Low-Complexity Linear Frequency Domain Equalization for Continuous Phase Modulation

Wim Van Thillo, *Member, IEEE,* François Horlin, *Member, IEEE,* Jimmy Nsenga, *Member, IEEE,* Valéry Ramon, *Member, IEEE,* André Bourdoux, *Member, IEEE,* and Rudy Lauwereins, *Member, IEEE*

***Abstract*—In this paper, we develop a new low-complexity linear frequency domain equalization (FDE) approach for continuous phase modulated (CPM) signals. As a CPM signal is highly correlated, calculating a linear minimum mean square error (MMSE) channel equalizer requires the inversion of a non-diagonal matrix, even in the frequency domain. In order to regain the FDE advantage of reduced computational complexity, we show that this matrix can be approximated by a block-diagonal matrix without performance loss. Moreover, our MMSE equalizer can be simplified to a low-complexity zero-forcing equalizer. The proposed techniques can be applied to any CPM scheme. To support this theory we present a new polyphase matrix model, valid for any block-based CPM system. Simulation results in a 60 GHz environment show that our reduced-complexity MMSE equalizer significantly outperforms the state of the art linear MMSE receiver for large modulation indices, while it performs only slightly worse for small ones.**

***Index Terms*—Continuous phase modulation (CPM), frequency domain equalization (FDE), minimum mean square error (MMSE) equalization, complexity reduction, polyphase representation**

I. Introduction

WE ARE WITNESSING an explosive growth in the demand for wireless connectivity. Short range wireless links will soon be expected to deliver bit rates of over 2 Gbits/s. Worldwide, recent regulation assigned a 3 GHz or wider frequency band at 60 GHz to this kind of applications [1]. Continuous phase modulation (CPM) has been proposed as an attractive modulation technique for 60 GHz communications in [2], [3] and [4] and in the IEEE 802.15.3c standardization committee.

Nevertheless, CPM is a nonlinear modulation technique. Therefore, it is mathematically less tractable. Fortunately, using the *Laurent decomposition* [5] and its extensions [6] and [7], any CPM signal can be decomposed into a sum of linearly modulated signals. In this decomposition, the data symbols are first nonlinearly mapped onto a set of *pseudocoefficients* (PCs) which then pass through a bank of linear pulse shaping filters called *Laurent pulses* (LPs). A maximum likelihood (ML) CPM receiver in additive white Gaussian noise (AWGN) based on the Laurent decomposition contains a Viterbi decoder which exploits the correlation properties of the LPs and PCs to perform ML sequence detection (MLSD) of the sent symbols [8].

Typical 60 GHz channels are severely frequency-selective for the targeted signal bandwidth. Equalizing such channels in the frequency domain (FD) rather than in the time domain (TD) can significantly lower the system complexity [9]. An excellent framework for frequency domain equalization (FDE) of CPM signals based on the Laurent decomposition is provided in [10]. A filter-based joint equalization of the channel and the LPs is perfomed. However, this is suboptimal. Indeed, the correlation properties of the LPs can in this case not be used anymore in the Viterbi decoder following the equalizer. This decoder only inverts the mapping of the input symbols on the PCs. A similar approach was presented in [11].

We take a different approach by separating channel equalization on the one hand and demodulation of the equalized CPM signal on the other. The application of this approach to the framework of [10] is the first contribution of this paper. A linear equalizer is first applied to filter out the intersymbol interference (ISI) introduced by the channel. The output of this equalizer is then fed to a CPM demodulator, which can still exploit the correlation properties of the LPs and of the PCs to perform MLSD. This approach has two main advantages. First, our MMSE channel equalizer followed by a CPM demodulator is shown to perform well for any modulation index, and to outperform the joint channel and LP equalizer from [10] significantly for large modulation indices. This is currently of interest for military systems using CPM [12]. Second, separating equalization and demodulation offers an advantage in terms of flexibility since all CPM demodulators already known in the literature can be used after the equalizer, including reduced-complexity ones.

Separating channel equalization and CPM demodulation has already been proposed in [13] and [14]. However, in these papers the *tilted-phase representation* [15] of CPM is used, whereas we use the Laurent decomposition as in [10]. In this latter, the nonlinear nature of CPM is completely captured in the mapping of the input symbols on the PCs. This allows us to construct a TD polyphase matrix model, valid for any block-based CPM system. The development of this matrix model is the second contribution of this paper. The model has two advantages.

First, it allows us to derive our new equalizers using the well-known framework for block-based FDE described in [16]. In this framework, a matrix model is first established in the

W. Van Thillo, J. Nsenga, V. Ramon, A. Bourdoux and R. Lauwereins are with the Interuniversity Micro-Electronics Center (IMEC), Kapeldreef 75, B-3001 Leuven, Belgium (e-mail: vthillo@imec.be). W. Van Thillo, J. Nsenga and R. Lauwereins are also with the Department of Electrical Engineering, Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Leuven, Belgium. F. Horlin is with the Université Libre de Bruxelles, Avenue F.D. Roosevelt 50, B-1050 Brussels, Belgium.

This paper was presented partly at the IEEE Global Telecommunications Conference (IEEE GLOBECOM), Washington, DC, USA, November 26-30, 2007.

TD and then transformed into the FD. The two equivalent TD and FD models allow us to perform all signal processing tasks where they can be done most efficiently, either in the TD or in the FD. This is not the case for the FD model of [10], which does not have a direct TD counterpart.

Second, the autocorrelation properties of the PCs are well-known [5]. This enables us to significantly reduce the complexity of our MMSE equalizer, which is the last contribution of the paper. We show that the elements beyond the main block diagonal of the CPM autocorrelation matrix can be neglected. The calculation of the resulting reduced-complexity MMSE equalizer then only requires the inversion of a block-diagonal matrix. This lowers the computational requirements significantly. We show that the approximation does not cause any noticeable performance loss. Moreover, it can be applied to any CPM scheme, independently of the modulation index. Finally, as it is based only on the structure of the CPM autocorrelation matrix, the conclusions are valid for CPM in general. The technique can therefore also be used to reduce the complexity of other known algorithms, such as the ones presented in [10].

The paper is structured as follows. In Section II, the CPM polyphase matrix models in both the TD and the FD are constructed. Section III introduces our new linear MMSE channel equalization and CPM demodulation approach. These techniques are compared to the state of the art linear CPM-FDE algorithm in Section IV. The equalizer complexity reduction techniques are introduced in Section V, followed by a complexity analysis in Section VI. Finally, simulation results are discussed in Section VII.

We still introduce some notation conventions. Vectors in the TD are represented by underlined lowercase letters $\underline{x}$, in the FD by uppercase letters $\underline{X}$. The n^{th} element of a vector $\underline{x}$ is x_n. Matrices in the TD are represented by doubly underlined lowercase letters $\underline{\underline{x}}$, in the FD by uppercase letters $\underline{\underline{X}}$. We do not use the classical boldface lowercase notation for vectors and uppercase for matrices as it does not allow us to distinguish between vectors and matrices both in TD and FD. The $(n,m)^{\text{th}}$ element of a matrix $\underline{\underline{x}}$ is $\left[\underline{\underline{x}}\right]_{(n,m)}$. In a matrix or vector, $[x,y,z]$ are elements on a row, whereas $[x;y;z]$ form a column. An identity matrix of size N is denoted by $\underline{\underline{I}}_N$, an $N\times M$ matrix containing all zeros by $\underline{\underline{0}}_{N\times M}$, and an $N\times M$ matrix containing all ones by $\underline{\underline{J}}_{N\times M}$. The conjugate transpose of a matrix is denoted by $(.)^H$ and $(.)^*$ denotes the complex conjugate of a scalar. The Kronecker product is denoted by $\otimes$ and $\odot$ denotes the Hadamard matrix product. A convolution is denoted by $\star$.

II. Polyphase Matrix Models for Block-Based CPM Systems

In Section II-A of this section, we briefly review the CPM waveform and its linear representation. We then construct a polyphase representation of the received signal in Section II-B. Next, we introduce the block-based CPM transmitter in Section II-C. Following the approach of [16], we subsequently model the system as a TD matrix model in Section II-D, which is afterwards converted into the FD in Section II-D.

A. CPM and its Linear Representation

A transmitted CPM signal has the form

$$s(t,\underline{a}) = \sqrt{\frac{2E_S}{T}} e^{j\,\psi(t,\underline{a})} \tag{1}$$

where $\underline{a}$ contains the sequence of M-ary Amplitude Shift Keyed (ASK) symbols $a_n \in \{\pm 1, \pm 3, ..., \pm M-1\}$ [17]. The symbol duration is T and E_S is the energy per symbol, normalized to $E_S = 1$. The transmitted information is contained in the phase:

$$\psi(t,\underline{a}) = 2\pi h \sum_n a_n \cdot q(t-nT) \tag{2}$$

where h is the modulation index and $q(t)$ is the *phase response*, related to the *frequency response* $f(t)$ by the relationship $q(t) = \int_{-\infty}^{t} f(\tau)\,\mathrm{d}\tau$. The pulse $f(t)$ is a smooth pulse shape over a finite time interval $0 \le t \le LT$ where L is an integer, and zero outside this interval. The function $f(t)$ is normalized such that $\int_{-\infty}^{\infty} f(t)\,\mathrm{d}t = \frac{1}{2}$.

Exploiting the Laurent decomposition [5], we can write (1) as a sum of $P = 2^{L-1}$ linearly modulated signals

$$s(t) = \sum_{p=0}^{P-1} \sum_n b_{p,n}\, l_p(t-nT) \tag{3}$$

where the PCs $b_{p,n}$ are given by

$$b_{p,n} = \exp\left[j\pi h \left(\sum_{i=0}^{n} a_i - \sum_{i=1}^{L-1} a_{n-i}\,\beta_{p,i} \right) \right] \tag{4}$$

with $\beta_{p,i}$ the i^{th} bit in the binary representation of p ($p = \sum_{i=1}^{L-1} 2^{i-1}\,\beta_{p,i}$). The LPs $l_p(t), p = 0, ..., P-1$ are real functions with nonzero values in the interval $L_pT, p = 0, ..., P-1$ respectively, where $L_p \le L+1$. Representation (3) is only valid for binary modulation formats ($M = 2$) and noninteger h. Using [6] and [7] though, the Laurent decomposition can be generalized to any CPM scheme, so that the techniques presented in this paper can be applied to all CPM schemes.

In this paper, we assume that the frequency components of the CPM signal above $1/T$ are negligible. This assumption allows us to construct a discrete version of (3) as

$$s_n = \sum_{p=0}^{P-1} \sum_m b_{p,m}\, l_{p,n-2m} \tag{5}$$

where $l_{p,n} \triangleq l_p(t)|_{t=nT/2}$ and $s_n \triangleq s(t)|_{t=nT/2}$. If the assumption made above is not satisfied for a particular CPM scheme, a higher oversampling rate should be chosen. The model in this paper can easily be adapted accordingly and thus stays valid.

We will construct our receiver based on the digital representation of the sent signal (5). It is shown in the left part of Fig. 1. In the next section we model the received signal by introducing the channel impulse response and the noise.

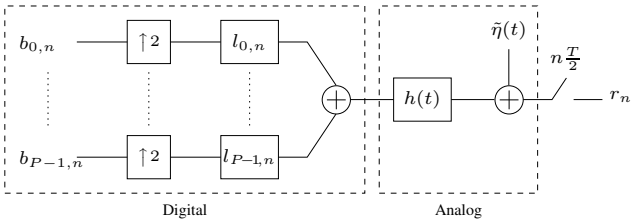

Fig. 1. Equivalent system model. The analog-to-digital and digital-to-analog converters are not shown.

B. Received Signal Model

Equivalent to the approach of [16], (5) is digital-to-analog converted and filtered by the transmitter filter $\psi_{tr}(t)$. It is then sent through a linear multipath channel $\psi_h(t)$ and through the receive filter $\psi_{rec}(t)$. Let $h(t) = \psi_{tr}(t) \star \psi_h(t) \star \psi_{rec}(t)$ denote the overall impulse response of the cascade transmit filter, linear channel and receive filter with maximum length $L_C T$. The received baseband signal can then be written as

$$r(t) = \sum_n s_n \, h(t - nT/2) + \tilde{\eta}(t) \tag{6}$$

where $\tilde{\eta}(t) = \eta(t) \star \psi_{rec}(t)$ with $\eta(t)$ the Additive White Gaussian Noise (AWGN). The received signal is sampled at $f_s = 2/T$ and split into two polyphase components $i = 0, 1$

$$r_n^i \triangleq r(t)\,\big|_{t=(2n+i)T/2} = \sum_m s_m^0 \, h_{n-m}^i + \sum_m s_m^1 \, h_{n-m-1+i}^{(1-i)} + \tilde{\eta}_n^i \tag{7}$$

where

$$h_n^i \triangleq h(t)\,\big|_{t=(2n+i)T/2} \tag{8}$$

$$\tilde{\eta}_n^i \triangleq \tilde{\eta}(t)\,\big|_{t=(2n+i)T/2} \tag{9}$$

$$s_n^i \triangleq s_{2n+i} = \sum_{p=0}^{P-1} \sum_m b_{p,m} \, l_{p,n-m}^i \tag{10}$$

$$l_{p,n}^i \triangleq l_{p,2n+i}. \tag{11}$$

We now have a well established polyphase model of the received CPM waveform in a multipath environment. As we want to perform block-based FDE on this signal, we review in the next section how the blocks should be constructed correctly.

C. Block-Based CPM and the Need of an Intrafix

The block-based CPM system is now introduced. A cyclic prefix (CP) is appended to each transmitted block to enable low-complexity equalization in the FD [9]. However, as a CPM signal contains memory, an *intrafix* of K data-dependent symbols has to be inserted in each block, in addition to the CP. This was shown in [10] but in this paper we follow a slightly different procedure, which is described in detail in [18]. As shown in Fig. 2, the input symbol stream $\underline{a}$ is first cut in blocks $\left[\underline{a}_f^{(l)}; \underline{a}_s^{(l)}\right]$ of length $N - K$, where the superscript (l) refers to the l^{th} block. Then, an intrafix $\underline{a}_c^{(l)}$ of length K is inserted in each block, yielding blocks $\left[\underline{a}_f^{(l)}; \underline{a}_c^{(l)}; \underline{a}_s^{(l)}\right]$ of length N. Finally, the CP of length N_P is inserted so that we obtain blocks of size $N_T = N + N_P$

$$\underline{a}^{(l)} = \left[\underline{a}_p^{(l)}; \underline{a}_f^{(l)}; \underline{a}_c^{(l)}; \underline{a}_s^{(l)}\right]. \tag{12}$$

The CP length N_P is chosen such that $N_P > L_C$ to avoid interblock interference (IBI). In [19], it is explained how precoded CPM can be combined with FDE by introducing a second intrafix in each block.

The intrafix insertion operation is nonlinear and can thus not be described in the matrix model below. We therefore assume that a correct intrafix has been inserted in each block. The CP insertion operation on the other hand is linear. It will therefore be incorporated in the matrix model, introduced in the next section.

D. Time Domain Matrix Model

In this section, we apply the approach of [16] to formulate the system (7) - (11) in matrix form. We define blocks of PCs

$$\underline{b}_p^{(l)} \triangleq \begin{bmatrix} b_{p,lN} & b_{p,lN+1} & \dots & b_{p,(l+1)N-1} \end{bmatrix}^T \tag{13}$$

for $p = 0, \dots, P-1$. As stated above, the blocks (13) do not contain the CP yet, but they do already contain an intrafix. As described in [16], the CP insertion is written as a matrix operation. We therefore define

$$\underline{\underline{T}}_{\text{CP}} \triangleq \left[\underline{\underline{0}}_{N_P \times (N-N_P)}, \; \underline{\underline{I}}_{N_P}; \; \underline{\underline{I}}_N\right] \tag{14}$$

such that the CP insertion can be written as a left multiplication of the blocks $\underline{b}_p$ with $\underline{\underline{T}}_{\text{CP}}$ as

$$\dot{\underline{b}}_p^{(l)} \triangleq \underline{\underline{T}}_{\text{CP}} \, \underline{b}_p^{(l)}. \tag{15}$$

We now stack all blocks of PCs into one large vector

$$\underline{b}^{(l)} \triangleq \left[\underline{b}_0^{(l)}; \quad \underline{b}_1^{(l)}; \quad \dots; \quad \underline{b}_{P-1}^{(l)}\right] \tag{16}$$

so that we can write

$$\dot{\underline{b}}^{(l)} \triangleq \left[\dot{\underline{b}}_0^{(l)}; \quad \dot{\underline{b}}_1^{(l)}; \quad \dots; \quad \dot{\underline{b}}_{P-1}^{(l)}\right] = \left(\underline{\underline{I}}_P \otimes \underline{\underline{T}}_{\text{CP}}\right) \, \underline{b}^{(l)}. \tag{17}$$

To describe the LPs and the linear channel in matrix form, we proceed as in [16] and define $N_T \times N_T$ convolution matrices as

$$\left[\underline{\underline{l}}_{p,0}^i\right]_{(n,m)} \triangleq l_{p,n-m}^i, \quad i = 0, 1 \tag{18}$$

$$\left[\underline{\underline{h}}_0^{i(l)}\right]_{(n,m)} \triangleq h_{n-m}^{i(l)}, \quad i = 0, 1 \tag{19}$$

$$\left[\underline{\underline{h}}_0^{1'(l)}\right]_{(n,m)} \triangleq h_{n-m-1}^{1(l)} \tag{20}$$

$$\left[\underline{\underline{l}}_{p,1}^i\right]_{(n,m)} \triangleq l_{p,N_T+n-m}^i, \quad i = 0, 1 \tag{21}$$

$$\left[\underline{\underline{h}}_1^{i(l)}\right]_{(n,m)} \triangleq h_{N_T+n-m}^{i(l-1)}, \quad i = 0, 1 \tag{22}$$

$$\left[\underline{\underline{h}}_1^{1'(l)}\right]_{(n,m)} \triangleq h_{N_T+n-m-1}^{1(l-1)}. \tag{23}$$

Here we assumed that the channel impulse response remains constant during the transmission of block l, and can therefore be written as $h_n^{(l)}$. Matrices (21) - (23) with subscript 1 will describe the IBI. Matrices (20) and (23) will be needed to represent the second term of (7) for $i = 0$. The structure

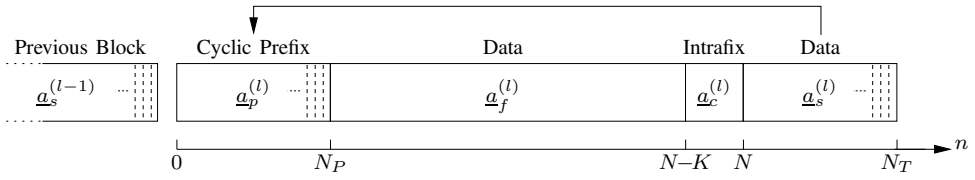

Fig. 2. Structure of an overall data block $\underline{a}^{(l)}$. The index N_P refers to the first data symbol, $N-K$ to the first intrafix symbol and N to the first symbol after the intrafix. The index of the last symbol in the block is N_T-1.

of these matrices is illustrated in Fig. 3 and will be further clarified below.

Using (18) - (23) and (8) - (11), we can now write the first polyphase component $i=0$ of the received signal (7) in vector form

$$\begin{aligned}\underline{\dot{r}}^{0(l)} &\triangleq \begin{bmatrix} r^0_{lN_T} & r^0_{lN_T+1} & \cdots & r^0_{(l+1)N_T-1}\end{bmatrix}^T \\ &= \underline{\underline{h}}_0^{0(l)} \left(\sum_{p=0}^{P-1} \underline{\underline{l}}^0_{p,0}\, \underline{\dot{b}}^{(l)}_p\right) + \underline{\underline{h}}_0^{1'(l)} \left(\sum_{p=0}^{P-1} \underline{\underline{l}}^1_{p,0}\, \underline{\dot{b}}^{(l)}_p\right) \\ &\quad + \underline{\underline{h}}_1^{0(l)} \left(\sum_{p=0}^{P-1} \underline{\underline{l}}^0_{p,1}\, \underline{\dot{b}}^{(l-1)}_p\right) + \underline{\underline{h}}_1^{1'(l)} \left(\sum_{p=0}^{P-1} \underline{\underline{l}}^1_{p,1}\, \underline{\dot{b}}^{(l-1)}_p\right) \\ &\quad + \underline{\tilde{\eta}}^{0(l)} \end{aligned} \tag{24}$$

whereas the second polyphase component $i=1$ can be written as

$$\begin{aligned}\underline{\dot{r}}^{1(l)} &\triangleq \begin{bmatrix} r^1_{lN_T} & r^1_{lN_T+1} & \cdots & r^1_{(l+1)N_T-1}\end{bmatrix}^T \\ &= \underline{\underline{h}}_0^{1(l)} \left(\sum_{p=0}^{P-1} \underline{\underline{l}}^0_{p,0}\, \underline{\dot{b}}^{(l)}_p\right) + \underline{\underline{h}}_0^{0(l)} \left(\sum_{p=0}^{P-1} \underline{\underline{l}}^1_{p,0}\, \underline{\dot{b}}^{(l)}_p\right) \\ &\quad + \underline{\underline{h}}_1^{1(l)} \left(\sum_{p=0}^{P-1} \underline{\underline{l}}^0_{p,1}\, \underline{\dot{b}}^{(l-1)}_p\right) + \underline{\underline{h}}_1^{0(l)} \left(\sum_{p=0}^{P-1} \underline{\underline{l}}^1_{p,1}\, \underline{\dot{b}}^{(l-1)}_p\right) \\ &\quad + \underline{\tilde{\eta}}^{1(l)} \end{aligned} \tag{25}$$

where

$$\underline{\tilde{\eta}}^{i(l)} = \begin{bmatrix} \tilde{\eta}^i_{lN_T} & \tilde{\eta}^i_{lN_T+1} & \cdots & \tilde{\eta}^i_{(l+1)N_T-1}\end{bmatrix}^T,\ i=0,1. \tag{26}$$

Stacking these components (24) and (25) into one vector, and writing the sums over p as a matrix multiplication using (17), we get

$$\begin{aligned}\underline{\dot{r}}^{(l)} &\triangleq \begin{bmatrix} \underline{\dot{r}}^{0(l)} \\ \underline{\dot{r}}^{1(l)}\end{bmatrix} \\ &= \begin{bmatrix} \underline{\underline{h}}_0^{0(l)} & \underline{\underline{h}}_0^{1'(l)} \\ \underline{\underline{h}}_0^{1(l)} & \underline{\underline{h}}_0^{0(l)}\end{bmatrix} \begin{bmatrix} \underline{\underline{l}}^0_{0,0} & \underline{\underline{l}}^0_{1,0} & \cdots & \underline{\underline{l}}^0_{P-1,0} \\ \underline{\underline{l}}^1_{0,0} & \underline{\underline{l}}^1_{1,0} & \cdots & \underline{\underline{l}}^1_{P-1,0}\end{bmatrix} \left(\underline{\underline{I}}_P \otimes \underline{\underline{T}}_{\mathrm{CP}}\right) \underline{b}^{(l)} \\ &\quad + \begin{bmatrix} \underline{\underline{h}}_1^{0(l)} & \underline{\underline{h}}_1^{1'(l)} \\ \underline{\underline{h}}_1^{1(l)} & \underline{\underline{h}}_1^{0(l)}\end{bmatrix} \begin{bmatrix} \underline{\underline{l}}^0_{0,1} & \underline{\underline{l}}^0_{1,1} & \cdots & \underline{\underline{l}}^0_{P-1,1} \\ \underline{\underline{l}}^1_{0,1} & \underline{\underline{l}}^1_{1,1} & \cdots & \underline{\underline{l}}^1_{P-1,1}\end{bmatrix} \left(\underline{\underline{I}}_P \otimes \underline{\underline{T}}_{\mathrm{CP}}\right) \underline{b}^{(l-1)} \\ &\quad + \begin{bmatrix} \underline{\tilde{\eta}}^{0(l)} \\ \underline{\tilde{\eta}}^{1(l)}\end{bmatrix}. \end{aligned} \tag{27}$$

This matrix model is visualised in Fig. 3. The useful information is contained in the first term, the IBI in the second one and the noise in the last one. The first term itself is a product of four factors: one matrix representing the channel convolution, a second one representing the LPs convolution, a third one representing the CP insertion and a final vector containing the P blocks of PCs.

The first operation in the receiver is the removal of the CP. This can be done by multiplying both polyphase components of the received signal (27) with the matrix

$$\underline{\underline{R}}_{\mathrm{CP}} \triangleq \begin{bmatrix} \underline{\underline{0}}_{N\times N_P},\ \underline{\underline{I}}_N\end{bmatrix}. \tag{28}$$

It can be seen that the second term of (27) becomes zero, which means that the IBI has been eliminated. We can therefore drop the block dependence (l) and the subscript 0 of the remaining term for simplicity:

$$\begin{aligned}\underline{r} &\triangleq \begin{bmatrix} \underline{r}^0 \\ \underline{r}^1\end{bmatrix} = \left(\underline{\underline{I}}_2 \otimes \underline{\underline{R}}_{\mathrm{CP}}\right)\, \underline{\dot{r}} \\ &= \begin{bmatrix} \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{h}}^0 & \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{h}}^{1'} \\ \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{h}}^1 & \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{h}}^0\end{bmatrix} \begin{bmatrix} \underline{\underline{l}}^0_0\, \underline{\underline{T}}_{\mathrm{CP}} & \underline{\underline{l}}^0_1\, \underline{\underline{T}}_{\mathrm{CP}} & \cdots & \underline{\underline{l}}^0_{P-1}\, \underline{\underline{T}}_{\mathrm{CP}} \\ \underline{\underline{l}}^1_0\, \underline{\underline{T}}_{\mathrm{CP}} & \underline{\underline{l}}^1_1\, \underline{\underline{T}}_{\mathrm{CP}} & \cdots & \underline{\underline{l}}^1_{P-1}\, \underline{\underline{T}}_{\mathrm{CP}}\end{bmatrix} \underline{b} \\ &\quad + \begin{bmatrix} \underline{\tilde{\eta}}^0 \\ \underline{\tilde{\eta}}^1\end{bmatrix}. \end{aligned} \tag{29}$$

If we choose $N_P > L_C + L$, it can be seen that

$$\underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{h}}^i\, \underline{\underline{l}}^j_p\, \underline{\underline{T}}_{\mathrm{CP}} = \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{h}}^i\, \underline{\underline{T}}_{\mathrm{CP}}\, \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{l}}^j_p\, \underline{\underline{T}}_{\mathrm{CP}} \tag{30}$$

for any $i,j \in \{0,1\}$, such that (29) can be written as

$$\begin{aligned}\underline{r} &= \begin{bmatrix} \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{h}}^0\, \underline{\underline{T}}_{\mathrm{CP}} & \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{h}}^{1'}\, \underline{\underline{T}}_{\mathrm{CP}} \\ \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{h}}^1\, \underline{\underline{T}}_{\mathrm{CP}} & \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{h}}^0\, \underline{\underline{T}}_{\mathrm{CP}}\end{bmatrix} \\ &\quad \cdot \begin{bmatrix} \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{l}}^0_0\, \underline{\underline{T}}_{\mathrm{CP}} & \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{l}}^0_1\, \underline{\underline{T}}_{\mathrm{CP}} & \cdots & \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{l}}^0_{P-1}\, \underline{\underline{T}}_{\mathrm{CP}} \\ \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{l}}^1_0\, \underline{\underline{T}}_{\mathrm{CP}} & \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{l}}^1_1\, \underline{\underline{T}}_{\mathrm{CP}} & \cdots & \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{l}}^1_{P-1}\, \underline{\underline{T}}_{\mathrm{CP}}\end{bmatrix} \underline{b} \\ &\quad + \begin{bmatrix} \underline{\tilde{\eta}}^0 \\ \underline{\tilde{\eta}}^1\end{bmatrix}. \end{aligned} \tag{31}$$

As explained in [16], left multiplication with $\underline{\underline{R}}_{\mathrm{CP}}$ and right multiplication with $\underline{\underline{T}}_{\mathrm{CP}}$ of an $N_T \times N_T$ convolution matrix $\underline{\underline{x}}$ results in a circulant $N \times N$ matrix $\underline{\underline{\dot{x}}}$

$$\underline{\underline{\dot{x}}} \triangleq \underline{\underline{R}}_{\mathrm{CP}}\, \underline{\underline{x}}\, \underline{\underline{T}}_{\mathrm{CP}} \tag{32}$$

where the dot denotes the circulant property, such that

$$\left[\underline{\underline{\dot{x}}}\right]_{(n,m)} = \left[\underline{\underline{x}}\right]_{((n-m+1) \bmod N,1)}. \tag{33}$$

Therefore, we can write (31) as

$$\begin{aligned}\underline{r} &= \begin{bmatrix} \underline{\dot{h}}^0 & \underline{\dot{h}}^{1'} \\ \underline{\dot{h}}^1 & \underline{\dot{h}}^0\end{bmatrix} \begin{bmatrix} \underline{\dot{l}}^0_0 & \underline{\dot{l}}^0_1 & \cdots & \underline{\dot{l}}^0_{P-1} \\ \underline{\dot{l}}^1_0 & \underline{\dot{l}}^1_1 & \cdots & \underline{\dot{l}}^1_{P-1}\end{bmatrix} \underline{b} + \begin{bmatrix} \underline{\tilde{\eta}}^0 \\ \underline{\tilde{\eta}}^1\end{bmatrix} \end{aligned} \tag{34}$$

$$\triangleq \underline{\underline{h}}\, \underline{\underline{l}}\, \underline{b} + \underline{\tilde{\eta}} \tag{35}$$

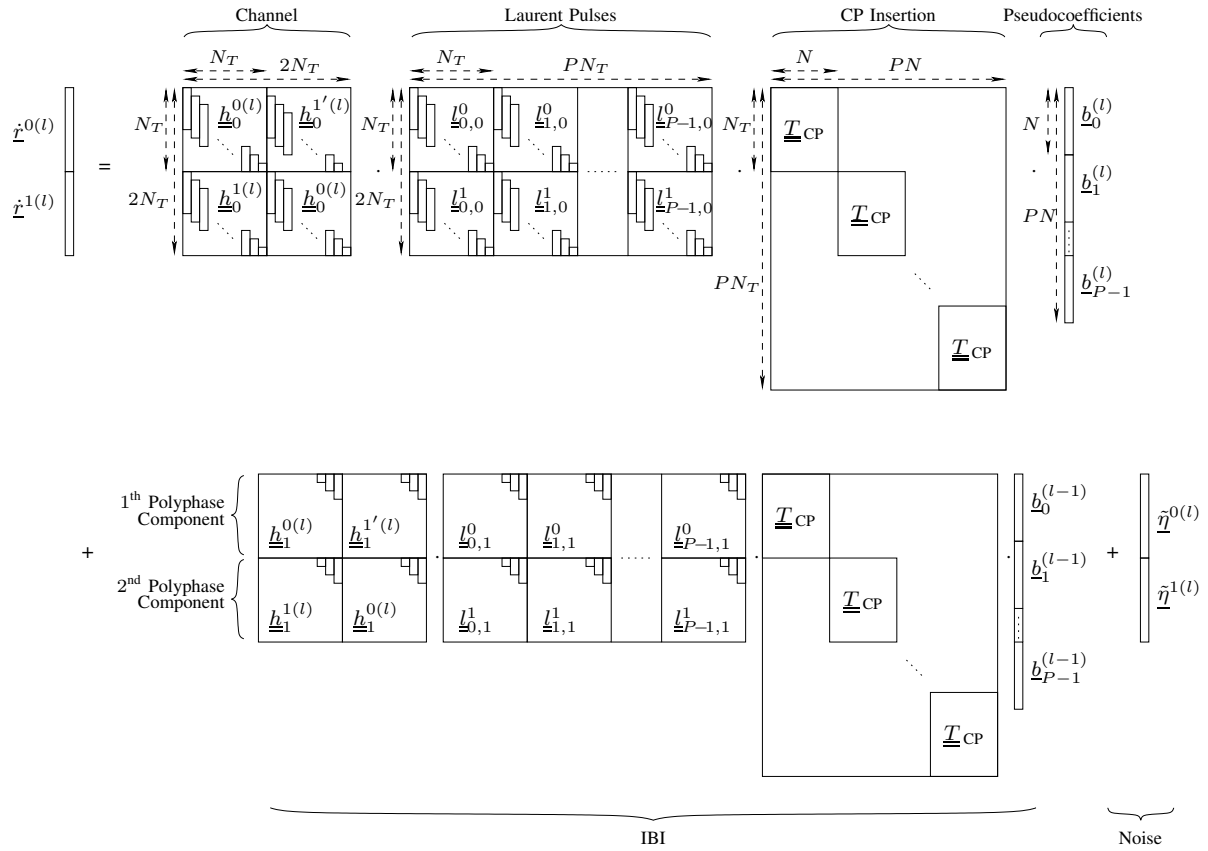

Fig. 3. Matrix model of the received signal, showing the channel and LP convolution matrices for both polyphase components, the CP insertion operation and the noise addition. The IBI (second term of the sum) is eliminated when the CP is discarded.

where all $N \times N$ submatrices $\underline{\underline{\dot{x}}}$ of $\underline{\underline{l}}$ and $\underline{\underline{h}}$ appearing in (34) are circulant.

We now have a well-structured polyphase TD matrix model which describes the received samples of any block-based cyclic-prefixed CPM system. In the next section, we will transform this model into the FD such that all circulant matrices become diagonalized.

E. Frequency Domain Matrix Model

Any circulant $N \times N$ matrix $\underline{\underline{\dot{x}}}$ can be transformed into a diagonal $N \times N$ matrix $\underline{\underline{X}}$ as

$$\underline{\underline{X}} = \underline{\underline{F}}_N \; \underline{\underline{\dot{x}}} \; \underline{\underline{F}}_N^H \tag{36}$$

where $\underline{\underline{F}}_N$ is an N-size discrete Fourier transform (DFT) matrix. Moreover,

$$\mathrm{diag}(\underline{\underline{X}}) = \underline{\underline{F}}_N \, \underline{x} \tag{37}$$

where $\underline{x}$ is the first column of $\underline{\underline{\dot{x}}}$ [16]. To diagonalize all circulant submatrices of $\underline{\underline{h}}$ and $\underline{\underline{l}}$, we therefore define the block diagonal $NM \times NM$ matrix $\underline{\underline{F}}_{N,M}$ as

$$\underline{\underline{F}}_{N,M} \triangleq \underline{\underline{I}}_M \otimes \underline{\underline{F}}_N, \tag{38}$$

such that the matrices $\underline{\underline{L}}$ and $\underline{\underline{H}}$ defined as

$$\underline{\underline{L}} \triangleq \underline{\underline{F}}_{N,2} \; \underline{\underline{l}} \; \underline{\underline{F}}_{N,P}^H \tag{39}$$

$$\underline{\underline{H}} \triangleq \underline{\underline{F}}_{N,2} \; \underline{\underline{h}} \; \underline{\underline{F}}_{N,2}^H \tag{40}$$

consist only of diagonal $N \times N$ submatrices. One of the properties of a DFT matrix is $\underline{\underline{F}}_N^{-1} = \underline{\underline{F}}_N^H$ such that $\underline{\underline{F}}_{N,M}^{-1} = \underline{\underline{F}}_{N,M}^H$. If we now define

$$\underline{B} \triangleq \underline{\underline{F}}_{N,P} \; \underline{b} \tag{41}$$

$$\underline{R} \triangleq \underline{\underline{F}}_{N,2} \; \underline{r} \tag{42}$$

$$\underline{N} \triangleq \underline{\underline{F}}_{N,2} \; \underline{\tilde{\eta}} \tag{43}$$

and use these definitions together with (39) and (40) in (35), we finally get the matrix model in the FD

$$\underline{R} = \underline{\underline{H}} \; \underline{\underline{L}} \; \underline{B} + \underline{N}. \tag{44}$$

This model describes the received samples of any block-based cyclic-prefixed CPM system in the FD. Three major differences exist between (44) and the FD model presented in [10].

First, we separate the channel matrix $\underline{\underline{H}}$ from the one representing the LPs $\underline{\underline{L}}$. This will allow us to treat the channel equalization separately from the CPM demodulation, as will be shown in Section III-A. In [10] on the other hand, the LPs are linearly equalized together with the channel. Therefore, their correlation properties cannot be exploited anymore in the Viterbi detector.

Second, we have used polyphase components to build completely equivalent matrix models in the TD and in the FD. This allowed us to apply the well-known framework of [16] to any block-based cyclic-prefixed CPM system. The FD model

proposed in [10] on the other hand, does not have a direct TD counterpart. Therefore, the familiar framework of [16] cannot be applied. The equivalent TD and FD models will also enable us to perform all signal processing tasks where they can be done most efficiently, in the TD or in the FD. For instance, we will implement the matched filterbank of our CPM demodulator in the FD, whereas its Viterbi detector operates in the TD, as explained in Section III-B.

Third, as can be seen from (42), we transform the received signal into the FD by two N-point DFT's, whereas in [10] one single $2N$-point DFT is taken. This has a computational complexity advantage, as will be explained in Section VI.

III. Our New Frequency Domain Equalization Approach for CPM

In this section, we use the FD matrix model (44) we derived so far to construct our new receiver. In Section III-A we introduce our new FDE approach, and in Section III-B the CPM demodulation. A block diagram of the whole system is provided in Fig. 4.

A. New Frequency Domain Channel Equalizer

As shown in Fig. 4, the first step of our approach is to equalize the channel $\underline{\underline{H}}$ in the FD. Let us define

$$\underline{S} \triangleq \underline{\underline{L}}\, \underline{B} \tag{45}$$

so that (44) can be written as $\underline{R} = \underline{\underline{H}}\, \underline{S} + \underline{N}$. According to [20], the MMSE matrix equalizer for this system is given by

$$\underline{\underline{G}}_{\text{MMSE}} = \underline{\underline{R}}_{SS}\, \underline{\underline{H}}^H \left[\underline{\underline{H}}\, \underline{\underline{R}}_{SS}\, \underline{\underline{H}}^H + \sigma_n^2\, \underline{\underline{I}}_{2N}\right]^{-1} \tag{46}$$

where σ_n^2 is the noise variance and

$$\underline{\underline{R}}_{SS} \triangleq \mathcal{E}\left\{\underline{S}\, \underline{S}^H\right\} = \underline{\underline{L}}\, \mathcal{E}\left\{\underline{B}\, \underline{B}^H\right\}\, \underline{\underline{L}}^H \triangleq \underline{\underline{L}}\, \underline{\underline{R}}_{BB}\, \underline{\underline{L}}^H \tag{47}$$

with $\mathcal{E}\{\cdot\}$ the expectation operator. As shown in [5], $\underline{\underline{R}}_{BB}$ can be calculated and stored once for a given CPM scheme and block size. We assume that the channel $\underline{\underline{H}}$ is always known at the receiver. Using the matrix inversion lemma, we can rewrite (46) as

$$\underline{\underline{G}}_{\text{MMSE}} = \left[\underline{\underline{R}}_{SS}^{-1} + \frac{1}{\sigma_n^2}\, \underline{\underline{H}}^H\, \underline{\underline{H}}\right]^{-1} \frac{1}{\sigma_n^2}\, \underline{\underline{H}}^H. \tag{48}$$

This equalizer produces an estimate $\hat{\underline{S}}$ of the sent signal in the FD

$$\hat{\underline{S}} = \underline{\underline{G}}_{\text{MMSE}}\, \underline{R}. \tag{49}$$

We emphasize that it only equalizes the channel $\underline{\underline{H}}$ but not the LPs $\underline{\underline{L}}$. The CPM demodulator following the equalizer can thus still exploit the memory introduced by the LPs, unlike the receiver proposed in [10]. The demodulator is now described.

B. CPM Demodulator

After (48), the noise is colored and residual ISI is present but we make the simplifying assumption that both can be modeled as AWGN. The equalizer can then be followed by any demodulator for CPM in AWGN available in the literature.

We construct a demodulator very similar to the one presented in [8]. This latter consists of a filterbank matched to the LPs and a Viterbi decoder. It operates completely in the TD. We follow its approach but implement the filter bank in the FD. Our Viterbi decoder is the same as in [8] and thus operates in the TD.

The demodulator decides that message $\tilde{\underline{S}}$ is transmitted if and only if it maximizes the metric

$$\Lambda = \tilde{\underline{S}}^H\, \hat{\underline{S}} \tag{50}$$

and substituting (41) and (45) into (50) yields

$$\Lambda = \tilde{\underline{b}}^H\, \underline{\underline{F}}_{N,P}^H\, \underline{\underline{L}}^H\, \hat{\underline{S}}. \tag{51}$$

We define

$$\underline{z} \triangleq \left[\underline{z}_0;\quad \underline{z}_1;\quad ...;\quad \underline{z}_{P-1}\right] = \underline{\underline{F}}_{N,P}^H\, \underline{\underline{L}}^H\, \hat{\underline{S}} \tag{52}$$

such that (51) can be written as

$$\Lambda = \tilde{\underline{b}}^H\, \underline{z}. \tag{53}$$

The vector $\underline{z}$ can be interpreted as the output of a bank of P filters matched to the LPs. This bank is represented in the FD by $\underline{\underline{L}}^H$ and its outputs are converted back to the TD by $\underline{\underline{F}}_{N,P}^H$.

As the length of $\underline{b}$ is NP, the number of possible hypotheses $\tilde{\underline{b}}$ grows exponentially with the block size N. To keep the decoding complexity under control, the search for the maximum Λ in (53) is therefore implemented in the TD using the Viterbi algorithm as follows. The memory in a CPM signal can be represented by a trellis [17]. A combination of P PCs $\tilde{\underline{b}}^i = [\tilde{b}_0^i, \tilde{b}_1^i, ..., \tilde{b}_{P-1}^i]$ corresponds to every branch $i = 1, ..., I$ of this trellis [8], where I is the total number of branches of a trellis section. The metric at time n associated with the branch i of the trellis is then calculated as

$$\lambda_n^i = \text{Re}\left(\sum_{p=0}^{P-1} z_{p,n} \tilde{b}_p^{i\,*}\right) \tag{54}$$

for all instants $n = 0, ..., N-1$ and for all trellis branches $i = 1, ..., I$ [8] where $\text{Re}(\cdot)$ denotes the real part. The Viterbi algorithm then finds the ML path through the trellis, i.e. the path with the highest total metric. The corresponding $\tilde{\underline{a}}$ is chosen as estimate of the sent symbols $\underline{a}$ in (1).

IV. Comparison with the State of the Art Linear CPM-FDE Receiver

In this section, we compare our new approach with the linear FD MMSE receiver presented in [10]. Below, the latter is called the state of the art (SoA) receiver. Similar algorithms for linear equalization of CPM in the FD have also been proposed in [11]. As mentioned in [10] though, these algorithms offer the same error performance at comparable complexities as the ones described in [10]. Therefore, we limit our comparison to this latter.

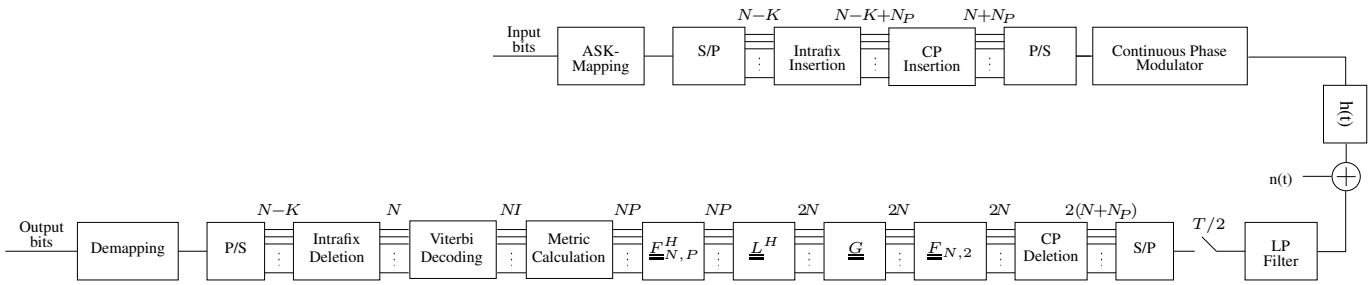

Fig. 4. Block diagram of the total system including transmitter (upper part) and receiver (lower part).

The main conceptual difference of our approach is clearly the separation of the channel matrix $\underline{\underline{H}}$ from the one representing the LPs $\underline{\underline{L}}$. As explained in Section II-E, another difference with [10] is our new matrix model, but to compare both equalization approaches we do not want to focus on this distinction. Therefore, we keep our model (44) and apply the equalization technique of [10] to it in Section IV-A. In Section IV-B we review the demodulator of [10].

A. Channel and Laurent pulse Equalizer

We first define as in [10]

$$\underline{\underline{M}} = \underline{\underline{H}}\,\underline{\underline{L}} \tag{55}$$

so that we can write (44) as $\underline{R} = \underline{\underline{M}}\,\underline{B} + \underline{N}$. According to [20], an MMSE equalizer for this system is given by

$$\underline{\underline{G}}_{\text{SoA}} = \underline{\underline{R}}_{BB}\underline{\underline{M}}^H \left[\underline{\underline{M}}\,\underline{\underline{R}}_{BB}\underline{\underline{M}}^H + \sigma_n^2\,\underline{\underline{I}}_{2N}\right]^{-1}. \tag{56}$$

This is the linear MMSE equalizer proposed in [10], applied to our new model (44). It produces an estimate $\underline{\hat{B}}$ of the PCs in the FD

$$\underline{\hat{B}} \triangleq \left[\underline{\hat{B}}_0;\quad \underline{\hat{B}}_1;\quad ...;\quad \underline{\hat{B}}_{P-1}\right] = \underline{\underline{G}}_{\text{SoA}}\,\underline{R}. \tag{57}$$

We emphasize that equalizer (56) jointly equalizes the channel $\underline{\underline{H}}$ *and* the LPs $\underline{\underline{L}}$, whereas our new equalizers (46) and (60) only equalize the channel but *not* the LPs. The demodulator after equalizer (46) can thus still exploit the correlation introduced by the LPs $\underline{\underline{L}}$, as it was explained in Section III-B. Moreover, an MMSE equalizer trades off residual interference versus noise power. Therefore, equalizer (56) pays a price in increased noise power by equalizing the LPs in addition to the channel.

B. CPM Demodulator

For completeness we briefly review the demodulator of [10]. The estimates of the PCs $\underline{\hat{B}}$ are transformed back into the TD

$$\underline{\hat{b}}_p = \underline{\underline{F}}_N^H\,\underline{\hat{B}}_p \tag{58}$$

for $p = 0, ..., P-1$, and fed to a CPM demodulator. It is assumed that the residual ISI after the equalizer can be modeled as AWGN. Therefore, the trellis structure used by the Viterbi decoder is the same as the one described in Section III-B. The metric at time n associated with the branch i of the trellis on the other hand, corresponding to the combination of Laurent coefficients $\underline{\tilde{b}}^i$, is now calculated as

$$\lambda_n^i = \sum_{p=0}^{P-1} \eta_p \left|\hat{b}_{p,n} - \tilde{b}_p^i\right|^2, \tag{59}$$

where η_p is the energy in the p^{th} LP $l_p(t)$ [10]. We can now see how our receiver exploits the correlation in the LPs by calculating the weights as in (54), whereas the SoA receiver cannot exploit it anymore, as shown by (59).

V. Equalization Complexity Reduction

In Section V-A of this section, we first explore how our MMSE equalizer (48) can be reduced to a ZF equalizer, and how this latter can be calculated efficiently. Afterwards, we study how the computational complexity of the MMSE equalizer (48) can be reduced in Section V-B.

A. Zero-Forcing Equalizer

Our MMSE equalizer (48) can easily be simplified to a ZF equalizer by letting $\sigma_n^2 \to 0$

$$\underline{\underline{G}}_{\text{ZF}} = \left[\underline{\underline{H}}^H\,\underline{\underline{H}}\right]^{-1}\underline{\underline{H}}^H. \tag{60}$$

In contrast to (48), the calculation of this equalizer does not depend on the structure of the CPM autocorrelation matrix. Therefore, we can exploit the knowledge of the structure of $\underline{\underline{H}}$, which consists of four diagonal submatrices, to calculate (60) efficiently. For this purpose, we define the $2N \times 2N$ permutation matrix

$$\underline{\underline{P}} \triangleq \left[\underline{\underline{I}}_N \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix},\ \underline{\underline{I}}_N \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix}\right] \tag{61}$$

so that $\underline{\underline{P}}^{-1} = \underline{\underline{P}}^H$, which allows us to transform $\underline{\underline{H}}$ into a block diagonal matrix $\underline{\underline{H}}_P$ as

$$\underline{\underline{H}}_P = \underline{\underline{P}}\,\underline{\underline{H}}\,\underline{\underline{P}}^H. \tag{62}$$

Using (62) in (60) yields

$$\underline{\underline{G}}_{\text{ZF}} = \underline{\underline{P}}^H \left[\underline{\underline{H}}_P^H\,\underline{\underline{H}}_P\right]^{-1}\underline{\underline{H}}_P^H\,\underline{\underline{P}}. \tag{63}$$

Unlike (60), calculating (63) only requires the inversion of a block diagonal matrix with 2×2 submatrices on its diagonal, which is a very low-complexity operation. Therefore, this ZF equalizer is a good alternative for the MMSE equalizer at high signal-to-noise ratios (SNR) or when the higher complexity of the MMSE equalizer calculation is not acceptable.

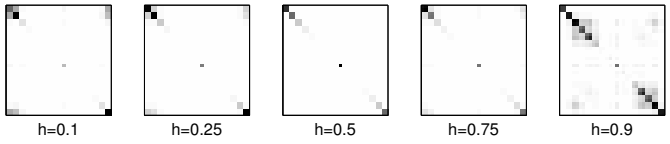

Fig. 5. Energy distribution of $\underline{\underline{R}}_{SS,P}^{-1}$ for different modulation indices and block length $N = 16$. Most energy is concentrated along the main block diagonal.

B. MMSE Equalizer Complexity Reduction

The primary aim of equalizing in the FD rather than in the TD is complexity reduction. By transforming the signal into the FD, we can diagonalize the channel matrix $\underline{\underline{H}}$ so that it can be inverted at very low complexity. However, in the MMSE equalizer (48) the inverted autocorrelation matrix of the CPM signal $\underline{\underline{R}}_{SS}^{-1}$ shows up. This matrix is not diagonal as a CPM signal is highly correlated. In this section, we study its structure and prove that we can approximate it by a block diagonal matrix without a noticeable performance loss. This way we regain the low complexity advantage of FDE. We first apply permutation $\underline{\underline{P}}$, defined in (61), to $\underline{\underline{R}}_{SS}^{-1}$ to obtain $\underline{\underline{R}}_{SS,P}^{-1}$

$$\underline{\underline{R}}_{SS,P}^{-1} = \underline{\underline{P}}\, \underline{\underline{R}}_{SS}^{-1}\, \underline{\underline{P}}^{H}. \tag{64}$$

Using (62) and (64) in (48) yields

$$\underline{\underline{G}}_{\text{MMSE}} = \underline{\underline{P}}^{H} \left[\underline{\underline{R}}_{SS,P}^{-1} + \frac{1}{\sigma_n^2} \underline{\underline{H}}_P^H\, \underline{\underline{H}}_P \right]^{-1} \frac{1}{\sigma_n^2}\, \underline{\underline{H}}_P^H\, \underline{\underline{P}} \tag{65}$$

and for simplicity we define

$$\underline{\underline{D}} \triangleq \underline{\underline{R}}_{SS,P}^{-1} + \frac{1}{\sigma_n^2} \underline{\underline{H}}_P^H\, \underline{\underline{H}}_P. \tag{66}$$

The complexity of the calculation of the MMSE equalizer will be dominated by the inversion of $\underline{\underline{D}}$. We therefore study its structure here. As stated above, $\underline{\underline{H}}_P$ is always block diagonal. The second term of $\underline{\underline{D}}$ is therefore also always block diagonal. The energy distribution of the first term $\underline{\underline{R}}_{SS,P}^{-1}$ is shown for different modulation indices in Fig. 5 for a block length $N = 16$. The darker the shade of gray, the more energy is concentrated in that part of the matrix. The value $N = 16$ is too small for practical systems, but it is chosen just for this illustration as it allows us to illustrate that most energy is concentrated along the main block diagonal.

To formalize this observation mathematically for practical values of N, we define

$$\underline{\underline{C}} \triangleq \underline{\underline{I}}_N \otimes \underline{\underline{J}}_2 \tag{67}$$

where $\underline{\underline{J}}_2$ is a 2×2 unit matrix. Using $\underline{\underline{C}}$, we can calculate

$$E_\varepsilon = \frac{\left\| \underline{\underline{C}} \odot \underline{\underline{R}}_{SS,P}^{-1} \right\|_F^2}{\left\| \left(\underline{\underline{J}}_{2N} - \underline{\underline{C}} \right) \odot \underline{\underline{R}}_{SS,P}^{-1} \right\|_F^2} \tag{68}$$

where $\|\cdot\|_F^2$ represents the Frobenius norm. In words, E_ε is the ratio of the energy in the block-diagonal elements to the energy in the remaining elements of $\underline{\underline{R}}_{SS,P}^{-1}$. Fig. 6 shows E_ε versus block size N ranging from 64 to 512, for $h = 0.25$ and $h = 0.9$. For $h = 0.5$, $\underline{\underline{R}}_{SS,P}^{-1}$ is purely block diagonal, so that $E_\varepsilon = \infty$. For all cases, high values for E_ε are obtained. We can thus expect that $\underline{\underline{R}}_{SS,P}^{-1}$ can be very well approximated by $\underline{\underline{C}} \odot \underline{\underline{R}}_{SS,P}^{-1}$. Moreover, $\underline{\underline{R}}_{SS,P}^{-1}$ is only the first term of $\underline{\underline{D}}$, the matrix that is actually to be inverted. Therefore, we propose a low-complexity approximation to $\underline{\underline{D}}^{-1}$ as

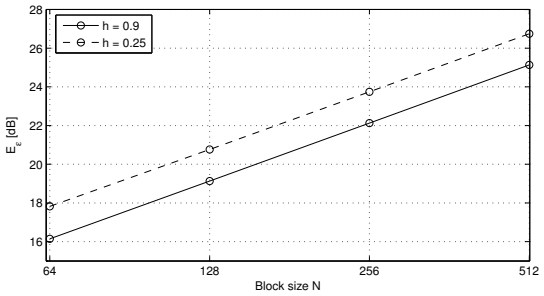

Fig. 6. Proportion of energy in the block diagonal elements of $\underline{\underline{R}}_{SS,P}^{-1}$ compared to the energy in the remaining elements. The major part of the energy is clearly always in the block diagonal elements (high E_ε values). This proportion grows with the blocksize.

$$\underline{\underline{D}}^{-1} \approx \left(\underline{\underline{C}} \odot \underline{\underline{D}} \right)^{-1} \tag{69}$$

which means that we neglect the elements outside the main block diagonal of $\underline{\underline{D}}$, and then invert this matrix.

To assess the performance of the equalizers (63) and (65) and of the approximation (69), extensive simulations were performed. In Section VII, we will present the results and see that the effect of approximation (69) is always negligible. But first we still present a complexity analysis in the next section.

VI. Computational Complexity Analysis

In this section, the computational complexity of the CPM-FDE techniques presented in this paper is compared to the SoA linear CPM-FDE algorithm, as well as to a system using TD equalization. We follow the approach of [10] and express the complexity in *floating-point operations* (flops) per block. As this paper focuses on the equalization complexity reduction, channel coding is not taken into account. We assume a digital oversampling rate $f_s = 2/T$ in the receiver. The results are presented in Table I and discussed below.

For this complexity analysis, we assume ML CPM demodulation. In other words, all P LPs are used in the receiver and the Viterbi detector operates over all S states of the chosen CPM scheme [8]. The Viterbi detector is the same in all receivers and requires $\mathcal{O}(SMN)$ flops. However, as explained in Section III-B, in our approach this ML demodulator is independent from the proposed equalization technique so that it can be replaced by any reduced-complexity CPM demodulator to further reduce the system complexity.

Calculating an MMSE equalizer operating in the TD requires the inversion of a $2N\times2N$ matrix, which takes approximately $(2N)^3$ flops. The equalizer is followed by a filterbank with P filters matched to the LPs, an operation taking

TABLE I
COMPUTATIONAL COMPLEXITY COMPARISON BETWEEN DIFFERENT RECEIVER TYPES. TYPICAL VALUES ARE $N \in \{64, 128, 512\}$, $S \in [4, 80]$, $M \in \{2, 3, 4\}$ AND $P \in [1, 16]$.

Receiver type	Number of flops per block: FFT and IFFTs	Equalizer calculation	Equalization and filtering	Viterbi decoding
Linear MMSE-TDE	0	$+8N^3$	$+\mathcal{O}(PN^3)$	$+\mathcal{O}(SMN)$
SoA linear MMSE-FDE	$2N\log_2(2N) + PN\log_2(N)$	$+8N^3$	$+\mathcal{O}(PN^3)$	$+\mathcal{O}(SMN)$
SoA decision feedback FDE	$2N\log_2(2N) + PN\log_2(N)$	$+(8+12P^2)N^3$	$+\mathcal{O}(PN^3)$	$+\mathcal{O}(SMN)$
New linear MMSE-FDE	$2N\log_2(N) + PN\log_2(N)$	$+8N^3$	$+\mathcal{O}(PN^3)$	$+\mathcal{O}(SMN)$
New reduced-complexity linear FDE (MMSE or ZF)	$2N\log_2(N) + PN\log_2(N)$	$+2N$	$+\mathcal{O}(PN)$	$+\mathcal{O}(SMN)$

$\mathcal{O}(PN^3)$ flops. To reduce the complexity of these operations, we will move them to the FD.

However, FDE requires FFT and inverse FFT (IFFT) operations in the receiver. First we look at the FFT. In the SoA receiver, the received signal is transformed into the FD by one size-$2N$ FFT, whereas in this paper the 2 polyphase components are transformed into the FD by 2 size-N FFTs. As the complexity of an FFT is $N\log_2 N$, this means that we save in the order of 10% in computational and memory requirements imposed on the receiver compared to the SoA receiver for typical block sizes. As we are aiming at low-complexity systems, this is a small but non-negligible saving. Concerning the IFFTs, there is no difference beween the SoA receiver and our new one as both transform the signal back into the TD by P size-N IFFTs. However, this number can be reduced by using reduced-complexity CPM demodulators, as mentioned above.

As explained in the previous section, the complexity of calculating both the SoA linear MMSE or decision feedback FD equalizer and our new MMSE FD equalizer depends on the structure of $\underline{\underline{R}}_{SS,P}^{-1}$. General conclusions about the structure of this matrix cannot be drawn, as it is related to the modulation index h in a very nontransparant way [5]. Therefore, if we do not take any information about its structure into account, the equalizer calculation requires a full $2N \times 2N$ matrix inversion. The complexity of this operation therefore remains $(2N)^3$ flops, the same as in the TD. Clearly we cannot exploit the complexity-reduction advantage of FDE in this way.

However, as could be seen from Fig. 5, $\underline{\underline{R}}_{SS,P}^{-1}$ is in general sparse. This inspired us to apply approximation (69). The matrix to be inverted is then always a $2N \times 2N$ block-diagonal matrix with 2×2 blocks on its diagonal. The equalizer computation complexity is therefore only $2N$. Moreover, as we implement both the equalizer and the matched filterbank in the FD, the complexity of these operations is only $\mathcal{O}(PN)$. We have thus regained the originally targeted complexity-reduction advantage of FDE. Moreover, as we will show in the next section, this approximation does not cause a noticeable performance loss.

VII. SIMULATION RESULTS

For simulations, the binary 3-RC CPM scheme was chosen. Here, 3-RC refers to the raised cosine pulse shape $f(t) = (1 - \cos\frac{2\pi t}{LT})/2LT$ with $L = 3$. Results with modulation index $h = 0.25$, $h = 0.5$ and $h = 0.9$ are presented. A huge bandwidth is available at 60 GHz. Hence, the bit rate $R_b = 1$ Gbit/s is chosen. For this system, the channel is severely frequency-selective. Therefore, a blocksize $N = 256$ and CP length $N_P = 64$ are chosen. The receiver lowpass (LP) filter is modeled as a raised cosine filter with roll-off factor $R = 0.5$. The multipath channel $h(t)$ is simulated using the Saleh-Valenzuela (SV) channel model [21] and the simulated 60 GHz indoor environment is described in [22]. The base station has an omni-directional antenna with 120° beam width and is located in the center of the room. The terminal has an omni-directional antenna with 60° beam width and is placed at the edge of the room. The corresponding SV parameters are $1/\Lambda = 75$ ns, $\Gamma = 20$ ns, $1/\lambda = 5$ ns and $\gamma = 9$ ns.

Fig. 7 shows the bit error rate (BER) of our ZF (63), MMSE (48) and approximated MMSE (69) equalizers in the 60 GHz environment. To verify the simulation framework, a reference curve in AWGN that can be compared to [23] is also provided. The BER is presented in function of E_b/N_0 where E_b is the energy per bit and N_0 is the noise one-sided power spectral density (PSD). First, we observe that increasing h lowers the BER. This is because the higher h, the higher the minimum Euclidean distance of the CPM scheme. Second, the MMSE equalizer always performs better than the ZF. The gap between ZF and MMSE performance becomes larger as h grows. This is because a larger h means that more correlation is introduced in the CPM signal. This correlation can be better exploited by the MMSE equalizer since it takes into account $\underline{\underline{R}}_{SS,P}^{-1}$ as it can be seen from (65). Third, there is no noticeable difference between the exact MMSE and the approximation: the curves almost perfectly coincide.

Moreover, we notice two flooring phenomena. First, for $h = 0.25$, the curves start to floor at high E_b/N_0 due to bad channels containing spectral zeros in the chosen set. This was verified by removing the 10% worst channels, which made the flooring disappear (not shown here). The linear equalizers presented in this paper cannot mitigate this effect. If it is important, iterative equalizers such as presented in [13] and [14] should be considered. Second, the BER curve of the $h = 0.9$ MMSE receiver also starts to floor. This is because part of the PSD of the $h = 0.9$ CPM scheme falls beyond the passband of the LP filter. Therefore, part of the information is lost. To mitigate this problem, a higher sampling rate should be used in the receiver, as it was explained in Section II-A.

In Fig. 8, we compare the BER performance of our new MMSE equalizer (48) to the SoA receiver. For $h = 0.25$, the

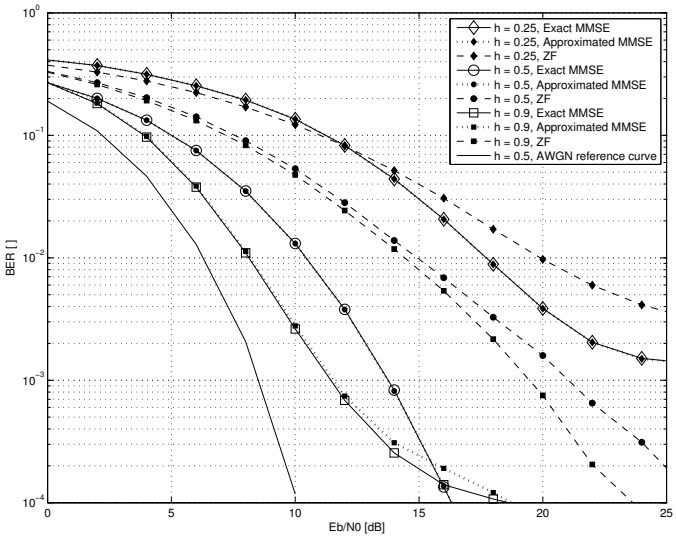

Fig. 7. BER in the 60 GHz multipath environment for our MMSE (solid lines), approximated MMSE (dotted lines) and ZF (dashed lines) receivers for $h = 0.25$, $h = 0.5$ and $h = 0.9$. A reference curve in AWGN for $h = 0.5$ is also provided.

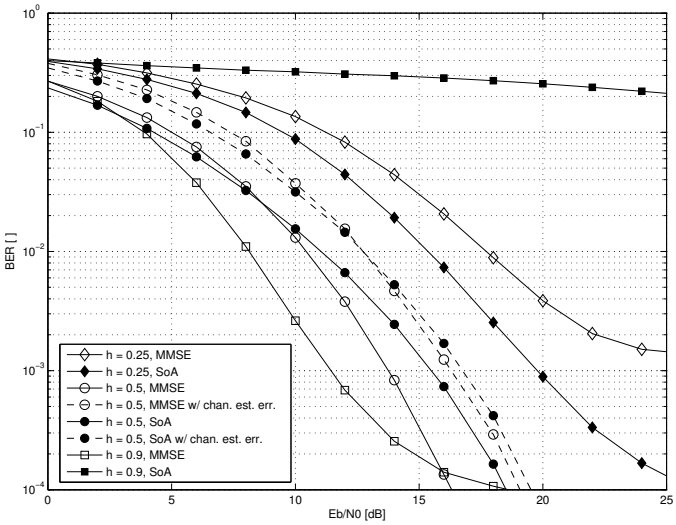

Fig. 8. BER in the 60 GHz multipath environment for our MMSE (white markers) and the SoA (black markers) receivers for $h = 0.25$, $h = 0.5$ and $h = 0.9$. For $h = 0.5$, the BER with channel estimation errors made in the receiver (dashed lines) is also provided for both receivers.

SoA receiver outperforms the MMSE receiver. For $h = 0.5$, the MMSE receiver outperforms the SoA receiver for high E_b/N_0 values. For $h = 0.9$, the SoA receiver cannot recover the information reliably anymore whereas the MMSE receiver keeps on performing very well. These observations can be explained by the correlation properties of the PCs. Studying (4), we note that the larger h, the more correlation is introduced in the PCs. Comparing (54) and (59), our new MMSE receiver exploits this correlation in the Viterbi detector by calculating the weights as in (54), whereas the correlation is partly lost and can therefore not be used anymore in the SoA receiver, which calculates the weights as in (59). For $h = 0.25$, the correlation is small and therefore so is the loss. The approximation made in our MMSE receiver that models the residual ISI and colored noise at the input of the CPM demodulator as AWGN is then more deteriorating. Therefore, the SoA receiver outperforms the MMSE receiver. For $h = 0.5$, the correlation exploitation starts to pay off, and the MMSE receiver starts to outperform the SoA receiver. For $h = 0.9$, the correlation becomes so important that it cannot be neglected by the receiver anymore. The SoA receiver therefore can no longer recover the information properly, whereas our new MMSE receiver still performs very well.

For $h = 0.5$ finally, results with channel estimation errors made in the receiver are also provided. The channel estimate used by the receiver is $\hat{h} = h + h_\epsilon$. Here, h is the perfect channel and h_ϵ is a white Gaussian error term with zero mean. Its variance satisfies $\sigma_h^2/\sigma_{h_\epsilon}^2 = E_b/N_0$. Our MMSE receiver suffers from a constant degradation of about 3 dB. It is more sensitive to channel estimation errors than the SoA receiver. For this latter, the gap between the perfect and channel estimate curves becomes smaller when E_b/N_0 grows. This can be explained as follows. In the low E_b/N_0 region, the SoA receiver suffers relatively more from channel estimation errors than from the lack of correlation exploitation in the Viterbi decoder. However, when the E_b/N_0 grows, this correlation exploitation becomes more important as the performance gets limited by ISI rather than by noise. The channel estimation error effect on the other hand, stays constant. Therefore, the gap narrows. For $h = 0.25$ and $h = 0.9$, the same results were obtained (not shown).

VIII. Conclusion

We presented a new high-performance, low-complexity approach to linear FDE of block-based CPM systems. To support our theory we first developed a new polyphase matrix model, valid for any block-based CPM system, according to the familiar framework of [16]. The main difference with respect to the SoA receiver is that we separated channel equalization on the one hand and CPM demodulation on the other. This enabled us to exploit the correlation properties of the CPM signal in the demodulator, after the channel equalizer. Calculating the MMSE equalizer required the inversion of a nondiagonal matrix. This defeats the primary objective of FDE, namely low-complexity equalization requiring only inversion of diagonal matrices in the frequency domain. Therefore, in order to restore the original advantage of FDE, we have shown that the CPM autocorrelation matrix can be approximated by a block-diagonal matrix. Finally, our new MMSE equalizer was also simplified to a low-complexity zero-forcing equalizer.

As shown by simulations in a 60 GHz environment, separating channel equalization and CPM demodulation leads to significantly better BER performance for modulation indices $h = 0.5$ and $h = 0.9$. Only for $h = 0.25$, we perform slightly worse than the state of the art receiver because in this case the correlation in the CPM signal is small. The gain of exploiting it is therefore smaller than the deterioration caused by our assumption that colored noise and residual ISI after the equalizer can be modeled as AWGN. Finally, simulations confirmed that our MMSE complexity reduction technique

can be applied for any modulation index without noticeable performance loss.

ACKNOWLEDGMENT

W. Van Thillo thanks the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) for funding this PhD research.

REFERENCES

[1] P. Smulders, "Exploiting the 60 GHz band for local wireless multimedia access: prospects and future directions," *IEEE Commun. Mag.*, vol. 40, no. 1, pp. 140–147, January 2002.

[2] N. Guo, R.C. Qiu, S.S. Mo, and K. Takahashi, "60 GHz millimeter-wave radio: principle, technology and new results," *EURASIP J. Wireless Commun. and Networking*, 2007.

[3] J. Nsenga, W. Van Thillo, F. Horlin, A. Bourdoux, and R. Lauwereins, "Comparison of OQPSK and CPM for communications at 60 GHz with a non-ideal front end," *EURASIP J. Wireless Commun. and Networking*, 2007.

[4] R.C. Daniels and R.W. Heath, Jr., "60 GHz wireless communications: emerging requirements and design recommendations," *IEEE Veh. Technol. Mag.*, vol. 2, no. 3, pp. 41–50, Sept. 2007.

[5] P.A. Laurent, "Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulses (AMP)," *IEEE Trans. Commun.*, vol. 34, no. 2, pp. 150–160, February 1986.

[6] 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, September 1995.

[7] W. Van Thillo, J. Nsenga, F. Horlin, A. Bourdoux, and R. Lauwereins, "The generalized linear decomposition of multilevel CPM signals," in *Proc. IEEE ICASSP*, April 2007.

[8] G.K. Kaleh, "Simple coherent receivers for partial response continuous phase modulation," *IEEE J. Sel. Areas Commun.*, vol. 7, no. 9, pp. 1427–1436, December 1989.

[9] 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, no. 4, pp. 58–66, April 2002.

[10] F. Pancaldi and G.M. Vitetta, "Equalization algorithms in the frequency domain for continuous phase modulations," *IEEE Trans. Commun.*, vol. 54, no. 4, pp. 648–658, April 2006.

[11] J. Tan and G.L. Stüber, "Frequency domain equalization for continuous phase modulation," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2479–2490, September 2005.

[12] P. Bianchi, P. Loubaton, and F. Sirven, "Non data-aided estimation of the modulation index of continuous phase modulations," *IEEE Trans. Signal Processing*, vol. 52, no. 10, pp. 2847–2861, October 2004.

[13] B. Özgül, M. Koca, and H. Deliç, "Frequency-domain doubly-iterative equalization of continuous phase modulation," in *Proc. IEEE Globecom*, November 2006.

[14] B. Özgül, M. Koca, and H. Deliç, "Doubly iterative equalization of continuous-phase modulation," *IEEE Trans. Commun.*, vol. 55, no. 11, pp. 2114–2124, November 2007.

[15] B.E. Rimoldi, "A decomposition approach to CPM," *IEEE Trans. Inf. Theory*, vol. 34, no. 2, pp. 260–270, March 1988.

[16] Z. Wang and G.B. Giannakis, "Wireless multicarrier communications," *IEEE Signal Process. Mag.*, vol. 17, no. 3, pp. 29–48, May 2000.

[17] C.-E. Sundberg, "Continuous phase modulation," *IEEE Commun. Mag.*, vol. 24, no. 4, pp. 25–38, April 1986.

[18] W. Van Thillo, J. Nsenga, R. Lauwereins, V. Ramon, A. Bourdoux, and F. Horlin, "A new symbol block construction for CPM with frequency domain equalization," in *Proc. IEEE ICC*, May 2008.

[19] W. Van Thillo, J. Nsenga, R. Lauwereins, V. Ramon, A. Bourdoux, and F. Horlin, "Applying frequency domain equalization to precoded CPM," in *Proc. IEEE ICASSP*, March 2008.

[20] A. Klein, G.K. Kaleh, and P.W. Baier, "Zero forcing and minimum mean-square-error equalization for multiuser detection in code-division multiple-access channels," *IEEE Trans. Veh. Technol.*, vol. 45, no. 2, pp. 276–287, May 1996.

[21] A. Saleh and R. Valenzuela, "A statistical model for indoor multipath propagation," *IEEE J. Sel. Areas Commun.*, vol. 5, no. 2, pp. 128–137, February 1987.

[22] J.H. Park, Y. Kim, Y.S. Hur, K. Lim, and K.H. Kim., "Analysis of 60 GHz band indoor wireless channels with channel configurations," in *Proc. IEEE PIMRC*, September 1998, pp. 617–620.

[23] K. Murota and K. Hirade, "GMSK modulation for digital mobile radio telephony," *IEEE Trans. Commun.*, vol. 29, no. 7, pp. 1044–1050, July 1981.