Interleaved coded modulation for non-binary codes: a factor graph approach

Henk Wymeersch, Heidi Steendam and Marc Moeneclaey
DIGCOM research group, TELIN Dept., Ghent University
Sint-Pietersnieuwstraat 41, 9000 GENT, BELGIUM
E-mail: {hwymeers,hs,mm}@telin.ugent.be

***Abstract*— This contribution deals with the problem of combining (non-binary) error-correcting codes with higher-order modulation schemes on AWGN and flat fading channels through interleaved coded modulation. We extend the idea of bit-interleaved coded modulation (BICM) to a more general form. With the aid of factor graph representations, we show how higher-order modulation can be combined with non-binary codes so that non-iterative detection becomes optimal. This is in contrast with conventional BICM, for which non-iterative detection can be far from optimal. Through computer simulations, we compare an optimal non-iterative scheme with conventional BICM. It turns out that the gain due to optimal detection is outweighed by the loss in time-diversity as compared to BICM-ID.**

I. Introduction

While state-of-the-art error-correcting codes have been shown to achieve near-capacity performance, their relatively low rates result in a significant bandwidth expansion. To combat this, a number of techniques combining channel coding with higher-order modulations have been proposed. They include trellis coded modulation (TCM), Turbo-TCM, multilevel coding (MLC) and bit-interleaved coded modulation with or without iterative decoding (BICM-ID and BICM resp.).

TCM was first proposed in [1] for convolutional codes whereby the signalling constellation was partitioned in subsets in such a way that the free Euclidean distance is increased. To obtain good performance in this technique, one must take care in the selection of bit-to-constellation mappings, avoiding parallel transitions, etc. In TTCM [2] this idea was extended to turbo codes with bit- or symbol-interleaving [3]. Despite good performance results, the computational complexity of (T-)TCM is fairly high. In MLC [4], the idea is to partition the signal constellation and to protect each level of the partition with separate binary codes. Good performance has been reported for MLC with both turbo- and low density parity check (LDPC) codes, but at the cost of high complexity [5].

A more attractive solution is BICM [6], [7]. Encoding and mapping are now completely disjoint, with coder and mapper separated by a bit-interleaver. When applied with iterative decoding [8], one iterates at the receiver side between decoding and demapping with the exchange of extrinsic information. Although this iterative decoding/demapping approach is suboptimal, it is shown in [9] that at a fixed complexity, BICM-ID has superior performance as compared to (T-)TCM in both AWGN and Rayleigh fading channels. Recently, in [10], a way to deal with this sub-optimality was proposed: by using symbol-decoding, rather than bit-decoding a 0.2 dB gain was observed for LDPC codes. One of the main issues in BICM-ID is finding the optimal mapping strategy. In [11], heuristics have been proposed to find optimum mapping strategies for BICM-ID.

Non-binary codes have been investigated in [12]–[15]. In particular, LDPC codes over finite fields have received a lot of attention [16]. Combining non-binary codes with interleaved modulation was considered in [13], [15] for LDPC codes and turbo codes, respectively. The main focus in [15] was on code design for codes over groups and rings. In [13] non-binary turbo codes and mapping were combined, but no comparisons with BICM were made.

In this contribution we propose an extension of BICM-ID for codes over binary extension fields based on factor graphs [17] and the sum-product (SP) algorithm. It can be shown that when a factor graph contains cycles, the performance of the SP algorithm is degraded [18]. By modifying the scheduling strategy of the SP algorithm, sensitivity to cycles can be reduced. We demonstrate that a cycle-free factor graph can be constructed when the number of constellation points in the signal set does not exceed the number of elements in the field over which the code is defined. This results in an non-iterative optimal receiver, which turns out to be fairly insensitive to mappings and does not require us to find an optimal message scheduling strategy. We compare, through computer simulations, the performance of such a scheme with a BICM and BICM-ID algorithm for a non-binary convolutional code with 8-PSK mapping. We show that the removal of cycles in the graph results in a loss of time-diversity as compared to BICM. Hence, the gain resulting from optimal detection is reduced because of a degradation due to diversity loss. This can be seen as a generalization of BICM from [6], where it was noticed that the gain in diversity thanks to bit-interleaving can more than compensate for the sub-optimal decoding as compared to symbol interleaving. The proposed framework for interleaved coded modulation allows the code designer to trade time-diversity for decoding complexity and decoding optimality.

II. Interleaved coded modulation

The ICM system under consideration consists of the following parts:

- a block *code* C over a binary extension field $GF(q)$, with $C \subset GF(q)^L$, where $L \in \mathbb{N}$ denotes the length of the codewords. We restrict ourselves to codes that have a practical soft-decoding algorithm based on the sum-product algorithm [18]; [1]
- a bijective *field-conversion function* $\varphi(.)$. This function maps a sequence of n_1 elements from $GF(q)$ to a sequence of n_2 elements in $GF(\tilde{q})$, with $q^{n_1} = \tilde{q}^{n_2}$;
- a pseudo-random *interleaver* $\Pi(.)$ over $GF(\tilde{q})$ of size Ln_2/n_1;
- a *mapping function* $\psi(.)$ from $GF(\tilde{q})^m$ onto a M-point complex signalling constellation Ω. This function is also bijective, which implies $\tilde{q}^m = M$.

Hence, the main system parameters are q (the size of the field over which the code is defined), $\tilde{q}$ (the size of the field in which the interleaving takes place) and M (the number of points in the signalling constellation). The code C can be seen as an outer code, while the mapping $\psi(.)$ can be interpreted as a inner code [19]. The transmitter operates as follows: bits are grouped to form elements in $GF(q)$. These elements are then block-encoded. A sequence $\mathbf{c}$ of L coded symbols is converted by $\varphi(.)$ to a sequence $\tilde{\mathbf{c}}$ of Ln_2/n_1 elements in $GF(\tilde{q})$, i.e., $\varphi(\mathbf{c}) = \tilde{\mathbf{c}}$ [2]. The sequence $\tilde{\mathbf{c}}$ is then interleaved resulting into $\Pi(\tilde{\mathbf{c}})$. The interleaved sequence $\Pi(\tilde{\mathbf{c}})$ is split up into groups of m elements; each group of m elements in $GF(\tilde{q})$ is mapped onto a point in the constellation Ω. This results in a sequence $\mathbf{x} \in \Omega^N$, with $\mathbf{x} = \psi(\Pi(\tilde{\mathbf{c}}))$. Note that $N = Ln_2/(n_1 m)$.

We will investigate both AWGN and flat fading channels. In both cases, the received discrete-time signal can be written as

$$\mathbf{r} = \mathbf{y} + \mathbf{n} \tag{1}$$

with $y_k = \alpha_k x_k$, where α_k are the complex gains, with $\mathrm{E}\left[|\alpha_k|^2\right] = 1$, while $\mathbf{n} = [n_0 \ldots n_{N-1}]$ is a vector of independent complex AWGN samples with $n_k \sim \mathcal{CN}\left(0, 2\sigma^2\right)$. In the case of a simple AWGN channel, the channel coefficients yield $\alpha_k = 1$, $\forall k$. We assume perfect channel knowledge at the receiver side.

III. ICM: A FACTOR GRAPH APPROACH

A. Factor graphs and indicator functions

We will use the normal graphs (NG) that were introduced in [17]. An NG is a diagram that represents the factorization of a function of several variables:

$$f(a_1, a_2, \ldots, a_N) = \prod_j f_j(A_j)$$

where A_j is a subset of $\{a_1, a_2, \ldots, a_N\}$. An NG consists of nodes, edges and half-edges (the latter are connected to only one node). The NG is related to the function $f(.)$ as follows: there is a node for every factor $f_j(.)$ and one (half-) edge for every variable a_k. Node f_j is connected to variable a_k iff $a_k \in A_j$. Finally, edges are connected to exactly two nodes.

We also introduce the notion of the so-called *indicator function*, $I[b]$: i.e., a binary valued function defined as follows. For a predicate b, $I[b] = 0$ if b is false and $I[b] = 1$ if b is true.

B. ICM as a factor graph

The valid configurations of transmitted frames and the fading coefficients from section II can be easily translated into function with $\mathbf{a} = \{\mathbf{c}, \tilde{\mathbf{c}}, \mathbf{x}, \mathbf{y}\}$

$$\begin{aligned} &f(\mathbf{a}) = \\ &I[\mathbf{c} \in C]\, I[\varphi(\mathbf{c}) = \tilde{\mathbf{c}}]\, I[\psi(\Pi(\tilde{\mathbf{c}})) = \mathbf{x}] \prod_k I[x_k \alpha_k = y_k]\,. \end{aligned} \tag{2}$$

As $f(\mathbf{a})$ is the product of indicator functions, it is itself an indicator function: it indicates that the codeword $\mathbf{c}$ must satisfy the code constraints, is converted to $\tilde{\mathbf{c}}$, interleaved and mapped to the transmitted sequence $\mathbf{x}$. Then $\mathbf{x}$ is undergoes flat fast fading with fading coefficients α_k. Only those values of $\mathbf{a}$ satisfying all conditions will evaluate to $f(\mathbf{a}) = 1$. Note that because of the assumption that the fading coefficients are known at the receiver, α_k is not part of $\mathbf{a}$. The function (2) can be represented by the factor graph from Fig. 1. This NG consists of the following nodes:

- the code constraints (CC) block, representing $I[\mathbf{c} \in C]$. In most cases this function can be factorized and represented by a NG;
- the field-conversion nodes (labeled φ) representing the factorization of $I[\varphi(\mathbf{c}) = \tilde{\mathbf{c}}]$;
- the mapper node (labeled ψ) representing the factorization of $I[\psi(\Pi(\tilde{\mathbf{c}})) = \mathbf{x}]$;
- the fading nodes (with labels α_k).

Note that the interleaver is not a node, just a re-ordering of the edges.

Factor graphs not only allow us to visualize the system, but can also be used to compute marginal probabilities: when all codewords are equiprobable, it can be shown that the indicator function $f(\mathbf{a})$ is proportional to the *prior* distribution $p(\mathbf{a})$ of the set of variables $\mathbf{a}$ [18]. Hence, the *posterior* distribution of $\mathbf{a}$ can be written as

$$\begin{aligned} p(\mathbf{a}|\,\mathbf{r}) &\propto p(\mathbf{r}|\,\mathbf{a})\, p(\mathbf{a}) \\ &\propto p(\mathbf{r}|\,\mathbf{a})\, f(\mathbf{a}) \\ &\propto \prod_k p(r_k|\,y_k)\, f(\mathbf{a})\,, \end{aligned}$$

where $p(r_k|\,y_k)$ is related to the distribution of the AWGN samples n_k, i.e., $p(r_k|\,y_k) = \mathcal{CN}\left(y_k, 2\sigma^2\right)$. Consequently, $p(\mathbf{a}|\,\mathbf{r})$ can be represented by the factor graph in Fig. 1. Note that the observations r_k are not variables in the factor graph: they are known at the receiver and should be considered as parameters.

[1] Such codes include convolutional and turbo codes, LDPC and Repeat-Accumulate (RA) codes

[2] Throughout this paper, we use the following notational shorthand: for a function $f : A \rightarrow B$, with $f(a_i) = b_i$, we denote by $f(\mathbf{a}) \triangleq [f(a_0) \ldots f(a_{L-1})]$, for $\mathbf{a} \triangleq [a_0 \ldots a_{L-1}]$.

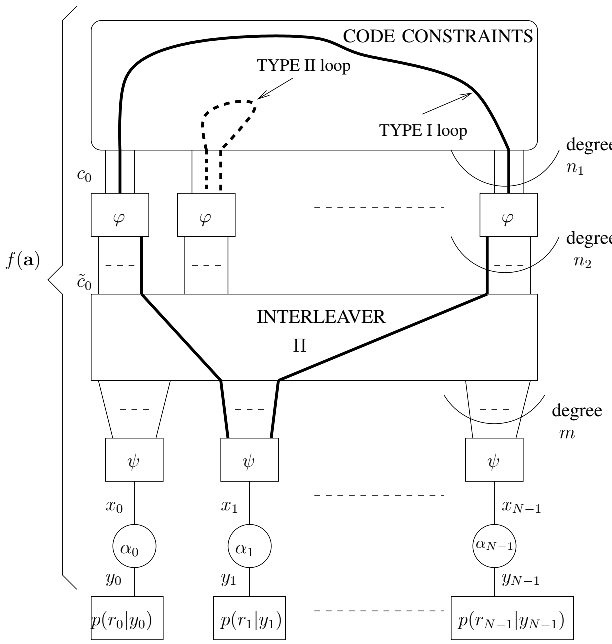

Fig. 1. Normal graph for general ICM. A Type I (II) cycle is shown in bold (dashed).

C. Marginalization and message passing

Suppose we are interested in computing the posterior pdf of the variable $p\left(a_k|\,\mathbf{r}\right)$. It is clear that there is a great deal of commonality in the computation of $p\left(a_k|\,\mathbf{r}\right)$ and $p\left(a_{k'}|\,\mathbf{r}\right)$, $k' \neq k$. To compute all $p\left(a_k|\,\mathbf{r}\right)$ in an efficient manner, we resort to the sum-product (SP) algorithm [18]: in this algorithm we pass *messages* over the edges in the NG. These messages are pdfs (or rather pmfs, in the case of discrete variables) of the variables associated with the given edge. The message from factor (node) f_n to variable (edge) a_k (with $a_k \in A_n$) is defined as:

$$\mu_{f_n \to a_k}\left(a_k = a\right) = \gamma \sum_{A_n : a_k = a} f_n\left(A_n\right) \prod_{l \neq k} \mu_{a_l \to f_n}\left(a_l\right) \quad (3)$$

with γ a normalization constant, such that $\sum_{a_k} \mu_{f_n \to a_k}\left(a_k\right) = 1$. In the special case when f_n has degree 1, (3) becomes

$$\mu_{f_n \to a_k}\left(a_k = a\right) = \gamma f_n\left(a\right). \quad (4)$$

The message from variable (edge) a_k to factor (node) f_n (with $a_k \in A_n$) is denoted by $\mu_{a_k \to f_n}\left(a_k\right)$. When a_k is a half-edge $\mu_{a_k \to f_n}\left(a_k\right) \equiv 1$. For 'real' edges, since an edge connects exactly two nodes, $\mu_{a_k \to f_n}\left(a_k\right) = \mu_{f_m \to a_k}\left(a_k\right)$ for some factor $f_{m \neq n}$. The message passing algorithm starts from the half-edges and nodes with degree 1. Only when all incoming messages have been received, nodes compute the outgoing messages. It can be shown (see, for example reference [18]) that for cycle-free NGs, this algorithm terminates after a finite number of steps and results in the exact marginal pdfs. The pdf $p\left(a_k|\,\mathbf{r}\right)$ is given by the product of two messages over the corresponding edge:

$$p\left(a_k|\,\mathbf{r}\right) = \gamma' \mu_{f_n \to a_k}\left(a_k\right) \mu_{a_k \to f_n}\left(a_k\right) \quad (5)$$

where γ' is a normalization constant, such that $\sum_{a_k} p\left(a_k|\,\mathbf{r}\right) = 1$.

When the graph contains cycles (loops), the SP algorithm has no natural initialization, nor termination. Initialization can be performed by setting all required messages to uniform distributions. After a given number of iterations, the entire system converges and the marginal pdfs can be computed according to (5). However, the resulting marginal pdfs are not the exact marginal pdfs, but an *approximation* of them. This also brings up the question of scheduling: as there are many possible ways to iterate (i.e., depending on how messages are scheduled), there are many ways to perform the SP algorithm. Different scheduling strategies will have different performances. Finding the optimal scheduling strategy is not a trivial task [20].

D. ICM through message passing

It is clear that the NG from Fig. 1 generally contains cycles. We can discern two types of cycles: type I cycles between the CC block and a mapping function nodes (shown in bold in Fig. 1) and type II cycles between the CC block and a field conversion node (shown in dashed in Fig. 1). This implies that the SP algorithm between the decoder and the demapper/field conversion will be iterative and therefore sub-optimal [3]. As a matter of fact, in code-design it is always important to avoid cycles in NG, or make them as long as possible [21]. From Fig. 1 is is also clear that type I cycles are less critical than type II cycles, as the latter are generally longer.

The computation of the posterior pdf $p\left(c_k|\,\mathbf{r}\right)$ of the coded symbols in the factor graph shown in Fig. 1 can be accomplished by applying the SP algorithm as follows. First messages to the variables y_k are computed using (4)

$$\mu_{p(\,r|y) \to y_k}\left(y_k\right) = \gamma p\left(r_k\,|y_k\,\right).$$

These messages are forwarded to the fading nodes where messages to the variables x_k are computed:

$$\mu_{\alpha_k \to x_k}\left(x_k\right) = \gamma' \mu_{y_k \to \alpha_k}\left(\alpha_k x_k\right) = p\left(r_k\,|\alpha_k x_k\,\right).$$

Suppose the k-th mapper node is connected to $\tilde{\mathbf{c}}_k \triangleq \left[\tilde{c}_{k_0}, \tilde{c}_{k_1}, \ldots, \tilde{c}_{k_{m-1}}\right]$, then (3) becomes

$$\mu_{\psi \to \tilde{c}_{k_i}}\left(a\right) = \gamma'' \sum_{x} \mu_{x_k \to \psi}\left(x\right) \left\{ \prod_{l \neq i} \mu_{\tilde{c}_{k_l} \to \psi}\left(\left[\psi^{-1}\left(x\right)\right]_{k_l}\right) \right\} \quad (6)$$

where the summation is taken over all x whose inverse mapping has a as k_i-th $GF\left(\tilde{q}\right)$ symbol, i.e., all $x : \left[\psi^{-1}\left(x\right)\right]_{k_i} =$

[3]In many cases the Code Constraint (CC) block in Fig. 1 will itself contain cycles. This is the case for LDPC codes, Turbo codes, RA codes... but not for convolutional codes. When cycles are present in the NG of the code, decoding itself is iterative (and sub-optimal).

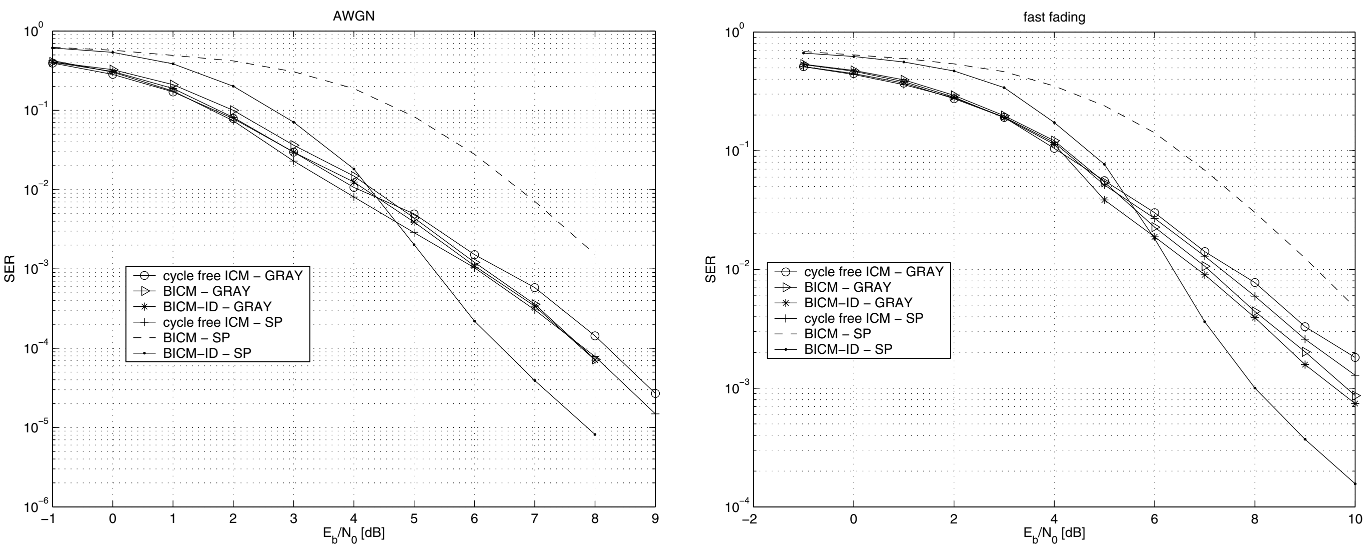

Fig. 2. Convolutional code: SER performance of cycle free ICM compared to BICM-ID and BICM for Gray and set-partitioning mapping. Left: AWGN channel; right: Fading channel

a with $a \in GF(\tilde{q})$. Because of the cycles in the graph, we need to initialize $\mu_{\tilde{c}_{k_l} \to \psi}(\tilde{c}_{k_l}) = 1/\tilde{q}$. Messages are then forwarded to the field conversion nodes and so forth.

After convergence, the probabilities $p(c_k | \mathbf{r})$ are used to make final decisions on the coded symbols.

E. Optimal detection vs. diversity

To avoid type I cycles in the graph we need $m = 1$. Similarly, to avoid type II cycles, we need to restrict n_1 to $n_1 = 1$. Hence, optimal (i.e., non-iterative) detection can only be achieved when $M \le q$ and $M = \tilde{q}$. On the other hand, it was shown in [6] that an interleaver in the binary field achieves the best possible time-diversity, assuming sufficiently large interleaver depth. This means that $\tilde{q}$ should be as small as possible. Clearly, these objectives are contradictory. To demonstrate this, we highlight three special cases.

Binary code with symbol-interleaving: This is equivalent to the "baseline" system from [6]: groups of 3 bits form symbols. These symbols are interleaved and mapped onto 8-PSK symbols. With our notations, this becomes $q = 2$, $n_1 = 3$, $\tilde{q} = 8$, $n_2 = 1$ and $m = 1$. Such a scheme contains type II cycles, but no type I cycles. In this case, we combine non-optimal detection with low diversity, which is clearly not a good situations. The principle of BICM from [6] can be interpreted as trading type II cycles for type I cycles.

Bit-Interleaved Coded Modulation (with Iterative Decoding): In the case of BICM and BICM-ID, $\tilde{q} = 2$, $n_1 = 1$, $n_2 = \log_2 q$ and $m = \log_2 M$: the field-conversion function converts each symbol in $GF(q)$ to a sequence of $\log_2 q$ bits. All $L \log_2 q$ bits are interleaved and mapped onto an M-point signaling constellation. Observe that no type II cycles are present and that $\tilde{q} = 2$, resulting in maximal time-diversity. The performance of BICM-ID depends on the mapping function $\psi(.)$, i.e., how bits are mapped onto the constellation. The performance gain of a given mapping compared to another mapping strongly depends on the SNR (and can even be a performance loss for certain SNRs) [11]. Because the NG corresponding to BICM-ID will contain type II cycles, performance will also depend on the scheduling strategy. For example, a possible strategy would be to not iterate between the mapper nodes and the CC block. When we schedule in this fashion, we end up with the standard BICM scheme from [6]. In that case, $\psi(.)$ should be Gray mapping.

Cycle-free ICM: We now set $m = n_1 = 1$. Since $\tilde{q}^m = M$, this implies that $\tilde{q} = M$. BICM-ID with BPSK modulation is a special case of this. More generally, for any code over $GF(q)$, a signalling constellation of size $M \le q$ exists such that the resulting factor graph contains no loops. In this case Eq. (6) becomes

$$\mu_{\psi \to \tilde{c}_{k_0}}(a) = \gamma \mu_{x_k \to \psi}(\psi(a)). \tag{7}$$

For example, an ICM scheme with a code over $GF(16)$, $\tilde{q} = 4$ and 4-PSK signaling will have a cycle-free NG[4]. In those cases, the SP algorithm can be performed with a single pass over the mapper nodes, passing messages to the CC-block and perform the SP algorithm within the CC-block. There is no need to return messages to the mapper nodes. Such an approach has several advantages: no cycles, so SP between mapper nodes and CC block is optimal. Moreover, when $\tilde{q} = q$, there is even no need for an interleaver. For codes that exhibit a lot of inherent randomness, performance should not be very sensitive to the specific mapping ψ. Note that since $\tilde{q} = M$, we cannot achieve maximal time-diversity.

[4]cycle-free in the sense that there are no cycles between the mapper nodes and the CC block. There may of course still be cycles within the CC block.

IV. Performance results

We can consider two types of outer codes: strong codes (e.g., turbo and LDPC codes) and weaker codes (such as convolutional codes). We will first discuss weaker codes and end by making some comments regarding powerful codes.

We have carried out computer simulations for a convolutional code over $GF(8)$ with 8-PSK signaling. The convolutional code is a rate 1/2 recursive systematic code over $GF(8)$ with constraint length 4 and block-size 60. We compare the conventional BICM-ID scheme with a cycle-free ICM version. For the BICM-ID scheme, we set $\tilde{q} = 2$, $n_1 = 1$, $n_2 = 3$ and $m = 3$. We have considered both Gray mapping and set-partitioning mapping. It is shown in [19] that 8-PSK set-partitioning mapping benefits from iterative demapping, while Gray mapping does not. In the cycle-free ICM, we set $\tilde{q} = 8$, $n_1 = 1$, $n_2 = 1$ and $m = 1$. Performance is measured in terms of the symbol error rate (SER). We show results for both an AWGN and a fast flat fading channel.

In the left part of Fig. 2, SER results are shown for the AWGN channel: cycle-free ICM and BICM with Gray mapping have similar performance. Observe the fairly low sensitivity of the SER to the mapping, in contrast to the significant impact of the mapping on BICM-ID. Compared with BICM-ID for SP, cycle-free ICM is not able to compensate for the time-diversity. In a fading environment (right part of Fig. 2), the situation is even more pronounced: now cycle free ICM is outperformed by all other schemes (except BICM with SP, which is never used in practice). This corresponds nicely to the results from [6] where BICM was introduced as an alternative to symbol-interleaving (i.e., group-wise interleaving) for binary codes.

When the outer codes is a powerful error-correcting code, it does not benefit from iterative detection [22]: in the SNR-range of interest, Gray mapping outperforms any other type of mapping. Hence, for powerful codes, BICM with simple Gray mapping achieves the best performance. We have verified for an LDPC code over $GF(8)$ with 8-PSK signaling (results not shown) that cycle-free ICM yields performance results that are independent of the mapping function. Moreover, the SER performance was very similar to BICM with Gray mapping. For fast fading channels, cycle-free ICM outperformed BICM with Gray mapping with about 0.2 dB.

V. Conclusions and remarks

We have considered the problem of combining non-binary codes with higher-order signaling constellations by means of interleaved coded modulation. With the aid of factor graph representations, we have investigated an optimal detection strategy based on cycle-free graphs. However, this approach results in a reduction in time-diversity. For fading channels, the net result is that cycle-free ICM gives rise to a degradation compared to conventional BICM, due to the loss in time-diversity. On the AWGN channel, cycle-free ICM and BICM with gray mapping have similar performance. For weak outer codes, BICM-ID with an optimized mapping is able to outperform cycle-free ICM for both channel types, albeit at a higher computational cost. Finally, for strong outer codes, BICM with Gray mapping and cycle-free ICM yield roughly the same performance.

Acknowledgement

This work has been supported by the Interuniversity Attraction Poles Program P5/11 - Belgian State - Belgian Science Policy.

References

[1] G. Ungerboeck. "Channel coding with multilevel/phase signal". *IEEE Trans. on Information Theory*, 28:55–66, January 1982.

[2] P. Robertson and T. Wörz. "Bandwidth-efficient turbo trellis-coded modulation using punctured component codes". *IEEE Journal on Selected Areas in Comm.*, 16:206–218, February 1998.

[3] C. Fragouli and R.D. Wesel. "Turbo-encoder design for symbol-interleaved parallel concatenated trellis-coded modulation". *IEEE Trans. on Comm.*, 49(3):425–435, March 2001.

[4] U. Wachsmann, R.F.H. Fischer and J.B. Huber. "Multilevel codes: theoretical concepts and practical design rules". *IEEE Trans. on Information Theory*, 45(5):1361–1391, July 1999.

[5] K.R. Narayanan and J. Li. "Bandwidth efficient low density parity check coding using multilevel coding and iterative multi stage decoding". In *Proc. 2nd Symp. on Turbo Codes*, Brest, France, July 2000.

[6] E. Zehavi. "8-PSK trellis codes for Rayleigh fading channels". *IEEE Trans. on Comm.*, 41:873–883, May 1992.

[7] G. Caire, G. Taricco and E. Biglieri. "Bit-interleaved coded modulation". *IEEE Trans. on Information Theory*, 44:927–946, May 1998.

[8] X. Li and J.A. Ritcey. "Trellis-coded modulation with bit interleaving and iterative decoding". *IEEE Journal on Selected Areas in Comm.*, 17(4), April 1999.

[9] S.X. Ng, T.H. Liew, L.L. Yang and L. Hanzo. "Comparative study of TCM, TTCM, BICM and BICM-ID schemes". In *Proc. of VTC2001 (Spring)*, pages 2450–2454, Rhodes, Greece, May 2001.

[10] B-Z. Shen, H. Tran and K. Cameron. "Low-density parity-check coded modulation using multiple signal maps and symbol decoding". In *Proc. IEEE International Conference on Communications (ICC)*, Paris, France, June 2004.

[11] F. Schreckenbach, N. Görtz, J. Hagenauer and G. Bauch. "Optimization of symbol mappings for bit-interleaved coded modulation with iterative decoding". *IEEE Comm. Letters*, 7(12):593–595, December 2003.

[12] J. Tan and G.L. Stüber. "A MAP equivalent SOVA for non-binary turbo codes". In *Proc. IEEE International Conference on Communications (ICC)*, pages 602–606, vol. 2, June 2000.

[13] M. Bingeman and A.K. Khandani. "Symbol-based turbo codes". *IEEE Comm. Letters*, 3(10):285–287, October 1999.

[14] J. Berkmann. "On turbo decoding of nonbinary codes". *IEEE Comm. Letters*, 2(4):94–96, April 1998.

[15] D. Sridhara and T. Fuja. "Low density parity check codes over groups and rings". In *Proc. 2002 IEEE Information Theory Workshop*, Bangalore, India, October 2002.

[16] M.C. Davey and D.J.C. MacKay. "Low density parity check codes over GF(q)". *IEEE Comm. Letters*, 2(6):pp.165–167, June 1998.

[17] G.D. Forney. "Codes on Graphs: Normal Realizations". *IEEE Trans. on Information Theory*, 47(2):520–545, February 2001.

[18] F. Kschinschang, B. Frey and H.-A. Loeliger. "Factor graphs and the sum-product algorithm". *IEEE Trans. Inform. Theory*, 47(2):pp.498–519, February 2001.

[19] X. Li, A. Chindapol and J.A. Ritcey. "Bit-interleaved coded modulation with iterative decoding and 8PSK signaling". *IEEE Trans. on Comm.*, 50(8):1250–1257, August 2002.

[20] H.-A. Loeliger. "Some remarks on factor graphs". In *Proc. 3rd International Symposium on Turbo Codes and Related Topics*, pages 111–115, Brest, France, 2003.

[21] J. Campello and D.S. Modha. "Extended bit-filling and LDPC code design". In *Proc. Globecom*, San Antonio, USA, November 2001.

[22] M. Gonzalez-Gopez and J. Garcia-Frias. "Bit interleaved coded modulation using low-density generator matrix codes". In *Proc. Baiona Workshop on Signal Processing in Communications*, Baiona, Spain, September 2003.