximation, obtained intuitively from the framework in.

4) The messages $u_n^{(A)}(a_n)$ are passed on to the interleaving and mapping constraint node of the FG.

The resulting receiver operation is schematized in Fig. 3(d) and summarized in Algorithm 3. As compared to Algorithm 2 the following is observed regarding Algorithm 3:

- The operation of the CPM-related SISO module has been altered (compare steps ALT-2 and COH-2).
- No initial phase estimate is needed to start the iterations in Algorithm 3.

Algorithm 3 Initialization and scheduling strategy of the A-SISO-sing receiver [15].

ALT-1 The FG messages are initialized as in step COH-1 of Algorithm 1.

For $i = 1, 2, ..., N_{it}$

ALT-2 An appropriately adapted SISO module performs
- a forward recursion from $n = 1$ to $n = N$ to jointly compute $f_{n+1}^{(A)}(\boldsymbol{\sigma}_{n+1})$, $\hat{p}\left(\mathbf{S}_n \left| \mathbf{r}_n^{(F)}; \hat{\boldsymbol{\theta}}_n^{(F,i)} \right.\right)$ and $\hat{\theta}_{n+1}^{(F,i)}$,
- a backward recursion from $n = N$ to $n = 1$ to jointly compute $b_n^{(A)}(\boldsymbol{\sigma}_n)$, $\hat{p}\left(\mathbf{S}_n \left| \mathbf{r}_n^{(B)}; \hat{\boldsymbol{\theta}}_n^{(B,i)} \right.\right)$ and $\hat{\theta}_{n-1}^{(B,i)}$,
- a computation of $u_n(a_n)$, $n = 1, 2, ..., N$.

ALT-3 Steps COH-3 to COH-5 of Algorithm 1 are executed.

□

ALT-4 The upward messages along the edges $\{b_k\}$ are computed and MAP bit detection is performed.

- Two different approximations of the CPM state transition APPs are computed in Algorithm 3, against only one approximation in Algorithm 2.

The net result is that Algorithm 3 is more complex than Algorithm 2. Similarly as for the proposed receiver algorithm, a forward-only version of the A-SISO-sing algorithm is obtained when the backward phase estimation procedure is removed. In that case, the backward phase estimate is set equal to the forward phase estimate and the binding factor disappears.

V. Complexity Considerations

The complexity of the proposed and the existing algorithms is now addressed. The number of operations (comparisons, additions, and multiplications) between two real arguments, per CPM symbol and per receiver iteration, is reported. The computational complexity of the front-end processing and of the SISO decoder and demapper algorithms is not considered, being the same for all algorithms. The computational complexity of the dp algorithm is evaluated assuming 8 discretization levels for the carrier phase and 5 possible transitions between these levels, per symbol period. Table I provides analytical closed-form expressions, whereas Table II presents absolute values for the CPM signals that are considered in Section VI (Table III). From the tables it is clear that the proposed receiver has a significantly lower complexity than most of receiver schemes from. The only exception is the A-SISO-sing algorithm from [15]. The complexity of the A-SISO-sing bialgorithm is only slightly higher than that of the SISO+PLL BiS receiver. The A-SISO-sing FO algorithm has the lowest complexity; it is also less complex than the proposed SISO+PLL FO receiver. This is due to the use of hard rather than soft decisions on the transmitted waveforms [(33) as opposed to (18)].

VI. Numerical Performance Results

Computer simulations have been run to assess the performance of the proposed detector, in terms of packet error rate

TABLE I
RELATIVE COMPUTATIONAL LOAD, PER CPM SYMBOL AND PER RECEIVER ITERATION

CPM signal model	SISO+PLL (proposed)		[18]SISO+ FF LMMSE	[13] non-coh.		[14], [15]A-SISO			
	BiS	FO		Tikh.-SISO	dp-SISO	bi	FO	sing, bi	sing, FO
Correct	$16N_\sigma M + N_\sigma + 6$	$16N_\sigma M + N_\sigma + 2$	$16N_\sigma M + N_\sigma + 2N$	$13N_\sigma^2 M + 6N_\sigma M + 12N_\sigma + 16$	$408N_\sigma M$	$16N_\sigma M + 14N_\sigma$	$12N_\sigma M + 3N_\sigma$	$16N_\sigma M + N_\sigma + 13$	$12N_\sigma M + N_\sigma + 2$
Tr. Lau.	$16pM + p + 6$	$16pM + p + 2$	$16pM + p + 2N$	$13p^2 M + 6pM + 12p + 16$	$408pM$	$16pM + 14p$	$12pM + 3p$	$16pM + p + 13$	$12pM + p + 2$

TABLE II
ABSOLUTE COMPUTATIONAL LOAD, PER CPM SYMBOL AND PER RECEIVER ITERATION

CPM		SISO+PLL (proposed)		[18] SISO+	[13] non-coh.		[14], [15] A-SISO			
signal model	from set-up #	BiS	FO	FF LMMSE	Tikh.-SISO	dp-SISO	bi	FO	sing, bi	sing, FO
Correct	0	72	68	4 166	1 68	1 632	92	54	79	52
	1	1 826	1 822	3 390	41 792	45 696	2 184	1 428	1 833	1 374
	2	5 206	5 202	6 640	335 696	130 560	6 240	4 080	5 213	3 922
Tr.	0	72	68	4 166	1 68	1 632	92	54	79	52
Lau.	1	461	457	2 025	2 816	11 424	546	357	468	345
approx.	2	331	327	1 765	1 496	8 160	390	255	338	247

TABLE III
DETAILS OF THE CONSIDERED BIC-CPM SCHEMES

#	T [sec.]	$f(t)$	L	h	M	code	N_d	i_{max}	$\frac{E_s}{N_0}$ @ PER 10^{-4}	PN model	R_s
0	T	REC	1	1/2	2	CC (7,5)	2050	10	-0.5	Wiener, σ_w (stdv increments over T)	16
1	16 μ	RC	2	3/7	4	(64,51) eBCH	720	15	3.85	mask Table IV	8
2	7.5 μ	RC	3	1/5	4	(64,51) eBCH	655	15	10.6	mask Table IV	8

(PER). Hard decisions on the information bits are made at every iteration, after which a genie checks for bit errors; the receiver stops iterating after a maximum number of iterations $i_{\max}$, or when all information bits have been detected correctly. In a practical system, the genie is replaced by a cyclic redundancy check. For comparison, the PER performance of the coherent detector (referred to as SISO) with ideal knowledge of the carrier phase is also generated. It serves as a lower bound to the PER performance of the proposed receiver. Three simulation setups, whose details are summarized in Table III, are considered. Each setup employs a different BIC-CPM scheme. No adjacent carriers are considered. Gray mapping and pseudorandom bit interleaving are applied. Rectangular (REC) and raised-cosine (RC) frequency pulses $f(t)$ are used

$$f(t) = \begin{cases} \frac{1}{2LT} & \text{REC} \\ \frac{1}{2LT}\left(1 - \cos\left(\frac{2\pi t}{LT}\right)\right) & \text{RC} \end{cases} \quad t \in [0, LT]. \tag{34}$$

For each setup, the number of samples per symbol period R_s is chosen such that most of the signal power is located in the frequency band $[-\frac{R_s}{2T}, \frac{R_s}{2T}]$. The setups are labeled (0, 1, 2) in order of increasing spectral falloff rate of the corresponding CPM signals[3]. For setup #0, R_s is set to 16, while for setups #1 and #2 we have taken $R_s = 8$. Two burst structures are employed; they are sketched in Fig. 4. Burst structure #1 is a simple one, with L_{pre} preamble symbols preceding the actual data symbols. Burst structure #2 was used in a recent project funded by the European Space Agency (ESA). Two different PN models are also considered. More details on the respective PN models will be provided later on.

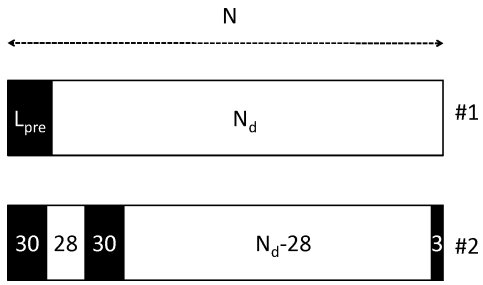

Fig. 4. Considered burst structures.

Let us first focus on simulation setup #0. This setup was also used in [14] and [13] to evaluate the performance of the A-SISO

[3]The smoother the phase response $q(t) = \int_0^t f(t)dt$, the faster the tail of the spectrum of the corresponding CPM signal drops to zero.

algorithms in which multiple PLLs are utilized (higher complexity). The system involves a nonrecursive rate-1/2 convolutional code with generator polynomials $[5_8, 7_8]$, and a minimum shift keying modulation, i.e., CPM with $h = \frac{1}{2}$, $M = 2, L = 1$ and a rectangular frequency pulse [see (34)]. The size of the information bit vector is 1024. A preamble of $L_{\rm pre}$ known symbols precedes the actual data symbols. The phase noise process is modeled as a (discrete) standard Wiener process, i.e.

$$\begin{aligned}\theta(l'T_s) &= \theta((l'-1)T_s) + w_{l'}, l' \\ &= R_s, R_s+1, \ldots, (N+1)R_s - 1\end{aligned}$$

with the initial carrier phase $\theta(T)$ uniformly distributed in $[-\pi, \pi]$ and $\{w_{l'}\}$ statistically independent Gaussian random variables with zero mean and standard deviation $\sigma_w/\sqrt{R_s}$, descriptive of the phase noise intensity; σ_w is the standard deviation of the PN increment over a symbol interval. Note that $R_s = 16$ samples of $\theta(t)$ are generated per symbol interval, whereas the receiver assumes $\theta(t)$ to be constant over a symbol interval. In the literature, the Wiener phase noise model is often used to describe the phase noise process of a free-running oscillator [14], [15], [13], [12], [19], although also more elaborate models exist [1]. For each set of parameters $(\frac{E_s}{N_0}, \sigma_w)$, the step size λ of the utilized PLLs is fixed to $\lambda = \sigma_w\sqrt{2E_s/N_0}$. This choice of λ is known to minimize the linearized steady-state MSPE caused by Wiener PN and AWGN, resulting from a data-aided (DA) PLL that knows the transmitted waveform in advance[4]. Burst structure #1 from Fig. 4 is used.

Fig. 5 shows, for $\sigma_w = 5$ degrees and $L_{\rm pre} = 30$, the PER performance obtained with the receiver algorithms: SISO+PLL BiS, SISO+PLL FO and A-SISO-sing FO. Results are presented after 5 and 10 receiver iterations. The PER performance after 10 iterations of the SISO+PLL FO receiver is also shown for $L_{\rm pre} = 7$ and $L_{\rm pre} = 0$ (no overhead). It was verified that a maximum of 10 iterations suffices for all receivers to converge, having phase information available (SISO) or not (SISO+PLL, A-SISO-sing) at the receiver. As far as this simulation setup is concerned, we make the following observations:

- The SISO+PLL FO receiver (triangular markers, continuous line) is seen to perform better than the SISO+PLL BiS receiver (squared markers). Similar observations were reported in [14], [13] with respect to the A-SISO bi and A-SISO FO algorithms. The basic explanation can be found in Section III-E. The considered BIC-CPM system operates at low values of E_s/N_0 and has a large block length (2052 coded data symbols); therefore, cycle slipping is a major performance limiting factor when using bidirectional phase smoothing.
- A-SISO FO (triangular markers, dashed lines) yields about the same performance as SISO+PLL FO (triangular markers, continuous line). With respect to the SISO benchmark curve (circular markers), a degradation of less than 1 dB is observed at a PER of 10^{-4}.

[4]Computing the linearized steady state MSPE for the iterative phase estimation algorithm seems impossible, because of the iterations involved and the dependence of the soft decisions on all the previous phase (vector) estimates. Instead we will proceed assuming that, at the normal operating SNR of the considered receiver, the MSPE resulting from the iterative CA phase estimator converges to the MSPE resulting from the fictitious DA phase estimator.

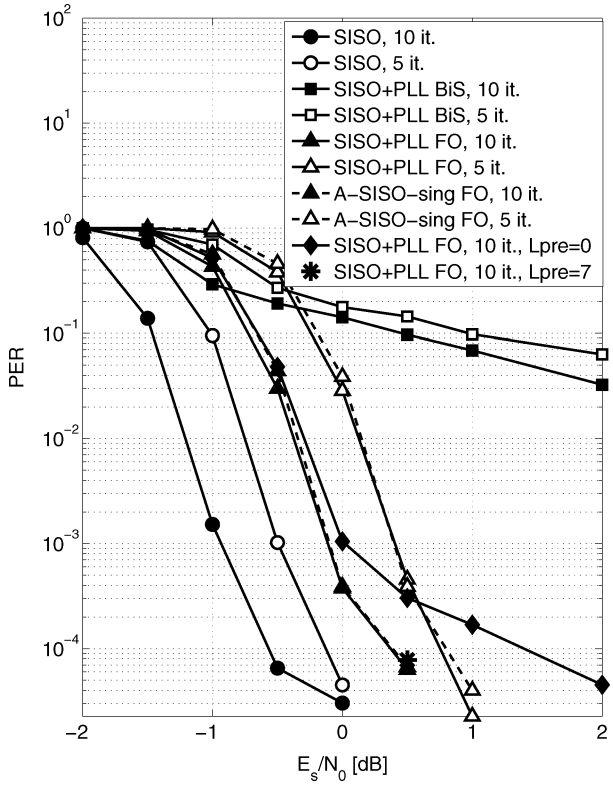

Fig. 5. PER versus E_s/N_0 for simulation setup #0 from Table III, Wiener phase noise with $\sigma_w = 5$ degrees.

- The SISO+PLL FO receiver does not strictly require an overhead of 30 preamble symbols (black triangular markers). Virtually the same performance can be achieved with a smaller overhead of only 7 preamble symbols. At $E_s/N_0 = 0.5$ dB, a marginal PER degradation is observed (gray asterisk). The complete absence of overhead symbols (black diamond-shaped markers) results in a E_s/N_0 degradation of about 2 dB at a PER of 10^{-4}, with respect to the SISO benchmark curve (circular markers). The use of 7 preamble symbols results in a E_s/N_0 power gain of about 1 dB at a PER of 10^{-4}. This power gain does not way against the power loss caused by these preamble symbols (and which is not included in this plot). This power loss amounts 0.01 dB for $L_{\rm pre} = 7$ (0.06 for $L_{\rm pre} = 30$).

Fig. 6 shows how the value of σ_w affects the receiver performance, giving an indication of how much phase noise we can tolerate. The FO variants of both the SISO+PLL and the A-SISO-sing receiver are considered. PERs are compared for $L_{\rm pre} = 30$ and for 10 iterations. We observe that both algorithms yield about the same PER performance for $\sigma_w = 3$ and 5 degrees. For higher values of σ_w, the PER curve of A-SISO-sing algorithm decays faster than that of the SISO+PLL algorithm at first and then flattens out. For $\sigma_w = 7$ degrees, SISO+PLL FO exhibits a small E_s/N_0 degradation of about 0.25 dB at a PER of 10^{-4} with respect to A-SISO-sing FO.

In Figs. 7–10, results are presented for setup #1 and setup #2 from Table III. The corresponding BIC-CPM schemes were recently proposed in [24]. Both schemes use quaternary $(M = 4)$

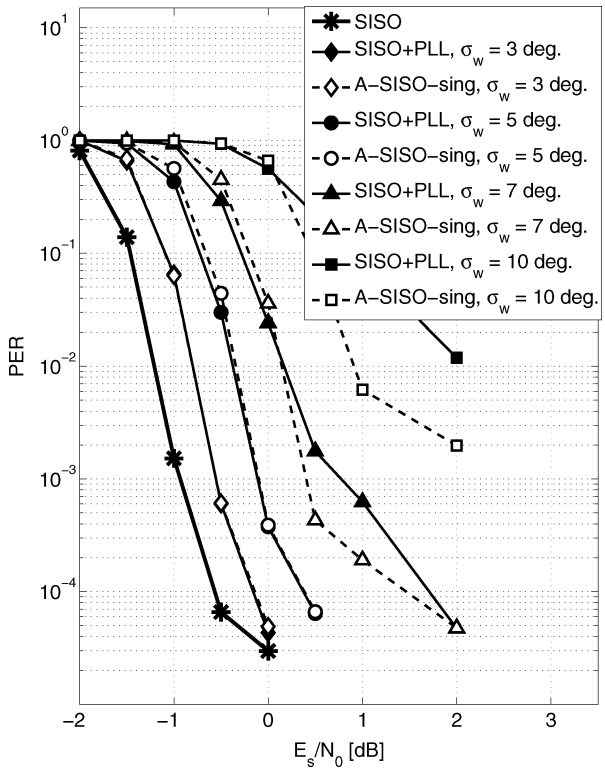

Fig. 6. PER versus E_s/N_0 for simulation setup #0 from Table III, FO PLL operation.

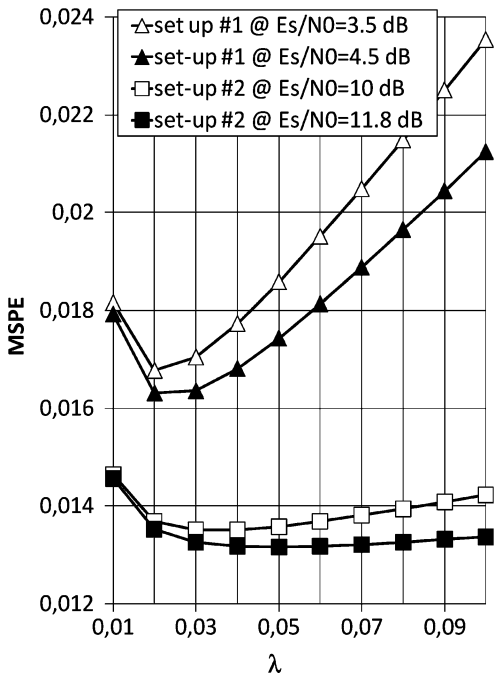

Fig. 7. Analytical linearized steady-state MSPE for DA FO PLL operation.

CPM with a raised-cosine frequency pulse [see (34)]. The information bit block size is 1024, the outer code is a (64, 51) extended BCH code. The receiver performs a maximum of 15 iterations. A different (more realistic) PN model is adopted. The corresponding single-sideband phase noise power spectral density (SSB PSD) mask $\mathcal{L}_{\rm mask}(\Delta f)$, which was specified for use in the previously referred ESA project, is outlined in Table IV; here, Δf denotes the frequency offset with respect to the carrier (oscillator) frequency. The SSB PSD $\mathcal{L}(\Delta f)$ defines the decay of the oscillator power spectral density (symmetrically) around the first harmonic; it is a practical measure to describe the oscillator behavior. The details of the PN generation (R_s=8 phase noise samples per symbol interval) are outlined in Appendix. The symbol durations T that are combined with this phase noise model were part of the same ESA project framework. The parameter λ that controls the PLL noise equivalent bandwidth is optimized in a similar way as before. The steady-state MSPE resulting from a data-aided PLL is computed analytically, taking into account the PN mask from Table IV. The resulting expression is plotted in Fig. 7 as a function of the loop parameter λ and for several values of E_s/N_0. There exists an optimum value for λ that minimizes the steady-state MSPE at the output of the PLL. This optimum value is quasi independent of E_s/N_0 within the normal operating range of the considered BIC-CPM schemes. We set: $\lambda_{\rm opt} \approx 0.02$, for setup #1 and $\lambda_{\rm opt} \approx 0.04$, for setup #2.

Fig. 8 applies to setup #1, for which $L = 2, h = 3/7$ and $T = 16 \cdot 10^{-6}$ seconds, yielding a spectral efficiency of 0.64 bits per second per Hertz. Both burst structures from Fig. 4 are considered. We make the following observations:

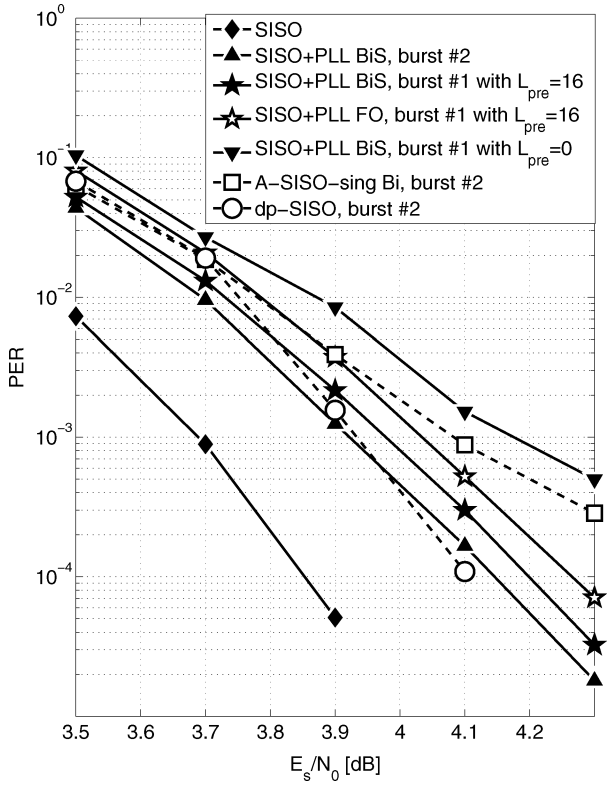

Fig. 8. PER versus E_s/N_0 for simulation setup #1 from Table III. Comparison of detection algorithms.

- For burst structure #2, which was employed in the referred ESA project, the E_s/N_0 degradation (with respect to the SISO benchmark) of the proposed SISO+PLL BiS receiver amounts to about 0.4 dB at a PER of 10^{-4}. This is about the same performance as achieved with the (much more complex) dp-SISO approach. The SISO+PLL BiS receiver

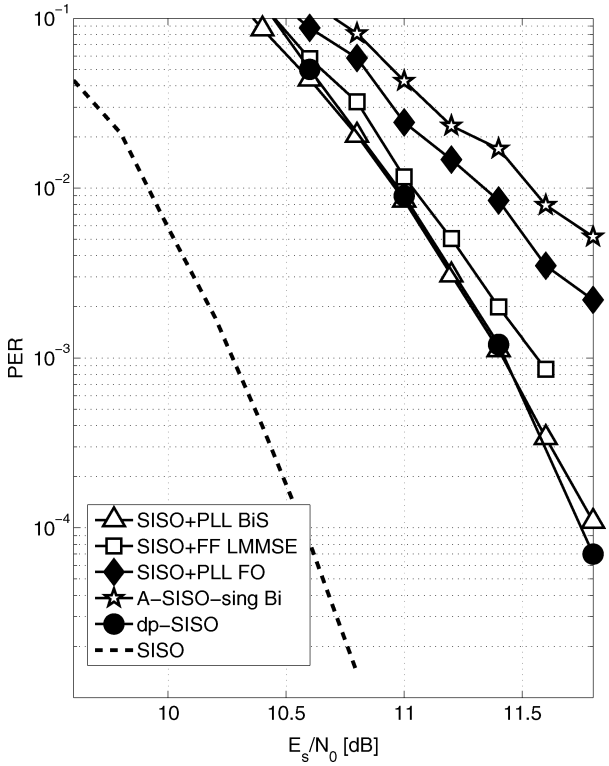

Fig. 9. PER versus E_s/N_0 for simulation setup #2 from Table III. Comparison of detection algorithms.

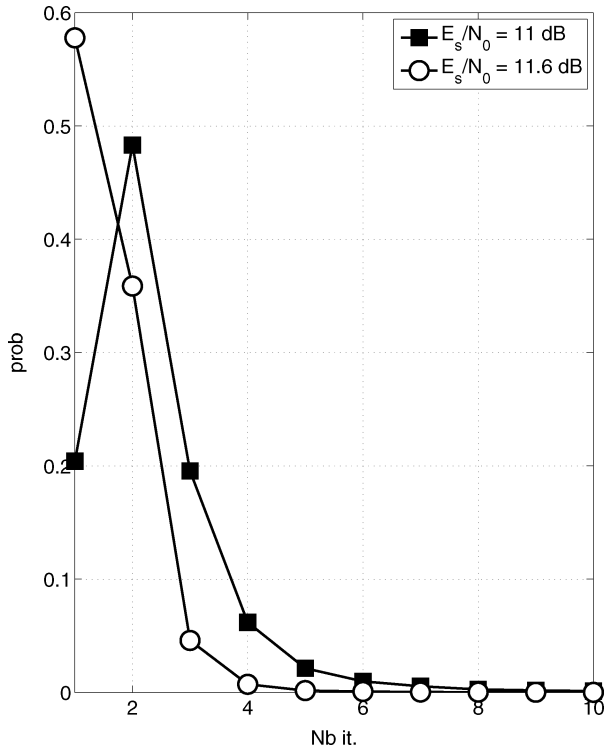

Fig. 10. Distribution of the number of iterations for simulation setup #1 from Table III.

is about 0.25 dB better than the A-SISO-sing Bi receiver at $\mathrm{PER} = 3 \cdot 10^{-4}$. We note that the overhead symbols cause an additional power loss of about 0.4 dB (this is not included in the plot).

- For burst structure #1 with $L_{\mathrm{pre}} = 16$, which yields a significantly lower overhead than burst structure #2, the bidirectional SISO+PLL BiS algorithm performs better than the receiver with SISO+PLL FO operation. The E_s/N_0 degradation of the SISO+PLL BiS algorithm as compared to using burst structure #2 is small.
- For the SISO+PLL BiS approach, the absence of overhead symbols results at a PER of $5 \cdot 10^{-4}$ in a E_s/N_0 degradation of 0.25 dB with respect to using burst structure #1 with $L_{\mathrm{pre}} = 16$ (while the power loss caused by this preamble amounts to only 0.1 dB).

TABLE IV
PHASE NOISE MASK

Δf [Hz]	10	10^2	10^3	$2 \cdot 10^4$	10^6
$\mathcal{L}_{mask}(\Delta f)$ [dBc/Hz]	-22	-48	-68	-68	-103

TABLE V
MEAN AND STANDARD DEVIATION OF THE NUMBER OF PERFORMED RECEIVER ITERATIONS

	E_s/N_0 = 11 dB	E_s/N_0 = 11.6 dB
mean	2.5	1.5
standard deviation	1.7	0.7

Figs. 9 and 10 present results for simulation setup #2 from Table III and burst structure #2 from Fig. 4. In this case, $L = 3, h = 1/5$ and $T = 7.5$ microseconds. Fig. 9 shows the PER versus E_s/N_0. The power loss caused by the overhead symbols (about 0.4 dB) is not included in this plot. The performance of the proposed algorithm (both the bidirectional and forward-only variant) is compared against that of the coherent SISO detector with ideal phase information and against that of the receivers adopting the A-SISO-sing bi, SISO+FF LMMSE and dp-SISO approaches. It can be observed that, as far as this simulation setup is concerned, the proposed bidirectional algorithm significantly outperforms its forward-only variant. Furthermore, it performs almost as well as the dp-algorithm, which is considerably more complex. The E_s/N_0 degradation compared to the case of coherent SISO detection is about 1.2 dB at a PER of 10^{-4}. The A-SISO-sing algorithm exhibits a noticeable performance degradation; at a PER of 10^{-2}, the E_s/N_0 degradation amounts to about 0.8 dB.

Finally, we address the number of iterations that needs to be performed by the receiver to achieve error-free detection. We focus on simulation setup #2 and on the proposed receiver with SISO+PLL BiS operation, which yields the best PER performance for this setup. The simulation setup is the same as in Fig. 9, except for the maximum number of receiver iterations, which was set to 30. For two particular values of E_s/N_0 ($E_s/N_0 = 11$ dB corresponding to a PER of 10^{-2} and $E_s/N_0 = 11.6$ dB corresponding to a PER of $3 \cdot 10^{-4}$), Fig. 10 shows the distribution of the number of iterations. The mean and standard deviation of the number of iterations is very small, as shown in Table V.

VII. Conclusion

In this paper, we have derived a practical PN tracking procedure that is well suited for application in the receiver of an interleaved coded CPM system. The technique combines ideas from [12], [19], and [14]. The resulting receiver is closely related to the single-estimator versions of the A-SISO detectors for PN affected CPM presented in [14] and [15]. However, as opposed to [14] and [15] a modification of the detector operation itself is not required, so that an off-the-shelf coherent bit detector can be used. The proposed detector yields about the same performance penalty as the receiver with suboptimal noncoherent dp-SISO operation from [13]. The computational overhead introduced by the PN tracking is negligibly small as compared to the overall detector complexity. A short preamble, preceding the actual data symbols, is desirable (but not strictly required) for initialization purposes. For at least two BIC-CPM schemes of practical importance the proposed algorithm outperforms the A-SISO-sing algorithm from [15], which also exhibits a slightly higher complexity.

Appendix

The PN generator is implemented as the output of a cascade of two first order filters driven by zero-mean unit-variance white Gaussian noise $\{w_k\}$

$$\begin{aligned}\theta(kT_s) &= \theta((k-1)T_s) + \phi_{k-1},\\ k &= R_s, R_s+1, \ldots, (N+1)R_s - 1 \qquad (35)\\ \phi_k &= z_{\rm pole}\phi_{k-1} + Cw_k - Cz_{null}w_{k-1}\\ z_{\rm null} &= e^{-2\pi f_{\rm null}T_s}, \quad \text{with } f_{\rm null} = 10^3{\rm Hz}\\ z_{\rm pole} &= e^{-2\pi f_{\rm pole}T_s}, \quad \text{with } f_{\rm pole} = 2\cdot 10^4{\rm Hz}\\ C &= 10^{-3.4}\sqrt{T_s}(1-z_{\rm pole}). \qquad (36)\end{aligned}$$

The process $\theta(kT_s)$ in (35) has a variance that linearly increases with time; the initial value $\theta(T)$ is modeled as a uniformly distributed random variable in $[-\pi, \pi]$. The steady-state variance of ϕ_k (36) saturates at

$$\sigma^2_{\phi,\infty} = C^2\left(\frac{(z_{\rm null}-z_{\rm pole})^2}{1-z^2_{\rm pole}}+1\right);$$

the initial value ϕ_{R_s} is generated as a random realization of a Gaussian distributed variable with zero-mean and variance $\sigma^2_{\phi,\infty}$. The corresponding phase noise power spectrum is given by

$$\begin{aligned}&S_\theta(e^{j2\pi fT_s})\\ &\quad= \frac{C^2|e^{j2\pi fT_s}-z_{\rm null}|^2}{|e^{j2\pi fT_s}-1|^2|e^{j2\pi fT_s}-z_{\rm pole}|^2}\\ &\quad\approx \frac{63.3957(|f|^2+10^6)}{|f|^2(|f|^2+4\cdot 10^8)}\cdot\frac{1}{T_s} \qquad (37)\end{aligned}$$

which is a good approximation of the PN mask from Table IV. The approximation in (37) holds for $f, f_{\rm pole}$ and $f_{\rm null}$ much smaller than $\frac{1}{2T_s}$.

References

[1] T. E. Parker, "Characteristics and sources of phase noise in stable oscillators," in *Proc. 41st Ann. Freq. Contr. Symp.*, Philadelphia, PA, May 1987, pp. 99–110.

[2] A. Demir, A. Mehrotra, and J. Roychowdhury, "Phase noise in oscillators: A unifying theory and numerical methods for characterization," *IEEE Trans. Circuits Syst.—I: Fund. Theory and Appl.*, vol. 47, no. 5, pp. 655–674, May 2000.

[3] T. Aulin, N. Rydbeck, and C. W. Sundberg, "Continuous phase modulation," *IEEE Trans. Commun.*, vol. 29, no. 3, pp. 196–225, Mar. 1981.

[4] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.

[5] H. Meyr, M. Moeneclaey, and S. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing*, ser. Wiley Series on Telecommun. and Signal Process. New York: Wiley, 1998.

[6] A. H. Loeliger, "An introduction to factor graphs," *IEEE Signal Process. Mag.*, vol. 21, no. 1, pp. 28–41, Jan. 2004.

[7] D. Petrovic, W. Rave, and G. Fettweis, "Effects of phase noise on OFDM systems with and without PLL: Characterization and compensation," *IEEE Trans. Commun.*, vol. 55, no. 8, pp. 1607–1616, Aug. 2007.

[8] V. Syrjälä, M. Valkama, N. N. Tchamov, and J. Rinne, "Phase noise modelling and mitigation techniques in OFDM communications systems," in *Proc. Wireless Telecommun. Symp. 2008*, Prague, Czech Republic, Apr. 2009.

[9] V. Syrjälä and M. Valkama, "Analysis and mitigation of phase noise and sampling jitter in OFDM radio receivers," *Int. J. Microw. Wireless Technol.*, vol. 2, pp. 193–202, Apr. 2010.

[10] J. Bhatti and M. Moeneclaey, "Iterative soft-decision-directed phase noise estimation from a DCT basis expansion," in *Proc. 20th Pers., Indoor and Mobile Radio Commun. Symp. (PIMRC 2009)*, Tokyo, Japan, Sep. 13–16, 2009.

[11] K. R. Narayanan and G. L. Stüber, "Performance of TCCPM with iterative demodulation and decoding," *IEEE Trans. Commun.*, vol. 49, no. 4, pp. 676–687, Apr. 2001.

[12] N. Noels, H. Steendam, and M. Moeneclaey, "Performance analysis of ML-based feedback carrier phase synchronizers for coded signals," *IEEE Trans. Signal Process.*, vol. 55, no. 3, pp. 1129–1136, Mar. 2007.

[13] A. Barbieri and G. Colavolpe, "Simplified soft-output detection of CPM signals over coherent and phase noise channels," *IEEE Trans. Commun.*, vol. 6, no. 7, pp. 2468–2496, Jul. 2007.

[14] Q. Zhao, H. Kim, and G. L. Stüber, "Innovations-based MAP estimation with application to phase synchronization for serially concatenated CPM," *IEEE Trans. Wireless Commun.*, vol. 5, no. 5, pp. 1033–1043, May 2006.

[15] A. Anastasopoulos and K. M. Chugg, "Adaptive iterative detection for phase tracking in turbo-coded systems," *IEEE Trans. Commun.*, vol. 49, no. 12, pp. 2135–2143, Dec. 2001.

[16] A. Anastasopoulos, "Adaptive soft-input soft-output algorithms for iterative detection," Ph.D., Univ. Southern Calif., Los Angeles, Aug. 1999.

[17] C. Herzet, N. Noels, V. Lottici, H. Wymeersch, M. Luise, M. Moeneclaey, and L. Vandendorpe, "Code-aided turbo synchronization," *Proc. IEEE*, vol. 95, pp. 1255–1271, June 2007.

[18] E. Chiavaccini and G. M. Vitetta, "An EM-based MAP detector for CPM signals transmitted over frequency-flat fading channels," in *Proc. IEEE GLOBECOM 2001*, pp. 1142–1146.

[19] J. Yang and B. Geller, "Near optimum low complexity smoothing loops for dynamical phase estimation—Application to BPSK modulated signals," *IEEE Trans. Signal Process.*, vol. 57, no. 9, pp. 3704–3711, Sep. 2009.

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

[21] L. R. Bahl, J. Cocke, F. Jelinek, and R. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 3, pp. 284–287, Mar. 1974.

[22] P. Ho and J. H. Kim, "Pilot symbol-assisted detection of CPM schemes operating in fast fading channels," *IEEE Trans. Commun.*, vol. 44, no. 3, pp. 337–347, Mar. 1996.

[23] U. Mengali and M. Morelli, "Decomposition of M-ary CPM signals into PAM waveforms," *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1265–1275, Sep. 1995.

[24] A. Graell i Amat, C. A. Nour, and C. Douillard, "Serially concatenated continuous phase modulation for satellite communications," *IEEE Trans. Wireless Commun.*, vol. 8, no. 6, pp. 3260–3269, Jun. 2009.

Nele Noels (M'11) received the diploma of electrical engineering and the Ph.D. degree in electrical engineering from Ghent University, Gent, Belgium, in 2001 and 2009, respectively.

She is currently a Postdoctoral Researcher whose work is supported by the Research Foundation-Flanders (FWO Vlaanderen), with the Department of Telecommunications and Information Processing, Ghent University. Her main research interests are in carrier and symbol synchronization. She is the author of several papers in international journals and conference proceedings.

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

He is a Professor with the Department of Telecommunications and Information Processing (TELIN), Gent University. His main research interests are in statistical communication theory, (iterative) estimation and detection, carrier and symbol synchronization, bandwidth-efficient modulation and coding, spread-spectrum, satellite, and mobile communication. He is the author of more than 400 scientific papers in international journals and conference proceedings. Together with Prof. H. Meyr (RWTH Aachen) and Dr. S. Fechtel (Siemens AG), he is a coauthor of the book *Digital Communication Receivers—Synchronization, Channel Estimation, and Signal Processing*.

Dr. Moeneclaey was Editor for Synchronization, for the IEEE TRANSACTIONS ON COMMUNICATIONS during 1992–1994. He served as co-Guest Editor for special issues of the *Wireless Personal Communications Journal* (on Equalization and Synchronization in Wireless Communications) and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (on Signal Synchronization in Digital Transmission Systems) in 1998 and 2001, respectively.

Frederik Simoens received the M.Sc. and Ph.D. degree in electrical engineering from Ghent University, Belgium, in 2003 and 2008, respectively. His Ph.D. focused on modulation technologies for MIMO systems and was sponsored by the Research Foundation-Flanders (FWO).

He then joined the R&D Department, Newtec, Belgium, a company active in the satellite communication industry, where he currently holds the position of Director Communication Technologies. He is coauthor of more than 40 papers in international journals and conference proceedings.

Daniel Delaruelle (M'03) received the M.Sc. degree in electrical engineering in 1983 from the University of Ghent, Belgium.

From 1985 to 1988, he was a Research Engineer with Alcatel, Belgium. In 1988, he joined Newtec, Belgium, where he was employed as researcher and R&D manager for satellite communications systems and networks. He served in the DVB standardization body, where he contributed to the DVB-DSNG, DVB-S2 , and DVB-RCS standards. His current research interests include satellite network architecture, resource scheduling, synchronization, modulation and coding, and digital signal processing.