COMPLEXITY REDUCTION IN BICM–ID SYSTEMS THROUGH SELECTIVE LOG-LIKELIHOOD RATIO UPDATES

S. Schwandter[1], *Z. Naja*[2], *P. Duhamel*[2], *G. Matz*[1]

[1]Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology
Gusshausstrasse 25/389, A-1040 Vienna, Austria, email: {sschwand,gmatz}@nt.tuwien.ac.at

[2]Laboratoire des Signaux et Systèmes (LSS), CNRS/Supélec
3 Rue Joliot Curie, F-91192 Gif-sur-Yvette, France, email: {ziad.naja,pierre.duhamel}@lss.supelec.fr

ABSTRACT

Bit-interleaved coded modulation with iterative decoding (BICM–ID) performs well under a variety of channel conditions. Based on a factor graph representation of BICM–ID, we propose a modification of the BICM-ID receiver that can reduce the computational complexity significantly without BER degredation. The basic idea is to avoid updates of code bit log-likelihood ratios whose magnitude is already sufficiently large. Simulation results illustrate the excellent performance-complexity trade-off of our method.

1. INTRODUCTION

Bit-interleaved coded modulation is a pragmatic approach to coded modulation. It has been first proposed by Zehavi [1] and a thorough analysis of the achievable information rates and error probabilities was given by Caire et al. [2]. BICM has been used in many recent standards like DVB-S2, wireless LANs, DSL and WiMAX because of its flexibility and simplicity. With BICM, modulation and coding are separated by a bitwise interleaver. At the receiver, the demodulator passes soft information about the transmitted code bits on to the decoder. This soft information usually consists of log-likelihood ratios (LLRs). The (small) performance loss of BICM relative to optimal coded modulation can be further decreased via iterative (turbo) decoding where LLRs are repeatedly exchanged between the decoder and the demodulator (see [3, 4]).

Since iterative decoding of BICM (BICM–ID for short) increases the computational complexity of the receiver, low-complexity implementations have received increasing attention. With regard to the demodulator, complexity reduction is often achieved using the max-log approximation. Approaches to further simplify the max-log demodulator are described in [5, 6]. BICM has also been studied in the context of multiple input multiple output (MIMO) systems [7]. Here, the demodulator complexity is very high, even with sophisticated sphere decoder implementations (e.g. [8]).

This work was funded by FWF Grant S10606 and the EU FP7 Network of Excellence NEWCOM++ (Grant No. 216715).

In this paper we propose a general approach to reduce the computational complexity necessary in the iterative decoding process. In contrast to previous work, we save computations not only at the demodulator but also at the decoder. Our approach builds on the interpretation of turbo demodulation as message passing (or loopy belief propagation) on a corresponding factor graph. The basic idea, referred to as "selective LLR update", is to stop updating the LLR for a code bit once the magnitude of this LLR exceeds a prescribed threshold. While being applicable to any turbo-like iterative receiver (see [9] for the application in a multi-user setup, and [10, 11] in LDCP decoding), we here restrict ourselves to BICM–ID and describe in detail the modifications to the usual message passing scheme. We further provide an interpretation of our approach as an *adaptive* message passing schedule. Numerical simulations corroborate the fact that our scheme achieves significant complexity savings.

The rest of the paper is organized as follows. Section 2 discusses BICM–ID. In Section 3, we describe the selective LLR update scheme. Simulation results are provided in Section 4; conclusions are presented in Section 5.

2. BICM–ID

2.1. System Model

Transmitter. A sequence of I information bits $u_i \in \{0,1\}$, $i = 1,\dots,I$, is encoded using a convolutional channel code $\mathcal{C}$ of rate R. The encoder generates $J = 1/R$ code bits $c_{i,j}$, $j = 1,\dots,J$, per information bit. Denoting the encoder states by s_i, we write $(s_i, \mathbf{c}_i) = \omega(s_{i-1}, u_i)$ to characterize the state transition and code bits $\mathbf{c}_i = (c_{i,1} \dots c_{i,J})$ generated when the encoder is in state s_{i-1} and is fed with the information bit u_i. The code bits $c_{i,j}$ are passed through an interleaver Π that provides the sequence $\tilde{c}_{k,l}$ with $(k,l) = \Pi(i,j)$. Here, $k = 1,\dots,K$ denotes the symbol time and $l = 1,\dots,L$ with $L = \log_2|\mathcal{X}|$, where $\mathcal{X}$ denotes the complex signal constellation. Each block $\tilde{\mathbf{c}}_k = (\tilde{c}_{k,1},\dots,\tilde{c}_{k,L})$ is modulated to a symbol $x_k = \mu(\tilde{\mathbf{c}}_k) \in \mathcal{X}$ using a mapping $\mu : \{0,1\}^L \to \mathcal{X}$. The symbols are transmitted over a fading channel with input-

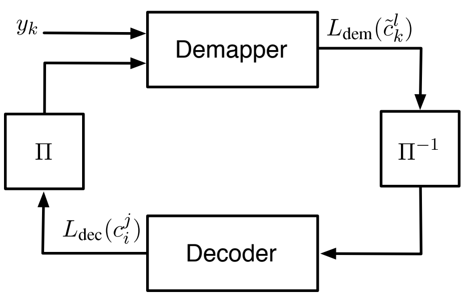

Fig. 1. *Block diagram of a BICM–ID receiver.*

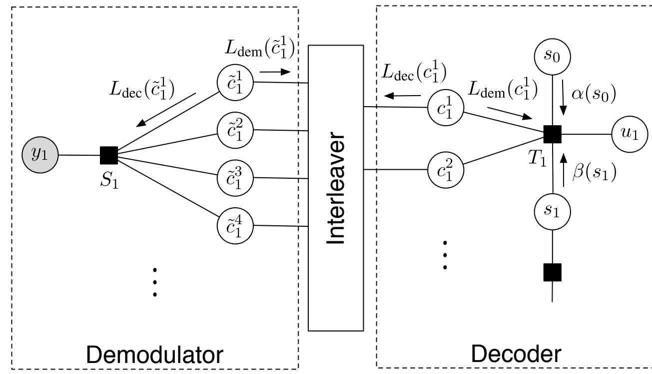

Fig. 2. *Factor graph representation of BICM–ID.*

output relation

$$y_k = h_k\, x_k + w_k, \qquad k = 1, \ldots, K.$$

Here, y_k is the received symbol, h_k is the fading coefficient and w_k is additive white Gaussian noise with variance N_0.

Receiver. The optimum receiver minimizing the block error probability is an ML receiver that jointly takes into account the channel code and the symbol mapping. Due to its high computational complexity such an ML receiver is practically infeasible. Therefore, practical BICM receivers use a suboptimal divide-and-conquer strategy that separates the demodulation and the channel decoding. The performance loss of this approach can be partly compensated by feeding back (soft) information from the decoder to the demodulator. This results in a BICM with iterative decoding (BICM–ID) scheme that is illustrated in Fig. 1. Here, the demodulator first computes extrinsic LLRs $\tilde{L}_{k,l}^{\text{dem}}$ for the interleaved code bits from the channel observations y_k. The LLR sequence is then deinterleaved according to $(i,j) = \Pi^{-1}(k,l)$, resulting in the permuted sequence $L_{i,j}^{\text{dem}} = \tilde{L}_{\Pi(i,j)}^{\text{dem}}$ which is passed on to the channel decoder; the decoder in turn calculates extrinsic LLRs $L_{i,j}^{\text{dec}}$ for the J code bits associated with each information bit u_i. The LLRs are re-interleaved and used as a priori information for the demodulator in the next iteration (in the first iteration, the demodulator uses $\tilde{L}_{k,l}^{\text{dec}} \equiv 0$). Through this iterative exchange of soft information, the information bit LLRs obtained after convergence represent an approximate solution to the decoding problem. This approximate solution is typically very close to the ML solution and hence achieves low bit error rate (BER).

2.2. Factor Graph Representation

Basic Principle. Factor graphs represent another way to study BICM–ID receivers [12, 13]. A factor graph captures the structure of the joint probability distribution (in other words, the statistical dependencies) of all variables involved. The decoding problem can be viewed as the calculation of the information bit probabilities, i.e., a marginalization of the joint distribution. This is usually accomplished by running the sum-product algorithm (i.e., loopy belief propagation, message passing) on the factor graph (cf. [12]). In our case, this amounts to updating the LLRs for the code bits and the information bits. A fundamental degree of freedom here is the scheduling of the message updates. Factor graphs have the advantage of capturing the receiver operation in a more fine-grained way than a turbo block diagram. This is useful since it allows to understand the receiver in more detail and think of modifications that might lead to algorithm improvements.

Fig. 2 shows a part of the factor graph for a BICM–ID receiver with rate $R = 1/2$ convolutional code (i.e., $J = 2$ code bits per information bit) and a 16-QAM symbol constellation ($L = 4$ bits per symbol). The left-hand side corresponds to the (de)modulator and the right-hand side to the (de)coder. These two parts are connected via the (de)interleaver. In the modulator, factor nodes S_k, $k = 1, \ldots, K$ (depicted as a black squares) connect the variable nodes $\tilde{c}_{k,1}, \ldots, \tilde{c}_{k,l}$, and y_k (depicted as white circles) and represent the constraints imposed by the modulator and the channel. The demodulator computes (outgoing) code bit LLRs $\tilde{L}_{k,l}^{\text{dem}}$ based on the (incoming) code bit LLRs $\tilde{L}_{k,l}^{\text{dec}}$ and the constraint node S_k. The decoder block consists mainly of a Markov chain of states s_i that reflects the trellis structure; here, trellis factor nodes T_i, $i = 1, \ldots, I$, describe how the previous state s_{i-1} and the current information bit u_i are mapped to the code bits $\mathbf{c}_i = (c_i^1, \ldots, c_{i,J})$ and the subsequent state s_i by the encoder mapping ω, i.e.,

$$T_i(u_i, s_{i-1}, s_i, \mathbf{c}_i) = \begin{cases} 1, & (s_i, \mathbf{c}_i) = \omega(s_{i-1}, u_i), \\ 0, & \text{else.} \end{cases}$$

The decoder processes the code bit LLRs $L_{i,j}^{\text{dem}}$ from the demodulator and combines those with the probabilities of the current state $\alpha(s_i)$ and of the subsequent state $\beta(s_i)$ to obtain improved LLRs $L_{i,j}^{\text{dec}}$.

LLR Updates. We next discuss in more detail the LLR updates at the factor nodes. Using the max-log approximation, the demodulator updates (assuming perfect channel state information) amount to computing the extrinsic LLRs

$$\tilde{L}^{\text{dem}}_{k,l} = \min_{x:\tilde{c}_{k,l}=1} \psi_{k,l}(x) + \min_{x:\tilde{c}_{k,l}=0} \psi_{k,l}(x) \tag{1}$$

with the demodulator metric

$$\psi_{k,l}(x) = \frac{1}{N_0}\big|y_k - h_k x\big| + \frac{1}{2}\sum_{l'\neq l}(2c_{l'}(x)-1)\,\tilde{L}^{\text{dec}}_{k,l'}, \tag{2}$$

where $c_l(x)$ denotes the lth bit in the bit label of x.

For the message updates at the decoder, the code bit LLRs are mapped to probabilities according to

$$\gamma(c_{i,j}) = \frac{1}{1+\exp\big((1-2c_{i,j})L^{\text{dem}}_{i,j}\big)}.$$

Furthermore, the forward and backward probabilities of the states s_i are determined as[1]

$$\alpha(s_i) = \sum_{\sim s_i} T_i(u_i, s_{i-1}, s_i, \mathbf{c}_i)\,\alpha(s_{i-1})\prod_{j=1}^{J}\gamma(c_{i,j}), \tag{3}$$

$$\beta(s_{i-1}) = \sum_{\sim s_{i-1}} T_i(u_i, s_{i-1}, s_i, \mathbf{c}_i)\,\beta(s_i)\prod_{j=1}^{J}\gamma(c_{i,j}). \tag{4}$$

The code bit LLRs then equal $L^{\text{dec}}_{i,j} = \log\frac{\bar{\gamma}(c_{i,j}=1)}{\bar{\gamma}(c_{i,j}=0)}$ with

$$\bar{\gamma}(c_{i,j}) = \sum_{\sim c_{i,j}} T_i(u_i, s_{i-1}, s_i, \mathbf{c}_i)\,\alpha(s_{i-1})\,\beta(s_i)\prod_{j'\neq j}\gamma(c_i^{j'}),$$

After the receiver iterations are terminated, the information bit LLRs are obtained as $L_i = \log\frac{\phi(u_i=1)}{\phi(u_i=0)}$ where

$$\phi(u_i) = \sum_{\sim u_i} T_i(u_i, s_{i-1}, s_i, \mathbf{c}_i)\,\alpha(s_{i-1})\,\beta(s_i)\prod_{j=1}^{J}\gamma(c_i^{j}),$$

We emphasize that L_i and $\phi(u_i)$ have to be computed only once at the end of the decoding process.

Schedule. The classical BICM–ID receiver described in Section 2.1 can be viewed as message passing scheme on the factor graph according to the following message schedule. First, all demodulator factor nodes T_k, $k = 1, \ldots, K$, compute the LLRs (i.e., "messages") $\tilde{L}^{\text{dem}}_{k,l}$ in (1), which are passed to the decoder factor nodes S_i via the interleaver ($L^{\text{dem}}_{i,j}$ denotes the messages after de-interleaving). Then, the decoder runs the BCJR algorithm [14] which performs the forward and backward recursions (3), (4) sequentially for all states s_i and then determines the decoder output LLRs $L^{\text{dec}}_{i,j}$ which are re-interleaved and passed to the demodulator nodes T_k. This completes one decoding iteration.

While this is just one possible schedule, apparently no other types of schedules for iterative decoding of BICM have been considered thus far. Alternative schedules have the potential of faster convergence and better approximation of the optimal ML solution; see [15] and [16] for examples that respectively consider sequential scheduling in the context of general belief propagation and LDPC decoding. The selective LLR update scheme we propose in the next section can be interpreted as an *adaptive* message passing schedule; our primary goal in this paper, however, is complexity reduction and not performance improvement. The latter is a natural next step for further research.

[1] The notation $\sum_{\sim x}$ indicates that the summation is with respect to all variables except x.

3. SELECTIVE LLR UPDATE

The computational complexity (measured in number of LLR updates) of the demodulator and decoder in a BICM–ID receiver can be prohibitive. From (1) we see that the calculation of the messages $\tilde{L}^{\text{dem}}_{k,l}$ is quite costly for higher-order constellations, and it becomes even higher in the case of a MIMO systems. The complexity of the decoder essentially depends on the code rate and on the number of states. Several techniques for reducing the overall complexity of BICM–ID have been proposed (see the introduction). Our proposed approach to reduce the BICM–ID complexity is based on the observation that LLRs that are large in magnitude correspond to almost certain knowledge about the corresponding bit and hence need not be further updated; this idea has previously been applied to the design of a low-complexity MIMO-IDMA receiver [9], and in the context of LDPC decoding [10, 11]. We adapt this approach to a BICM–ID system in order to reduce the number of LLR/message updates both in the demodulator and the decoder.

Our proposed scheme, termed "selective LLR update", consists of the following steps:

1) Initialization: The demodulator first calculates all code bit LLRs $\tilde{L}^{\text{dem}}_{k,l}$ according to (1) and (2) using the initialization $\tilde{L}^{\text{dec}}_{k,l} \equiv 0$. The de-interleaved LLR sequence $L^{\text{dem}}_{i,j} = \tilde{L}^{\text{dem}}_{\Pi^{-1}(i,j)}$ is passed to the decoder which runs the BCJR algorithm once to obtain the LLRs $L^{\text{dec}}_{i,j}$.

2) Update Criterion: The code bit LLRs which are going to be updated in the next iteration are characterized by the index set (a justification for this is provided below)

$$\mathcal{U} = \left\{ i \,\middle|\, \sum_{j=1}^{J}\big|L^{\text{dec}}_{i,j}\big| \leq L_{\text{th}} \right\} \times \{1, \ldots, J\}. \tag{5}$$

Here, L_{th} is an "LLR update threshold" (this threshold can either be fixed or may change in the course of the receiver iterations).

3) LLR Updates: The demodulator uses the interleaved decoder LLRs $\tilde{L}^{\text{dec}}_{k,l} = L^{\text{dec}}_{\Pi^{-1}(k,l)}$ to perform the LLR updates (1), (2) for the code bits $\tilde{c}_{k,l}$ with $(k,l) \in \widetilde{\mathcal{U}} =$

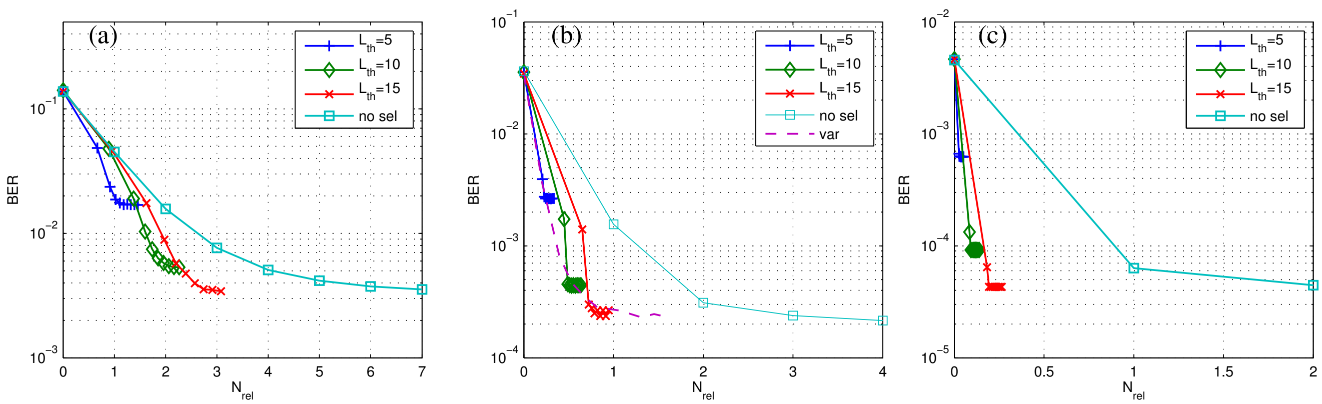

Fig. 3. *BER versus complexity of selective LLR updates for (a) $E_b/N_0 = 6$ dB, (b) $E_b/N_0 = 8$ dB, (c) $E_b/N_0 = 10$ dB.*

$\Pi\,\mathcal{U}$. The rest of the demodulator LLRs are no longer updated, i.e., their value remains unchanged. After performing the forward/backward recursion for $\alpha(s_i)$ and $\beta(s_i)$, the decoder also updates only the code bit LLRs $L_{i,j}^{\text{dec}}$ for which $(i,j) \in \mathcal{U}$ and leaves the other LLRs (i.e., those satisfying the inequality in (5)) unchanged.

4) Termination: If a given stopping criterion is satisfied, the algorithm calculates the information bit LLRs L_i and terminates. Otherwise, the algorithm goes back to step 2 and determines a new update set $\mathcal{U}$ according to (5).

Our method reduces the receiver complexity by performing only $2|\mathcal{U}|$ instead of $2IJ$ LLR updates per iteration. With a fixed LLR threshold L_{th}, the update set $\mathcal{U}$ becomes smaller after each pass of LLR updates (all LLRs not belonging to $\mathcal{U}$ remain frozen forever); hence, in this case the complexity (number of LLR updates) per iteration is monotonically decreasing (although not strictly) with the iteration count.

The update criterion (5) can be justified as follows. The final information bit decisions and their reliabilities are respectively given by the signs and magnitudes of the information bit LLRs L_i. Hence, the decision whether to update or not specific LLRs should ideally be based on the LLR magnitude $|L_i|$. However, such a direct approach is impractical since the information bit LLRs need to be calculated only once, at the end of the receiver iterations; calculating the information bit LLRs would actually increase the complexity. Hence, we estimate the reliability of the information bit LLRs in terms of the code bit LLRs according to the sum $\sum_{j=1}^{J} |L_{i,j}^{\text{dec}}|$; if this sum is large, our uncertainty about the respective information bit tends to be low and hence updating the code bit LLRs becomes less useful. This heuristic reliability measure was observed to result in excellent performance (see Section 4).

Our selective LLR update scheme can alternatively be interpreted as an alternative message passing schedule on the associated factor graph in which unreliable code bit LLRs are updated while reliable ones remain frozen (temporarily or forever). The unusual aspect of this message schedule is its *adaptive* nature, i.e., the order and frequency of the LLR updates is not prescribed but depends on the specific realization of channel and noise.

4. SIMULATION RESULTS

We next provide numerical simulations illustrating the complexity savings and performance of the proposed selective LLR update scheme. We used a (13,15) convolutional code of rate $R = 1/2$ to encode $I = 512$ information bits into $I/R = 1024$ code bits. The code bits were passed through a random interleaver and modulated to 16-QAM symbols using set partition mapping [17]. A fast Rayleigh fading channel was simulated via i.i.d. complex Gaussian channel coefficients $h_k \sim \mathcal{CN}(0,1)$. The signal-to-noise ratio is defined as $\text{SNR} = \frac{E_b}{N_0}$, where $E_b = \text{E}\{|x_k|^2\}/(LR)$ denotes the energy per information bit and N_0 is the noise variance.

To measure algorithm complexity, we use the quantity N_{rel} which equals the total number of LLR updates after initialization, normalized by twice the block length. For conventional BICM–ID, N_{rel} is always integer and equals the number of turbo iterations minus one. In contrast, our algorithm can realize fractional N_{rel}. Fig. 3 shows BER versus complexity N_{rel} for SNRs of 6, 8, and 10 dB (markers indicate one pass of LLR updates). The different curves correspond to conventional BICM–ID (labeled 'no sel') and our selective LLR update with $L_{\text{th}} = 5$ ('+'), $L_{\text{th}} = 10$ ('$\diamond$'), and $L_{\text{th}} = 15$ ('$\times$'). It is seen that for small LLR thresholds the BER initially decreases faster but also levels off at high values while for large LLR thresholds a lower BER saturation level is payed for by a slower convergence rate. An LLR threshold that gradually increases from 5 to 15 (labeled 'var' in Fig. 3(b)) leads to fast initial convergence and low BER saturation level. Compared to conventional BICM–ID, our scheme can realize enormous complexity savings. As an example consider the practically relevant case of SNR $= 10$ dB, where conventional BICM–ID requires $N_{\text{rel}} = 2$ iterations (4096 LLR updates) after ini-

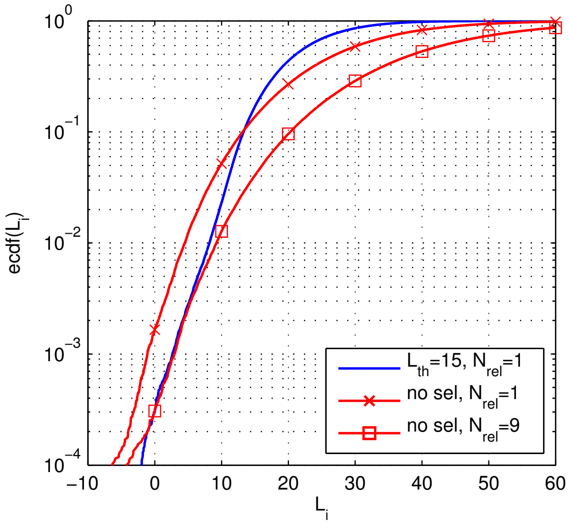

Fig. 4. *Empirical cdfs of information bit LLRs.*

tialization to achieve a BER of $4.4 \cdot 10^{-5}$. Our proposed algorithm achieves the same BER with only $N_{\text{rel}} = 0.2$ (400 LLR updates), amounting to a complexity reduction by a factor of about 10. At lower SNRs, complexity is reduced by factors of about 5 (at 8 dB) and 2.5 (at 6 dB).

The reason why much fewer updates can lead to identical BER performance can be understood from the distribution of the information bit LLRs which are used for the final bit decisions. Fig. 4 shows the empirical cumulative probability distribution function (ecdf) of the information bit LLRs L_i conditioned on $u_i = 1$ at an SNR of 8 dB for BICM–ID with $N_{\text{rel}} = 1$ (dashed line) and $N_{\text{rel}} = 9$ (dash-dotted) and for our selective update scheme with $N_{\text{rel}} = 1$ (solid). It is seen that our scheme tends to produce smaller LLR magnitudes (e.g., the frequency of LLRs less than 20 equals 44% for our scheme but only 26% and 10% for conventional BICM–ID with $N_{\text{rel}} = 1$ and $N_{\text{rel}} = 9$, respectively). However, our selective update simultaneously decreases the number of negative LLRs which correspond to bit errors, i.e., the frequency of LLRs below 0 (which equals the BER) is about $3 \cdot 10^{-4}$ for our scheme with $N_{\text{rel}} = 1$ and for BICM–ID with $N_{\text{rel}} = 9$. This confirms the usefulness of our reliability measure in (5). Apparently, conventional BICM–ID spends a lot of computational effort to generate large LLRs that do not improve the BER.

5. CONCLUSIONS

We proposed a pragmatic selective LLR update scheme for BICM–ID systems with the goal of reducing computational complexity. This scheme can be interpreted as an adaptive schedule for loopy belief propagation on the BICM factor graph. Simulation results revealed that a significant complexity reduction can be achieved without performance loss. While this paper is intended as a proof of concept, our future research will be concerned with an analytical treatment of the convergence properties of the scheme and the derivation of design guidelines for the LLR threshold. Instead of updating a variable number of LLRs whose magnitude lies below a given threshold, a practically interesting alternative may be to update the smallest (in magnitude) U LLRs, with U a fixed number determined by hardware constraints.

ACKNOWLEDGEMENTS

We thank Clemens Novak for sharing his ideas about selective LLR updates and Florence Alberge for helpful discussions.

REFERENCES

[1] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 873–884, May 1992.

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

[3] X. Li and J. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 1, no. 6, pp. 169–171, 1997.

[4] S. ten Brink, J. Speidel, and R.-H. Han, "Iterative demapping for QPSK modulation," *Electronics Letters*, vol. 34, no. 15, pp. 1459–1460, Jul. 1998.

[5] F. Tosato and P. Bisaglia, "Simplified soft-output demapper for binary interleaved COFDM with application to HIPERLAN/2," in *Proc. IEEE Int. Conf. Communications*, vol. 2, May 2002, pp. 664–668.

[6] E. Akay and E. Ayanoglu, "Low complexity decoding of bit-interleaved coded modulation for M-ary QAM," in *Proc. IEEE Int. Conf. Communications*, vol. 2, Paris, France, Jun. 2004, pp. 901–905.

[7] E. Biglieri, G. Taricco, and E. Viterbo, "Bit-interleaved time-space codes for fading channels," in *Proc. Conf. on Information Sciences and Systems*, Princeton, NJ, Mar. 2000, pp. WA 4.1–4.6.

[8] C. Studer and H. Bölcskei, "Soft-input soft-output single tree-search sphere decoding," *submitted to IEEE Transactions on Information Theory*, Jun. 2009.

[9] C. Novak, F. Hlawatsch, and G. Matz, "Low-complexity factor graph receivers for spectrally efficient MIMO-IDMA," in *Proc. IEEE ICC*, Beijing, China, Apr. 2008, pp. 770–774.

[10] E. Zimmermann, P. Pattisapu, P. Bora, and G. Fettweis, "Reduced complexity LDPC decoding using forced convergence," in *Proc. 7th International Symposium on Wireless Personal Multimedia Communications*, vol. 3, Abano Terme, Italy, Sep. 2004, pp. 243–246.

[11] A. Blad, O. Gustafsson, and L. Wanhammar, "An early decision decoding algorithm for LDPC codes using dynamic thresholds," in *Proc. European Conference on Circuit Theory and Design*, vol. 3, Cork, Ireland, Aug. 2005, pp. 285–288.

[12] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[13] A. G. I. Fàbregas, A. Martinez, and G. Caire, "Bit-interleaved coded modulation," *Foundations and Trends in Communications and Information Theory*, vol. 5, no. 1-2, pp. 1–153, Jan. 2007.

[14] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate (corresp.)," *IEEE Transactions on Information Theory*, Jan. 1974.

[15] G. Elidan, I. McGraw, and D. Koller, "Residual belief propagation: Informed scheduling for asynchronous message passing," in *Proc. Conf. Uncertainty in Artificial Intelligence*, vol. 6, no. 4, Cambridge, USA, Jul. 2006, pp. 6–4.

[16] A. Vila Casado, M. Griot, and R. Wesel, "Informed dynamic scheduling for belief-propagation decoding of LDPC codes," in *Proc. IEEE Int. Conf. Communications*, Glasgow, Scotland, Jun. 2007, pp. 932–937.

[17] A. Chindapol and J. A. Ritcey, "Design, analysis, and performance evaluation for BICM-ID with square QAM constellations in Rayleigh fading channels," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 5, pp. 944–957, May 2001.