ON THE PERFORMANCE AND NUMERICAL STABILITY OF SOFT-DECISION REED-SOLOMON DECODING

Marcel Bimberg, Emil Matúš, Gerhard P. Fettweis

Vodafone Chair Mobile Communications Systems
Technische Universität Dresden, D-01069 Dresden, Germany
Email:{bimberg, matus, fettweis}@ifn.et.tu-dresden.de

ABSTRACT

In this paper we numerically analyze soft-decision Reed-Solomon decoding based on adaptive Belief Propagation (ABP) algorithm in an additive white Gaussian noise (AWGN) and Rayleigh fading environment. We compare modifications of ABP in terms of frame-error-rate performance and decoding latency, while focusing on efficient implementations in software or hardware. Main outcome of the paper is the analysis of quantization effects when ABP is implemented employing fixed-point values.

1. INTRODUCTION

Since their discovery in 1960 [1], Reed-Solomon (RS) 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 [2]. 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 relative 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 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 recently proposed in [3] an iterative, adaptive BP (ABP) algorithm. 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 [4] ABP with the Koetter-Vardy [5] algebraic soft-decision decoding (ASD) algorithm, which enabled them to achieve near optimal performance for relatively short, high-rate codes. In [6] a combination of ABP and the ordered-statistics-decoding (OSD) was used to improve the error correcting capability of medium length codes.

Due to the significant performance gains ABP and it's modifications are able to achieve when decoding RS codes, we investigate in this paper performance-cost trade-offs when ABP based algorithms are considered for implementation in software or hardware. Moreover, since these algorithms are based on the same operations as decoding algorithms for LDPC codes, future multi-standard, mobile devices could easily extend their decoding capabilities into a domain usually demanding dedicated hardware units, thereby saving cost-intensive chip area and implementation time. Beside performance studies for the AWGN and Rayleigh fading channel, we therefore present as a first step towards an embedded application, numerical results regarding quantization effects and explore modifications which generally result in considerable complexity and latency reduction.

The remainder of this work is organized as follows. Some preliminaries including the system model are given in the following section 2. Simulation results and corresponding discussions are provided in section 3. Finally, section 4 concludes the paper and suggests further research directions.

2. ITERATIVE DECODING OF REED-SOLOMON CODES USING ABP

2.1 Description 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 $N = 2^q - 1$. The minimum distance of the RS code can be denoted by $d_{min} = N - K + 1$. Let $\mathbf{m} = [m_o, m_1, \ldots, m_{K-1}]$ be a message of K information symbols. These symbols can be associated with an information polynomial $m(x) = m_0 + m_1 x + \ldots + m_{K-1}x^{K-1}$, which is encoded through multiplication by a generator polynomial $g(x) = \prod_{j=1}^{N-K}(x - \beta^j)$ employing the equation $c(x) = m(x)g(x)$. By this definition, each codeword $\mathbf{c} = [c_0, c_1, \ldots, c_{N-1}], c_i \in GF(2^q)$ is interpreted as a code polynomial $c(x)$. On decoder side, one could verify the validity of $\mathbf{c}$ by evaluating the well known parity-check equation $\mathbf{c} \cdot \mathbf{H}_q^T = \mathbf{0}$, with $\mathbf{H}_q$ being a $(N-K) \times N$ dimensional matrix consisting of $(N-K)$ codewords spanning the dual code of $RS(N,K)$ [7]:

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

Because the ABP algorithm operates on bit-level reliability values, $\mathbf{H}_q$ as well as $\mathbf{c}$ need to be transformed into an equivalent binary image representation $\mathbf{H}_b$ and $\mathbf{c}_b$, respectively. Depending on the chosen basis for representing the symbols of $GF(2^q)$ one could get different binary images of $\mathbf{H}_q$. In this paper, we assume normal basis representation for which the transformation can be found e.g. in [8]. Correspondingly, each element of $\mathbf{H}_q$ will be represented by a $q \times q$ sub-matrix in the binary image $\mathbf{H}_b$. Hence, $\mathbf{H}_b$ will be of size $(N-K)q \times Nq$. Now let $n = N \cdot q$ and $k = K \cdot q$ be the length of the codeword and the information at bit level, respectively. Again, we can compute the syndrome $\mathbf{c}_b \cdot \mathbf{H}_b^T$, which becomes zero if $\mathbf{c}_b = [c_0, c_1, \ldots, c_{n-1}], c_i \in GF(2)$ is a valid codeword.

2.2 System Model

Before describing the ABP algorithm formally, we first assume that the encoded bits c_j, $(j = 0, \ldots, n-1)$ are modulated using antipodal BPSK with a sent symbol taking the value $s_j = (-1)^{c_j}$. The binary symbols are either transmitted over an AWGN or a Rayleigh fading channel (Fig. 1). On receiver side, the input values can be specified by $r_j = |a_j| s_j + n_j$, with n_j being real valued additive white Gaussian noise and a_j being the complex fading factor which becomes $a = 1$ for the AWGN, and $|a_j|$ being Rayleigh distributed otherwise. By setting $E(|a_j|^2) = 1$ we suppose there is no loss of energy during transmission. Additionally, for the Rayleigh channel, a_j is assumed to be constant for q consecutive BPSK symbols, i.e. a slowly fading channel model is applied.

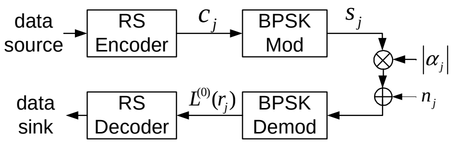


Figure 1: System Model

2.3 Adaptive Belief Propagation

The BP algorithm as original proposed in [9] has found numerous application in the decoding of linear block codes with sparse graph representations, as e.g. LDPC codes. In [10] it was shown that using BP decoding for LDPC codes can approach the Shannon limit. However, the binary image $\mathbf{H}_b$ of the parity-check matrix of a RS code is of high-density, leading to many short cycles in the corresponding Tanner graph representation. These short cycles counteract the BP algorithm in converging to the correct codeword, as they introduce correlation between exchanged belief information. Messages, sent from check nodes to variable nodes and vice versa, are therefore loosing their independence, resulting in poor error-correcting performance of the BP algorithm. In [3] a modified version of BP, namely ABP, was proposed where the parity-check sub-matrix corresponding to the $n-k$ 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 soft-decoding performance of linear block codes having HDPC matrices with a large number of short cycles.

2.3.1 BP decoding

In relation to [11], each parity-check matrix $\mathbf{H}_b$ can be described by a bipartite graph consisting of n variable and $(n-k)$ check nodes, that are connected with each other by edges when the corresponding entry $H_{(i,j)}$ in $\mathbf{H}_b$ (i being the row index, j being the column index) equals one. Decoding using BP can be accomplished by iteratively exchanging reliability information (called messages) between variable and check nodes. Messages $m_{x \to y}$ passed along the edges are calculated in two stages according to the following decoding equations:

- **Check node update:**

$$m_{i \to j}^{(l+1)} = \prod_{j' \in V_{i \setminus j}} \text{sign}(m_{j' \to i}^{(l)}) \cdot f\Big(\sum_{j' \in V_{i \setminus j}} f(|m_{j' \to i}^{(l)}|)\Big) \quad (2)$$

- **Variable node update:**

$$m_{j \to i}^{(l+1)} = L^{(0)}(r_j) + \theta \cdot \sum_{i' \in C_{j \setminus i}} m_{i' \to j}^{(l+1)} \quad (3)$$

Symbols:

l	count index for inner BP iterations
$L^{(0)}(r_j)$	initial log-likelihood ratio of bit j
θ	variable node attenuation coefficient $(0 < \theta \leq 1)$
$f(x)$	$f(x) = log\frac{exp(x)+1}{exp(x)-1}$
$m_{j \to i}$	message passed from variable node j to check node i
$m_{i \to j}$	message 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

Before the BP algorithm iterates on a matrix $\mathbf{H}_b$, variable node messages are initialized by the current bit-reliability values $m_{j \to i}^{(0)} = L^{(0)}(r_j)$. In the first iteration of BP decoding these values can be expressed in terms of log-likelihood ratios (LLRs) as observed from the channel:

$$L^{(0)}(r_j) = log\frac{Pr(r_j|c_j = 0)}{Pr(r_j|c_j = 1)} = 4|a_j| R \frac{E_b}{N_0} r_j, \quad (4)$$

with $R = K/N$ being the rate of the $RS(N,K)$ code and E_b/N_0 the signal-to-noise ratio used during transmission. We assume that a_j is known to the receiver (perfect channel state information). In the last iteration of BP, variable node updates are skipped and new bit-reliability values are computed according to:

$$L^{(k+1)}(r_j) = L^{(k)}(r_j) + \alpha \cdot \sum_{i \in C_j} m_{i \to j}^{(l+1)}, \quad (0 < \alpha \leq 1) \quad (5)$$

with k being the outer iteration index and $L^{(k)}(r_j)$ the LLR of bit j at iteration k. As the BP algorithm is only suboptimal when decoding dense parity check matrices, the parameters α and θ, which attenuate the influence of the extrinsic information, need to be chosen carefully for each individual code (this is different from LDPC codes, where α and θ can usually be set to one as the BP decoding is considered to be optimal due to the nonexisting short cycles).

2.3.2 ABP decoding

As mentioned above, the ABP algorithm first adopts the matrix $\mathbf{H}_b$ according to the LRBs. This can be accomplished by first sorting the LLRs, followed by Gaussian elimination

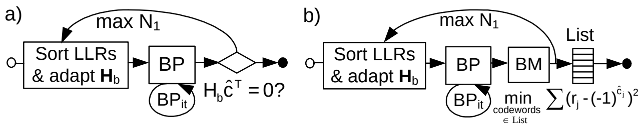


Figure 2: Decoding chains incorporating the ABP algorithm:
a) ABP-HD b) ABP-BM

(GE) of the columns which correspond to the LRBs. Afterwards, the BP decoding algorithm is applied to the adopted matrix $\mathbf{H}_b^{(k+1)} = \phi(\mathbf{H}_b^{(k)}, |\mathbf{L}^{(k)}|)$ with a preset number of inner iterations $l_{max} = BP_{it}$. Depending on a chosen stopping criterium there are then several possibilities to continue:

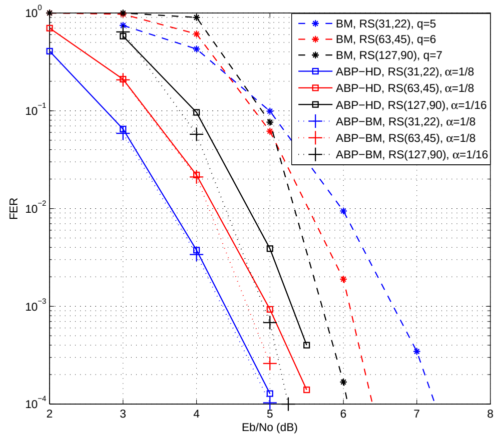
a) ABP-HD:
Perform hard decisions on the updated bit-reliability values $\hat{c}_j = \begin{cases} 0 & \text{if} \quad L^{(k+1)}(r_j) \geq 0 \\ 1 & \text{else} \end{cases}$ and check if BP decoding has been successful by computing $\hat{\mathbf{c}} \cdot \mathbf{H}_b^T$. Decoding is finished if it is equal to zero, otherwise ABP continues with a new outer iteration unless a maximum number of outer iterations $k_{max} = N_1$ has been reached.

b) ABP-BM/KV:
After each bit-reliability update (5), the corresponding word is added to a global list which can be used as input for a second decoder D (e.g. a hard-in hard-out Berlekamp-Massay (BM) or a soft-in hard-out Koetter-Vardy (KV) decoder). Decoding stops if the maximum number of outer iterations has been reached. Finally, the codeword $\hat{\mathbf{c}}$ is selected from the list of valid codewords decoded by the decoder D, using e.g. minimum Euclidean distance. In this paper, we restrict our comparison to the ABP-HD against the ABP-BM decoder as they are pictured in Fig. 2.

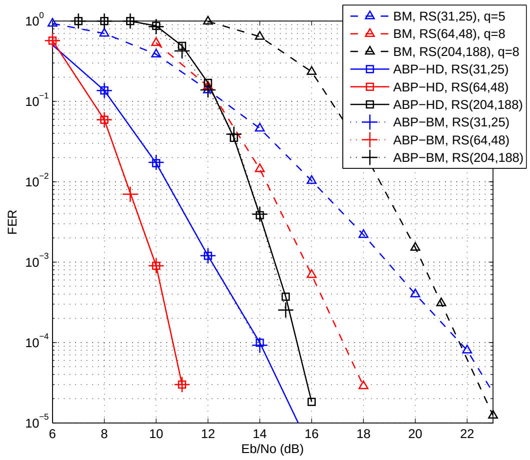
3. SIMULATION SETUP AND RESULTS

3.1 Comparison of ABP-BM with ABP-HD

Considering the two decoding chains from an implementation point of view, it gives reason to the question on the coding gain loss if the less complex ABP-HD is deployed instead of ABP-BM. Independent of the decision if the ABP algorithm is going to be implemented as software on a digital signal processor (DSP) or if a dedicated hardware accelerator is going to be used, ABP-BM would require a second decoder D which would demand at least more computational power and result in higher latency. In particular the fact that both the Berlekamp-Massay as well as the Koetter-Vardy algorithm are based on algebraic operations over higher order fields, would not only require additional instructions for the corresponding field arithmetic, but also much more area due to dedicated computational units (e.g. for Galois field multiplication) [12]. Therefore, as ABP-HD is just based on operations over $GF(2)$, it would fit the requirement for universality of future multi-standard mobile devices very well. Hence, an implementation would not be restricted to some specific code parameters q and $g(x)$ as it usually becomes necessary for the BM as well as the KV algorithm. In Fig. 3(a) the BM, ABP-HD and the ABP-BM algorithms are compared with each other for the AWGN and in Fig. 3(b) for the slowly fading Rayleigh channel. N_1 we set to a maximum number of 50 iterations, while BP_{It} was set to one, hence variable node updates were skipped and instead bit reliabilities were updated immediately according to (5) after completion of check-node updates.



(a) AWGN channel



(b) Rayleigh channel

Figure 3: Performance of BM, ABP-HD and ABP-BM for different RS codes

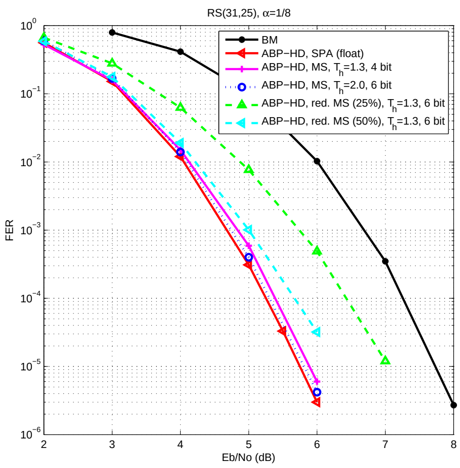
If we compare ABP-HD with ABP-BM under AWGN conditions, one can observe for larger block lengths the growing coding gain loss of ABP-HD as the SNR increases. At FER of 10^{-3} the loss compared to ABP-BM is about ≈ 0.3 dB for the $RS(63,45)$ and ≈ 0.4 dB for $RS(127,90)$. For smaller block lengths, as e.g. for $RS(31,22)$, we observed almost no difference between both algorithms down to the FERs presented here. As all codes in Fig. 3(a) were chosen to have $R \approx 0.71$, one can also notice the shrinking coding gain of ABP-based algorithms compared to classical BM as the block length becomes larger . At a FER of 10^{-3} the gain is ≈ 2.3 dB for $RS(31,22)$, while for $RS(127,90)$ it shrinks down to ≈ 0.8 dB. From wireless system design point of view, this suggests to use e.g. LDPC codes for larger block lengths, while RS codes decoded by ABP would be well suited for smaller block lengths. For the Rayleigh channel we investigated two shortened RS codes which are included e.g. in WiMAX ($RS(64,48), q = 8$) and in DVB standards ($RS(204,188), q = 8$). One can observe the large coding gain improvement of ≈ 6 dB which ABP-HD is able to achieve at a FER of 10^{-3} for this more practical system model. Unlike for the AWGN channel, we could not notice valuable coding gain when ABP-BM is used instead. From the simulations we also observed, that the larger the number of parity checks, the more carefully the damping coefficient α needs to be chosen. For a $RS(31,25)$ code α could be set to $\frac{1}{32}$ or $\frac{1}{8}$ resulting in almost no difference in FER performance, while for the RS(64,48) and RS(204,188) we found the best results only for $\alpha = \frac{1}{16}$ (we restricted α to values of power of two which could be easily implemented on DSPs using shift registers). Moreover, considering especially shorter block lengths with FER being less sensitive to α, we recommend to set α as large as possible, as this reduces the number of iterations necessary to find a valid codeword, hence decoding latency can be decreased.

3.2 Sum-Product vs. quantized Min-Sum Algorithm

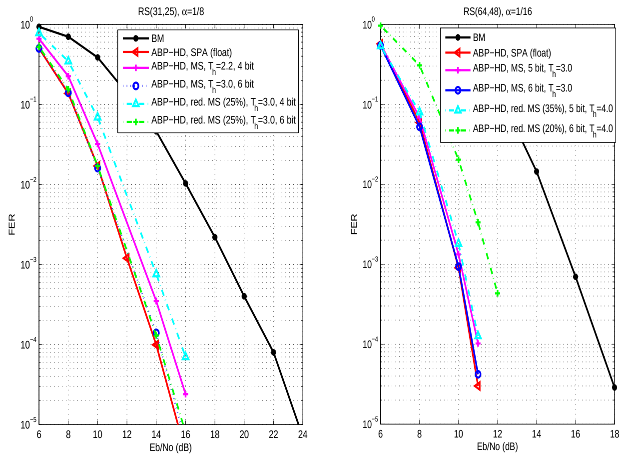
Even if some state-of-the-art DSPs possess complex floating point operations, one is still faced with the need to implement the function $f(x)$ and its inverse for check-node operation (2). Especially if a parallel implementation of the ABP algorithm is aspired, the approximation of $f(x)$ by means of lookup-tables would increase the area as well as power consumption significantly. As already known from LDPC codes, equation (2) can be approximated by the Min-Sum (MS) algorithm which overcomes the restriction to use lookup-tables with:

$$m_{i\to j}^{(l+1)} = \prod_{j' \in V_{i\setminus j}} \operatorname{sign}(m_{j'\to i}^{(l)}) \cdot \min_{j' \in V_{i\setminus j}} (|m_{j'\to i}^{(l)}|). \qquad (6)$$

In Fig. 4(a) and 4(b) we investigated the FER performance of the ABP-HD algorithm using the Sum-Product algorithm (SPA) and its quantized MS approximation for a $RS(31,25)$ and the $RS(64,48)$ code. For all the quantized simulations we skipped the scaling factor 4 in equation (4) and applied instead $L^{(0)}(r_j) = |a_j| R \frac{E_b}{N_0} r_j$. We further used a uniform quantization scheme which clips the soft-input values and exchanged messages into the range spanned by $-T_h \leq L^{(0)}(r_j) \leq +T_h$, with T_h being the optimal threshold found by simulations. Depending on the number of bits W used for quantization of soft-information, the quantization interval is given by $\Delta = \frac{2T_h}{2^W - 1}$. Due to the summation in (5) we assumed the addition to be more precise than the given bit-width W of the exchanged messages. This does not cost much complexity as the summed values can be stored locally in registers. For $RS(31,25)$ it can be observed that 6 bit message quantization (with 8 bits for summation) results in negligible performance loss for the AWGN (≈ 0.05 dB $@FER = 10^{-4}$) as well as for the Rayleigh channel (≈ 0.15 dB $@FER = 10^{-4}$) if one compares it with ≈ 2 dB and 7 dB coding gain respectively over BM algorithm. For $RS(64,48)$ 6 bit message quantization even coincides with SPA floating point implementation in a Rayleigh fading environment. Considering the smaller $RS(31,25)$ code with only 4 bit message quantization and 6 bit summation, the FER of MSA compared to SPA degrades by ≈ 0.2 dB for AWGN and ≈ 1.0 dB for the Rayleigh channel at FER=10^{-4}.



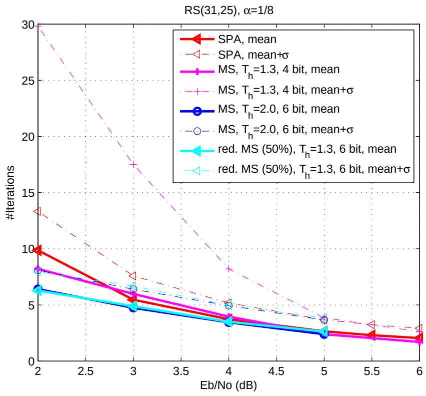
(a) AWGN channel



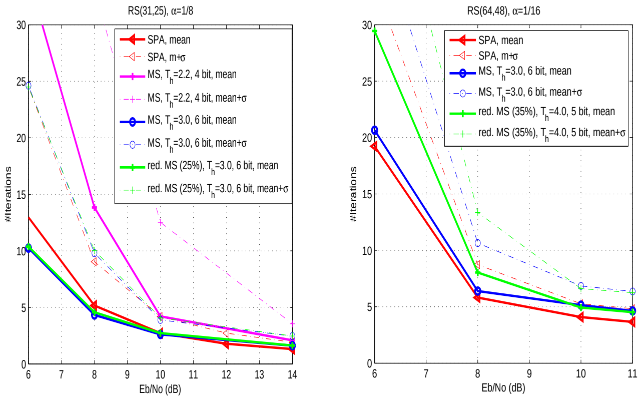
(b) Rayleigh channel

Figure 4: Performance of quantized MSA in comparison to SPA

For the larger $RS(64,48)$ code, we found a minimum of 5 bits adequate for message quantization. This corresponds to the quantization which is usually employed for LDPC codes. In Fig. 5(a) and 5(b) we also compare the average number of iterations executed until a valid codeword was found. Clearly, the number of iterations decreases as the SNR becomes larger. For the $RS(31,25)$ in a low SNR, Rayleigh fading environment we found a large increase in the number of iterations if only 4 bits are used for message quantization. Considering the variance, which is also plotted for the sake of clarity, one can also notice a large increase for the AWGN channel. Much better results we obtain if 6 bits are employed, which not only improves FER but also reduces the average number of iterations for low SNR values compared to the SPA. Although MSA is originally used as an approximation of the SPA, SPA is not optimal for decoding HDPC matrices. Hence, MSA seems to be better suited at least for low SNR values as it can reduce latency. As the operating point is moved towards lower FERs, the difference between SPA and quantized MSA diminishes. A similar behavior can be observed for the $RS(64,48)$ code.



(a) AWGN channel



(b) Rayleigh channel

Figure 5: Average number of iterations using SPA and MSA

3.3 Performance of Complexity Reduced MSA

Especially if BP_{It} is set to one and the LLRs are therefore re-ordered every iteration, one can restrict the bit-reliability updates to the LRBs, hence reducing the number of operations. Besides, we can take advantage of the fact that we already know the sorting index of the input LLRs from the GE step. Therefore, no extra sorting operations become necessary. Updating only the LRBs can be justified by the fact that the corresponding bits are with higher probability in error than the most-reliable bits (MRBs). Especially for the Rayleigh fading channel, where the fading factor a_j dominates the reliability of the received bits, the deployment of complexity reduced MSA can save significant amount of computation while still providing competitive FER performance. For the $RS(31,25)$ code we simulated the MSA with a partial reliability update incorporating only 25% of the bits with the lowest LLR. From Fig. 4(b) it can be observed that with 6 bit message quantization the results almost coincident with full MS update. The same we obtained for the $RS(64,48)$ code using 6 bit message quantization while updating only 35% of the LLRs (not shown in the figures). Reducing quantization to 5 bits still results in good FER, while the number of iterations increases slightly for low SNR values. Further restriction of updated LLRs leads to noticeable performance loss. However, updating only 20% of all LLRs still results in ≈ 4.1 dB coding gain over BM at a FER of 10^{-3}. In order to emphasize the large reduction in decoding time when the reduced MSA is employed, we provide the average decoding time per iteration for several RS codes in table 1. We would like to point out, that the saving in decoding time for reduced complexity MSA would be even higher if the GE step would be optimized as e.g. proposed in [4]. In order to achieve the same FER as with the full MS update in a AWGN channel, the reduction could not be made as small as for the Rayleigh channel. The reason for that basically comes from the AWGN distribution, which leads to a smaller difference between least- and most reliable bits, hence more bits are required to be updated for better belief propagation.

	SPA	**full MS**	**red. MS (25%)**
RS(31,25)	0.02187 (100%)	0.00496 (23%)	0.00087 (4%)
RS(64,48)	0.92191 (100%)	0.20617 (22%)	0.08740 (9%)
RS(204,188)	13.77440 (100%)	2.78330 (20%)	0.53878 (4%)

Table 1: Average decoding time per iteration for ABP-HD (in seconds), measured on a Intel Xeon CPU (2.8 GHz) using Microsoft Visual C++ 2005 Compiler

4. CONCLUSIONS

In this paper, we investigated soft-decision decoding of RS codes based on the ABP algorithm. We focused on the performance-cost trade-offs that become necessary if ABP is employed in future wireless, embedded systems and provided adequate parameter settings in order to achieve acceptable error-correcting performance. We showed for short to medium length codes, that ABP-HD based on 5 to 6 bit quantized MS approximation can provide significant lower decoding complexity than the SP algorithm at almost no loss of FER performance in Rayleigh fading environments. Besides, under Rayleigh conditions, decoding latency of MS algorithm can be further decreased employing partial reliability updates. Future work will investigate the trade-offs when the number of BP iterations is not restricted to be one as in the analysis presented here. Also, timing measurements of the proposed quantized MSA in an actual, embedded system would be of interest.

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

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

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

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

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

[7] S. Lin, D. J. Costello, "Error Control Coding," 2nd edition, 2004

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

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

[10] 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.

[11] R. Tanner, "A recursive approach to low complexity codes," in *IEEE Trans. Inf. Theory,* vol. 27, no. 5, Sep. 1981

[12] J. Dielissen, N. Engin, S. Sawitzki, K. van Berkel "Multistandard FEC Decoders for Wireless Devices," in *IEEE Trans. Circuits and Systems,* vol. 55, no. 3, March 2008