Performance Study of Non-Binary Belief Propagation for Decoding Reed-Solomon Codes

Bimberg, Marcel; Lentmaier, Michael; Fettweis, Gerhard

Published in:
2010 International ITG Conference on Source and Channel Coding (SCC)

2010

Performance Study of Non-Binary Belief Propagation for Decoding Reed-Solomon codes

Marcel Bimberg, Michael Lentmaier, Gerhard P. Fettweis
Vodafone Chair Mobile Communications Systems
Technische Universität Dresden
D-01069 Dresden, Germany
Email: {bimberg, lentmaier, fettweis}@ifn.et.tu-dresden.de

***Abstract*—In this paper we investigate the frame-error correcting performance of two newly developed, non-binary belief propagation based, soft-in/soft-out decoding algorithms when decoding Reed-Solomon codes. We present results for the AWGN channel indicating that non-binary belief propagation can come close to the performance of the binary adaptive Belief-Propagation algorithm or even slightly better when decoding short Reed-Solomon codes.**

I. Introduction

Since their discovery in 1960 [1], Reed-Solomon (RS) codes have been probably the most widely applied error-correction codes (ECCs) in many digital communications and recording systems. Besides numerous applications in the past, RS codes are still incorporated into today's state-of-the-art communications systems such as WiMAX, DVB, DAB and the newly developed WirelessHD standard [3]. Due to their ability to correct burst errors and the existence of efficient hard-decision based algebraic en- and decoding algorithms, RS codes are still favored over other codes in environments where delay sensitive services combined with robust communication become necessary. Moreover, in order to cope with low latency requirements in multimedia networks, PHY headers and control messages of MAC protocols demand the usage of shorter block lengths. With their property of being maximum-distance-separable (MDS) codes, RS codes therefore provide good error-correcting performance also at relatively small block lengths.

As the rediscovery of low-density parity-check (LDPC) codes in the early 90's showed, the use of long block codes (in order of tens of thousands bits) combined with iterative, message-passing algorithms allows to asymptotically approach the capacity of the AWGN channel. However, applying the standard belief propagation (BP) decoding to RS codes leads to poor error-correcting capability due to many short cycles in the high-density parity-check (HDPC) matrices. In order to improve the performance of BP also for HDPC matrices, Jiang and Narayanan proposed in [4] the iterative, adaptive belief propagation (ABP) algorithm which operates in $GF(2)$. This algorithm compares favorably with other soft-decision decoding algorithms and can be regarded as a fundamental step towards message passing decoding of RS codes. El-Khamy and McEliece later concatenated in [6] ABP with the Koetter-Vardy [7] algebraic soft-decision decoding (ASD) algorithm, which enabled them to achieve near optimal performance for relatively short, high-rate codes. In [8] a combination of ABP and the ordered-statistics-decoding (OSD) was used to improve the error correcting capability of medium length codes. However, except for the original proposed ABP decoder all of them are restricted to provide only hard decisions as output information (SIHO).

In this paper, we investigate two newly developed soft-in/soft-out (SISO) decoding algorithms. One of them utilizes the matrix adaption step while performing belief propagation decoding in higher order Galois fields (GF-ABP). It has been demonstrated in [10] that GF-BP based decoding can improve error correcting performance compared to binary BP algorithms especially for shorter LDPC codes. As bits are grouped together in the GF-BP decoding approach, it is quite reasonable to assume that GF-BP decoding can perform better than binary decoding on channels with noise bursts. In a second approach we extend the idea of multiple-bases belief propagation (MBBP) [12] also to higher order Galois fields (GF-MBBP). Utilizing MBBP could especially become useful in next-generation's many-core based wireless communications systems as the decoding algorithm can be mapped easily on several computing cores.

The remainder of this work is organized as follows. Some preliminaries including the system model are given in Section II. The investigated algorithms are presented in Section III. Simulation results and discussions are provided in Section IV-B. Finally, Section V concludes the paper and suggests further research directions.

II. Preliminaries

A. Notation and properties of RS codes

Let $RS(N, K)$ be a Reed-Solomon code which is defined over a finite field $GF(2^q), q \in \mathbb{N}$ and let β be a primitive element of the field. K represents the number of information symbols, while the block length N is equal to $N = 2^q - 1$. Let $\mathbf{m} = [m_1, m_2, \ldots, m_K]$ be a message of K information symbols. These symbols can be associated with an information polynomial

$$m(x) = m_1 + m_2 x + \ldots + m_K x^{K-1}, \tag{1}$$

which is encoded through multiplication by a generator polynomial

$$g(x) = \prod_{j=1}^{N-K} (x - \beta^j) \tag{2}$$

resulting in the polynomial $c(x) = m(x)g(x)$. By this definition, each codeword $\mathbf{c} = [c_1, c_2, \ldots, c_N], c_i \in GF(2^q)$ is interpreted as a code polynomial $c(x)$. The decoder verifies the validity of $\mathbf{c}$ by evaluating the well known parity-check equation $\mathbf{c}\mathbf{H}_q^T = \mathbf{0}$, with $\mathbf{H}_q$ being an $(N-K) \times N$ dimensional matrix consisting of $(N-K)$ codewords spanning the dual code of $RS(N, K)$:

$$\mathbf{H}_q = \begin{bmatrix} 1 & \beta & \ldots & \beta^{(N-1)} \\ 1 & \beta^2 & \ldots & \beta^{2(N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \beta^{(N-K)} & \ldots & \beta^{(N-1)(N-K))} \end{bmatrix}. \tag{3}$$

B. System model

Before describing the investigated decoding algorithms formally, we first assume that the encoded symbols c_j, $(j = 1, \ldots, N)$ are modulated using antipodal BPSK. Therefore, each symbol $c_j \in GF(2^q)$ is mapped into q binary symbols (we assume normal basis representation for which the transformation can be found, e.g., in [9]). The binary symbols are transmitted over an AWGN channel (Fig. 1). On the receiver side, the input values can be specified by:

$$r_{j,t} = s_{j,t} + n_{j,t}, \quad (t = 1, \ldots, q) \tag{4}$$

with n_j being real valued additive white Gaussian noise.

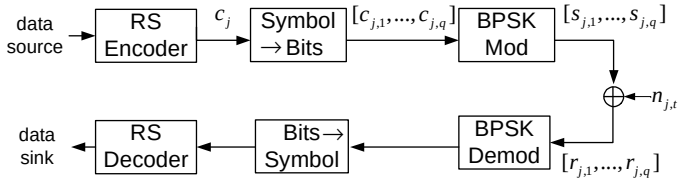

Fig. 1. System Model

C. Non-binary belief propagation - GF-BP

The binary BP algorithm as originally proposed in [2] has found numerous application in the decoding of linear block codes with sparse graph representations, as, e.g., LDPC codes. In [14] it was shown that BP decoding for LDPC codes can approach the Shannon limit for large block lengths. For shorter ones an error-correcting improvement was first observed in [10] by extending the BP algorithm to non-binary LDPC codes. However, this improvement was achieved at the expense of increased decoding complexity. In [11] a reduced complexity GF-BP algorithm in the order of $O(q \log_2 q)$ was therefore proposed, which is based on Fast-Fourier-Transformation (FFT), enabling efficient decoding even of non-binary LDPC codes defined over very large order Galois fields. In the following the Fourier transform decoding algorithm is described.

Suppose a non-binary LDPC code is given, having an $(N-K) \times N$ parity-check matrix $\mathbf{H}_q$. According to [11], $\mathbf{H}_q$ can be described by a graph consisting of N variable and $N-K$ check nodes that are connected with each other by edges if the corresponding entry $h_{ij} \in GF(2^q)$ (i being row, j being column index) in $\mathbf{H}_q$ is non-zero. Decoding can be accomplished by iteratively exchanging reliability information (called messages) between variable and check nodes. In contrast to the binary algorithm each message is now a vector representing a 2^q-point discrete probability set rather than a single value. Furthermore, these vectors are now permuted and reordered each time they are sent from variable to check nodes and vice versa (Fig. 2). Check node messages $\mathbf{R}_{i\to j}^{(l)}$

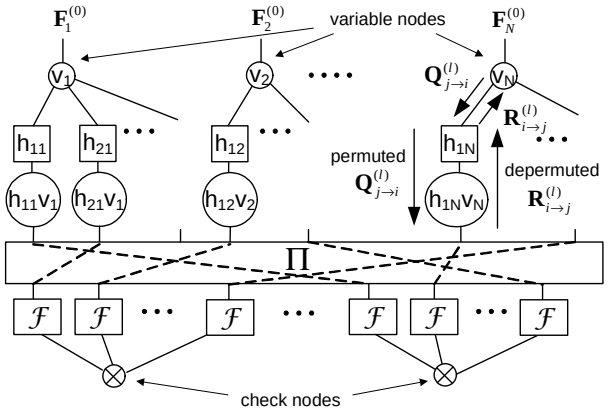

Fig. 2. Graph of a non-binary LDPC code

and variable node messages $\mathbf{Q}_{j\to i}^{(l)}$ passed along the edges are calculated in two stages according to the following decoding equations:

1) **Check node update:**

a) Permutation and Fourier transform:

$$\tilde{\mathbf{R}}_{i\to j}^{(l+1)} = \prod_{j' \in V_{i\setminus j}} \text{FFT}[P(\mathbf{Q}_{j'\to i}^{(l)})] \tag{5}$$

b) Inverse Fourier transform and depermutation

$$\mathbf{R}_{i\to j}^{(l+1)} = P^{-1}(\text{IFFT}[\tilde{\mathbf{R}}_{i\to j}^{(l+1)}]) \tag{6}$$

2) **Variable node update:**

$$\mathbf{Q}_{j\to i}^{(l+1)} = \gamma_{ij}\, \mathbf{F}_j^{(0)} \prod_{i' \in C_{j\setminus i}} \mathbf{R}_{i'\to j}^{(l+1)} \tag{7}$$

Notation:

l	count index for GF-BP iterations
$\mathbf{F}_j^{(0)}$	initial prior probability vector of symbol j
γ_{ij}	normalization coefficient
$\mathbf{Q}_{j\to i}$	message vector passed from variable node j to check node i
$\mathbf{R}_{i\to j}$	message vector passed from check node i to variable node j
$V_{i\setminus j}$	set of all variable nodes connecting to check node i except for node j
$C_{j\setminus i}$	set of all check nodes connecting to variable node j except for node i
$P(.)/P^{-1}(.)$	permutation / inverse permutation

Due to the structure of the Galois fields, the permutation block $P(\mathbf{Q}_{j'\to i}^{(l)})$ can actually be implemented by cyclically shifting downwards the column vector $\mathbf{Q}_{j'\to i}^{(l)}$, with the exception of the first likelihood, which corresponds to the probability of the coded symbol c_j being zero. The number of cyclic shifts is equal to the power of the primitive element that corresponds to the entry h_{ij}. The $\prod(.)$-operation then performs the term-by-term multiplication of the Fourier transformed probability values, where FFT(.) is a q-dimension two-point FFT. After inverse transformation and depermutation (cyclically upshifting) the results are sent back to the variable nodes.

Before the algorithm iterates on a matrix $\mathbf{H}_q$, variable node messages $\mathbf{Q}_{j\to i}^{(l)}$ are initialized by the current symbol-reliability values. In the first iteration of GF-BP decoding these probabilities are given by the channel values:

$$\mathbf{F}_j^{(0)} = \begin{bmatrix} p(v_j|c_j=0) \\ p(v_j|c_j=\beta^0) \\ \vdots \\ p(v_j|c_j=\beta^{2^q-2}) \end{bmatrix}. \tag{8}$$

As we assume BPSK transmission and all transmitted bits being independent, they can be computed by:

$$p(v_j|c_j=x) = \prod_{t=1}^{q} p(r_{j,t}|c_{j,t}=x_t) \tag{9}$$

$$p(r_{j,t}|c_{j,t}=1) \propto e^{-\frac{(r_t-\alpha)^2}{2\sigma^2}}, \qquad p(r_{j,t}|c_{j,t}=0) \propto e^{-\frac{(r_t+\alpha)^2}{2\sigma^2}} \tag{10}$$

with x_t being the tth bit of the binary representation of $x, x \in GF(2^q)$. The value of α is assumed to be known to the receiver. During the last iteration (or meanwhile each variable node update) posterior symbol-reliability values are calculated according to:

$$\mathbf{F}_j^{(k+1)} = \gamma_j \, \mathbf{F}_j^{(k)} \prod_{i'\in C_j} \mathbf{R}_{i'\to j}^{(l+1)}. \tag{11}$$

The normalization factor γ_j ensures that

$$\sum_{x\in \mathrm{GF}(2^q)} p(v_j|c_j=x) \doteq 1. \tag{12}$$

Based on these posterior likelihoods a tentative decoding decision can be made such that

$$\hat{c}_j^{(k+1)} = \arg \max_{x\in \mathrm{GF}(2^q)} p(v_j|c_j=x). \tag{13}$$

If $\hat{\mathbf{c}}\mathbf{H}_q^T = \mathbf{0}$ decoding stops and outputs $\hat{\mathbf{c}}$; otherwise a new iteration starts until a valid codeword is found or a maximum number l_{max} of iterations is reached.

III. Iterative decoding of Reed-Solomon codes utilizing non-binary belief propagation

A. Non-binary adaptive belief propagation - GF-ABP

Due to the high density that parity-check matrices of RS codes possess, off-the-shelf BP based decoding leads to poor error-correcting performance as the large number of short cycles counteracts the independence of messages exchanged between variable and check nodes. To overcome this problem, in [4] a modified version of the binary BP algorithm - namely ABP - was proposed. Using ABP, the binary parity-check sub-matrix corresponding to the $(N-K)q$ least reliable bits (LRBs) in $\mathbf{H}_b$ is reduced into an identity matrix before standard BP decoding is applied. Hence, error propagation from the unreliable bits is effectively reduced, which facilitates ABP decoding also to improve the decoding performance of linear block codes having HDPC matrices. Here we extend the idea of adaptively changing the parity-check matrix to the non-binary BP algorithm.

As pictured in Fig. 3 we are now separating between inner GF-BP iterations indexed by l and outer matrix adaption steps indexed by k. The maximum number of iterations therefore amounts to $k_{max}l_{max}$. Before GF-BP is going to be applied, the kth parity-check matrix $\mathbf{H}_q^{(k)}$ is adapted in order to minimize the adverse influence of less reliable symbols onto belief propagation. This is accomplished by diagonalization of the sub-matrix corresponding to the $N-K$ least reliable symbols (LRS), (as the matrix $\mathbf{H}_q$ has full rank it is always possible to eliminate exactly $N-K$ columns). Each of the LRS therefore participates only in one parity-check equation, which improves extrinsic information exchange originating from the more reliable symbols. One general method to perform diagonalization is to employ standard Gaussian-elimination over $GF(2^q)$ having a polynomial complexity of $O((N-K)N^2)$. A second, and more powerful method in particular for RS codes, is to employ erasure decoding of the dual code $RS^{\perp}(N, N-K)$ of a $RS(N,K)$ code. As we know $N-K$ positions in each of the $N-K$ parity check rows (corresponding to the LRS part), the missing K positions can be determined using the dual. Hence, each codeword found by dual code can serve as a row in the parity-check matrix of $RS(N,K)$. This method provides a beneficial way for high throughput implementations since all rows can be determined in parallel employing, e.g., the Forney algorithm [5].

Prior to the matrix adaption step, the set of LRS has to be identified. For this purpose the most likely value of each symbol

$$p_{j,max} = \max_{x\in GF(2^q)} p(v_j|c_j=x) \tag{14}$$

is used for determining the overall order of the N symbols. Employing appropriate sorting algorithms the complexity of this step amounts to $O(N\log_2 N)$. Different from GF-BP presented in Section II-C we introduce two additional damping coefficients δ and θ for GF-ABP. As the GF-BP algorithm is only suboptimal when decoding dense parity-check matrices, the additional parameters δ and θ attenuate the influence of the extrinsic information. This is different from LDPC codes, where both parameters are usually set to one as the GF-BP decoding is considered to be optimal due to ideally nonexisting short cycles.

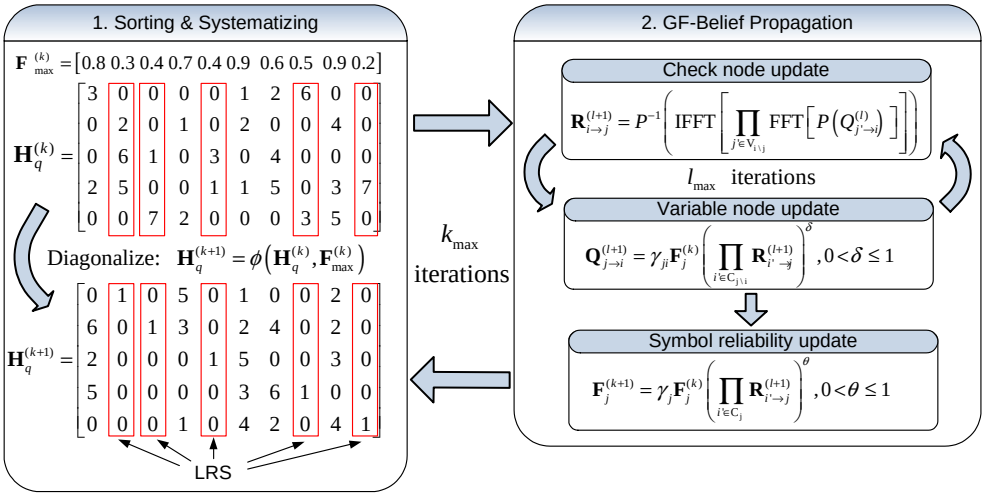

Fig. 3. Non-binary adaptive belief propagation (GF-ABP)

B. Non-binary multiple-bases belief propagation - GF-MBBP

Although from a practical implementation point of view the different decoding steps of GF-ABP can be pipelined, the MBBP approach introduced in [12] appears suitable for efficient parallel implementations in next-generation many-core systems. Instead of adapting $\mathbf{H}_q^{(k)}$ in a sequential manner, MBBP employs BP decoding on k parallel parity-check matrices. From the so generated list of codewords one is finally selected by some metric criterion. In order to distinguish different parity-check matrices [12] introduces the so called cyclic group generators (CGG). A CGG is representative of one group, which allows the generation of all other codewords in this group by multiplication or shifting operations. The CGG description can also be extended to RS codes. As they are cyclic codes, the dual code is also cyclic. This means we can construct an $(N-K)\times N$ full-rank parity-check matrices over $GF(2^q)$ just by cyclically shifting a codeword belonging to the dual of $RS(N,K)$. Each row of the matrix then corresponds to a different shift of the selected codeword. In [12] it was observed that the use of redundant rows can not only be advantageous for the BEC but also for the AWGN channel. We therefore employed $N \times N$ matrices, where each row represents one of the N possible shifts of the CGG. Employing this method equal error protection is assured. Note that other matrices of one group can be generated by multiplying the CGG with elements from $GF(2^q)\backslash\{0\}$. However, for GF-BP decoding purposes they are equivalent with each other, as the multiplication implies a simply common downward/upward shifting operation of the corresponding message vectors participating in a parity-check equation.

The optimal choice of concurrent parity-check matrices for GF-MBBP decoding is an open research problem. Recent results for binary codes in [13] indicate that the stopping set size of a parity-check matrix should be small in order to obtain good error-correcting performance. Nevertheless, finding stopping sets is a difficult task, in particular for codes

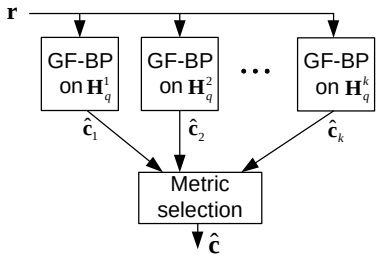

Fig. 4. Non-binary multiple-bases belief propagation (GF-MBBP)

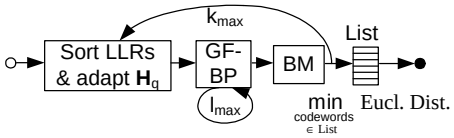

Fig. 5. GF-ABP concatenated with RS hard-decision decoder

defined over $GF(2^q)$. For the codes investigated within this paper, we therefore focused on parity-check matrices generated by CGGs from the dual, having smallest minimum distance $d_{min}^{\perp} = K+1$. Employing this restriction, the constructed matrices have a small number of short cycles, which improves BP decoding. Beside decoding aspects, the use of matrices constructed by CGGs also saves significant amount of storage memory since only one row (corresponding to N symbols in $GF(2^q)$) has to be stored for each matrix. As metric criterion we selected the final codeword to be the most frequent one from the list of codewords provided by the k concurrent GF-BP decoders.

C. Extension by RS hard-decision decoder

In order to further improve decoding performance, [4] suggested to extend the ABP algorithm with an additional, concatenated RS hard-decision decoder, as, e.g., the Berlekamp Massey algorithm. This second decoder operates on an input

list of words assembled from the outputs of the ABP decoder after each BP iteration. The final codeword $\hat{\mathbf{c}}$ is selected from this newly generated list by comparing the Euclidean distances (Fig. 5). Although such a decoder looses the desired soft-output property, we also provide frame error rate curves for a corresponding GF-ABP-BM decoder.

IV. Results and Discussion

A. Cyclic group generators

In order to find the CGGs of a particular RS code, we applied a modified brute-force search to the dual code. Instead of comparing each and every value of a codeword while searching the space of $2^{q(N-K)}$ codewords of the dual, we compared only the positions of the zero values of the current indexed word (for each such zero-pattern one codeword as a representative - which is the CGG - has to be stored). As described in the previous section, all other codewords can be computed from this CGG by shift and/or multiplication operations. Employing this method, we were able to search even the large space of approximately 10^9 codewords for the dual of a $RS(31, 25)$ code within one day. Table I summarizes the results for the $N \times N$ parity-check matrices constructed from these CGGs.

code	rate	#parity-check matrices	#length-4 cycles
RS(7,5)	0.71	1	210
RS(15,11)	0.73	24	4185
RS(31,25)	0.81	546	104160

TABLE I
Constructed parity-check matrices for GF-MBBP

As for the $RS(7, 5)$ only one $N \times N$ matrix exists, we investigated a second approach employing 7 parallel $(N - K) \times N$ matrices. Each of them corresponds to one of the 7 possible shifts of the matrix:

$$\mathbf{H}_q^{(0)} = \begin{bmatrix} 0 & 2 & 7 & 2 & 7 & 5 & 5 \\ 5 & 0 & 2 & 7 & 2 & 7 & 5 \end{bmatrix}, \qquad (15)$$

and posses 10 length-4 cycles.

B. Simulation results

In Fig. 6-8 we compare the FER performance of the ABP, GF-ABP and GF-MBBP decoding algorithms. In all simulations the maximum number of outer iterations k_{max} was set to 50. For the smaller $RS(7, 5)$ code ($\hat{=}$21 bits blocklength) we can observe almost no difference between ABP and GF-ABP when the number of inner iterations is $l_{max} = 1$. Increasing the number of inner iterations to $l_{max} = 5$ leads to an improvement of ≈ 0.2 dB for the GF-ABP approach. Employing the GF-MBBP algorithm on the unique $N \times N$ matrix degrades the performance compared to the GF-ABP approach by ≈ 0.2 dB. However, observe that the 7 parallel $(N - K) \times N$ matrices formed from equation (15) facilitate the same good error-correcting capability as the GF-ABP with $l_{max} = 5$.

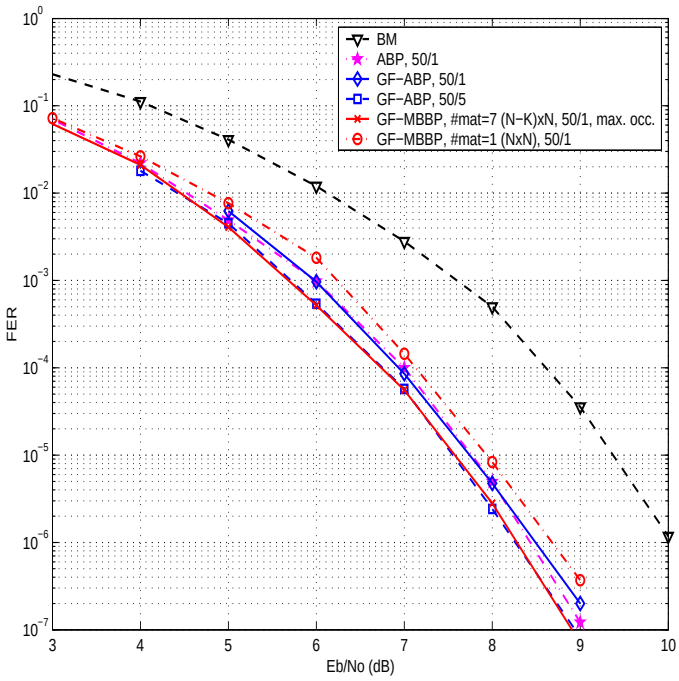

Fig. 6. Comparison of FER for BM, ABP, GF-ABP and GF-MBBP decoding applied to a RS(7,5) code using AWGN channel

Considering the $RS(15, 11)$ code ($\hat{=}$60 bits blocklength), the GF-ABP algorithm is not able to provide the same FER as the binary counterpart in FER regions above 10^{-6}. As can be observed, the extension of ABP by a hard-decision decoder improves the error-correcting capability especially in lower FER regions. Nevertheless, for the GF-ABP extended version (GF-ABP-BM) we could not observe that large improvements. Again, the GF-MBBP approach employing now 24 parallel matrices gives the best FER performance without the usage of an additional second decoder. Employing only half of the matrices still provides comparable results with the ABP decoder. Particularly the promising results in low FER regions give rise to the question if the GF-MBBP decoding will exhibit the same error-floor as it was observed for the ABP algorithm, e.g., in [4].

For the $RS(31, 25)$ code ($\hat{=}$155 bits blocklength) neither the GF-ABP nor the GF-MBBP decoder are able to reach the ABP error-correcting performance. Compared with ABP the coding gain of the GF-ABP decoder decreases by ≈ 1 dB and for the GF-MBBP by ≈ 0.2 dB. Due to long term simulations for the huge amount of 546 concurrent matrices, we are not yet able to provide results in lower FER regions for that code. However, we would like to point out here, that GF-BP based decoding is much faster than the binary case. This can be justified by two facts. First, employing higher order GF-BP demands less permutations for the check-node operations as the absolute number of entries in the parity check matrix is much lower. Secondly, this lower number of permutations facilitates more localized operations like the FFT/IFFT operation on each variable-to-check/check-to-variable node message. For

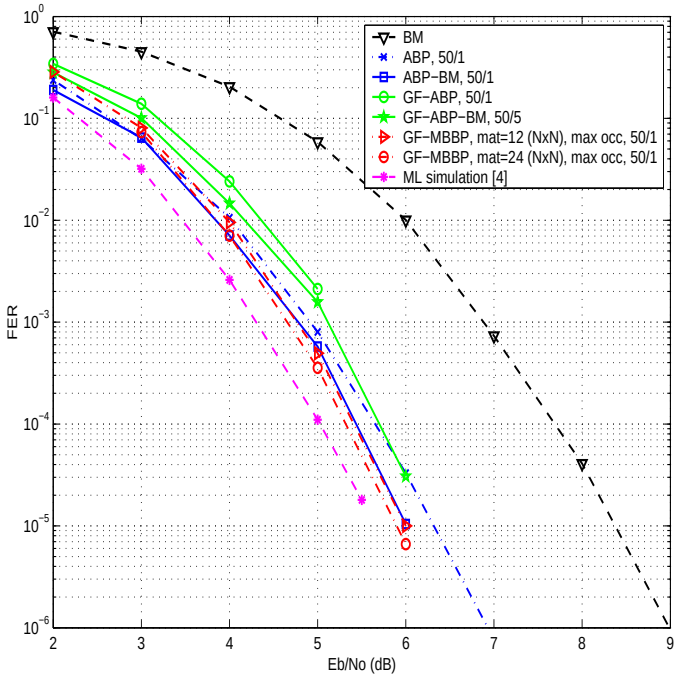

Fig. 7. Comparison of FER for BM, ABP, GF-ABP and GF-MBBP decoding applied to a RS(15,11) code using AWGN channel

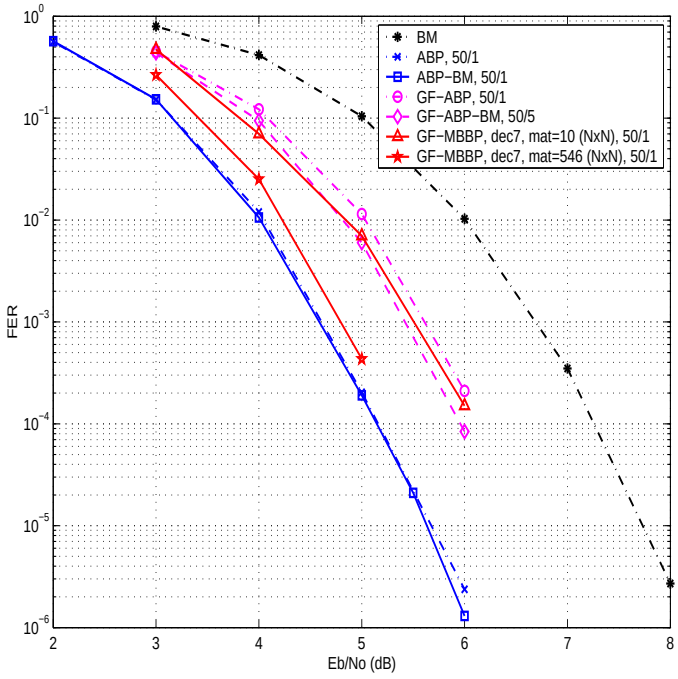

Fig. 8. Comparison of FER for BM, ABP, GF-ABP and GF-MBBP decoding applied to a RS(31,25) code using AWGN channel

practical VLSI implementations this property is quite useful. Instead of routing each message in an individual way as for the binary case, now, message vectors of size 2^q are routed the same way. Hence, the permutation network can be more simplified and structured.

V. Conclusion

In this paper, we investigated the newly developed GF-ABP and GF-MBBP algorithm for the SISO decoding of RS codes. We compared them with the binary ABP decoder and showed that the GF-MBBP approach provides very good results at least for small blocklengths. We believe that the GF-MBBP decoding approach is also very attractive for future many-core systems, as it provides an inherent way for parallelization. Moreover, employing GF-BP based decoding might be an efficient way for coded modulation techniques when non-binary modulation as, e.g, 64-QAM for bandwith efficient communication is desired. Future research directions may also consider the concatenation of the presented decoders with the Koetter-Vardy [7] algebraic soft-decision decoding algorithm.

VI. Acknowledgment

The authors are grateful for the use of the high performance computing facilities of the ZIH at the TU Dresden.

References

[1] I. S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," *Journal of Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300-304, June 1960.

[2] R. Gallager, "Low-Density Parity-Check Codes," Ph.D. Thesis, MIT Press, Cambridge, MA, 1963

[3] WirelessHD Specification Version 1.0, www.wirelesshd.org, 2007

[4] J. Jiang and K. R. Narayanan, "Iterative soft decision decoding of Reed-Solomon codes based on adaptive parity check matrices," *Proc. IEEE Intl. Symp. Inf. Theory,* p. 261, 2004

[5] B. Friedrichs, "Kanalcodierung,", Springer, 1994

[6] M. El-Khamy and R. J. McEliece, "Iterative Algebraic Soft-Decision List Decoding of Reed-Solomon codes," *IEEE Journal on Selected Areas in Communications* , vol. 24, no. 3, March 2006

[7] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory,* vol. 49, no. 11, pp. 2809-2825, Nov. 2003

[8] A. Kothiyal, O. Y. Takeshita, W. Jin, M. Fossorier, "Iterative Reliability-Based Decoding of Linear Block Codes with Adaptive Belief Propagation," *IEEE Comm. Lett.,* vol. 9, no. 12, Dec. 2005

[9] J. Bellorado, "Low-Complexity Soft Decoding Algorithms for Reed-Solomon Codes," PhD thesis, Harvard University, 2006

[10] M. Davey and D. MacKay, "Low density parity check codes over GF(q)," *IEEE Comm. Letters,* vol. 2, no. 6, pp. 165-167, June 1998

[11] L. Barnault and D. Declercq, "Fast decoding algorithm for LDPC over GF(2^q)," *Proc. Inf. Theory Workshop*, pp. 70-73, Mar. 2003

[12] T. Hehn, J. Huber, S. Laendner, O. Milenkovic, "Multiple-bases belief-propagation decoding for short block-codes," *Proc. IEEE Intl. Symp. Inf. Theory,* pp. 311-315, June 2007

[13] T. Hehn, J. Huber, O. Milenkovic, S. Laendner, "Multiple-Bases Belief-Propagation Decoding of High-Density Cyclic Codes," *arXiv:0905.0079v1*, May 2009

[14] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density partiy check codes," *IEE Electron. Lett.*, vol. 32, no. 18, pp. 1645-1646, Aug. 1996.