Iterative Multiuser Detection of Spectrally Efficient FDMA CPM

Nele Noels, *Member, IEEE*, and Marc Moeneclaey, *Fellow, IEEE*

***Abstract*—This paper presents two new iterative multiuser detection algorithms for spectrally efficient frequency division multiple access continuous phase modulation in additive white Gaussian noise. The detectors are derived from the sum-product algorithm and the factor graph framework, performing approximate maximum *a posteriori* (MAP) bit detection. A factor graph of the actual multiuser detection problem is considered, rather than only factor graphs of the individual single user detection problems combined with *ad-hoc* inter-user interference cancellation. The practical implementation of the detectors is based on the assumption that a suitable set of sum-product messages can be approximated by a Gaussian distribution. This methodology allows to considerably reduce the computational complexity and memory size requirements as compared to a straightforward application of the sum-product rules. The proposed algorithms further succeed in achieving a good error performance.**

***Index Terms*—Continuous phase modulation, frequency division multiaccess, satellite communication.**

I. Introduction

THE current paper is concerned with the derivation of practical multiuser (MU) receivers for uplink satellite communication of bit-interleaved coded continuous-phase modulated (BIC-CPM) signals using a spectrally efficient frequency division multiple access (FDMA) scheme.

A FDMA scheme defines a number of carrier frequencies and allocates a different carrier frequency to each individual user. In spectrally efficient FDMA systems, the spacing between the carrier frequencies that are assigned to different users is kept small, such that the leakage of the neighboring signal energy into the desired frequency band cannot be ignored. This leakage signal is referred to as inter-user interference (IUI). In order to avoid a severe performance degradation, the IUI must be dealt with by the receiver.

BIC-CPM is a combination of continuous-phase modulation (CPM) and bit-interleaved coded modulation (BICM). CPM is a modulation method commonly used in wireless modems [1]. The transmitted CPM waveform has a constant envelope, and its phase is a continuous function of time that changes according to the digital information to be transmitted. CPM is attractive because of its high power and spectral efficiency, and because of its robustness to non-linearities. CPM is also well-suited for use in BICM schemes.

BICM is a pragmatic yet powerful coding technique to improve system performance [2]. Well-known binary channel encoders are combined with basic off-the-shelf modulators, with a bit interleaver connecting these two entities. Although the optimal (coherent) symbol-by-symbol detection [3] of a BICM signal is prohibitively complex, there exist approximate iterative detectors with reasonable complexity that yield a very good performance. Such practical detectors can be derived from the sum-product (SP) algorithm and the factor graph (FG) framework [4].

In the past few years, several advanced techniques have been proposed for iterative MU bit detection in BIC-CPM systems using spectrally efficient FDMA [5], [6].

In [6], an intuitive approach to MU detection is proposed. Conventional FG-based single user (SU) detectors are employed. To cope with the presence of IUI in the received signal, the receiver is equipped with a separate module for IUI cancellation. The latter module is characterized by a low computational complexity and limited memory requirements. By iterating between the SU detectors and the IUI cancellation module, the receiver executes a practical form of MU detection. When the SU detectors themselves involve an iterative process (of demapping and decoding), the IUI cancellation iterations are intertwined with those of the detectors to reduce the overall receiver complexity. The performance improvement over standard SU detection is significant. A similar detection algorithm is also considered in [5].

A more fundamental approach to MU detection is the following. Consider a FG of the actual MU detection problem. Such a FG represents a suitable factorization of the joint *a-posteriori* probability (APP) of the bit sequences from all the users, given the observed received MU signal. Running the SP algorithm on this graph yields an estimate of the marginal bit APPs. The latter APPs can be used to perform maximum *a posteriori* (MAP) bit detection.

Optimal MU detection is obtained when the employed FG corresponds to a tree. In that case, running the SP algorithm is straightforward and yields the exact marginal bit APPs. Using these APPs to perform MAP bit detection results in minimum bit error rate performance. Unfortunately, the complexity of this optimal MU detector is extremely high and exponential in the number of interfering users [5]. It is therefore not suited for use in practice and one has to resort to approximations.

The practical MU detector proposed in [5] is based on a FG that contains cycles (i.e., paths from a node to itself via other

Manuscript received July 29, 2011; revised February 10, 2012 and June 18, 2012; accepted June 20, 2012. Date of publication June 28, 2012; date of current version September 11, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Peter J. Schreier. The work of N. Noels was supported by the Research Foundation-Flanders (FWO Vlaanderen). Part of this work was supported by Newtec Cy N.V. and by the European Space Agency, ESA-ESTEC, Noordwijk, The Netherlands, under Contract number 22397/09/NL/AD.

The authors are with the Department of Telecommunications and Information Processing, Ghent University, 9000 Gent, Belgium (e-mail: nnoels@telin.UGent.be; mm@telin.UGent.be).

Digital Object Identifier 10.1109/TSP.2012.2206586

nodes). When applied to a FG with cycles, the SP algorithm becomes an iterative process that yields, even after convergence, only an approximation of the marginal bit APPs. The complexity of the receiver from [5] increases only proportional to the number of interfering users. Unfortunately, the proportionality coefficient is typically quite large for CPM schemes yielding a high spectral efficiency. In [5], the authors therefore propose making the simplifying assumption that only the users in the most adjacent frequency bands significantly contribute to the IUI. This approximation may substantially reduce the computational complexity and the memory requirements of the receiver. It is accurate only when the carrier spacing between adjacent users is sufficiently large.

Two new approximate MU detection algorithms are considered in this paper. They are referred to as g6-MU-FG-GA and g3-MU-FG-GA. The g6-MU-FG-GA algorithm results from applying a different approximation to (a slight variation of) the MU FG with cycles from [5]: a suitable set of SP messages are assumed to be Gaussian distribution functions. The IUI from all the users is accounted for. The computational complexity and memory requirements per user of the g6-MU-FG-GA receiver is considerably lower than for the simplest version of the receiver from [5], that considers only 2 interfering users. The proposed g3-MU-FG-GA algorithm is even less complex than the g6-MU-FG-GA algorithm. It results from applying the same Gaussian approximation methodology to a modified MU FG that results from clustering a suitable set of function nodes in order to reduce the cardinality of the variable alphabets in the graph. This, in turn, simplifies the SP process.

II. System Model

The following system model is considered. The transmitter of user u encodes the N_b-element information bit vector $\mathbf{b}^{(u)} = \left\{b_k^{(u)}\right\}$ into a vector of N_c coded bits $\mathbf{c}^{(u)} = \left\{c_l^{(u)}\right\}$. This vector of coded bits is subsequently interleaved and mapped to an N element symbol vector $\mathbf{a}^{(u)} = \left(a_0^{(u)}, \ldots, a_{N-1}^{(u)}\right)$, with $a_n^{(u)}$ taking values from the M-ary alphabet $\Omega_M = \{\pm 1, \pm 3, \ldots, \pm(M-1)\}$. The resulting symbol vector is then used to generate the complex envelope $s_{SU}^{(u)}(t)$, for $0 \leq t < NT$, of the CPM signal from user u:

$$s_{SU}^{(u)}(t) = e^{j\psi\left(t;\mathbf{a}^{(u)}\right)}, \tag{1}$$

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

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

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

where $n = 0, 1, \ldots, N-1$. The quantity $\mathbf{S}_n^{(u)}$ in (3) describes the CPM state transition during the nth symbol interval $[nT, (n+1)T[$ of the transmitter signal $s_{SU}^{(u)}(t)$ from user u:

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

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

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

The first element $\sigma_{n,0}^{(u)}$ of the CPM state vector $\boldsymbol{\sigma}_n^{(u)}$ is referred to as the phase state. When K is even, $\sigma_{n,0}^{(u)} \in \frac{2\pi}{P} \cdot \{0, 1, \ldots, P-1\}$. When K is odd, $\sigma_{n,0}^{(u)}$ belongs alternatingly to $\frac{2\pi}{P} \cdot \{0, 1, \ldots, P-1\}$ and $\frac{\pi}{P} \cdot \{1, 3, \ldots, 2P-1\}$. For $i > 0$, $\sigma_{n,i}^{(u)}$ take values from Ω_M. At each time instant n the CPM transmission scheme has PM^{L-1} possible states. Given the symbol vector $\mathbf{a}^{(u)}$ and starting from a given initial CPM state $\boldsymbol{\sigma}_0^{(u)}$, the vectors $\boldsymbol{\sigma}_n^{(u)}$, with $n = 1, 2, \ldots, N$, can be computed recursively according to the following equations:

$$\sigma_{n,0}^{(u)} = \left[\sigma_{n-1,0}^{(u)} + \pi h \sigma_{n-1,1}^{(u)}\right]_{2\pi}, \tag{6}$$

$$\sigma_{n,i}^{(u)} = \sigma_{n-1,i+1}^{(u)}, 1 \leq i \leq L-2, \tag{7}$$

$$\sigma_{n,L-1}^{(u)} = a_{n-1}^{(u)}, \tag{8}$$

where $[x]_{2\pi}$ denotes modulo-2π reduction of x to the interval $[0, 2\pi[$. The complete sequence of CPM state vectors $\boldsymbol{\sigma}_n^{(u)}$ is grouped in the vector $\boldsymbol{\sigma}^{(u)} = \left(\boldsymbol{\sigma}_0^{(u)}, \boldsymbol{\sigma}_1^{(u)}, \ldots, \boldsymbol{\sigma}_N^{(u)}\right)$. Similarly, the complete sequence of CPM state transition identifiers $\mathbf{S}_n^{(u)}$ is grouped in the vector $\mathbf{S}^{(u)} = \left(\mathbf{S}_0^{(u)}, \mathbf{S}_1^{(u)}, \ldots, \mathbf{S}_{N-1}^{(u)}\right)$.

A group of U users are simultaneously active. They asynchronously transmit frequency-shifted versions of their signals $s_{SU}^{(u)}(t)$, $u = 1, 2, \ldots, U$ over an additive white Gaussian noise (AWGN) channel, which is typical for satellite communications. The short-hand notations $\mathbf{b}$, $\mathbf{c}$, $\mathbf{a}$, $\boldsymbol{\sigma}$ and $\mathbf{S}$ respectively collect the information bit sequences, the coded bit sequences, the data symbol sequences, the CPM state vector sequences and the CPM state transition identifier sequences from all the users: $\mathbf{b} = \left\{\mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \ldots, \mathbf{b}^{(U)}\right\}$, $\mathbf{c} = \left\{\mathbf{c}^{(1)}, \mathbf{c}^{(2)}, \ldots, \mathbf{c}^{(U)}\right\}$, $\mathbf{a} = \left\{\mathbf{a}^{(1)}, \mathbf{a}^{(2)}, \ldots, \mathbf{a}^{(U)}\right\}$, $\boldsymbol{\sigma} = \left\{\boldsymbol{\sigma}^{(1)}, \boldsymbol{\sigma}^{(2)}, \ldots, \boldsymbol{\sigma}^{(U)}\right\}$ and $\mathbf{S} = \left\{\mathbf{S}^{(1)}, \mathbf{S}^{(2)}, \ldots, \mathbf{S}^{(U)}\right\}$. The phase offset, frequency shift and time delay associated with user $u (u = 1, 2, \ldots, U)$ are equal to $\theta^{(u)}$, $\frac{f^{(u)}}{T}$ and $\tau^{(u)}T$, with $f^{(u)}$ and $\tau^{(u)}$ denoting the dimensionless relative (to the symbol interval T) frequency shift and time delay. Equal nominal frequency spacings are considered such that: $f^{(u)} = \left(\frac{u-(U+1)}{2}\right) \cdot NSP$, where NSP is the normalized nominal spacing. The latter spacing is applied at the transmitter in order to facilitate the separation between the contributions of the different users. The values of $\theta^{(u)}$ and $\tau^{(u)}$ are assumed to be known at the start of the detection process. The

overall spectral efficiency $\eta_{spec,U}$ of a system with groups of U users is defined as the number of transmitted information bits per second per Hertz. It amounts to

$$\eta_{spec,U} = \frac{U \cdot N_b}{N \cdot (\Delta f_g + (U-1) \cdot NSP)}, \tag{9}$$

with Δf_g the normalized (with respect to the symbol interval T) spectral spacing between different groups of users. As in [6], we assume that, within a given group of users, all signals are received with the same power. This assumption is not essential -extension of the derivation in Section III to the case of non-equal receive powers is trivial (see e.g., [5])- but it simplifies the equations. Besides, in typical satellite systems, the use of extensive power control and a smart assignment of the available carrier frequencies to the different users allow to minimize the power difference between the users of a same group. It is therefore also reasonable to model the complex baseband representation $s(t)$ of the received signal component, which aggregates the contributions of all users from a group, as $s(t) = \sum_{u=1}^{U} s^{(u)}(t)$, with

$$s^{(u)}(t) = s_{SU}^{(u)}\left(t - \tau^{(u)}T\right) e^{j2\pi f^{(u)}\frac{t}{T}} e^{j\theta^{(u)}} \tag{10}$$

the contribution from the uth user. The latter contribution differs from zero only for $t \in \left[\tau^{(u)}T, \left(\tau^{(u)} + N\right)T\right[$. During this time interval, $s^{(u)}(t)$ from (10) is a unit-power data-dependent process. Without loss of generality, we assume that $\min_u \tau^{(u)} = 0$, implying that all $\tau^{(u)}$, $u = 1, 2, \ldots, U$, are nonnegative and $s(t) = 0$ for $t < 0$. It is further assumed that the signal $s(t)$ can be well-approximated by the sampled sequence $s(kT_s)$, $k = 0, 1, \ldots, N_s - 1$, with $N_s = \left(\left\lceil \max_u \tau^{(u)} \right\rceil + N + 1\right) R_s$, using sampling rate $R_s = \frac{T}{T_s}$, even though $s(t)$ is not band-limited, by choosing R_s sufficiently large. The received baseband signal is applied to a low-pass anti-aliasing filter with bandwith $\frac{R_s}{2T}$ and sampled at R_s samples per symbol interval. It is assumed that the spacing Δf_g between the central frequency of the last user of a group and that of the first user of the adjacent group is sufficiently large, such that the leakage of the signal energy from neighboring groups into the frequency band $\left[-\frac{R_s}{2T}, \frac{R_s}{2T}\right]$ can be safely ignored. The resulting samples r_k, for $k = 0, 1, \ldots, N_s - 1$, can then be modeled as follows:

$$r_k = \sum_{u=1}^{U} s_k^{(u)} + n_k, \tag{11}$$

where $s_k^{(u)}$, $k = 0, 1, \ldots, N_s - 1$, are samples of the receiver signal contribution $s^{(u)}(t)$ from (10), taken at $t = kT_s$ and n_k are zero-mean complex AWGN samples with variance equal to $\frac{N_0 R_s}{E_s}$, with N_0 the noise power spectral density and E_s the energy per symbol period. The samples r_k are conveniently grouped into the vectors $\mathbf{r}_l = (r_l, r_{l+1}, \ldots, r_{l+R_s-1})^T$ and $\mathbf{r} = \left(\mathbf{r}_0^T, \mathbf{r}_{R_s}^T, \ldots, \mathbf{r}_{(N_s-R_s)}^T\right)^T$.

III. MAP Bit Detection

A. Detection Rule

When an information bit is detected erroneously at the receiver, a bit error occurs. Optimal detection, which minimizes the bit error probability is achieved by MAP bit decision of the individual information bits $b_k^{(u)}$. The corresponding bit-by-bit MAP detector maximizes the APP $p\left(b_k^{(u)} | \mathbf{r}\right)$ with respect to $b_k^{(u)}$[3]

$$\hat{b}_k^{(u)} = \arg \max_{b \in \{0,1\}} p\left(b_k^{(u)} = b \,|\mathbf{r}\right). \tag{12}$$

B. FG-Based Detector Design

The APPs $p\left(b_k^{(u)} | \mathbf{r}\right)$ involved in (12) are the marginals of $p(\mathbf{b}|\mathbf{r})$, with $p(\mathbf{b}|\mathbf{r})$ the probability of the set $\mathbf{b}$ of information bit sequences from all the users, given the vector $\mathbf{r}$ of received samples. An efficient way to jointly compute these marginals is to apply the sum-product (SP) algorithm to a suitable factor graph (FG) [4].

A FG is a graphical representation of a function factorization. It consists of a set of function nodes (one for each factor in the factorization) that are connected by variable edges (one for each variable in the factorization). The SP algorithm provides a set of rules for computing messages in the nodes of a FG and for passing them on to neighboring nodes along the edges of the graph. Message passing on a FG that corresponds to a tree is straightforward and yields the exact marginals of all the variables in the graph. When a FG contains cycles (i.e., paths from a node to itself via other nodes) the message passing is an iterative process and one must decide on a particular initialization and scheduling strategy. In that case, the procedure also yields, even after convergence, only an approximation of the variable marginals. Nevertheless, the approximate nature of the SP algorithm when applied to a FG with cycles is often a more than acceptable price for performing efficient function marginalization.

Not every function is easily factorized. A possible solution to circumvent this problem is to introduce additional variables in such a way that the joint function of the original and the additional variables leads to a convenient FG representation. Besides the fact that the choice of additional variables is not unique, in many cases, there is also more than one way to factorize a function, given a set of (original and additional) variables. Each of these function factorizations corresponds to a different FG, leading to a different way to compute the marginals. Choosing a function factorization is therefore a matter of trading off the accuracy of the resulting marginals against the computational complexity associated with their evaluation. In general, decreasing the minimum cycle length of a graph deteriorates the accuracy. However, this is not always true and it sometimes even improves results (as we will see later on). The complexity of the SP algorithm primarily depends on the number of edges and the size of the variable alphabets in the graph.

Several FGs for computing the APPs $p\left(b_k^{(u)} | \mathbf{r}\right)$ will be considered next.

C. State-of-the-Art Solution

We start with (a slight variation of) the FG from [5], extended to the more general case of U asynchronous users. This FG is shown in Fig. 1. The additional variables $\mathbf{s}_x^{(u)}$, $x = 0, 1, \ldots, N_s - 1$, $u = 1, 2, \ldots, U$, in Fig. 1 contain the R_s samples $s^{(u)}(iT_s + xT)$, $i = 0, 1, \ldots, R_s - 1$, that correspond

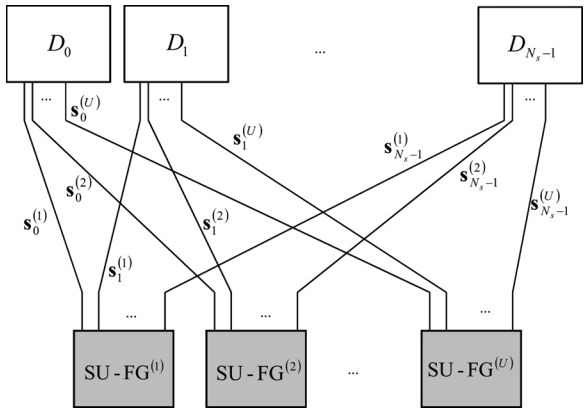

Fig. 1. FG representing the factorization of the joint APP $p(\mathbf{b}|\mathbf{r})$ of the bit sequences of all users from a same group, as proposed in [5].

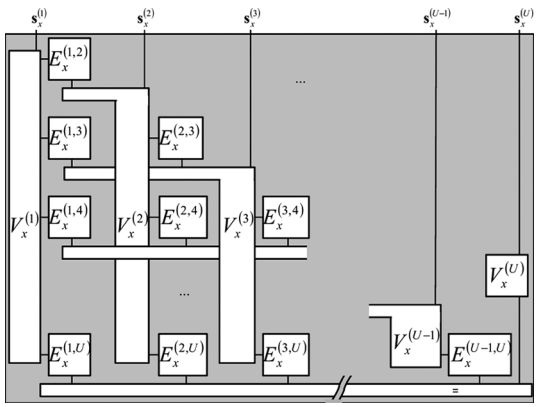

Fig. 2. Further decomposition of the nodes D_x, $u = 1, 2, \ldots, U$, in the FG of Fig. 1, as proposed in [5].

to the xth symbol interval $[xT, (x+1)T[$ of the receiver signal $s^{(u)}(t)$ from (10). It follows from (1)–(8) and (10) that $\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}$, with $N_\tau^{(u)}$ the largest integer smaller than $\tau^{(u)}$, can take PM^{L+1} possible values: one for each possible realization of $\left(\mathbf{S}^{(u)}_{n-1}, a^{(u)}_n\right)$. The nodes D_x, $x = 0, 1, \ldots, N_s - 1$, in Fig. 1 further decompose as shown in Fig. 2. In Fig. 2, the nodes $E_x^{(u,v)}$, $(u, v) \in \{1, 2, \ldots, U\}^2$ with $u < v$, implement the function $F\left(\mathbf{s}_x^{(u)}, \mathbf{s}_x^{(v)}\right)$, with

$$F\left(\mathbf{s}_x^{(u)}, \mathbf{s}_x^{(v)}\right) = \exp\left\{-\frac{2E_s}{N_0 R_s}\Re\left\{\left(\mathbf{s}_x^{(u)}\right)^H \mathbf{s}_x^{(v)}\right\}\right\}, \quad (13)$$

while the nodes $V_x^{(u)}$, $u = 1, 2, \ldots, U$ represent the function $L\left(\mathbf{s}_x^{(u)}; \mathbf{r}_{xR_s}\right) I_{eq,x}^{(u)}[.]$. The indicator function $I_{eq,x}^{(u)}[.]$ equals one when all the variables corresponding to edges connected to node $V_x^{(u)}$ are equal, and zero otherwise. The function $L\left(\mathbf{s}_x^{(u)}; \mathbf{r}_{xR_s}\right)$ is given by:

$$L\left(\mathbf{s}_x^{(u)}; \mathbf{r}_{xR_s}\right) = \exp\left\{\frac{2E_s}{N_0 R_s}\Re\left\{\mathbf{r}_{xR_s}^H \mathbf{s}_x^{(u)}\right\}\right\}. \quad (14)$$

The nodes labeled $\mathrm{SU-FG}^{(u)}$, $u = 1, 2, \ldots, U$, in Fig. 1 enforce the coding and mapping constraints of user u on the samples of the signal transmitted by that user. Using $a_n^{(u)}$, $\boldsymbol{\sigma}_0^{(u)}$, $\boldsymbol{\sigma}_N^{(u)}$ and $\mathbf{S}_n^{(u)}$ as additional variables, these nodes are

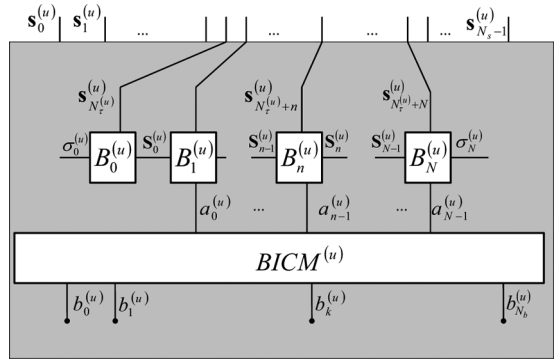

Fig. 3. Further decomposition of the nodes $\mathrm{SU-FG}^{(u)}$, $u = 1, 2, \ldots, U$, in the FG of Fig. 1.

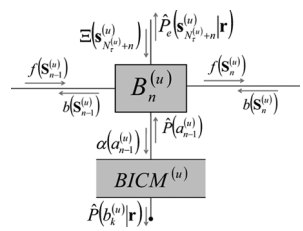

Fig. 4. Messages traveling on the edges of the FG from Fig. 3.

further decomposed as shown in Fig. 3, where $N_\tau^{(u)}$ denotes the largest integer smaller than $\tau^{(u)}$. The node $BICM^{(u)}$ in Fig. 3 imposes the BICM constraint $P\left(\mathbf{a}^{(u)} \middle| \mathbf{b}^{(u)}\right)$ to the information bits and the data symbols of the uth user, and the nodes $B_0^{(u)}$, $B_N^{(u)}$ and $B_n^{(u)}$, $n = 1, 2, \ldots, N-1$, represent the functions $P\left(\boldsymbol{\sigma}_0^{(u)}\right) I_{tr,0}^{(u)}[.]$, $P\left(\boldsymbol{\sigma}_N^{(u)}\right) I_{tr,N}^{(u)}[.]$ and $I_{tr,n}^{(u)}[.]$, $n = 1, 2, \ldots, N-1$, respectively. Here, $P\left(\boldsymbol{\sigma}_0^{(u)}\right)$ is the *a priori* probability of $\boldsymbol{\sigma}_0^{(u)}$, $P\left(\boldsymbol{\sigma}_N^{(u)}\right)$ is the *a priori* probability of $\boldsymbol{\sigma}_N^{(u)}$ and the indicator function $I_{tr,n}^{(u)}[.]$ equals one when the variables corresponding to edges connected to node $B_n^{(u)}$ satisfy the trellis (1)–(8) and (10), and zero otherwise. In the FG resulting from Figs. 1–3, the cycles with minimum length are of length 6, for example, between the nodes $V_x^{(1)}$, $E_x^{(1,2)}$, $V_x^{(2)}$, $E_x^{(2,3)}$, $V_x^{(3)}$ and $E_x^{(1,3)}$ in Fig. 2. The obvious initialization and message passing scheduling strategy for the corresponding iterative SP procedure is outlined in Algorithm 1. The notation for the messages traveling along the edges of the FG of Figs. 1–3 is introduced in Fig. 4. The rules for computing these messages are the standard SP rules from [4], followed by a normalization step such that all messages communicated along the edges of the FG can be interpreted as probability mass functions. This normalization alleviates numerical problems due to finite precision and fixed point arithmetic. The message $\hat{P}\left(b_k^{(u)}|\mathbf{r}\right)$ traveling on the half-edge corresponding to the variable $b_k^{(u)}$ provides an estimate of the APP of the kth information bit of the uth user. A further motivation for the notation of the messages in Fig. 4 is provided as a remark in Section III-C. The per user processing step in Algorithm 1 involves the same operations as one iteration of a conventional FG-based single user detector, except for the additional

evaluation of the messages $\hat{P}_e\left(\mathbf{s}^{(u)}_{N^{(u)}_\tau+n}|\mathbf{r}\right)$. The equations for computing the messages $f\left(\mathbf{S}^{(u)}_n\right)$, $b\left(\mathbf{S}^{(u)}_n\right)$, $\alpha\left(a^{(u)}_n\right)$ and $\hat{P}_e\left(\mathbf{s}^{(u)}_{N^{(u)}_\tau+n}|\mathbf{r}\right)$ follow from applying the SP algorithm to the FG of Fig. 3. These equations are provided in the Appendix. It follows that the number of elementary operations between two real arguments that is required for the evaluation of $f\left(\mathbf{S}^{(u)}_n\right)$, $b\left(\mathbf{S}^{(u)}_n\right)$, $\alpha\left(a^{(u)}_n\right)$ and $\hat{P}_e\left(\mathbf{s}^{(u)}_{N^{(u)}_\tau+n}|\mathbf{r}\right)$ for all u, for all n and for all possible realizations of $\mathbf{S}^{(u)}_n$, $a^{(u)}_n$ and $\mathbf{s}^{(u)}_{N^{(u)}_\tau+n}$ amounts to $11PM^{L+1}$, per user, per CPM symbol and per receiver iteration. Applying the SP algorithm to the FG of Fig. 2 yields the following procedure for updating the messages $\Xi\left(\mathbf{s}^{(u)}_x\right)$ in the multiuser processing step of Algorithm 1. It consists of 2 steps. First, new values are computed for the internal messages (inside the cluster node D_x). Using the short-hand notation $\overset{(v)}{\phi}{}^{\rightarrow}\left(\mathbf{s}^{(u)}_x\right)$ for the SP message travelling from node $E^{(u,v)}_x$ (if $v>u$) or $E^{(v,u)}_x$ (if $v<u$) to node $V^{(u)}_x$ in Fig. 2, we can write:

$$\overset{(v)}{\phi}{}^{\rightarrow}\left(\mathbf{s}^{(u)}_x\right)=\sum_{\mathbf{s}^{(v)}_x}\hat{P}_e\left(\mathbf{s}^{(v)}_x|\mathbf{r}\right)F\left(\mathbf{s}^{(u)}_x,\mathbf{s}^{(v)}_x\right)\times\cdot\,\Xi_{prev}\left(\mathbf{s}^{(v)}_x\right)\left(\overset{(u)}{\vec{\phi}}_{prev}\left(\mathbf{s}^{(v)}_x\right)\right)^{-1},\quad(15)$$

where $\sum_{\mathbf{s}^{(v)}_x}$ denotes the summation over all possible values of $\mathbf{s}^{(v)}_x$, and $\Xi_{prev}\left(\mathbf{s}^{(v)}_x\right)$ and $\overset{(u)}{\vec{\phi}}_{prev}\left(\mathbf{s}^{(v)}_x\right)$ are the values of the messages $\Xi\left(\mathbf{s}^{(v)}_x\right)$ and $\overset{(u)}{\phi}{}^{\rightarrow}\left(\mathbf{s}^{(v)}_x\right)$ computed during the previous iteration. Then, the outgoing messages $\Xi\left(\mathbf{s}^{(u)}_x\right)$ are updated as follows:

$$\Xi\left(\mathbf{s}^{(u)}_x\right)=L\left(\mathbf{s}^{(u)}_x;\mathbf{r}_{xR_s}\right)\prod_{v\neq u}\overset{(v)}{\phi}{}^{\rightarrow}\left(\mathbf{s}^{(u)}_x\right).\quad(16)$$

Evaluating the messages $\Xi\left(\mathbf{s}^{(u)}_x\right)$ according to (15)–(16) requires that the function values of $L\left(\mathbf{s}^{(u)}_x;\mathbf{r}_{xR_s}\right)$ and $F\left(\mathbf{s}^{(u)}_x,\mathbf{s}^{(v)}_x\right)$ are available, for all u, for all $v>u$, for all relevant x and for all possible realizations of $\mathbf{s}^{(u)}_x$ and $\mathbf{s}^{(v)}_x$. The values of $L\left(\mathbf{s}^{(u)}_x;\mathbf{r}_{xR_s}\right)$ and $F\left(\mathbf{s}^{(u)}_x,\mathbf{s}^{(v)}_x\right)$ need to be evaluated only once, at the start of the iterations, but their straightforward computation according to (1)–(8), (10) and (13)–(14) involves a huge amount of elementary operations. It is therefore imperative to employ precomputed look-up tables, referred to as LUT-1 and LUT-2.ℓ, for $\ell=1,2,\ldots,U-1$, to efficiently compute accurate approximations of $L\left(\mathbf{s}^{(u)}_x;\mathbf{r}_{xR_s}\right)$ and $F\left(\mathbf{s}^{(u)}_x,\mathbf{s}^{(v)}_x\right)$, respectively. To limit the size of these tables, the time delays $\tau^{(u)}$ need to be rounded to the nearest integer multiple of a suitable sub-sampling period $T_{ss}=\frac{T}{R_{ss}}=\frac{T_s}{\delta}$, with δ strictly positive and integer valued. More details on the computation of $L\left(\mathbf{s}^{(u)}_x;\mathbf{r}_{xR_s}\right)$ and $F\left(\mathbf{s}^{(u)}_x,\mathbf{s}^{(v)}_x\right)$ using LUT-1 and LUT-2.ℓ, for $\ell=1,2,\ldots,U-1$, are provided in the appendix. The sub-sampling parameter δ affects the performance and the memory requirements of the detector. A larger value of δ results in a more accurate computation of $L\left(\mathbf{s}^{(u)}_x;\mathbf{r}_{xR_s}\right)$ and $F\left(\mathbf{s}^{(u)}_x,\mathbf{s}^{(v)}_x\right)$, but the number of terms in (13)–(14) increases proportionally to δ and the size of the look-up tables LUT-1 and LUT-2.ℓ, for $\ell=1,2,\ldots,U-1$, also increases proportionally to δ and δ^2, respectively. A particular issue of the state-of-the-art detection algorithm from [5] is the large amount of memory that is required to store the look-up tables, in particular the look-up tables LUT-2.ℓ, for $\ell=1,2,\ldots,U-1$. The latter tables each have $2\left(R_{ss}PM^{L+1}\right)^2$ real-valued entries and serve only to evaluate the interference function values $F\left(\mathbf{s}^{(u)}_x,\mathbf{s}^{(v)}_x\right)$.

Algorithm 1: Initialization and Message Passing Scheduling Strategy

Initialization. The iteration counter i is reset to 0. For all u and n, the messages $\left\{\hat{P}\left(a^{(u)}_n\right)\right\}$ and $\left\{\hat{P}_e\left(\mathbf{s}^{(u)}_{N^{(u)}_\tau+n}|\mathbf{r}\right)\right\}$ are initialized to 1.

Iterative procedure. For $i=1,2,\ldots i_{\max}$:

Increase iteration counter: $i=i+1$.

Multiuser processing. For all u and n, compute the messages $\left\{\Xi\left(\mathbf{s}^{(u)}_{N^{(u)}_\tau+n}\right)\right\}$.

Per user processing. For $u=1,2,\ldots,U$:

- For all n, compute the messages $\left\{f\left(\mathbf{S}^{(u)}_n\right)\right\}$ and $\left\{b\left(\mathbf{S}^{(u)}_n\right)\right\}$.
- For all n and k, compute the messages $\left\{\alpha\left(a^{(u)}_n\right)\right\}$ and $\left\{\hat{P}\left(b^{(u)}_k|\mathbf{r}\right)\right\}$.
- For all n and l, compute $\left\{\hat{P}\left(a^{(u)}_n\right)\right\}$ and $\left\{\hat{P}_e\left(\mathbf{s}^{(u)}_{N^{(u)}_\tau+n}|\mathbf{r}\right)\right\}$.

Use the estimates $\left\{\hat{P}\left(b^{(u)}_k|\mathbf{r}\right)\right\}$ of $\left\{p\left(b^{(u)}_k|\mathbf{r}\right)\right\}$ to compute $\left\{\hat{b}^{(u)}_k\right\}$ using **(12)**. If all detected bits are error-free: end iterations.

Evaluating $\overset{(v)}{\phi}{}^{\rightarrow}\left(\mathbf{s}^{(u)}_x\right)$ from (15) for all u, for all $v\neq u$, for all x and for all possible values of $\mathbf{s}^{(u)}_x$ requires $4\left(PM^{L+1}\right)^2(U-1)$ elementary operations on two real arguments, per user, per CPM symbol and per receiver iteration. Obtaining the messages $\Xi\left(\mathbf{s}^{(u)}_x\right)$ from (16) involves another $PM^{L+1}(U-1)$ real multiplications, per user, per CPM symbol and per receiver iteration. Besides the values of the messages $\Xi\left(\mathbf{s}^{(u)}_x\right)$, all PM^{L+1} values of each of the $U(U-1)$ helper messages $\overset{(v)}{\phi}{}^{\rightarrow}\left(\mathbf{s}^{(u)}_x\right)$, for $u=1,2,\ldots,U$ and $v=1,2,\ldots,U$ with $v\neq u$, also need to be stored.

Furthermore, the initialization procedure in Algorithm 1 needs to be extended in the sense that the messages $\Xi\left(\mathbf{s}_x^{(u)}\right)$ and $\phi^{(u)}_{\rightarrow}\left(\mathbf{s}_x^{(v)}\right)$ need to be initialized before the start of the iterative process.

In [5], the authors propose to simplify the detector by neglecting the interference among non-adjacent users u and v with $|v-u| > U_I$. This approximation is accurate only for values of the NSP that are sufficiently large. It allows to eliminate the function nodes $E_x^{(u,v)}$ with $v > u + U_I$ in the FG from Fig. 2. Applying the SP algorithm to the resulting FG yields a simplified detector. This simplified detector basically follows the strategy outlined in Algorithm 1; the per user processing step remains unmodified, but the computational complexity and the memory requirements of the multiuser processing step are significantly reduced as compared to the detector that takes into account the interference from all the users. The product in (16) is restricted to the $2U_I$ users v that are most adjacent to user u. The number of helper messages $\phi^{(v)}_{\rightarrow}\left(\mathbf{s}_x^{(u)}\right)$ is reduced by a factor of $\frac{(U-1)}{(2U_I)}$. All this results in a computational complexity of only $8U_I\left(PM^{L+1}\right)^2 + 2PM^{L+1}$ elementary operations on two real arguments, per user, per CPM symbol and per receiver iteration, while less than $2U_I$ helper messages and only U_I pre-computed look-up tables LUT-2.ℓ, $\ell = 1, 2, \ldots, U_I$, need to be stored. The most radical simplification results from assuming $U_I = 1$. In this specific case, the messages $\phi^{(u-1)}_{\rightarrow}\left(\mathbf{s}_x^{(u)}\right)$ and $\phi^{(u+1)}_{\rightarrow}\left(\mathbf{s}_x^{(u)}\right)$ can alternatively be computed in a recursive way, without increasing the computational complexity. Starting from $\phi^{(0)}_{\rightarrow}\left(\mathbf{s}_x^{(1)}\right) = 0$, we obtain

$$\phi^{(u-1)}_{\rightarrow}\left(\mathbf{s}_x^{(u)}\right) = \sum_{\mathbf{s}_x^{(u-1)}} \hat{P}_e\left(\mathbf{s}_x^{(u)}\,|\mathbf{r}\right) F\left(\mathbf{s}_x^{(u)}, \mathbf{s}_x^{(u-1)}\right) \phi^{(u-2)}_{\rightarrow}\left(\mathbf{s}_x^{(u-1)}\right), \quad (17)$$

for $u = 1, 2, \ldots, U$, and from the boundary condition $\phi^{(U+1)}_{\rightarrow}\left(\mathbf{s}_x^{(U)}\right) = 0$, we can in a similar way compute

$$\phi^{(u+1)}_{\rightarrow}\left(\mathbf{s}_x^{(u)}\right) = \sum_{\mathbf{s}_x^{(u+1)}} \hat{P}_e\left(\mathbf{s}_x^{(u+1)}\,|\mathbf{r}\right) F\left(\mathbf{s}_x^{(u+1)}, \mathbf{s}_x^{(u)}\right) \phi^{(u+2)}_{\rightarrow}\left(\mathbf{s}_x^{(u+1)}\right), \quad (18)$$

for $u = U, U-1, \ldots, 1$. This approach has the advantage that the messages $\phi^{(u)}_{\rightarrow}\left(\mathbf{s}_x^{(u+1)}\right)$ and $\phi^{(u)}_{\rightarrow}\left(\mathbf{s}_x^{(u-1)}\right)$ do not rely on message values obtained during previous iterations. The corresponding detector will be referred to as g10-MU-FG-2IU because it is based on the assumption of only 2 interfering users (IU) and because the corresponding MU FG turns out to have girth 10 (instead of 6). In spite of the substantial simplifications obtained by considering only two (instead of $(U-1)$) interfering users, the number of elementary operations on two real arguments and the amount of memory that is required by g10-MU-FG-2IU for the evaluation of the messages $\Xi\left(\mathbf{s}_x^{(u)}\right)$ is still rather high.

Remarks:

- In the absence of BICM, the constraint between the data symbols and the information bits of the uth user decomposes into $P\left(\mathbf{a}^{(u)}\,\middle|\mathbf{b}^{(u)}\right) = \prod_n P\left(a_n^{(u)}\,\middle|\mathbf{b}_n^{(u)}\right)$, with $\mathbf{b}_n^{(u)} = \left(b_{Mn}^{(u)}, b_{Mn+1}^{(u)}, \ldots, b_{Mn+M-1}^{(u)}\right)$. In the absence of IUI, the factors $F\left(\mathbf{s}_x^{(u)}, \mathbf{s}_x^{(v)}\right)$ from (13) disappear, such that the nodes D_x in Fig. 1 decompose into a set of U non-connected nodes $V_x^{(u)}$ representing the factors $L\left(\mathbf{s}_x^{(u)}; \mathbf{r}_{xR_s}\right)$ from (14). At the same time, the factors $L\left(\mathbf{s}_x^{(u)}; \mathbf{r}_{xR_s}\right)$ reduce (within a normalization constant not depending on $\mathbf{s}_x^{(u)}$) to the likelihood function of $\mathbf{s}_x^{(u)}$: $L\left(\mathbf{s}_x^{(u)}; \mathbf{r}_{xR_s}\right) \propto p\left(\mathbf{r}_{xR_s}\,\middle|\mathbf{s}_x^{(u)}\right)$, with $p\left(\mathbf{r}_{xR_s}\,\middle|\mathbf{s}_x^{(u)}\right)$ the APP of $\mathbf{r}_{xR_s}$, given $\mathbf{s}_x^{(u)}$. Applying the SP algorithm to the resulting cycle-free FG takes the form of a forward/backward algorithm, also known as the BCJR algorithm [7]. This algorithm returns, in the form of the messages traveling on the edge $\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}$ from node $B_n^{(u)}$ to node $D_{N_\tau^{(u)}+n}$, and within a normalization constant not depending on $\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}$, the exact *extrinsic* signal waveform probabilities, i.e.

$$P_e\left(\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}\,|\mathbf{r}\right) \propto \frac{P\left(\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}\,|\mathbf{r}\right)}{L\left(\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}; \mathbf{r}_{\left(N_\tau^{(u)}+n\right)R_s}\right)}$$

with $P\left(\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}\,|\mathbf{r}\right)$ the exact APP of $\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}$, given $\mathbf{r}$. This motivates the notations employed for the messages in Fig. 4.

- In [5], the factors $L\left(\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}; \mathbf{r}_{\left(N_\tau^{(u)}+n\right)R_s}\right)$ are associated to the FG nodes $B_n^{(u)}$ instead of to the nodes $V^{(u)}_{N_\tau^{(u)}+n}$. This leaves the detection algorithm from [5] unmodified. However, in the absence of IUI, the edges associated with the waveform $\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}$ become half-edges, such that the BCJR algorithm yields the exact *a posteriori* probabilities $P\left(\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}\,|\mathbf{r}\right)$. Combined with the fact that the waveform $\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}$ is uniquivocally determined by $\left(\boldsymbol{\sigma}_{n-1}^{(u)}, a_{n-1}^{(u)}, a_n^{(u)}\right)$, this explains the notation $\hat{P}\left(\boldsymbol{\sigma}_{n-1}^{(u)}, a_{n-1}^{(u)}, a_n^{(u)}\,|\mathbf{r}\right)$ in [5] (instead of $\hat{P}_e\left(\mathbf{s}^{(u)}_{N_\tau^{(u)}+n}\,|\mathbf{r}\right)$) for the messages from node $B_n^{(u)}$ to node $V^{(u)}_{N_\tau^{(u)}+n}$.

D. Novel Receivers

1) g6-MU-FG-GA: An alternative approach to simplify the detector is to approximate an appropriate set of messages in the SP algorithm by a canonical distribution. We will apply this approach to the FG from Fig. 1, which will lead to a novel receiver structure.

Based on the discussion in [8], we propose to approximate the messages $\hat{P}_e\left(\mathbf{s}_x^{(u)}|\mathbf{r}\right)$, for $u = 1,2,\ldots,U$ and $x = N_\tau^{(u)}, N_\tau^{(u)}+1,\ldots,N_\tau^{(u)}+N$, in Fig. 4 by the product of R_s univariate Gaussian distribution functions of complex-valued circularly symmetric random variables with means $\left\{\mu_{xR_s+k}^{(u)}\right\}$ and variances $\left\{\upsilon_{xR_s+k}^{(u)}\right\}$, $k = 0,1,\ldots,R_s-1$:

$$\hat{P}_e\left(\mathbf{s}_x^{(u)}|\mathbf{r}\right) \approx \exp\left\{-\sum_{k=0}^{R_s-1}\frac{\left|s_{xR_s+k}^{(u)}-\mu_{xR_s+k}^{(u)}\right|^2}{\upsilon_{xR_s+k}^{(u)}}\right\}, \quad (19)$$

with

$$\mu_{xR_s+k}^{(u)} = \sum_{\mathbf{s}_x^{(u)}} s_{xR_s+k}^{(u)}\hat{P}_e\left(\mathbf{s}_x^{(u)}|\mathbf{r}\right), \quad (20)$$

for $k = 0,1,\ldots,R_s-1$ and

$$\upsilon_l^{(u)} = \begin{cases} 1-\left|\mu_l^{(u)}\right|^2 & ,\tau^{(u)}R_s \le l < \left(\tau^{(u)}+N\right)R_s \\ 0 & ,\text{otherwise}\end{cases}. \quad (21)$$

The approximation (19) significantly simplifies the multiuser processing in Algorithm 1. When the cluster nodes D_x in Fig. 1 are not further decomposed, they actually implement the function $F(\mathbf{s}_x)$, with $\mathbf{s}_x = \left(\mathbf{s}_x^{(1)},\mathbf{s}_x^{(2)},\ldots,\mathbf{s}_x^{(U)}\right)$ and

$$F(\mathbf{s}_x) = \exp\left\{-\frac{E_s}{N_0R_s}\sum_{k=0}^{R_s-1}\left|r_{xR_s+k}-\sum_{u=1}^{U}s_{xR_s+k}^{(u)}\right|^2\right\}.$$

In the FG that results from combining Figs. 1 and 3 (without Fig. 2) the cycles with minimum length are still of length 6, i.e., between the nodes $B_{n-1}^{(u)}$, $B_n^{(u)}$, $D_{N_\tau^{(u)}+n}$, $B_{N_\tau^{(u)}-N_\tau^{(v)}+n}^{(v)}$, $B_{N_\tau^{(u)}-N_\tau^{(v)}+n-1}^{(v)}$ and $D_{N_\tau^{(u)}+n-1}$, for $u \ne v$. Exploiting the property that the sum of independent Gaussian variables (signal samples from different users $v \ne u$) is Gaussian with mean equal to the sum of the means and variance equal to the sum of the variances, the SP messages $\Xi\left(\mathbf{s}_x^{(u)}\right)$ are easily obtained in closed-form. Applying the SP rules from [4] to the FG of Fig. 1, we find:

$$\Xi\left(\mathbf{s}_x^{(u)}\right) \propto \exp\left\{2\Re\left\{\left(\mathbf{y}_x^{(u)}\right)^H\mathbf{s}_x^{(u)}\right\}\right\}, \quad (22)$$

where $\mathbf{y}_x^{(u)}$ is a size-R_s column vector with components $y_q^{(u)}$, for $q = x, x+1,\ldots,x+R_s-1$ given by:

$$y_q^{(u)} = \frac{r_q-\left(\mu_{MU,q}-\mu_q^{(u)}\right)}{\frac{N_0}{E_s}R_s+\left(\upsilon_{MU,q}-\upsilon_q^{(u)}\right)}, \quad (23)$$

where $\mu_{MU,q} = \sum_{u=1}^{U}\mu_q^{(u)}$ and $\upsilon_{MU,q} = \sum_{u=1}^{U}\upsilon_q^{(u)}$. Expression (23) indicates that a soft interference estimate $\left(\mu_{MU,q}-\mu_q^{(u)}\right)$ is subtracted from the observation r_q, and the estimation error variance $\left(\upsilon_{MU,q}-\upsilon_q^{(u)}\right)$ is added to the noise variance $\frac{N_0R_s}{E_s}$. The particular structure of (22) indicates that the proposed multiuser detector can be decomposed into an equivalent set of U iterative single user detectors, with a separate module for IUI parameter estimation (means and variances). The single user detectors are operated in parallel, with the uth detector accepting the samples $\left\{y_q^{(u)}\right\}$ from (23) as equivalent observations. The per user processing step in Algorithm 1 is the same as for the g10-MU-FG-2IU detector. At each iteration, the messages $\left\{\hat{P}_e\left(\mathbf{s}_x^{(u)}|\mathbf{r}\right)\right\}$ computed at the end of the per user processing step are subsequently used to update the IUI parameters $\left\{\mu_q^{(u)}\right\}$ and $\left\{\upsilon_q^{(u)}\right\}$, which in turn are used to update the equivalent observation samples $\left\{y_q^{(u)}\right\}$. Finally, the messages $\Xi\left(\mathbf{s}_x^{(u)}\right)$ are evaluated according to (22). In (20) and (22), we need $\mathbf{s}_x^{(u)}$ for all u, for all relevant x and for all possible realizations of $\mathbf{s}_x^{(u)}$. Rounding $\tau^{(u)}T$ to the nearest integer multiple of $T_{ss} = \frac{T}{R_{ss}}$, with R_{ss} integer valued, an approximation of $\mathbf{s}_x^{(u)}$ can efficiently be obtained from (1)–(3), (10) and the look-up table LUT-1 that contains the $R_{ss}PM^L$ precomputed (real-valued) values of $\Psi\left(lT_{ss};\mathbf{S}_n^{(u)}\right)$ for $l = 0,1,\ldots,R_{ss}-1$ and for all PM^L possible realizations of $\mathbf{S}_n^{(u)}$. This needs to be done only once, at the start of the iterations. The number of elementary operations that is required to compute $\Xi\left(\mathbf{s}_x^{(u)}\right)$ from (20)–(23) amounts to $7R_sPM^{L+1}+2PM^{L+1}+13R_s$, per user, per CPM symbol and per receiver iteration. The above receiver algorithm will be referred to as g6-MU-FG-GA, because it is derived from a MU FG with girth 6 (g6) and relies on a Gaussian approximation (GA).

2) g3-MU-FG-GA: The variables with the largest alphabets in the FG of Figs. 1–3 are $\{\mathbf{s}_{N_\tau^{(u)}+n}^{(u)} : n = 0,1,\ldots,N\}$ and $\{\mathbf{S}_n^{(u)} : n = 0,1,\ldots,N-1\}$. An equivalent FG for deriving the bit APPs $p\left(b_k^{(u)}|\mathbf{r}\right)$ in (12) results from clustering in Fig. 1 the function nodes D_x, for $x = 0,1,\ldots,N_s-1$. The main motivation for this clustering operation is that, after some basic manipulations, we obtain the FG from Figs. 5–6.

This FG contains $\{\tilde{\mathbf{s}}_n^{(u)} : x = 0,1,\ldots,N-1\}$ and $\{\boldsymbol{\sigma}_n^{(u)}, n = 1,2,\ldots,N-1\}$ as additional variables, rather than $\{\mathbf{s}_x^{(u)} : x = 0,1,\ldots,N_s-1\}$ and $\{\{\mathbf{S}_n^{(u)} : n = 0,1,\ldots,N-1\}$. The vector variables $\tilde{\mathbf{s}}_n^{(u)}$ results from stacking the R_s samples $s^{(u)}\left(\left(K_\tau^{(u)}+i\right)T_s+nT\right)$, $i = 0,1,\ldots,R_s-1$, that correspond to the nth symbol interval $[nT,(n+1)T[$ of the transmitter signal $s_{SU}^{(u)}(t)$(1) from the uth user. Here, the quantity $K_\tau^{(u)}$ denotes the smallest integer larger than $\tau^{(u)}R_s$. The difference between the variables $\mathbf{s}_x^{(u)}$ in Fig. 1 and the variables $\tilde{\mathbf{s}}_n^{(u)}$ in Fig. 5 is illustrated in Fig. 7, assuming $R_s = 8$. The time indications in the figure are relative to the symbol interval T. The figure shows the location $\left[\tau^{(u)},\tau^{(u)}+N\right[$ of the transmitter signal from user u, with a duration of N symbol intervals, within the observation interval $[0,N_sT_s[$ at the receiver, with a duration of $\frac{N_s}{R_s}$ symbol intervals. The waveform corresponding to the nth symbol interval of the transmitter signal from user u is fully determined by the state transition $\mathbf{S}_n^{(u)}$. The normalized (with respect to the symbol interval T) sampling time instants, $\frac{k}{R_s}$,

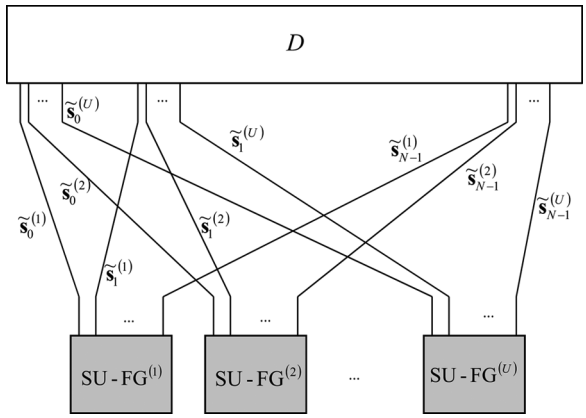

Fig. 5. FG representing an alternative factorization of the joint APP $p(\mathbf{b}\,|\mathbf{r}\,)$ of the bit sequences of all users from a same group, obtained by clustering the set of function nodes D_x in the FG of Fig. 1 and regrouping the variable edges.

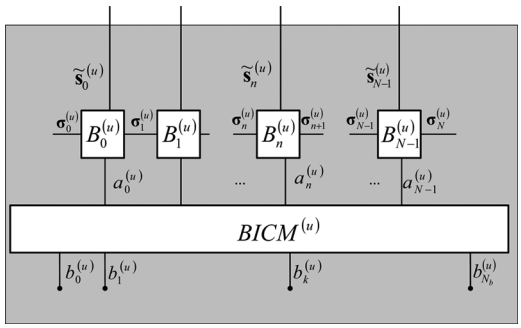

Fig. 6. Further decomposition of the nodes $\mathrm{SU-FG}^{(u)}$, $u = 1, 2, \ldots, U$ in the FG of Fig. 5.

are indicated, the meaning of the quantities $N_\tau^{(u)}$ and $K_\tau^{(u)}$ is illustrated and the grouping of samples into vectors of the type $\mathbf{s}_x^{(u)}$ and $\tilde{\mathbf{s}}_n^{(u)}$ is explicitated. Because $\tilde{\mathbf{s}}_n^{(u)}$ is fully determined by $\mathbf{S}_n^{(u)} = \left(\boldsymbol{\sigma}_n^{(u)}, a_n^{(u)}\right)$, whereas $\mathbf{s}_{N_\tau^{(u)}+n}^{(u)}$ is fully determined by $\left(\mathbf{S}_{n-1}^{(u)}, a_n^{(u)}\right)$, the cardinality of the largest variable alphabets in Figs. 5–6 (cardinality PM^L and PM^{L-1} for alphabet of $\tilde{\mathbf{s}}_n^{(u)}$ and $\boldsymbol{\sigma}_n^{(u)}$, respectively, with $u = 1, 2, \ldots, U$ and $n = 1, 2, \ldots, N-1$) is M times smaller than the cardinality of the largest variable alphabets in Figs. 1–3 (cardinality PM^{L+1} and PM^L for alphabet of $\mathbf{s}_{N_\tau^{(u)}+n}^{(u)}$ and $\mathbf{S}_n^{(u)}$, respectively, with $u = 1, 2, \ldots, U$ and $n = 1, 2, \ldots, N-1$). This potentially results in a lower computational complexity for the SP algorithm.

The cluster node D in Fig. 5 implements the function $\tilde{F}\,(\tilde{\mathbf{s}})$, with

$$\tilde{F}\,(\tilde{\mathbf{s}}) = \exp \times \left\{ -\frac{E_s}{N_0 R_s} \sum_{k=0}^{N_s-1} \left| r_k - \sum_{u:\, \tau^{(u)} \in \left]\frac{k}{R_s}-N, \frac{k}{R_s}\right]} s_k^{(u)} \right|^2 \right\},$$

where $\tilde{\mathbf{s}}_n = \left(\tilde{\mathbf{s}}_n^{(1)}, \tilde{\mathbf{s}}_n^{(2)}, \ldots, \tilde{\mathbf{s}}_n^{(U)}\right)$ and $\tilde{\mathbf{s}} = (\tilde{\mathbf{s}}_0, \tilde{\mathbf{s}}_1, \ldots, \tilde{\mathbf{s}}_{N-1})$. The node $BICM^{(u)}$ in Fig. 6 that imposes the constraints from the BICM unit of the uth user to the information bits and the data symbols of that user is the same as in Fig. 3. The nodes $B_0^{(u)}$, $B_{N-1}^{(u)}$ and $B_n^{(u)}$, $n = 1, 2, \ldots, N-2$, in Fig. 6 represent the following functions: $P\left(\boldsymbol{\sigma}_0^{(u)}\right)\tilde{I}_{tr,0}^{(u)}[.]$, $P\left(\boldsymbol{\sigma}_N^{(u)}\right)\tilde{I}_{tr,N}^{(u)}[.]$ and $\tilde{I}_{tr,n}^{(u)}[.]$, $n = 1, 2, \ldots, N-2$, with $P\left(\boldsymbol{\sigma}_0^{(u)}\right)$ the *a priori* probability of $\boldsymbol{\sigma}_0^{(u)}$, $P\left(\boldsymbol{\sigma}_N^{(u)}\right)$ the *a priori* probability of $\boldsymbol{\sigma}_N^{(u)}$, and $\tilde{I}_{tr,n}^{(u)}[.]$ equal to one when the variables corresponding to edges connected to node $B_n^{(u)}$ satisfy the trellis (1)–(8) and (10), and zero otherwise. In the FG of Figs. 5–6, cycles with minimum length 3 (instead of 6) are found between the nodes $B_{n-1}^{(u)}$, $B_n^{(u)}$ and D. Such very short cycles can be disastrous for the accuracy of the SP algorithm. Provided that in Fig. 4, in Algorithm 1 and in (25)–(28) the variables ($\mathbf{S}_{n-1}^{(u)}$, $\mathbf{S}_n^{(u)}$, $\mathbf{s}_{N_\tau^{(u)}+n}^{(u)}$, $a_{n-1}^{(u)}$) are replaced with ($\boldsymbol{\sigma}_n^{(u)}$, $\boldsymbol{\sigma}_{n+1}^{(u)}$, $\tilde{\mathbf{s}}_n^{(u)}$, $a_n^{(u)}$), the notation for the messages, the initialization and message passing scheduling strategy and the equations describing the per user processing remain valid. As in Section III-D-I, we approximate the messages $\hat{P}_e\,(.\,|\mathbf{r}\,)$ in Fig. 4 by the product of R_s univariate Gaussian distribution functions in the elements of $\tilde{\mathbf{s}}_n^{(u)}$, in order to reduce the complexity of the multiuser processing of the resulting detector. This derivation leads to similar equations as in Section III-D-I. It suffices to replace in (19) and (20) the message $\hat{P}_e\left(\mathbf{s}_x^{(u)}\,|\mathbf{r}\right)$ by $\hat{P}_e\left(\tilde{\mathbf{s}}_n^{(u)}\,|\mathbf{r}\right)$ and the subscript $xR_s + k$ by $K_\tau^{(u)} + nR_s + k$ and in (22) the message $\Xi\left(\mathbf{s}_n^{(u)}\right)$ by $\Xi\left(\tilde{\mathbf{s}}_n^{(u)}\right)$, the variable $\mathbf{s}_x^{(u)}$ by $\tilde{\mathbf{s}}_n^{(u)}$ and the quantity $\mathbf{y}_x^{(u)}$ by $\mathbf{y}_{K_\tau^{(u)}+nR_s}^{(u)}$. The resulting receiver will be referred to as g3-MU-FG-GA.

E. Relation to the AD-HOC Detector With Soft IUI Cancellation From the Literature

Ad-hoc variants of the g3-MU-FG-GA detector are reported in [5] and [6]. They operate essentially in the same manner, but use (approximate) signal waveform APPs $\hat{P}\left(\tilde{\mathbf{s}}_n^{(u)}\,|\mathbf{r}\right)$ rather than the extrinsics $\hat{P}_e\left(\tilde{\mathbf{s}}_n^{(u)}\,|\mathbf{r}\right)$ to perform IUI parameter estimation. In [5], [6], the required signal waveform APPs are computed by applying the SP algorithm to a set of U FGs that each represent a suitable factorization of the joint APP $p\left(\mathbf{b}^{(u)}\,\middle|\mathbf{r}; \left\{\left(\mu_q^{(v)}, \upsilon_q^{(v)}\right), \forall v \neq u\right\}\right)$, where $\mu_q^{(v)}$ and $\upsilon_q^{(v)}$, for $v \neq u$, are to be considered as parameters (rather than arguments) of the joint probability. The APP $p\left(\mathbf{b}^{(u)}\,\middle|\mathbf{r}; \left\{\left(\mu_q^{(v)}, \upsilon_q^{(u)}\right), \forall v \neq u\right\}\right)$ corresponds to a single user context where only the bit sequence of the uth user is detected and the IUI interference is treated as additional white Gaussian noise with mean $\sum_{v\neq u}\mu_q^{(v)}$ and variance $\sum_{v\neq u}\upsilon_q^{(v)}$. In practice, this boils down to replacing in (20) the messages $\hat{P}_e\left(\tilde{\mathbf{s}}_n^{(u)}\,|\mathbf{r}\right)$ by the product $\hat{P}_e\left(\tilde{\mathbf{s}}_n^{(u)}\,|\mathbf{r}\right) L\left(\tilde{\mathbf{s}}_n^{(u)}; \mathbf{r}_{K_\tau^{(u)}+nR_s}\right)$. The corresponding receiver will be referred to as SU-FG-GA.

IV. Complexity Considerations

The memory requirements and the computational complexity of the novel detection algorithms g3-MU-FG-GA and g6-MU-FG-GA are now addressed. The results are compared to those of the simplified detector g10-MU-FG-2IU from [5]. The SU-FG-GA detection algorithm is not explicitly considered. Its memory requirements and computational load are about the same as for the g3-MU-FG-GA detector.

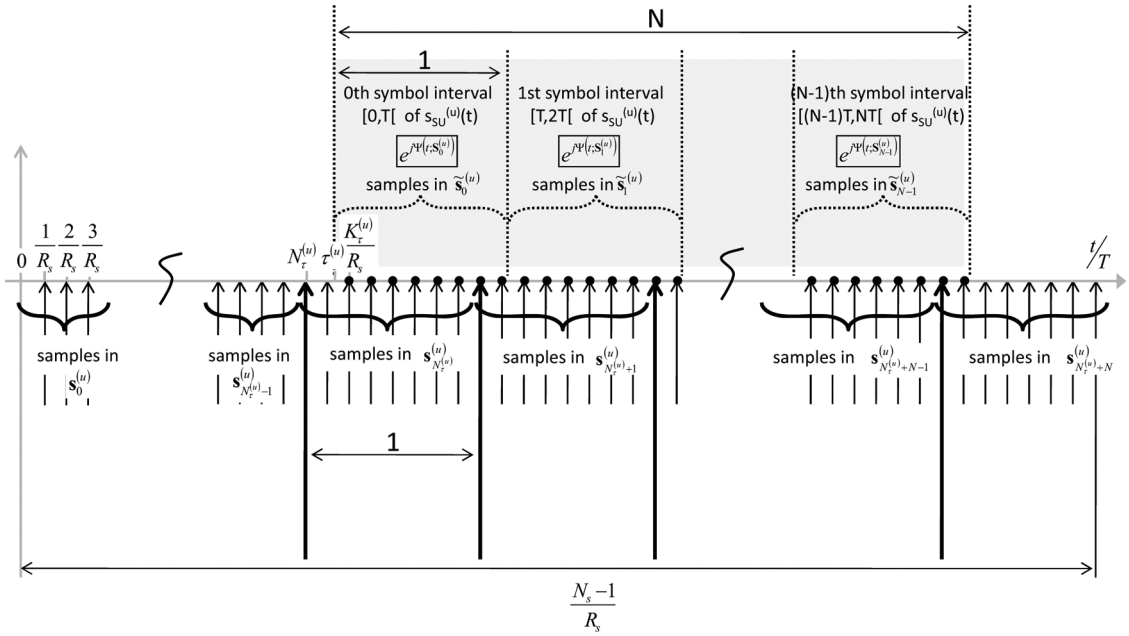

Fig. 7. Construction of the signal sample vectors.

TABLE I
DETAILS OF THE CONSIDERED SPECTRALLY EFFICIENT BIC-CPM SCHEMES.

#	$\dot{q}(t)$	L	h	M	code	NSP	$\eta_{spec,\infty}$
1	REC	2	1/3	2	(64,51) eBCH	0.3	2.6
2	RC	2	1/3	4	(128,115) eBCH	0.8	2.2

The g3-MU-FG-GA detector has a larger complexity per user u and per detector iteration than a conventional single user detector. This increase in complexity results from computing, at each iteration, the messages $\left\{\hat{P}_e\left(\tilde{\mathbf{s}}_n^{(u)}|\mathbf{r}\right)\right\}$, the IUI parameters $\left\{\mu_q^{(u)}\right\}$ and $\left\{v_q^{(u)}\right\}$, the equivalent observation samples $\left\{y_q^{(u)}\right\}$ and the messages $\left\{\Xi\left(\tilde{\mathbf{s}}_n^{(u)}\right)\right\}$. The g6-MU-FG-GA detector has the same structure as the g3-MU-FG-GA detector, but it is slightly more complex. This is mainly due to the fact that the cardinality of the variable alphabets is larger in the FG with girth 6 of (Figs. 1, 3) than in the FG with girth 3 of (Figs. 5, 6) (see Section III). As a result, the messages $f\left(\mathbf{S}_n^{(u)}\right)$, $b\left(\mathbf{S}_n^{(u)}\right)$, $\hat{P}_e\left(\mathbf{s}_x^{(u)}|\mathbf{r}\right)$ and $\Xi\left(\mathbf{s}_x^{(u)}\right)$ computed in the g6-MU-FG-GA detector take M times more values than the messages $f\left(\boldsymbol{\sigma}_n^{(u)}\right)$, $b\left(\boldsymbol{\sigma}_n^{(u)}\right)$, $\hat{P}_e\left(\tilde{\mathbf{s}}_n^{(u)}|\mathbf{r}\right)$ and $\Xi\left(\tilde{\mathbf{s}}_n^{(u)}\right)$ computed in the g3-MU-FG-GA detector. All these values have to be re-computed and stored at each iteration. Furthermore, the number of operations that is required to update the messages $\alpha(a_n)$ according to (27), see Appendix A, and the number of additions that is needed to evaluate the means $\mu_q^{(u)}$ from (20) is also M times larger for g6-MU-FG-GA than for g3-MU-FG-GA. The structure of the g10-MU-FG-2IU detector differs substantially from that of the proposed detectors g3-MU-FG-GA and g6-MU-FG-GA. The messages $f\left(\mathbf{S}_n^{(u)}\right)$, $b\left(\mathbf{S}_n^{(u)}\right)$, $\alpha(a_n)$ and $\hat{P}_e\left(\mathbf{s}_x^{(u)}|\mathbf{r}\right)$ are computed in the same way as for g6-MU-FG-GA. The messages $\Xi\left(\mathbf{s}_x^{(u)}\right)$ have to be computed according to (15)–(16). The size of the pre-computed LUT-2.1 tabel required for, and the number of operations involved in, the computation of $\Xi\left(\mathbf{s}_x^{(u)}\right)$ is large and contributes significantly to the overall memory requirement and the overall complexity of the g10-MU-FG-2IU detector, respectively. Besides, $2(U-1)PM^{L+1}$ helper message values $\phi^{(v)}\left(\mathbf{s}_x^{(u)}\right)$ need to be additionally stored.

In the following, we only take into account the contribution to the required memory and the number of operations that results from the processing steps in which the g3-MU-FG-GA, g6-MU-FG-GA and g10-MU-FG-2IU algorithms differ from each other. These contributions provide a solid basis for comparing the considered algorithms because they dominate the detector's total memory requirements and total computational complexity, respectively. General closed-form expressions are provided, as well as numerical results for 4 specific simulation set-ups. The 4 set-ups involve 2 different spectrally efficient BIC-CPM schemes, whose parameters are summarized in Table I. Rectangular (REC) and raised-cosine (RC) frequency pulses $\dot{q}(t)$ with a duration of $L=2$ symbol intervals are used:

$$\dot{q}(t)=\begin{cases}\frac{1}{2LT} & ,\text{REC}\\ \frac{1}{2LT}\left(1-\cos\left(\frac{2\pi t}{LT}\right)\right) & ,\text{RC}\end{cases},\ t\in[0,LT]. \quad (24)$$

The outer code is a (64,51) or a (128,115) extended BCH code. Both schemes are amongst the schemes with the highest asymptotic (for $U\to\infty$) spectral efficiency reported in [9]. For each of these two BIC-CPM schemes, practical systems with groups of $U=5$ and $U=17$ simultaneous users are considered.

Table II shows the number of real values to be stored at the receiver. The first three columns show the memory that needs to be allocated dynamically by each of the three considered algorithms to store the required pre-computed function values and the most recent values of the SP messages, assuming the transmission of 1024 information bits. The last column

TABLE II
NUMBER OF REAL VALUES TO BE STORED, ASSUMING $R_{ss} = \lceil 3 + (U-1) \cdot NSP \rceil$ AND THE TRANSMISSION OF 1024 INFORMATION BITS.

	g3-MU-FG-GA	g6-MU-FG-GA	g10-MU-FG-2IU	
generic	$2NUPM^{L-1}(1+M)$	$2NUPM^{L}(1+M)$	$2NUPM^{L}(1+2M)$	$2\left(R_{ss}PM^{(L+1)}\right)^2$
#1, U=5	233 460	466 920	778 200	57 600
#1, U=17	793 764	1 587 528	2 645 880	73 728
#2, U=5	352 200	1 408 800	2 535 840	3 612 671
#2, U=17	1 197 480	4 789 920	8 621 856	18 874 368

TABLE III
NUMBER OF ELEMENTARY OPERATIONS BETWEEN TWO REAL ARGUMENTS, PER USER, PER ITERATION AND PER TRANSMITTED SYMBOL, ASSUMING $R_s = \lceil 3 + (U-1) \cdot NSP \rceil$.

	g3-MU-FG-GA	g6-MU-FG-GA	g10-MU-FG-2IU
generic	$13R_s + 7R_sPM^L + 13PM^L$	$13R_s + 7R_sPM^{L+1} + 13PM^{L+1}$	$13PM^{L+1} + 8P^2M^{2(L+1)}$
#1, U=5	641	1217	4920
#1, U=17	932	1760	4920
#2, U=5	3067	11995	297408
#2, U=17	6208	24208	297408

indicates the static memory that is required to store the look-up table LUT-2.1, which is only required for the g10-MU-FG-2IU detector from [5], assuming $R_{ss} = \lceil 3 + (U-1) \cdot NSP \rceil$. The storage of the messages $\alpha(.)$ and of the table LUT-1 is not taken into account; these quantities need to stored for all three algorithms and require a negligibly small amount of memory as compared to the messages ($f(.), b(.), \hat{P}_e(.|\mathbf{r}), \Xi(.)$, \ldots) and to the table LUT-2.1, respectively. We clearly observe that the multiuser receiver g10-MU-FG-2IU from [5] requires a considerably larger amount of memory than the proposed detectors.

Table III considers the number of elementary operations (additions, multiplications, etc\ldots) on two real arguments, per CPM symbol and per receiver iteration, assuming $R_s = \lceil 3 + (U-1) \cdot NSP \rceil$. Operations that are executed only once, at the start of the iterative process are not taken into account. We observe that, as opposed to the complexity of the g10-MU-FG-2IU detector from [5], the complexity of the proposed g3-MU-FG-GA and g6-MU-FG-GA detectors increases (less than proportional) with the number of users in a group. The complexity of the g6-MU-FG-GA detector is about M times as large as that of the g3-MU-FG-GA detector. Both detectors are significantly less complex than the g10-MU-FG-2IU receiver from [5]. A remarkable complexity reduction is observed for the BIC-CPM scheme #2 with $M = 4$. In this case, the g3-MU-FG-GA and g6-MU-FG-GA detectors are at least 10 times less complex (per user, per iteration and per transmitted symbol) than the g10-MU-FG-2IU detector, which becomes almost prohibitively complex. A fair complexity comparison of the different algorithms must take into account the number of iterations that actually needs to be performed by the receiver (in order to meet some given performance specifications). The required number of iterations for the different algorithms will be considered in Section V.

V. NUMERICAL PERFORMANCE RESULTS

Computer simulations have been run to evaluate the performance of the proposed g3-MU-FG-GA and g6-MU-FG-GA detectors. For comparison, the performance of the simple *ad-hoc* SU-FG-GA detector and the performance of the more complex MU graph-based g10-MU-FG-2IU receiver from [5] are evaluated as well. We consider the same 4 simulation set-ups as in the previous section: the two different spectrally efficient BIC-CPM schemes from Table I, each with $U = 5$ and $U = 17$ active users. Interference from adjacent user groups is not generated. Each user asynchronously transmits an information bit vector of size 1024. Gray mapping and pseudorandom bit interleaving are employed. The accuracy of the multiuser communication is assessed in terms of the packet error rate (PER) of the middle user ($u = \frac{(U+1)}{2}$) at a given value of $\frac{E_b}{N_0}$, with $E_b = \frac{NE_s}{N_b}$. At every iteration, hard decisions about the information bits are made from the corresponding *a posteriori* information bit probabilities, after which a genie checks for bit errors; the receiver stops iterating after a maximum number of iterations $i_{\max}$, or when all information bits have been detected correctly. In a practical system, the genie is replaced by a cyclic redundancy check. Several values of $i_{\max}$ (up to $i_{\max} = 30$) are tested. For each value of $i_{\max}$, a large number of simulations (N_{sim}) is performed and the number of packet errors (with 1 packet containing 1024 information bits) encountered by the middle user at the $i_{\max}$ th iteration (N_{err}) is recorded. In each simulation new time delays $\tau^{(u)}T$, $u = 1, 2, \ldots, U$, are taken independently from a random uniform distribution over [0, 20] μs. We consider a fixed carrier spacing of $SP =$ 128 kHz for all scenarios; the corresponding symbol interval T equals $\frac{NSP}{SP}$ where NSP is given in Table I. The performance of the algorithms is investigated for actual continuous-valued time delays, whereas the receiver approximates $\tau^{(u)}$ by the nearest integer multiple of T_s (case $\delta = 1$). The PER is estimated as $\frac{N_{err}}{N_{sim}}$. Figs. 8–11 plot the resulting PER values as a function of $\frac{E_b}{N_0}$ in dB. The data points are based on at least 100 packet error events; except for the most complex detectors at high SNR, where N_{err} is smaller than 100 but at least equal to 10. The values of N_{err} smaller than 100 are specified in the figure captions. Smaller size grey-filled markers, above and below the corresponding data points, indicate the 95% confidence interval of the obtained estimate.

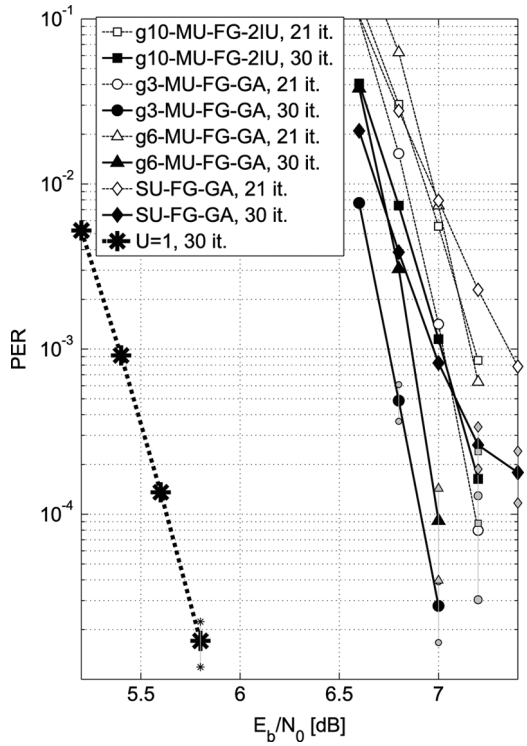

Fig. 8. PER versus $\frac{E_b}{N_0}$ for scheme #1 from Table I and $U = 5$. Data points based on N_{err} error events, with N_{err} smaller than 100: for g10-MU-FG-2IU with 30 iterations $N_{err} = 18$ at $\frac{E_b}{N_0} = 7.2$ dB, for g3-MU-FG-GA with 30 iterations $N_{err} = 24$ at $\frac{E_b}{N_0} = 7$ dB and $N_{err} = 61$ at $\frac{E_b}{N_0} = 6.8$ dB, for g3-MU-FG-GA with 21 iterations $N_{err} = 10$ at $\frac{E_b}{N_0} = 7.2$ dB, for g6-MU-FG-GA with 30 iterations $N_{err} = 12$ at $\frac{E_b}{N_0} = 7$ dB, for SU-FG-GA with 30 iterations $N_{err} = 37$ at $\frac{E_b}{N_0} = 7.2$ dB and $N_{err} = 26$ at $\frac{E_b}{N_0} = 7.4$ dB and for U=1 with 30 iterations $N_{err} = 29$ at $\frac{E_b}{N_0} = 5.8$ dB.

Let us first focus on the BIC-CPM scheme #1 from Table I. The normalized spacing between users is 0.3, yielding a maximum value for $\tau^{(u)}$ equal to $\left(\tau^{(u)}T\right)_{\max}\frac{SP}{NSP} = 8.5333$.

Fig. 8 shows performance results for $U = 5$, obtained with $R_s = 5$. The PER obtained after maximum 21 and 30 iterations of the g3-MU-FG-GA, g6-MU-FG-GA, SU-FG-GA and g10-MU-FG-2UI algorithms is plotted as a function of $\frac{E_b}{N_0}$. It was verified that all considered algorithms are reasonably close to convergence after 30iterations. The performance of the conventional SU detector, executing 30 iterations, for $U = 1$ and $R_s = 3$, is included as an additional reference. The most important observations are listed below:

- The degradation of all considered MU detectors with respect to the single user case (and hence, in a sense due to the remaining IUI), is considerable: about 1.5 dB at a PER of 10^{-4}. This loss in power efficiency is the price to pay for the gain in spectral efficiency that is realized. Assuming a normalized spectral spacing of $\Delta f_g = 1$, and for the considered BIC-CPM scheme, the spectral efficiency (9) increases from 0.8 for $U = 1$ to 1.8 for $U = 5$.
- For the considered range of $\frac{E_b}{N_0}$ and PER values, and executing the same number of iterations for all algorithms, g3-MU-FG-GA outperforms g6-MU-FG-GA and g10-MU-FG-2UI. From all MU FG-based algorithms, surprisingly, the algorithm based on the FG with the *smallest* girth requires the lowest $\frac{E_b}{N_0}$ for achieving a given PER. Conveniently, this algorithm also has the *lowest* computational complexity per iteration, per user and per symbol period. For the considered range of $\frac{E_b}{N_0}$ and PER values, and executing 30 iterations for both algorithms, g6-MU-FG-GA also significantly outperforms g10-MU-FG-2IU.
- At $\frac{E_b}{N_0} = 7$ dB, g3-MU-FG-GA with only 21 iterations yields about the same PER performance as g10-MU-FG-2IU with 30 iterations and SU-FG-GA with 30 iterations, namely, PER $\approx 10^{-3}$. At $\frac{E_b}{N_0} = 7.2$ dB, g3-MU-FG-GA with only 21 iterations achieves a lower PER than SU-FG-GA with 30 iterations and is likely to perform at least as well as g10-MU-FG-2IU with 30 iterations.
- The slope of the PER curves obtained with SU-FG-GA is noticeably smaller than with g6-MU-FG-GA and g3-MU-FG-GA. The SU-FG-GA algorithm is therefore not suited for use at high $\frac{E_b}{N_0}$.

An explanation for g3-MU-FG-GA and g6-MU-FG-GA yielding a better PER performance than g10-MU-FG-2IU can be found in the fact that the 2IU approximation from [5] is not very accurate for values of NSP as small as 0.3. It is less obvious, however, why g3-MU-FG-GA outperforms g6-MU-FG-GA. In the considered simulation set-up, users have a random temporal misalignment at the receiver. The comparison with simulation results (not shown here) obtained for the ideal situation that all users are synchronously received seems to suggest that g3-MU-FG-GA, in contrast to g6-MU-FG-GA, benefits somehow from this randomness in the signal[1]

We recall, from (9), that increasing U yields a larger spectral efficiency η_{spec} of the considered multiuser communication system. A larger U also requires a larger minimum sample rate $\frac{R_s}{T}$ at the receiver, which in turn slightly augments the computational complexity of the g3-MU-FG-GA, g6-MU-FG-GA and SU-FG-GA algorithms (see Table III). Fig. 9 illustrates the PER performance of the considered algorithms for BIC-CPM scheme #1, when the number of users per group is increased to $U = 17$ and the sample rate is $R_s = 8$. We make the following observations:

- In general, a significant performance degradation is observed as compared to the case where $U = 5$. At a PER of about 10^{-3}, this degradation amounts to about 1.25 dB in $\frac{E_b}{N_0}$. The degradation with respect to the SU case, increases from about 1.5 dB, for $U = 5$ (see Fig. 8), to about 2.25 dB, for $U = 17$, at a PER of about 10^{-3}. On the other hand, assuming $\Delta f_g = 1$, an increase in spectral efficiency is realized from 1.8, for $U = 5$, to 2.3, for $U = 17$ (see (9)).
- Again, the proposed g3-MU-FG-GA and g6-MU-FG-GA algorithms outperform the reference g10-MU-FG-2IU algorithm. g10-MU-FG-2IU with 28 iterations obtains a

[1]For scheme #1, 5 users and the considered range of $\frac{E_s}{N_0}$, the PER performance of g3-MU-FG-GA turns out to be slightly better with asynchronous users than with synchronous users; moreover, the PER performance of g3-MU-FG-GA with synchronous users equals that of g6-MU-FG-GA with synchronous users, which in turn is found to virtually equal that of g6-MU-FG-GA with asynchronous users.

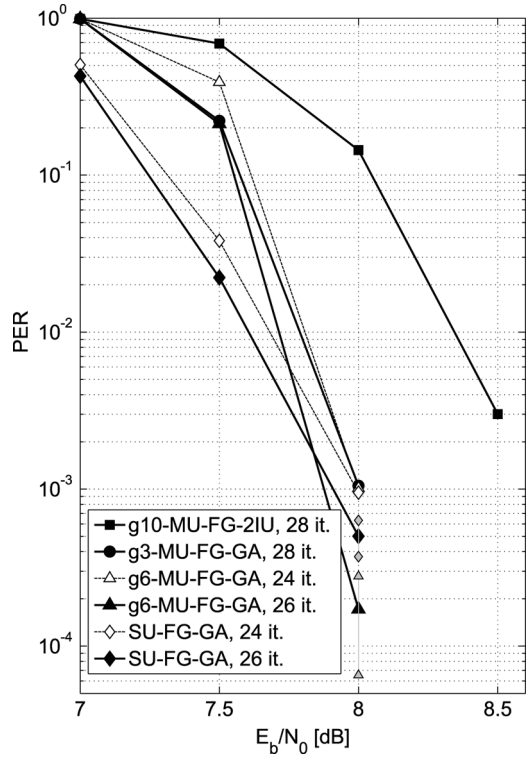

Fig. 9. PER versus $\frac{E_b}{N_0}$ for scheme #1 from Table I and $U = 17$. Data points based on N_{err} error events, with N_{err} smaller than 100: for g6-MU-FG-GA with 26 iterations $N_{err} = 10$ at $\frac{E_b}{N_0} = 8$ dB and for SU-FG-GA with 26 iterations $N_{err} = 46$ at $\frac{E_b}{N_0} = 8$ dB.

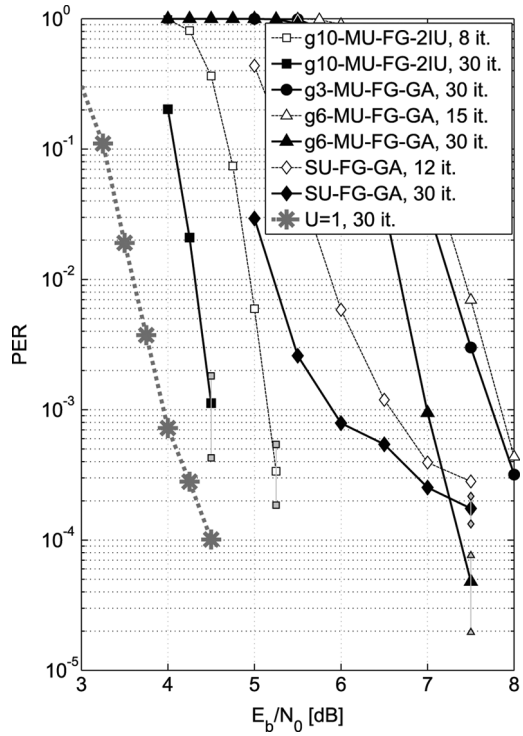

Fig. 10. PER versus $\frac{E_b}{N_0}$ for scheme #2 from Table I and $U = 5$. Data points based on N_{err} error events, with N_{err} smaller than 100: for g10-MU-FG-2IU with 30 iterations $N_{err} = 10$ at $\frac{E_b}{N_0} = 4.5$ dB, for g10-MU-FG-GA with 8 iterations $N_{err} = 16$ at $\frac{E_b}{N_0} = 5.25$ dB, for g3-MU-FG-GA with 30 iterations $N_{err} = 67$ at $\frac{E_b}{N_0} = 7.5$ dB and for g6-MU-FG-GA with 30 iterations $N_{err} = 11$ at $\frac{E_b}{N_0} = 7.5$ dB.

PER of $3 \cdot 10^{-3}$ at $\frac{E_b}{N_0} = 8.5$ dB, whereas SU-FG-GA with 24 iterations, g3-MU-FG-GA with 28 iterations and g6-MU-FG-GA with only 24 iterations achieve about a 1/3 of that PER value of at a 0.5 dB lower value of $\frac{E_b}{N_0}$.

- At $\mathrm{PER} = 10^{-3}$, SU-FG-GA with 24 iterations, g3-MU-FG-GA with 28 iterations and g6-MU-FG-GA with 24 iterations yield about the same performance. It should be noted however that 24 iterations of g6-MU-FG-GA requires more computation time than 28 iterations of g3-MU-FG-GA. This is because g3-MU-FG-GA is only about half as complex per iteration.
- When executing 26 iterations for both algorithms, SU-FG-GA outperforms g6-MU-FG-GA at $\frac{E_b}{N_0} = 8$ dB, whereas the opposite holds for lower $\frac{E_b}{N_0}$ values.

In Figs. 10–11, results are presented for scheme #2 from Table I. The normalized carrier spacing NSP is 0.8, yielding a maximum value for $\tau^{(u)}$ equal to $\left(\tau^{(u)}T\right)_{\max} \frac{SP}{NSP} = 3.2$.

Fig. 10 applies to groups of $U = 5$ users and $R_s = 7$. The ranking of the algorithms in terms of error performance is different from when scheme #1 is used. For scheme #2, g10-MU-FG-2IU outperforms the simpler g3-MU-FG-GA, g6-MU-FG-GA and SU-FG-GA, even after significantly less iterations. The degradation of this detector with respect to the conventional SU detector for $U = 1$ is rather small: about 0.5 dB at a PER of about 10^{-3} and about 1 dB at a PER of about $3 \cdot 10^{-4}$. This power efficiency loss stands against a spectral efficiency gain of about 20% (assuming $\Delta f_g = 1$)). However, for this particular scheme, g10-MU-FG-2IU (even with as little as 8 iterations) is an order of magnitude more complex than any of the other algorithms (with up to 30 iterations). The former algorithm can therefore be excluded for practical use. Out of the three remaining algorithms, the *ad-hoc* SU-FG-GA is to be preferred for PERs not smaller than about $2 \cdot 10^{-4}$. Below $\mathrm{PER} = 2 \cdot 10^{-4}$, g6-MU-FG-GA yields a better performance, provided that a sufficient amount of iterations is performed. Yet, the degradation with respect to the PER obtained with the (overly complex) g10-MU-FG-2IU amounts to several dB. The slope of the PER curve resulting from the *ad-hoc* SU-FG-GA decreases significantly above $\frac{E_b}{N_0} = 6$ dB. For $\frac{E_b}{N_0}$ above 6.5 dB, the PER of SU-FG-GA also hardly decreases between the 12th and the 30th iteration. The simpler g3-MU-FG-GA with 30 iterations achieves about the same performance as g6-MU-FG-GA (which is about 4 times as complex per iteration) with 15 iterations.

Fig. 11 presents similar results for BIC-CPM scheme #2 from Table I and $U = 17$. As for $U = 5$, the impractically complex g10-MU-FG-2IU performs best. For PERs above 10^{-4}, the second best performance is achieved with the simple *ad-hoc* SU-FG-GA. The g3-MU-FG-GA algorithm turns out to be the worst option. This algorithm has a very low convergence speed. The difference between the PER of the g3-MU-FG-GA detector with 30 iterations and that of the g6-MU-FG-GA detector with 30 iterations is much larger than for the case where $U = 5$.

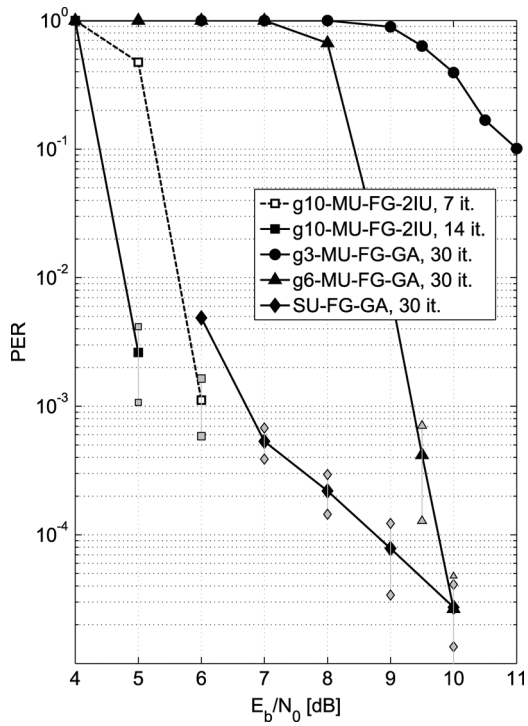

Fig. 11. PER versus $\frac{E_s}{N_0}$ for scheme #2 from Table I and $U = 17$. Data points based on N_{err} error events, with N_{err} smaller than 100: for g10-MU-FG-2IU with 7 iterations $N_{err} = 17$ at $\frac{E_b}{N_0} = 6$ dB, for g10-MU-FG-GA with 14 iterations $N_{err} = 11$ at $\frac{E_b}{N_0} = 5$ dB, for g6-MU-FG-GA with 30 iterations $N_{err} = 8$ at $\frac{E_b}{N_0} = 9.5$ dB and $N_{err} = 6$ at $\frac{E_b}{N_0} = 10$ dB, and for SU-FG-GA with 30 iterations $N_{err} = 52$ at $\frac{E_b}{N_0} = 7$ dB, $N_{err} = 26$ at $\frac{E_b}{N_0} = 8$ dB, $N_{err} = 10$ at $\frac{E_b}{N_0} = 9$ dB an $N_{err} = 15$ at $\frac{E_b}{N_0} = 10$ dB.

VI. Conclusions

In this paper we have derived two new multiuser detection algorithms for spectrally efficient bandpass communication of BIC-CPM signals over a typical satellite channel. General asynchronous transmission is considered.

The proposed techniques implement approximate minimum bit error rate MU detection. They are derived from the SP and FG framework. Two different FGs are employed. To reduce the computational complexity of the SP procedure a suitable set of SP messages are assumed to be Gaussian distribution functions.

The computational complexity and memory requirements of the proposed receivers are considerably lower than for the FG-based receiver from [5] (even with the approximation that the IUI is only due to the two adjacent channels) and about the same as for the receiver with simple *ad-hoc* IUI cancellation from [6].

Our simulation results show that the ranking of the different algorithms in terms of error performance strongly depends on the transmission system parameters. For all considered sets of system parameters, yielding a high spectral efficiency [about 1.8, 2.3, 2.08, and 2.15 transmitted information bits per second per Herz (b/s/Hz)], the proposed algorithms can provide a valuable alternative to more complex algorithms when the main concerns are the computational complexity and the memory requirements. At least one of the proposed detectors achieves low PER values at a lower SNR than the simple *ad-hoc* detector from [6]. For two sets of system parameters, with a carrier spacing between users of only 30% of the symbol rate (and a spectral efficiency of 1.8 and 2.3 b/s/Hz), a significant performance gain is observed with respect to the more complex approximate MU detector from [5], which ignores all non-adjacent users.

Appendix A

The per user processing part of the g10-MU-FG-GA and g6-MU-FG-GA detectors involves evaluating (25)–(28), with

$$f\left(\mathbf{S}_n^{(u)}\right) = \hat{P}\left(\dot{a}_{n-1}^{(u)}\right) \sum_{a_{n-L}^{(u)}} f\left(\dot{\mathbf{S}}_{n-1}^{(u)}\right) \Xi\left(\dot{\mathbf{s}}_{N_\tau^{(u)}+n}^{(u)}\right), \quad (25)$$

where $\sum_{a_{n-L}^{(u)}}$ denotes the summation over all possible values of $a_{n-L}^{(u)}$ and $\left(\dot{\mathbf{S}}_{n-1}^{(u)}, \dot{a}_{n-1}^{(u)}, \dot{\mathbf{s}}_{N_\tau^{(u)}+n}^{(u)}\right)$ is the only possible value of $\left(\mathbf{S}_{n-1}^{(u)}, a_{n-1}^{(u)}, \mathbf{s}_{N_\tau^{(u)}+n}^{(u)}\right)$ for which $\left(\mathbf{S}_{n-1}^{(u)}, a_{n-1}^{(u)}, \mathbf{s}_{N_\tau^{(u)}+n}^{(u)}, \mathbf{S}_n^{(u)}, a_{n-L}^{(u)}\right)$ satisfy the trellis (1)–(8) and (10),

$$b\left(\mathbf{S}_{n-1}^{(u)}\right) = \sum_{a_n^{(u)}} b\left(\ddot{\mathbf{S}}_n^{(u)}\right) \hat{P}\left(\ddot{a}_{n-1}^{(u)}\right) \Xi\left(\ddot{\mathbf{s}}_{N_\tau^{(u)}+n}^{(u)}\right), \quad (26)$$

where $\sum_{a_n^{(u)}}$ denotes the summation over all possible values of $a_n^{(u)}$ and $\left(\ddot{\mathbf{S}}_n^{(u)}, \ddot{a}_{n-1}^{(u)}, \ddot{\mathbf{s}}_{N_\tau^{(u)}+n}^{(u)}\right)$ is the only possible value of $\left(\mathbf{S}_n^{(u)}, a_{n-1}^{(u)}, \mathbf{s}_{N_\tau^{(u)}+n}^{(u)}\right)$ for which $\left(\mathbf{S}_n^{(u)}, a_{n-1}^{(u)}, \mathbf{s}_{N_\tau^{(u)}+n}^{(u)}, \mathbf{S}_{n-1}^{(u)}, a_n^{(u)}\right)$ satisfy the trellis (1)–(8) and (10),

$$\alpha\left(a_{n-1}^{(u)}\right) = \sum_{a_{n-L}^{(u)}, \mathbf{S}_n^{(u)}; a_{n-1}^{(u)}} b\left(\grave{\mathbf{S}}_n^{(u)}\right) f\left(\grave{\mathbf{S}}_{n-1}^{(u)}\right) \Xi\left(\grave{\mathbf{s}}_{N_\tau^{(u)}+n}^{(u)}\right), \quad (27)$$

where $\sum_{a_{n-L}^{(u)}, \mathbf{S}_{n-1}^{(u)}; a_{n-1}^{(u)}}$ denotes the summation over all possible values of $\left(a_{n-L}^{(u)}, \mathbf{S}_{n-1}^{(u)}\right)$ for a given value of $a_{n-1}^{(u)}$ and $\left(\grave{\mathbf{S}}_{n-1}^{(u)}, \grave{\mathbf{S}}_n^{(u)}, \grave{\mathbf{s}}_{N_\tau^{(u)}+n}^{(u)}\right)$ is the only possible value of $\left(\mathbf{S}_{n-1}^{(u)}, \mathbf{S}_n^{(u)}, \mathbf{s}_{N_\tau^{(u)}+n}^{(u)}\right)$ for which $\left(\mathbf{S}_{n-1}^{(u)}, \mathbf{S}_n^{(u)}, \mathbf{s}_{N_\tau^{(u)}+n}^{(u)}, a_{n-1}^{(u)}, a_{n-L}^{(u)}\right)$ satisfy the trellis (1)–(8) and (10), and

$$\hat{P}_e\left(\mathbf{s}_{N_\tau^{(u)}+n}^{(u)} | \mathbf{r}\right) = \begin{cases} b\left(\check{\mathbf{S}}_n^{(u)}\right) f\left(\check{\mathbf{S}}_{n-1}^{(u)}\right) \hat{P}\left(\check{a}_{n-1}^{(u)}\right) & asyn \\ b\left(\check{\mathbf{S}}_n^{(u)}\right) \left(\sum_{a_{n-L}} f\left(\check{\mathbf{S}}_{n-1}^{(u)}\right)\right) \hat{P}\left(\check{a}_{n-1}^{(u)}\right) & sync \end{cases}, \quad (28)$$

where we have to distinguish between the synchronous and the asynchronous case. For the asynchronous case, $\left(\check{\mathbf{S}}_{n-1}^{(u)}, \check{a}_{n-1}^{(u)}, \check{\mathbf{S}}_n^{(u)}\right)$ is the only possible value of $\left(\mathbf{S}_{n-1}^{(u)}, a_{n-1}^{(u)}, \mathbf{S}_n^{(u)}\right)$ for which $\left(\mathbf{S}_{n-1}^{(u)}, a_{n-1}^{(u)}, \mathbf{S}_n^{(u)}, \mathbf{s}_{N_\tau^{(u)}+n}^{(u)}\right)$ satisfy the trellis (1)–(8) and (10). For the asynchronous case, $\left(\check{\mathbf{S}}_{n-1}^{(u)}, \check{a}_{n-1}^{(u)}, \check{\mathbf{S}}_n^{(u)}\right)$ is the only

possible value of $\left(\mathbf{S}_{n-1}^{(u)}, a_{n-1}^{(u)}, \mathbf{S}_{n}^{(u)}\right)$ for which $\left(\mathbf{S}_{n-1}^{(u)}, a_{n-1}^{(u)}, \mathbf{S}_{n}^{(u)}, \mathbf{s}_{N_\tau^{(u)}+n}^{(u)}, a_{n-L}\right)$ satisfy the trellis (1)–(8) and (10). The above equations hold for $u = 1, 2, \ldots, U$, for $n = 1, 2, \ldots, N-1$ and for $L > 1$.

APPENDIX B

The function values of $L\left(\mathbf{s}_x^{(u)}; \mathbf{r}_{xR_s}\right)$ and $F\left(\mathbf{s}_x^{(u)}, \mathbf{s}_x^{(v)}\right)$ in (15)–(16) can be evaluated according to (14) and (13), and involves computing the quantities $\mathbf{s}_x^{(u)}$ and $\left(\mathbf{s}_x^{(u)}\right)^H \mathbf{s}_x^{(v)}$, for all u, for all v, for all relevant x and for all possible realizations of $\mathbf{s}_x^{(u)}$ and $\left(\mathbf{s}_x^{(u)}, \mathbf{s}_x^{(v)}\right)$. The values of $\mathbf{s}_x^{(u)}$ and $\left(\mathbf{s}_x^{(u)}\right)^H \mathbf{s}_x^{(v)}$, in their turn, can be efficiently obtained, using table look-ups. To limit the size of these look-up tables, we round the time delays $\tau^{(u)}T$, $u = 1, 2, \ldots, U$, to the nearest integer multiple of $T_{ss} = \frac{T}{R_{ss}}$, with R_{ss} integer valued. The vector $\mathbf{s}_x^{(u)}$ in (14) then follows directly from (1)–(3) and (10) and a precomputed look-up table (LUT-1) that stores $R_{ss}PM^L$ real-valued numbers, namely, the value of $\Psi\left(lT_{ss}; \mathbf{S}_n^{(u)}\right)$ for $l = 0, 1, \ldots, R_{ss}-1$ and for all PM^L possible realizations of $\mathbf{S}_n^{(u)}$. The vector product $\left(\mathbf{s}_x^{(u)}\right)^H \mathbf{s}_x^{(v)}$ in (13) can be computed as:

$$\left(\mathbf{s}_x^{(u)}\right)^H \mathbf{s}_x^{(v)} = \mathcal{L}_x^{(u,v)} \mathcal{T}_x^{(u,v)}, \tag{29}$$

with

$$\mathcal{L}_x^{(u,v)} = \exp\left\{-j\left(2\pi\left(f^{(u)} - f^{(v)}\right)x + \theta^{(u)} - \theta^{(v)}\right)\right\} \tag{30}$$

and

$$\mathcal{T}_x^{(u,v)} = \sum_{k=0}^{R_s-1} \left(\check{s}_{xR_s+k}^{(u)}\right)^* \check{s}_{xR_s+k}^{(v)} e^{-j2\pi(u-v)NSP\frac{k}{R_s}}, \tag{31}$$

where $\check{s}_{xR_s+k}^{(u)} = s_{xR_s+k}^{(u)} e^{-j\left(2\pi f^{(u)}\frac{k}{R_s} + \theta^{(u)}\right)}$. It is easily verified from (1)–(3) and (10) that the quantity $\mathcal{T}_x^{(u,v)}$ from (31) is completely determined by the value $(u-v)$ and by the sixtuple $\mathbf{e}_\mathcal{T}$, with

$$\mathbf{e}_\mathcal{T} = \left(\mathbf{S}_{x-N_\tau^{(u)}-1}^{(u)}, a_{x-N_\tau^{(u)}}^{(u)}, \mathbf{S}_{x-N_\tau^{(v)}-1}^{(v)}, a_{x-N_\tau^{(v)}}^{(v)}, d_\tau^{(u)}, d_\tau^{(v)}\right), \tag{32}$$

with $d_\tau^{(u)} = \tau^{(u)} - N_\tau^{(u)} \in [0, T[$, $N_\tau^{(u)}$ being the largest integer smaller than or equal to $\tau^{(u)}$. Since $\mathcal{T}_x^{(v,u)} = \left(\mathcal{T}_x^{(u,v)}\right)^*$ it suffices to precompute $\mathcal{T}_x^{(u,v)}$ for positive values of $(u-v)$. For a given value of $|u-v|$, the values of $\mathcal{T}_x^{(u,v)}$ from (31) can therefore to be stored in a precomputed look-up table (LUT-2.ℓ, with $\ell = |u-v|$) that contains $2\left(R_{ss}PM^{L+1}\right)^2$ real-valued numbers, namely, the real and imaginary parts of the complex-valued result of the right-hand side of (31), evaluated for all $\left(R_{ss}PM^{L+1}\right)^2$ possible values of $\mathbf{e}_\mathcal{T}$ from (32). In general, $(U-1)$ such look-up tables are required by the detector from [5]: LUT-2.1, LUT-2.2, \ldots , LUT-2.$(U-1)$. The simplified detector, which only takes into account the interference from the $2U_I$ most adjacent users, needs only U_I look-up tables, namely, LUT-2.1, LUT-2.2, \ldots , LUT-2.U_I.

REFERENCES

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

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

[3] H. Meyr, M. Moeneclaey, and S. Fechtel, *Digital Communication Receivers: Synchronization, Channel Estimation, and Signal Processing, Wiley Series in Telecommunications and Signal Processing*. New York: Wiley, 1998.

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

[5] A. Piemontese and G. Colavolpe, "A novel graph-based suboptimal multiuser detector for FDM-CPM transmissions," *IEEE Trans. Wireless Commun.*, vol. 9, no. 9, pp. 2812–2819, Sep. 2010.

[6] A. Perotti, S. Benedetto, and P. Remlein, "Spectrally efficient multiuser continuous-phase modulation systems," in *Proc. IEEE Int. Conf. Commun. (ICC2010)*, Cape Town, South Africa, May 23–27, 2010.

[7] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 3, pp. 248–287, Mar. 1974.

[8] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1772–1793, Jul. 2002.

[9] A. Piemontese, A. G. i. Amat, and G. Colavolpe, "Information-theoretic analysis and practical coding schemes for spectrally efficient FDM-CPM systems," in *Proc. 2010 6th Int. Symp. Turbo Codes and Iterative Inf. Process.*, Brest, France, Sep. 6–10, 2010.

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

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

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

He is a Professor with the Department of Telecommunications and Information Processing (TELIN), Gent University. His main research interests are in statistical communication theory, (iterative) estimation and detection, carrier and symbol synchronization, bandwidth-efficient modulation and coding, spread-spectrum, satellite, and mobile communication. He is the author of more than 400 scientific papers in international journals and conference proceedings. He is the coauthor (together with Prof. H. Meyr, RWTH Aachen, and Dr. S. Fechtel, Siemens AG), of the book *Digital Communication Receivers—Synchronization, Channel Estimation, and Signal Processing* (Hoboken, NJ: Wiley, 1998).

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