

# Double Turbo Equalization of Continuous Phase Modulation with Frequency Domain Processing

Barış Özgül, *Member, IEEE*, Mutlu Koca, *Member, IEEE*, and Hakan Deliç, *Senior Member, IEEE*

**Abstract**—In this paper, a doubly-iterative linear receiver, equipped with a soft-information aided frequency domain minimum mean-squared error (MMSE) equalizer, is proposed for the combined equalization and decoding of coded continuous phase modulation (CPM) signals over long multipath fading channels. In the proposed receiver architecture, the front-end frequency domain equalizer (FDE) is followed by the soft-input, soft-output (SISO) CPM demodulator and channel decoder modules. The receiver employs double turbo processing by performing back-end demodulation/decoding iterations per each equalization iteration to improve the *a priori* information for the front-end FDE. As presented by the computational complexity analysis and simulations, this process provides not only a significant reduction in the overall computational complexity, but also a performance improvement over the previously proposed iterative and non-iterative MMSE receivers.

**Index Terms**—Continuous phase modulation, double turbo processing, frequency domain equalization, intersymbol interference.

## I. INTRODUCTION

CONTINUOUS phase modulation (CPM) is a feasible transmission scheme for power- and bandwidth-limited wireless applications owing to its constant envelope property and spectral efficiency [1]. However, optimal detection of CPM signals under frequency-selective multipath fading encounters complexity issues due to the intensive search performed over a single super-trellis for the combined equalization/demodulation (CED) operation [2]. The suboptimal reduced-state soft-input soft-output (SISO) trellis-search algorithms offer lower complexity solutions for CED of CPM as presented in [3] and [4], but the computational load is still exponentially constrained with the modulation and/or channel

Paper approved by L. K. Rasmussen, the Editor for Iterative Detection Decoding and ARQ of the IEEE Communications Society. Manuscript received December 8, 2006; revised June 6, 2007 and December 27, 2007.

This work was supported by TÜBİTAK (Scientific and Technical Research Council of Turkey) under Contract 105E077, and the Boğaziçi University Research Fund under Contract 06A204D. B. Özgül was also supported by the TÜBİTAK Integrated Doctorate Program (BDP). Part of this paper was presented at the IEEE Global Telecommunications Conference, San Francisco, CA, November 2006.

B. Özgül was with the Wireless Communications Laboratory, Department of Electrical and Electronics Engineering, Boğaziçi University, Bebek 34342 Istanbul, Turkey. He is now with the Centre for Telecommunications Value-Chain Research (CTVR), University of Dublin, Trinity College, Ireland (e-mail: baris.ozgul@ctvr.ie).

M. Koca and H. Deliç are with the Wireless Communications Laboratory, Department of Electrical and Electronics Engineering, Boğaziçi University, Bebek 34342 Istanbul, Turkey. H. Deliç is currently visiting the Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands (e-mail: {mutlu.koca, delic}@boun.edu.tr).

Digital Object Identifier 10.1109/TCOMM.2009.02.060674

memory. To alleviate this problem and motivated by the near-optimum error performance of the iterative receivers, an alternative CPM receiver is proposed in [5], where the equalization, CPM demodulation and channel decoding operations are assigned to three separate SISO blocks and the central CPM module is coupled with both the front-end equalizer and the back-end decoder in a doubly-iterative architecture. The most important feature of this receiver is that a soft-information-aided minimum mean-squared error (MMSE) equalizer is used at its front-end instead of a trellis-based algorithm, which presents a low complexity alternative while still achieving a performance close to the “no interference” bound. Notice that computing the MMSE equalizer coefficients requires some cumbersome matrix inversions causing the computational load to be still relatively large. As presented in [6] and [7], by doing the same computation in the frequency domain the complexity can be reduced further while attaining the same and often better performance.

The frequency domain equalization (FDE) approach has also been extended to the equalization of CPM in [8], where the FDE is not equipped with any SISO capability and thus is not suitable for turbo processing. The advantages of frequency domain processing and iterative information exchange are combined in [9], where a turbo linear equalizer (TLE) is presented in which a SISO block-form FDE (BFDE) is followed by the SISO CPM demodulator and channel decoder modules. Here, the soft CPM signal information to start the subsequent equalization iterations are computed from the code bit probabilities obtained from the back-end channel decoder. However, this produces long error bursts due to the inherent modulation memory and thus, the CPM signal probabilities are delivered to BFDE only at certain epoches to break up the error propagation at the expense of obtaining only a slight turbo gain. Moreover, because the proposed FDE operates on blocks of information, it still involves matrix inversions which result in an increased computational cost. For this reason, we propose herein a soft-information-aided FDE for CPM which overcomes the disadvantages of that in [9] and which is used in a doubly-iterative joint CPM equalization and demodulation architecture similar to the one in [5], so as to achieve a better error performance with lower computational complexity compared to the methods in both [5] and [9].

In the proposed receiver, the frequency domain processing of CPM signals is made possible by inserting a cyclic guard interval longer than the channel memory while maintaining the phase continuity of CPM. The FDE is equipped with an *a priori* soft interference canceller (SIC) and an *a posteriori*

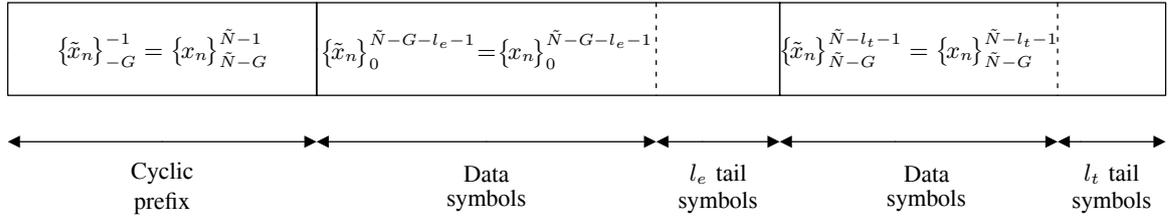


Fig. 1. Modulating sequence with the cyclic prefix.

ori probability mapper, similar to that presented for linear modulations in [10], to generate soft information for the central CPM demodulator. Because the modulator can be represented by a trellis-diagram, a SISO trellis-based decoder is used for demodulation where the decoder computes extrinsic information at its output on both the discrete CPM signals and the coded bits. Then, these two soft outputs are employed in a doubly-iterative information exchange where the CPM demodulator is coupled with both the front-end FDE and the back-end decoder. Because the CPM signal probabilities are not computed from the code bit probabilities, the error bursts of [9] due to the modulation memory are not encountered, which results in a significant performance improvement. Moreover, because its implementation does not involve any matrix inversion, the proposed SISO FDE is computationally less complex than the equalizers in [5] and [9]. The doubly-iterative CPM receiver with the FDE is also more feasible in attaining faster convergence to low bit-error rates (BERs) because the number of equalization iterations are decreased by performing several demodulation/decoding iterations per each equalization iteration to improve the equalizer *a priori* information. This behavior can be justified by a three dimensional extrinsic information transfer (EXIT) chart analysis similar to the one in [5] presented for the doubly-iterative receiver with a time domain MMSE equalizer.

The rest of the paper is organized as follows. In Section II, the CPM signal model is described followed by the presentation of the proposed doubly-iterative CPM receiver and the soft-information-aided FDE algorithm in Section III. Here, we also present the computational complexity analysis of the proposed equalizer and the doubly-iterative receiver, and the comparisons with alternative receiver structures. We also provide a similar comparison in terms of the BER simulations in Section IV and end the paper with concluding remarks in Section V.

## II. SIGNAL MODEL

At the transmitter, a length- $L_d$  data bit sequence with elements  $d_i \in \{0, 1\}$ ,  $i = 0, \dots, L_d - 1$ , is encoded by a rate- $L_d/L_b$  convolutional code to form  $b_l \in \{-1, +1\}$ , where  $l = 0, \dots, L_b - 1$ . Then,  $b_l$  are interleaved to  $c_l$ , which are mapped onto  $M$ -ary symbols  $x_n$ ,  $x_n \in \{\pm 1, \pm 3, \dots, \pm M-1\}$ , and  $n = 0, \dots, N - 1$ , where  $N = L_b/\log_2 M$ . The  $M$ -ary CPM signal with unit amplitude is

$$y(t) = e^{j\varphi(t, \mathbf{x}_0^{N-1})}, \quad 0 < t < NT, \quad (1)$$

where  $\mathbf{x}_0^{N-1}$  denotes the symbol sequence  $\{x_n\}_0^{N-1}$ ,  $\varphi(t, \mathbf{x}_0^{N-1}) = 2\pi h \sum_{n=0}^{N-1} x_n q(t - nT)$  and  $T$  is the symbol interval. Here,  $h$  is the modulation index and  $q(t) = \int_0^t g(\tau) d\tau$

is the phase smoothing function such that  $q(t) = 0$  for  $t < 0$  and  $q(t) = 1/2$  for  $t \geq LT$ , where  $L \geq 1$  denotes memory length of CPM. Similar to the approach in [11], the CPM modulator is represented as the combination of a continuous phase encoder (CPE) with a time-invariant trellis and a memoryless modulator. The CPE considers a tilted phase  $\psi(t, \mathbf{x}_0^{N-1})$  where

$$\psi(t, \mathbf{x}_0^{N-1}) = \varphi(t, \mathbf{x}_0^{N-1}) + \pi h(M-1)t/T. \quad (2)$$

The baseband tilted-phase CPM signal is denoted as

$$z(t) = e^{j\psi(t, \mathbf{x}_0^{N-1})}, \quad 0 < t < NT, \quad (3)$$

where the carrier frequency is adjusted as  $f_c - h(M-1)/2T$  before the transmission to compensate for the frequency shift in (2). Assuming that  $h = K/P$  is a fraction where  $K$  and  $P$  are relatively prime integers, the number of all signals generated by CPE trellis is found as  $Q = PM^L$  [11].

The receiver observes a noisy linear convolution of the transmitted signals with the multipath fading channel. To make frequency domain processing possible and to make the linear and circular convolutions equivalent to each other, a cyclic prefix needs to be appended to the transmitted signal sequence at the expense of an increased redundancy. Here the length of the cyclic prefix,  $G$ , is chosen as the minimum number of symbol periods to avoid interblock interference due to the multipath channel effects. However, the transmission of CPM signals also requires the preservation of phase continuity within each transmitted block and between consecutive transmitted blocks. Observing the similarity of the CPE to a recursive convolutional encoder, this phase continuity can be attained by inserting two sets of tail symbols of lengths  $l_e$  and  $l_t$  after the first  $N - G + l_t$  modulating symbols and to the end of the whole symbol sequence, respectively. Please notice that, choosing  $l_t$  as the minimum number of symbols to return to the initial state from any trellis state and  $l_e \geq l_t$  as the number of symbols to return to and stay at the initial state, this operation assures that the CPE trellis path returns to the initial state at  $n = N - G + l_t + l_e - 1$  and  $n = N + l_t + l_e - 1$ . The symbol sequence with tail symbols becomes  $\{x_n\}_0^{\tilde{N}-1}$  where  $\tilde{N} = N + l_t + l_e$ . Then, the length- $G$  cyclic prefix, selected as the last  $G$  symbols, is appended to the beginning of the symbol block as depicted in Fig. 1 and the new symbol sequence becomes  $\tilde{x}_n = x_{(n)_{\tilde{N}}}$ ,  $n = -G, \dots, -1, 0, 1, \dots, \tilde{N} - 1$ , where  $(\cdot)_{\tilde{N}}$  stands for the modulo- $\tilde{N}$  operation. By compensating for the frequency shift in (2), the CPM signal in (1) produced by the new symbol sequence  $\tilde{\mathbf{x}}_{-G}^{\tilde{N}-1} = \{\tilde{x}_n\}_{-G}^{\tilde{N}-1}$  is denoted as

$$y(t) = e^{j\psi(t, \tilde{\mathbf{x}}_{-G}^{\tilde{N}-1}) - j\pi h(M-1)t/T}, \quad -GT < t < \tilde{N}T. \quad (4)$$

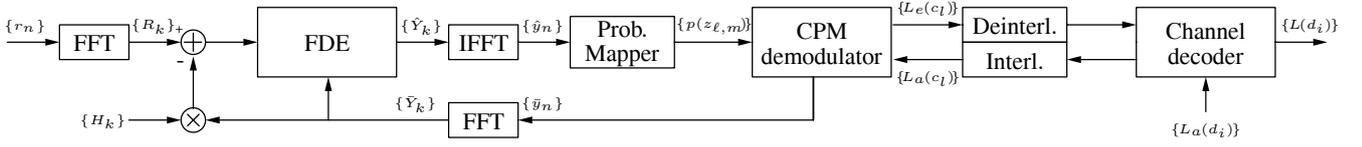


Fig. 2. Receiver structure.

Note that this new sequence ensures that the CPE trellis path for each packet begins and ends at the initial state and therefore no phase discontinuities are encountered during the transitions between consecutive packets. Furthermore, within each packet, the trellis path returns to the initial state after the first  $G$  symbols so that the cyclic guard interval does not disrupt the phase continuity. When  $l_e$  is chosen properly such that  $h\tilde{N}$  is an even integer, (4) yields  $y(t) = y(t + \tilde{N}T)$  on the interval  $-GT < t < 0$ , as also described in detail in [14].

The multipath channel is assumed to be time-invariant throughout the one packet duration such that it can be modelled as

$$h(t) = \sum_{m=0}^{N_c-1} \rho_m \delta(t - \tau_m) \quad (5)$$

where  $\rho_m$  and  $\tau_m$  are the time-invariant complex gain and the propagation delay for the  $m$ th path, respectively, and  $N_c$  is the number of the channel paths. For practical purposes, CPM signal can be considered as band-limited to  $|f| \leq W/2$ . Then, choosing a sampling period  $T_s$  such that  $T_s = T/n_s \leq 1/W$  where  $n_s \in \mathbb{Z}^+$ , the path delays  $\tau_m$  in (5) can be approximated to the integer multiples of  $T_s$ . Accordingly, the fractionally spaced channel impulse function can be written as

$$h(t) = \sum_{\ell=0}^{L_c-1} h_\ell \delta(t - \ell T_s) \quad (6)$$

where  $L_c = \tau_{N_c-1}/T_s + 1$  with  $\tau_{N_c-1}$  being the maximum path delay, and  $h_\ell = \rho_m$  for  $\ell = \tau_m/T_s$  and 0 for all other  $\ell$  values. Then the received signal can be expressed as

$$r(t) = \sum_{\ell=0}^{L_c-1} h_\ell y(t - \ell T_s) + v(t), \quad -GT < t < \tilde{N}T, \quad (7)$$

where  $v(t)$  is the zero-mean additive white Gaussian noise (AWGN) term with variance  $\sigma_v^2$ . After the removal of the prefix and low-pass filtering with a two-sided bandwidth of  $1/T_s$  Hertz, the discrete symbols are obtained by sampling the filter output every  $T_s$  seconds as in [9] such that the additive noise is still white and there is no aliasing. The corresponding discrete-time signal is

$$r_n = \sum_{\ell=0}^{L_c-1} h_\ell y_{(n-\ell)\tilde{N}} + v_n, \quad n = 0, \dots, \tilde{N} - 1, \quad (8)$$

where  $\tilde{N} = n_s \tilde{N}$  and  $r_n \triangleq r(nT_s)$ ,  $y_n \triangleq y(nT_s)$ ,  $v_n \triangleq v(nT_s)$ . Notice that by defining  $\mathbf{h}_0^{\tilde{N}-1} \triangleq \{h_n\}_0^{\tilde{N}-1}$  that is obtained through zero padding after the first  $L_c$  terms and  $\mathbf{y}_0^{\tilde{N}-1} \triangleq \{y_n\}_0^{\tilde{N}-1}$ , (8) can be observed as a noisy circular convolution and can be rewritten as

$$r_n = [\mathbf{h} \star \mathbf{y}]_n + v_n, \quad n = 0, \dots, \tilde{N} - 1. \quad (9)$$

Here  $[\mathbf{h} \star \mathbf{y}]_n$  denotes the  $n$ th element of the circular convolution of the sequences  $\mathbf{h}_0^{\tilde{N}-1}$  and  $\mathbf{y}_0^{\tilde{N}-1}$  whose indices are dropped for notational simplicity.

### III. DOUBLY-ITERATIVE CPM RECEIVER WITH FREQUENCY DOMAIN EQUALIZATION

The proposed doubly-iterative CPM receiver is shown in Fig. 2. Initially the FDE iteration starts with no *a priori* information, and no interference cancellation takes place. At the output of this process, within each symbol interval,  $n_s$  samples are mapped onto a  $Q$ -ary vector of extrinsic probabilities so as to start the back-end iterations between the CPM demodulator and the channel decoder. Both modules are implemented by the log-domain *a posteriori* probability (APP) algorithm ([12], pp. 570). The demodulator generates extrinsic information on both the coded bits  $c_n$  in the form of log-likelihood ratios (LLRs) and the tilted-phase CPM signals in the form of  $Q$ -ary vectors. The former is exchanged within the back-end iterations, where the latter is used to compute the expected values,  $\bar{y}_n$ , to start the next equalization iteration after any number of back-end iterations. Both the log-domain APP algorithm and its application to CPM demodulation are well-known, and the interested reader is referred to [12] for the former and [5] for the latter. Below, we describe only the operation of the front-end FDE algorithm.

The front-end equalizer applies discrete Fourier transform (DFT), SIC/FDE, and inverse DFT (IDFT) operations consecutively to obtain the outputs from which the soft information to the demodulator is calculated by the probability mapper. To describe the derivation of the SIC/FDE algorithm, we first present its time domain equivalent and then obtain the corresponding representation in the frequency domain. Let us define  $\mathbf{w} \triangleq \{w_n\}_0^{\tilde{N}-1}$ ,  $\bar{\mathbf{y}} \triangleq \{\bar{y}_n\}_0^{\tilde{N}-1}$ , and  $\mathbf{v} \triangleq \{v_n\}_0^{\tilde{N}-1}$  as the length- $\tilde{N}$  sequences collecting the equalizer coefficients, the mean values for the discrete-time CPM symbols and the noise samples, respectively, and assume that  $y_n$ , the sample of  $y(t)$  taken at  $t = nT_s$ , is the target symbol at the  $n$ th time instant. In a conventional time domain SIC/MMSE equalizer receiver, the discrete time signal observed at the output of the equalizer after the SIC and equalization operations are applied to the received signal in (9) is found as

$$\hat{y}_n = [\mathbf{w} \star \mathbf{h} \star \mathbf{y}]_n + [\mathbf{w} \star \mathbf{v}]_n - [\mathbf{w} \star \mathbf{h} \star \bar{\mathbf{y}}]_n + \mu \bar{y}_n \quad (10)$$

where  $\mu = \sum_{\ell=0}^{\tilde{N}-1} w_\ell h_{\tilde{N}-\ell}$  is the total gain of the  $y_n$  terms in  $[\mathbf{w} \star \mathbf{h} \star \mathbf{y}]_n$  and  $\mu \bar{y}_n$  prevents the cancellation of the symbol information at time  $n$ . This value can be computed in both the time domain and frequency domain by employing Plancherel's Theorem, which leads to

$$\mu = \sum_{\ell=0}^{\tilde{N}-1} w_\ell h_{\tilde{N}-\ell} = \frac{1}{\tilde{N}} \sum_{k=0}^{\tilde{N}-1} W_k H_k \quad (11)$$

TABLE I  
COMPLEXITY OF THE SISO MODULES USED BY THE PROPOSED RECEIVER, THE RECEIVERS IN [5] AND [9], AND TAE

	FDE	TDE [5]	BFDE [9]	CED [2]	CPM Demodulator	Channel Decoder
Real multiplications and additions	$O(B)+$ $O(n_i B \log_2 B)+$ $O(n_i B P M^L)$	$O(L_f^3)+$ $O(n_i B L_f)+$ $O(n_i B P M^L)$	$O(n_i B^3)+$ $O(n_i B P M^L)$	$O(n_i B P M^{L+M_c-1})$	$O(n_i B P M^L)$	$O(n_i B 2^{m_c+1})$
$B, n_i, L_f, M_c, L, M, P,$ and $m_c$ denote the block length, number of iterations, length of the TDE filter, length of the multipath channel in terms of symbol intervals, memory length of CPM, modulation order, denominator of the modulation index, and the memory length of the convolutional code, respectively.						

where  $W_k$  and  $H_k, k = 0, 1, \dots, \bar{N}-1$ , are the  $\bar{N}$ -point DFTs of  $w_n$  and  $h_n$ , respectively. Notice that the computation of the MSE-optimum equalizer coefficients in time domain requires large matrix inversions causing a computational burden. That is why the equalization method proposed in this paper is the frequency domain equivalent of the time domain operation in (10). In frequency domain, (10) corresponds to

$$\hat{Y}_k = W_k R_k - W_k H_k \bar{Y}_k + \mu \bar{Y}_k, \quad k = 0, 1, \dots, \bar{N} - 1, \quad (12)$$

where  $R_k = H_k Y_k + V_k$  and  $Y_k, \hat{Y}_k, \bar{Y}_k, R_k,$  and  $V_k$  are the  $\bar{N}$ -point DFTs of  $y_n, \hat{y}_n, \bar{y}_n, r_n,$  and  $v_n$ , respectively. Here, the frequency domain filter coefficients  $\{W_k\}$  having the MMSE solution are computed by minimizing

$$E \left[ \left| Y_k - \hat{Y}_k \right|^2 \right] = E \left[ \left| Y_k - W_k R_k + W_k H_k \bar{Y}_k - \mu \bar{Y}_k \right|^2 \right]. \quad (13)$$

Notice that the  $W_k$  values need to be updated at each equalizer iteration since different  $\bar{Y}_k$  values are delivered from the CPM demodulator at each front-end iteration. This complexity can be reduced by computing two different sets of equalizer coefficients under either zero *a priori* information (ZAI) or full *a priori* information (FAI) assumptions as in [13] and by starting the initial front-end iterations with the ZAI coefficients and then switching to the FAI coefficients after a few iterations.

The optimum coefficients minimizing (13) for the ZAI and FAI cases which correspond to  $\bar{Y}_k = 0$  and  $\bar{Y}_k = Y_k$  assumptions, respectively, can be computed as

$$W_k^{\text{ZAI}} = \frac{E \left[ |Y_k|^2 \right] H_k^*}{\bar{N} \sigma_v^2 + E \left[ |Y_k|^2 \right] |H_k|^2}, \quad (14)$$

$$\begin{aligned} W_k^{\text{FAI}} &= \frac{\left(1 - \frac{1}{\bar{N}} \sum_{\ell=0}^{\bar{N}-1} W_\ell H_\ell\right) E \left[ |Y_k|^2 \right] H_k^*}{\bar{N}^2 \sigma_v^2} \\ &= \frac{(1 - \mu) E \left[ |Y_k|^2 \right] H_k^*}{\bar{N}^2 \sigma_v^2}, \end{aligned} \quad (15)$$

where  $*$  denotes the complex conjugation. Notice that, for simplicity,  $E \left[ |Y_k|^2 \right]$  terms in (14) and (15) can be replaced by the average

$$\frac{1}{\bar{N}} \sum_{k=0}^{\bar{N}-1} |Y_k|^2 = \frac{1}{\bar{N}} \sum_{k=0}^{\bar{N}-1} \sum_{n=0}^{\bar{N}-1} \sum_{\ell=0}^{\bar{N}-1} y_n y_\ell^* e^{-j2\pi(n-\ell)k/\bar{N}} = \bar{N}, \quad (16)$$

which is independent of the index  $k$  and obtained through the symmetry of the DFT operation and the unit-amplitude CPM signals. Then, using this replacement and the joint solution of (11) and (15) for  $\mu$  together, the equalizer coefficients in (14)

and (15) for the ZAI and FAI cases simplify to

$$W_k^{\text{ZAI}} = \frac{H_k^*}{\sigma_v^2 + |H_k|^2}, \quad (17)$$

$$W_k^{\text{FAI}} = \frac{H_k^*}{\bar{N} \sigma_v^2 + (1/\bar{N}) \sum_{\ell=0}^{\bar{N}-1} |H_\ell|^2}, \quad (18)$$

respectively.

The SIC exploits the mean values,  $\bar{y}_n$ , which are delivered from the demodulator as follows. Denoting the mean value of  $z(t)$  in (3) as  $\bar{z}(t)$ ,

$$\bar{z}_n \triangleq \bar{z}(nT_s) = \sum_{m=0}^{Q-1} p_{\ell,m} Z_{i,m}, \quad n = 0, 1, \dots, \bar{N} - 1, \quad (19)$$

where  $n = \ell n_s + i, \ell = 0, \dots, \bar{N} - 1, i = 0, \dots, n_s - 1$ ,  $p_{\ell,m}$  is the probability generated by the demodulator at the  $\ell$ th symbol interval for the  $m$ th tilted-phase CPM signal, and  $Z_{i,m}$  is the value of the  $m$ th signal at time instant  $iT_s$  over the symbol interval  $[0, T]$ . Using the relation in (2),  $\bar{y}_n$  is obtained as

$$\bar{y}_n = \bar{z}_n e^{-j\pi h(M-1)n/n_s}, \quad n = 0, 1, \dots, \bar{N} - 1. \quad (20)$$

After performing the DFT operations on (9) and (20), and computing the frequency domain outputs with (12) by either using (17) or (18), the time domain outputs,  $\hat{y}_n$ , are found by the IDFT operation. Then, the soft information for CPM demodulator is computed by the probability mapper. For this purpose, the equalizer output with the tilted-phase is found as  $\hat{z}_n = \hat{y}_n e^{j\pi h(M-1)n/n_s}, n = 0, 1, \dots, \bar{N} - 1$ , which can be viewed as  $\hat{z}_n = \mu z_n + \nu_n$  where  $\mu$  is the symbol gain and  $\nu_n$  is the zero-mean complex Gaussian noise term with variance  $\sigma_\nu^2$  [10], [13]. At each equalizer iteration,  $\sigma_\nu^2$  is estimated as

$$\hat{\sigma}_\nu^2 = \frac{1}{\bar{N}} \sum_{n=0}^{\bar{N}-1} \sum_{m=0}^{Q-1} p_{\ell,m} |\hat{z}_n - \mu Z_{i,m}|^2$$

for  $\ell = 0, \dots, \bar{N} - 1, i = 0, \dots, n_s - 1$ , and  $n = \ell n_s + i$ . Here,  $p_{\ell,m}$  are set to  $1/Q$  under the ZAI assumption and the probabilities delivered from CPM demodulator are taken into account for the FAI scenario. Then, defining  $\Gamma_{\ell,m} \triangleq -\sum_{i=0}^{n_s-1} |\hat{z}_n - \mu Z_{i,m}|^2 / \hat{\sigma}_\nu^2$  where  $\ell = 0, \dots, \bar{N} - 1, n = \ell n_s + i$ , and  $m = 0, 1, \dots, Q - 1$ , the probability mapper generates

$$\log [p(z_{\ell,m})] = \Gamma_{\ell,m} - \log \left[ \sum_{m=0}^{Q-1} e^{\Gamma_{\ell,m}} \right]$$

where  $p(z_{\ell,m})$  is the probability of the  $m$ th signal to be fed to CPM demodulator at the  $\ell$ th symbol interval.

TABLE II  
COMPUTATIONAL COMPLEXITY PER SIGNAL BLOCK FOR THE PROPOSED RECEIVER, THE TURBO RECEIVERS IN [5] AND [9], AND TAE

Receiver	Computational Complexity
Turbo FDE with $n_i$ front-end and $n_i$ back-end iterations	$O(n_s \tilde{N}) + O(n_i n_s \tilde{N} \log_2 n_s \tilde{N}) + O(n_i n_s \tilde{N} P M^L) + O(n_i L_b P M^L) + O(n_i L_b 2^{m_c+1})$
Turbo FDE with $n_f$ front-end and $n_i$ back-end iterations	$O(n_s N) + O(n_f n_s N \log_2 n_s N) + O(n_f n_s N P M^L) + O(n_i L_b P M^L) + O(n_i L_b 2^{m_c+1})$
Turbo TDE [5] with $n_i$ front-end and $n_i$ back-end iterations	$O(L_f^3) + O(n_i n_s N L_f) + O(n_i n_s N P M^L) + O(n_i L_b P M^L) + O(n_i L_b 2^{m_c+1})$
Turbo TDE [5] with $n_f$ front-end and $n_i$ back-end iterations	$O(L_f^3) + O(n_f n_s N L_f) + O(n_f n_s N P M^L) + O(n_i L_b P M^L) + O(n_i L_b 2^{m_c+1})$
TLE [9] with $n_i$ iterations	$O(n_i n_s^3 \tilde{N}^3) + O(n_i n_s \tilde{N} P M^L) + O(n_i L_b P M^L) + O(n_i L_b 2^{m_c+1})$
TAE with $n_i$ iterations	$O(n_i N P M^{L+M_c-1}) + O(n_i L_b 2^{m_c+1})$

$N, \tilde{N}, n_s, L_f, M_c, L, M, P,$  and  $m_c$  denote the length of the modulation sequence, length of the modulation sequence with tail symbols, number of samples per symbol period, length of the TDE filter, length of the multipath channel in terms of symbol intervals, memory length of CPM, modulation order, denominator of the modulation index, and the memory length of the convolutional code, respectively.

The computational complexity analysis for the proposed FDE is presented in Table I, along with those of the TDE in [5], the BFDE in [9] and the optimal CED in [2] using the APP algorithm. Also included in the table are the complexities of the SISO CPM demodulator and the channel decoder so that complexity comparisons of different turbo architectures using these blocks can be obtained. Here  $n_i$ ,  $M_c = \lfloor L_c/n_s \rfloor + 1$ ,  $m_c$ , and  $L_f$  denote the number of iterations, length of the multipath channel in terms of symbol intervals with  $\lfloor \cdot \rfloor$  being the floor operator, the memory length of the convolutional code, and the length of the TDE filter which depends on  $L_c$  linearly, respectively. The example below presents the approximate number of real multiplications and additions required throughout a signal block by four different turbo receivers: 1) the proposed turbo receiver with FDE (turbo FDE) in Fig. 2, 2) the turbo receiver with TDE (turbo TDE) in [5] replacing the front-end FDE in Fig. 2 by a TDE filter, 3) the TLE in [9] in which the BFDE is followed by the CPM demodulator and the channel decoder modules, 4) a turbo APP equalizer (TAE) exchanging soft information between the optimal CED module and the channel decoder.

*Example:* It is assumed that  $L_d = 125$  data bits are encoded by a rate-1/2 convolutional code so that  $L_b = 250$  code bits are obtained, where the memory length of the convolutional code is  $m_c = 3$ . The parameters for the CPM scheme is  $M = 2$ ,  $L = 3$  and  $h = 0.5$  with  $P = 2$ . To apply the proposed turbo FDE and the TLE in [9], a cyclic guard interval is added to the transmitted block. For this purpose, four tail symbols are inserted as shown in Fig. 1 where  $l_e = 3$  and  $l_t = 3$ , and the length of the modulating sequence is  $\tilde{N} = 256$ . The turbo receivers with time domain equalization methods in [2] and [5] do not require tail symbols and cyclic guard addition so that the length of the modulating sequence is  $N = 250$ . For the multipath fading channel, it is assumed that  $L_c = 11$ , the channel resolution is  $T_s = T/2$  with  $n_s = 2$ , and, therefore, the maximum tap delay is  $5T$ . After the addition of the cyclic guard interval, the duration of the transmitted packet is  $261T$  with  $G = 5$ . Thus, the number of extra tail and guard symbols employed by the frequency domain methods is only  $l_e + l_t + G = 11$  compared to the time domain

methods. The computational load for the aforementioned turbo receivers are presented in Table II by using the approximate complexity values in Table I for the SISO modules deployed by these receivers. At the first and third rows of Table II, the complexity values for the turbo FDE and turbo TDE are given, respectively, where each of the  $n_i$  front-end iterations is followed by one back-end demodulator/decoder iteration so that  $n_i$  front-end and  $n_i$  back-end iterations are performed in total. As in [5], the TDE filter length is  $L_f = 2L_c$ . Depending on the doubly-iterative architecture of these receivers, it is possible to attain the same performance by fewer equalizer iterations where each front-end iteration is performed after improving the *a priori* information by  $n_i/n_f$  back-end iterations so that the total number of front-end and back-end iterations are  $n_f$  and  $n_i$ , respectively, as shown at the second and fourth rows of Table II. Because the TLE and TAE cannot perform double iterations, all the SISO modules at these receivers are employed throughout  $n_i$  iterations, as presented at the fifth and sixth rows of Table II, respectively. In this example,  $n_f = 4$  and  $n_i = 12$ , which are the same as the values used in the BER simulations in Section IV.

By using the aforementioned parameter values in Table II, it is observed that the computations for the proposed turbo FDE at the first and second rows are less than the ones for the turbo TDE at the third and fourth rows, respectively, by adding only nine extra symbols to the transmitted packet. This is because of the cubic complexity that comes from the matrix inversion operation for TDE and the dependency of the filter length  $L_f$  on the channel length  $L_c$ . Note that the redundancy for turbo FDE to add the cyclic prefix increases linearly in longer channel impulse responses without any computational change whereas the turbo TDE encounters higher cubic complexity. The complexity of TLE does not depend on the length of the channel impulse response. However it is computationally more demanding compared to the proposed turbo FDE as shown at the fifth row of Table II, depending on the matrix inversions required by the BFDE at each iteration. Moreover, it is not possible to perform double iterations at this receiver to reduce the complexity for equalization. Last row of Table II shows that the TAE is also more complex compared to

the proposed method. Here the complexity of the optimal CED in [2] applied by TAE increases exponentially with the length of the channel impulse response. For the proposed turbo FDE, the length of channel impulse response does not have impact on the complexity at the expense of adding redundancy which increases linearly with the channel memory length. Furthermore, the performance of the turbo FDE is better compared to the turbo TDE and TLE and is close to that of TAE, as shown in Section IV.

#### IV. SIMULATION RESULTS

In this section, the BER performance of the proposed turbo FDE is presented for different number of front- and back-end iterations, and is also compared to those of the TAE employing the CED in [2], the turbo TDE in [5], the TLE in [9], and the performance in AWGN channel. The binary three raised-cosine (3RC) CPM with  $L = 3$  and  $h = 0.5$  is considered as in [1] with  $n_s = 2$ , the two-sided bandwidth of the low-pass filter is  $2/T$ , and the channel resolution is  $T_s = T/2$ , as in [9]. The rate-1/2 convolutional code with generator polynomial  $(64, 74)_8$  is used, and random interleaving is applied. First, the performance gap between the optimal and the proposed receiver is observed in a mild Proakis' A channel with coefficients  $[0.04 \ -0.05 \ 0.07 \ -0.21 \ -0.5 \ 0.72 \ 0.36 \ 0.00 \ 0.21 \ 0.03 \ 0.07]$ , where the delay of the  $m$ th path is  $mT_s$  for  $m = 0, 1, \dots, 10$ . The channel coefficients are normalized to have unit total energy. The aforementioned receivers are also compared in a more severe eleven-tap quasi-static channel (channel I) environment with deep spectral nulls, where the tap coefficients are zero-mean complex white Gaussian random variables with exponentially decaying power profile such that the variance of the  $m$ th path coefficient is  $e^{-m/2}/(\sum_{l=0}^{10} e^{-l/2})$  and the corresponding path delay is  $mT_s$  for  $m = 0, 1, \dots, 10$ . Furthermore, the six-tap typical urban channel (channel II) model in [8] is considered, where the variances of the complex Gaussian path coefficients are  $[0.189 \ 0.379 \ 0.255 \ 0.090 \ 0.055 \ 0.032]$  and the corresponding path delays are  $[0 \ T_s \ 2T_s \ 8T_s \ 12T_s \ 25T_s]$ . For all the scenarios considered, the information packets start and terminate at the zero state and consist of 256 symbols including the tail coefficients with  $l_e = 3$  and  $l_t = 3$ . For channels I and II, the duration of the cyclic guard intervals is  $G = 5$  and  $G = 13$  symbol periods, respectively.

The switching condition between ZAI and FAI coefficients is determined by using the corresponding transfer characteristic curves of the FDE. The method in [13] to obtain the equalizer characteristic curves for linear modulations is not applicable here since it relies on the independence of the transmitted symbols, whereas CPM signals are correlated. Thus, the average of the amplitudes of the mean values ( $A$ ) at the input and the average of the squared distances between the equalizer outputs and the transmitted symbols ( $D$ ) at the output of the FDE are considered as the information measures for the characteristic curves, which are computed as

$$A = \frac{1}{N} \sum_{n=0}^{\tilde{N}-1} \left| \sum_{m=0}^{Q-1} p_{\ell,m} Z_{i,m} \right|, \quad D = \frac{1}{N} \sum_{n=0}^{\tilde{N}-1} |z_n - \hat{z}_n/\mu|^2, \quad (21)$$

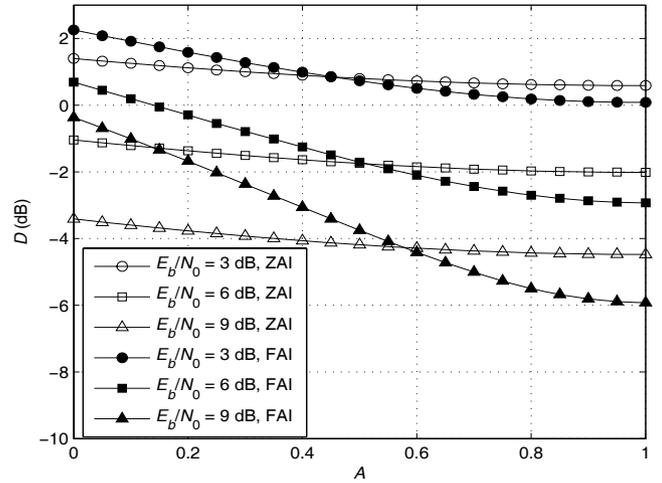


Fig. 3. Characteristic curves for the FDE ( $D$  is the average squared distance between the equalizer outputs and the samples for the transmitted CPM signal and  $A$  is the average of the amplitudes of the mean values used by SIC).

respectively, where  $n = \ell n_s + i$ ,  $\ell = 0, \dots, \tilde{N} - 1$ , and  $i = 0, \dots, n_s - 1$ . As described in [5], for CPM schemes where  $P$  is an even integer,  $A$  is zero for the equally likely CPM symbols, and assuming unit-amplitude symbols, it goes from zero to one if any of the signal probabilities converges to one. At the output of FDE, smaller  $D$  values are obtained with more accurate equalizer outputs. In Fig. 3, the characteristic curves of the equalizers using (17) and (18) are shown for channel I at  $E_b/N_0 = 3, 6$ , and  $9$  dB, by averaging the results for eleven-tap quasi-static channels. The symbol probabilities  $p_{\ell,m}$  from CPM demodulator are generated artificially, where the probability of the transmitted signal at each symbol interval is set as  $p_t$ , and the probabilities corresponding to the remaining  $Q - 1$  CPM signals are set as  $(1 - p_t)/(Q - 1)$  for  $1/Q \leq p_t \leq 1$ . It is unnecessary to perform a front-end equalizer iteration after each back-end demodulator/decoder iteration. The delivery of the *a priori* information to the SIC/equalizer after being improved by a few back-end iterations is adequate for convergence, which is determined by a three dimensional EXIT chart analysis, as in [5].

In Fig. 4, the BER performance of turbo FDE with respect to TAE is depicted for Proakis' A channel and channel I. Moreover, the performance of turbo FDE in channel I is compared to those of turbo TDE and TLE. Both turbo FDE and TDE exploit double iterations where each front-end iteration (FIT) is followed by one back-end iteration (BIT) between the CPM demodulator and the channel decoder. For TLE, the channel decoder feeds soft information to the front-end at each iteration, where there is no significant turbo gain after two iterations, as also described in [9]. The TAE conducts twelve iterations between the optimal combined equalizer/demodulator and the channel decoder. In AWGN channel scenario, the demodulator and the decoder exchange soft information by performing twelve iterations. The proposed receiver yields very close performance compared to TAE in the mild Proakis' A channel. In more severe channels with deep spectral nulls such as the ones in channel I scenario, TAE performs much better at the expense of large complexity

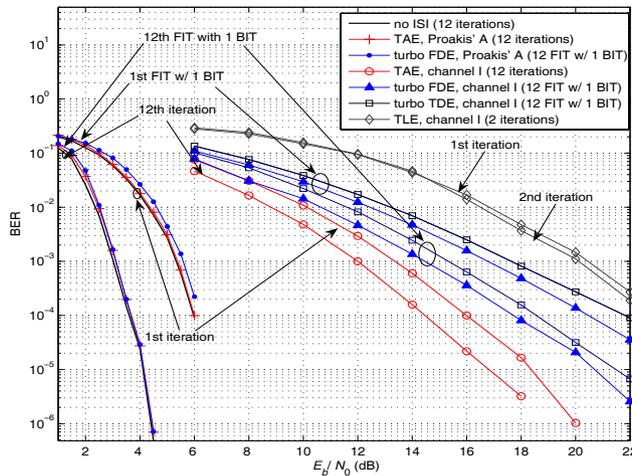


Fig. 4. BER performance for no ISI, Proakis' A channel and channel I.

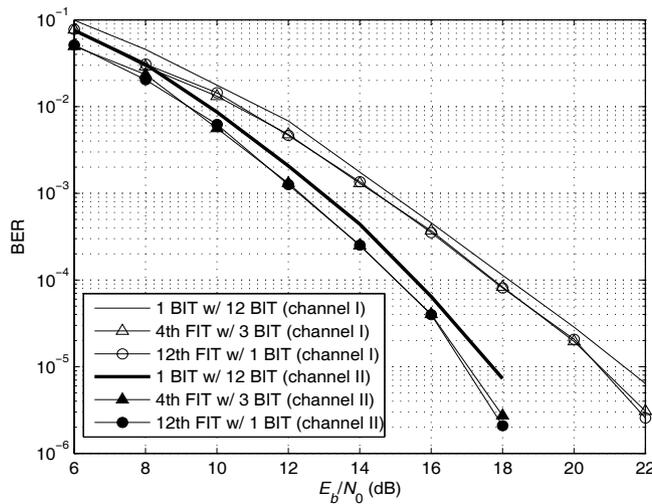


Fig. 5. BER performance in channels I and II.

as shown by the last row of Table II. The turbo FDE performs better than turbo TDE and TLE with less computation which can be verified by comparing the computational load at the first row with those at the third and fifth rows of Table II, respectively, as also described by the example at the end of Section III.

The complexity of the proposed receiver can be further reduced by feeding *a priori* information to the SIC/equalizer after a few back-end iterations, as shown in Fig. 5. For both channel I and II, four FITs where each one is followed by three BITs result in the same performance compared to

twelve FITs where each one is followed by one FIT, while the equalizer complexity is three times less for the former scenario. Moreover, by exploiting the SISO capability of FDE, the performance gain of the aforementioned scenarios compared to one FIT followed by twelve BITs is about 1 dB after  $\text{BER} = 1 \times 10^{-5}$  in both channels I and II.

## V. CONCLUSION

In this paper, a doubly-iterative receiver with a SISO frequency domain MMSE equalizer is proposed for CPM with higher performance and less complexity than its counterparts also employing linear equalizers in [5] and [9]. This effectiveness of the new receiver is presented with comparisons in terms of the overall computational complexity and the BER simulations. The constant envelope and phase continuity of CPM are maintained while appending the cyclic prefix to the transmitted signal.

## REFERENCES

- [1] T. Aulin and C. E. Sundberg, "Continuous phase modulation—parts I and II," *IEEE Trans. Commun.*, vol. 29, pp. 196-225, Mar. 1981.
- [2] L. Yin and G. L. Stüber, "MLSE and soft-output equalization for trellis-coded continuous phase modulation," *IEEE Trans. Commun.*, vol. 45, pp. 651-659, June 1997.
- [3] K. Chung, K. M. Chugg, and J. Heo, "Reduced state adaptive SISO algorithms for serially concatenated CPM over frequency-selective fading channels," in *Proc. GLOBECOM*, Nov. 2001, pp. 1162-1166.
- [4] A. Hansson and T. Aulin, "Generalized APP detection of continuous phase modulation over unknown ISI channels," *IEEE Trans. Commun.*, vol. 53, pp. 1615-1619, Oct. 2005.
- [5] B. Özgül, M. Koca, and H. Deliç, "Doubly-iterative equalization of continuous phase modulation," *IEEE Trans. Commun.*, vol. 55, pp. 2114-2124, Nov. 2007.
- [6] H. Sari, G. Karam, and I. Jeanclaude, "Transmission techniques for digital terrestrial TV broadcasting," *IEEE Commun. Mag.*, vol. 33, pp. 100-109, Feb. 1995.
- [7] D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, "Frequency domain equalization for single-carrier broadband wireless systems," *IEEE Commun. Mag.*, vol. 40, pp. 58-66, Apr. 2002.
- [8] J. Tan and G. L. Stüber, "Frequency domain equalization for continuous phase modulation," *IEEE Trans. Wireless Commun.*, vol. 4, pp. 2479-2490, Sep. 2005.
- [9] F. Pancaldi and G. M. Vitetta, "Equalization algorithms in the frequency domain for continuous phase modulations," *IEEE Trans. Commun.*, vol. 54, pp. 648-658, Apr. 2006.
- [10] M. Tüchler and J. Hagenauer, "Turbo equalization using frequency domain filters," in *Proc. Allerton Conf.*, Oct. 2000, pp. 144-153.
- [11] B. Rimoldi, "A decomposition approach to CPM," *IEEE Trans. Inform. Theory*, vol. 34, pp. 260-270, Mar. 1988.
- [12] S. Lin and D. J. Costello, Jr., *Error Control Coding*, 2nd ed. New Jersey: Pearson-Prentice Hall, 2004.
- [13] M. Tüchler, A. C. Singer, and R. Koetter, "Turbo equalization: principles and new results," *IEEE Trans. Commun.*, vol. 50, pp. 754-767, May 2002.
- [14] B. Özgül, M. Koca, and H. Deliç, "Frequency domain doubly-iterative equalization of continuous phase modulation," in *Proc. GLOBECOM*, Nov. 2006, pp. 1-5.