International Symposium on Information Theory and its Applications, ISITA2004
Parma, Italy, October 10–13, 2004

Iterative Algebraic Soft Decision Decoding of Reed-Solomon Codes

Mostafa El-Khamy † and Robert J. McEliece‡

California Institute of Technology
Pasadena
CA 91125, USA
†E-mail: mostafa@systems.caltech.edu ‡E-mail: rjm@systems.caltech.edu

Abstract

In this paper, we propose an iterative soft decision decoding scheme for Reed Solomon codes with near maximum likelihood performance. The advantage of this decoding algorithm over previously proposed algorithms is its fast convergence in terms of the number of iterations required. This is achieved by combining two powerful soft decision decoding techniques which were previously regarded in the literature as competitive ones, namely, algebraic soft decision decoding and belief propagation based on adaptive parity check matrices. This algorithm could also be viewed as a multiplicity assignment scheme for the Guruswami-Sudan list decoding algorithm.

1. Introduction

Maximum likelihood (ML) decoding of general (n, k) linear codes, and specifically Reed Solomon (RS) codes, is NP-Hard [1, 2]. It remains an open problem to find efficient polynomial time algorithms with near maximum likelihood performance. Guruswami and Sudan (GS) [3] invented a polynomial time list decoding algorithm for RS codes capable of correcting errors beyond half the minimum distance of the code. In [4], Kötter and Vardy (KV) developed an algebraic soft decision decoding (ASD) algorithm for RS codes based on a multiplicity assignment (MA) scheme for the GS algorithm. Alternative multiplicity assignment schemes with better performance were proposed in [5] and [6,7]. In [8], Fossorier combined BP with order statistics decoding (OSD) for soft decoding of LDPC codes which led to faster convergence. Jiang and Narayanan [9] proposed an adaptive belief propagation (ABP) algorithm and reported near ML performance for RS codes. In this paper we combine the ABP algorithm with ASD for faster convergence and performance enhancement.

This research was supported by NSF grant no. CCR-0118670 and grants from Sony, Qualcomm, and the Lee Center for Advanced Networking

2. Preliminaries

An (n, k) RS codeword $\boldsymbol{u} = [u_0, u_1, .., u_{n-1}]$ over the finite field F_q of q elements is generated by evaluating a data polynomial $M(x)$ of degree $k-1$ at n elements of the field, that is

$$\boldsymbol{u} = [M(1), M(\alpha^1), ..., M(\alpha^{n-1}] \tag{1}$$

where α is a primitive element of the field and $n = q-1$. The set $T = \{1, \alpha, \alpha^2, ..., \alpha^{n-1}\}$ is called the support set of the code and is vital for the GS interpolation step. From the following lemma (the proof is omitted)

Lemma 1 *The polynomial $U(x) = \sum_{i=0}^{n-1} u_i x^i$ associated with a codeword $\boldsymbol{u}$ generated as in (1) has $\alpha, \alpha^2, .., \alpha^{n-k}$ as zeros.*

it follows that

$$\sum_{i=0}^{n-1} u_i \alpha^{ij} = 0 \text{ for } j = 1, 2, .., n-k, \tag{2}$$

and a valid PC matrix $\mathcal{H}$, such that $\mathcal{H}\boldsymbol{u}^T = 0$, is [10]

$$\mathcal{H} = \begin{bmatrix} 1 & \alpha & \dots & \alpha^{n-1} \\ 1 & \alpha^2 & \dots & \alpha^{2(n-1)} \\ \vdots & \vdots & \dots & \vdots \\ 1 & \alpha^{n-k} & \dots & \alpha^{(n-k)(n-1)} \end{bmatrix}. \tag{3}$$

Let $W = [W_i(\beta)]$, where $i = 0, 1, ..., n-1$ and $\beta \in F_q$, be an $q \times n$ array of real numbers. The *cost* of W is defined to be $|W| \triangleq \frac{1}{2} \sum_{i=0}^{n-1} \sum_{\beta \in F_q} W_i(\beta) (W_i(\beta) + 1)$ and the score of $\boldsymbol{u}$ with respect to W is $\langle \boldsymbol{u}, W \rangle \triangleq \sum_{i=0}^{n-1} W_i(u_i)$. Assuming that a codeword $\boldsymbol{u}$ is transmitted and the corresponding channel output is $\boldsymbol{y}$, then the reliability matrix Π of a a-posteriori probabilities is

$$\Pi_i(\beta) = Pr\{u_i = \beta | y_i\}.$$

3. Algebraic Soft Decoding

Given Π, a multiplicity assignment (MA) algorithm generates an $q \times n$ *multiplicity* matrix, M, of non-negative integers. The multiplicity matrix M is then passed to the GS algorithm. A decoding success is signaled if the transmitted codeword is on the generated list. A sufficient condition for a codeword $\boldsymbol{u}$ to be on the GS generated list is [4],

$$\langle \boldsymbol{u}, M \rangle > D_{k-1}(|M|), \tag{4}$$

where $D_v(\gamma) = \left\lfloor \frac{\gamma}{m} + \frac{v(m-1)}{2} \right\rfloor$ for $m = \left\lfloor \sqrt{\frac{2\gamma}{v} + \frac{1}{4}} + \frac{1}{2} \right\rfloor$ [7].

A reduced complexity KV MA algorithm is [4]

$$M_i(\beta) = \lfloor \lambda \Pi_i(\beta) \rfloor \tag{5}$$

where $\lambda > 0$ is a complexity parameter determined by $|M|$. For $|M| \leq \gamma$, it could be shown that $\lambda = (-1 + \sqrt{1 + 8\gamma/n})/2$. Asymptotically, for large interpolation costs, a sufficient condition for a codeword $\boldsymbol{u}$ to be on the KV-GS generated list is

$$\frac{\langle \boldsymbol{u}, \Pi \rangle}{\|\Pi\|_2} > \sqrt{k-1}. \tag{6}$$

Whereas, the KV MA algorithm attempts to maximize the mean of the score, the algorithms in [5] and [6] attempt to minimize the error probability directly. The algorithm of [5] (Gauss) assumes a gaussian distribution of the score, while that of [6] (Chernoff) minimizes a Chernoff bound on the error probability. The later has the best performance at the expense of computational complexity [7]. However in this paper, we will focus on using the KV algorithm due to its relatively low complexity.

4. Binary Image of the Reed Solomon Code

Let $p(x) = a_o + a_1 x + a_{m-1} x^{m-1} + x^m$ be a primitive polynomial in $F_2[x]$. Let α be a root of $p(x)$, then α is a primitive element in F_{2^m}. The companion matrix of $p(x)$ is given by the $m \times m$ matrix

$$C = \left[\begin{array}{ccc|c} 0 & \dots & 0 & a_o \\ & & & a_1 \\ I_{m-1} & & & \vdots \\ & & & a_{m-1} \end{array}\right],$$

where I_m is the $m \times m$ identity matrix [11]. Representing the primitive element, α, by its binary companion matrix C, the mapping $\alpha^i \leftrightarrow C^i$, $\{i = 0, 1, 2, ..\}$ induces a field isomorphism. So every element in the parity check (PC) matrix of (3) can be replaced with an $m \times m$ matrix resulting in a binary PC matrix, H. Also, any element, $\beta \in F_{2^m}$, has an m-tuple representation $\{\beta_0, \beta_1, ..., \beta_{m-1}\}$ where $\beta = \beta_0 + \beta_1 \alpha + ... + \beta_{m-1}\alpha^{m-1}$ and $\beta_i \in F_2$. Thus the binary image of a codeword $\boldsymbol{u}$ is given by the nm tuple $\boldsymbol{u_b}$ where

$$\boldsymbol{u_b} = [u_{0,0}, u_{0,1}, ..., u_{0,m-1}, \dots, u_{n-1,0}, u_{n-1,1}, ..., u_{n-1,m-1}].$$

Such a mapping results in $H\boldsymbol{u_b}^T = 0$.

5. Log-Belief Propagation

BP was originally invented by Gallager for decoding LDPC codes [12]. $H_{i,j}$ will denote the element in the ith row and jth column of the binary PC matrix H. Define the sets, $J(i) \triangleq \{j \mid H_{i,j} = 1\}$ and $I(j) \triangleq \{i \mid H_{i,j} = 1\}$. Given the vector $\boldsymbol{\Lambda^{in}}$ of initial log-likelihood ratios (LLRs), the BP algorithm outputs the extrinsic LLRs $\boldsymbol{\Lambda^x}$.

Algorithm 1 *Log Belief Propagation*
For all (i, j) *such that* $H_{i,j} = 1$*:*
Initialization: $Q_{i,j} = \Lambda_j^{in}$
DO
Horizontal Step (HS):

$$R_{i,j} = \log\left(\frac{1 + \prod_{k \in J(i)\backslash j} \tanh(Q_{i,k}/2)}{1 - \prod_{k \in J(i)\backslash j} \tanh(Q_{i,k}/2)}\right) \tag{7}$$

$$= 2\tanh^{-1}\left(\prod_{k \in J(i)\backslash j} \tanh(Q_{i,k}/2)\right) \tag{8}$$

Vertical Step (VS):

$$Q_{i,j} = \Lambda_j^{in} + \theta \sum_{k \in I(j)\backslash i} R_{k,j}$$

While *stopping criteria is not met.*
Extrinsic Information: $\Lambda_j^x = \sum_{k \in I(j)} R_{k,j}$.

The factor θ is termed the *vertical step damping factor* and $0 < \theta \leq 1$. θ is typically 0.5. Eq.(8) is specifically useful for fast hardware implementations where the tanh function will be implemented as a lookup table. In our implementation, BP is run for a small number of iterations on the same PC matrix, so the stopping criteria is the number of iterations. In case only one iteration is run, the VS is eliminated.

6. Adaptive Belief Propagation

Let the received vector to be $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{\eta}$, where $\boldsymbol{x} = 1 - 2\boldsymbol{u_b}$ is the BPSK modulation of a codeword $\boldsymbol{u}$.

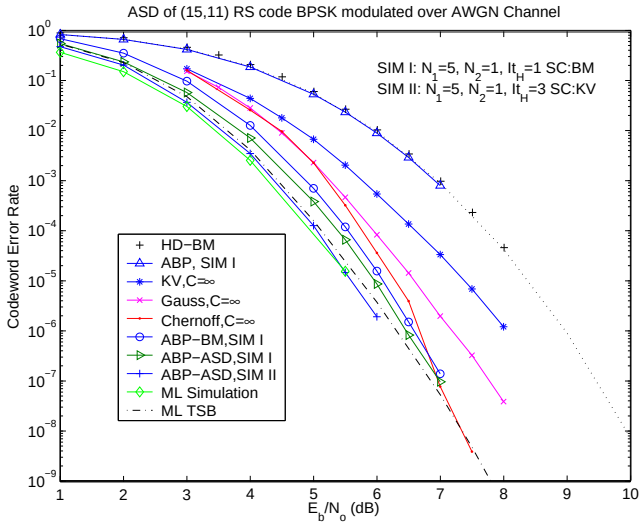

Figure 1: Iterative ASD of (15,11) RS code BPSK modulated over an AWGN channel, infinite cost, $\alpha_1 = 0.1$

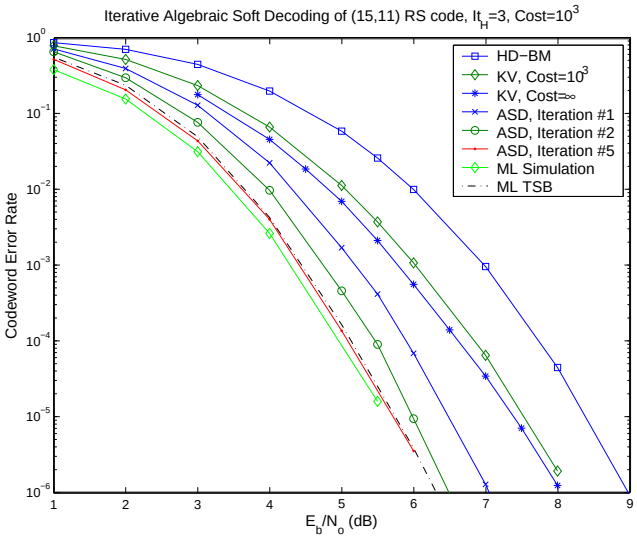

Figure 2: Iterative ASD of (15,11) RS code BPSK modulated over an AWGN channel, finite cost, $\alpha_1 = 0.1$

$\boldsymbol{\eta}$ is the AWGN vector with variance σ^2. The channel LLRs are $\boldsymbol{\Lambda^{ch}} := 2\boldsymbol{y}/\sigma^2$. In concatenated systems, where the RS code is implemented as an outer code, the 'channel' LLRs will be the soft output of an inner decoder which could be for example an BCJR, SOVA or another BP decoder.

Algorithm 2 *Adaptive Belief Propagation [9]*
Initialization $\boldsymbol{\Lambda^p} := \boldsymbol{\Lambda^{ch}}$
DO

1. *Sort* $\boldsymbol{\Lambda^p}$ *in ascending order of magnitude. The resulting vector of sorted LLRs is*

$$\boldsymbol{\Lambda^{in}} = [\Lambda_1^{in}, \Lambda_2^{in}, ..., \Lambda_{nm}^{in}],$$

$|\Lambda_k^{in}| \leq |\Lambda_{k+1}^{in}|$ *for* $k = 1, 2, ..., nm-1$ *and* $\boldsymbol{\Lambda^{in}} := P\boldsymbol{\Lambda^p}$ *where* P *is a permutation matrix.*

2. *Rearrange the columns of the binary parity check matrix* H *to form a new matrix* H_P *where the rearrangement is defined by the permutation* P*.*

3. *Perform Gaussian elimination (GE) on the matrix* H_P *from left to right. GE will reduce the first independent* $(n-k)m$ *columns in* H_P *to an identity sub-matrix. The columns which are dependent on previously reduced columns will remain intact. Let this new matrix be* $\hat{H}_P$*.*

4. *Run Log Belief Propagation on the parity check matrix* $\hat{H}_P$ *with initial LLRs* $\boldsymbol{\Lambda^{in}}$ *for a maximum number of iterations* It_H *and a vertical step damping factor* θ*. The log BP algorithm outputs extrinsic LLRs* $\boldsymbol{\Lambda^x}$*.*

5. *Update the LLRs,* $\boldsymbol{\Lambda^q} = \boldsymbol{\Lambda^{in}} + \alpha_1\boldsymbol{\Lambda^x}$ *where* $0 < \alpha_1 \leq 1$ *is called the ABP damping factor.* $\boldsymbol{\Lambda^p} := P^{-1}\boldsymbol{\Lambda^q}$ *where* P^{-1} *is the inverse of* P*.*

While *Stopping criteria not satisfied.*

To reduce the complexity, GE could be run on the parity check matrix already reduced by GE at the previous the step (after appropriate permutations). As suggested in [9], if the algorithm with the specified SC fails for N_1 inner iterations, the algorithm is repeated again until it succeeds and for a maximum number of N_2 outer iterations. Each one of these N_2 iterations starts with a different permutation of the sorted channel LLRs in the first inner iteration. We propose doing this in a systematic way to ensure that most bits will have a chance to pass their information by being in the identity sub-matrix of the eliminated PC matrix. In a real time implementation, these iterations could be run on N_2 parallel processors and the decoding stops once the stopping criteria is satisfied for any of the N_2 parallel decoders. Let $z = \lfloor mn/N_2 \rfloor$, then at the $(r+1)^{th}$ outer iteration, $r > 0$, the initial LLR vector at the first inner iteration is

$$[\Lambda_{rz+1}^{in}, .., \Lambda_{(r+1)z}^{in}, \Lambda_1^{in}, ..., \Lambda_{rz}^{in}, \Lambda_{(r+1)z+1}^{in}, ..., \Lambda_{nm}^{in}], \quad (9)$$

where $\boldsymbol{\Lambda^{in}}$ is the vector of sorted channel LLRs. The columns of H_P will also be rearranged according to the same permuatation. If $rz < (n-k)m$, then it is less likely that this permutation will introduce new columns into the identity submatrix area of the PC

matrix reduced at the first inner iteration and thus it is recommended to continue with $r > (n-k)m/z$.

The performance of this algorithm as well as the convergence speed depends largely on the stopping criteria (SC). Different stopping criteria are suggested:

- **SC:HD**: Perform hard decisions on the LLRs, $\hat{\boldsymbol{u}} = (1 - \text{sign}(\boldsymbol{\Lambda^p}))/2$. Stop if $H\hat{\boldsymbol{u}}^T = 0$ or the number of iterations is N_1.

- **SC:BM**: Perform hard decisions on the LLRs, $\hat{\boldsymbol{u}} = (1-\text{sign}(\boldsymbol{\Lambda^p}))/2$. The vector $\hat{\boldsymbol{u}}$ is input to the Berlekamp-Massey (BM) algorithm. If this vector is within a Hamming distance of $\lfloor \frac{n-k}{2} \rfloor$ from a valid codeword, $\boldsymbol{g}$, output $\boldsymbol{g}$ and stop. Otherwise if the number of iterations is N_1 stop.

- **SC:ASD** Using $\boldsymbol{\Lambda^p}$ generate an $q \times n$ reliability matrix $\hat{\Pi}$ which is then used as an input to an MA algorithm to generate multiplicities according to the required interpolation cost. This multiplicity matrix is passed to the GS list decoding algorithm. Stop if the generated codeword list is not empty or if the maximum number of iterations, N_1, is reached. If more than one codeword is on the list choose the one with the highest reliability with respect to the channel LLR's $\boldsymbol{\Lambda^{ch}}$.

In this paper, the KV algorithm is used as the MA scheme and we will denote this stopping criteria by 'SC:KV'. More efficient but more complex MA schemes could also be used [6]. The SC should also be checked before any ABP iteration is carried out. If the SC is satisfied, then the decoding stops.

We highlight the main differences between our proposed algorithm and that proposed in [9]. We propose running a small number of iterations, It_H, on the same PC matrix using a damped vertical step. This has the effect of updating the LLRs using the information from other parity checks and the independence assumption used by the BP algorithm still holds. A small number of iterations is not enough for the information passed to start to propagate in the loops of the reduced PC matrix. The main contribution in this paper is the utilization of the pseudo-posterior LLRs output from the ABP algorithm as the soft information input to an ASD algorithm. SC:KV improves the convergence speed of the ABP algorithm in terms of the number of iterations and thus could render it more practical. Since our algorithm transforms the channel LLRs into interpolation multiplicities for the GS algorithm, then by definition it is an interpolation multiplicity assignment algorithm.

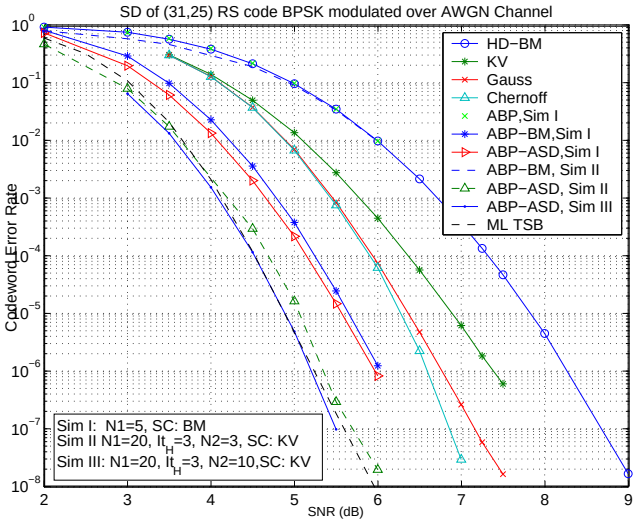

Figure 3: Iterative ASD of (31,25) RS code BPSK modulated over an AWGN channel, $\alpha_1 = 0.1$

7. Optimality of ABP-ASD

The authors of [4] point out that it is hard to maximize the mean of the *score* according to the true posteriori channel reliabilities. Previous MA algorithms [4–7] assumed approximate a-posteriori reliabilities. The problem is simplified by assuming that the transmitted codeword is drawn uniformly from F_q^n. Also, the n received word symbols are assumed to be independent. In such a case, the a-posteriori probabilities are approximated to be a scaling of the channel transition probabilities,

$$\Pi_i^{ch}(\beta) = \frac{Pr\{y_i|u_i = \beta\}}{\sum_{\omega \in F_q} Pr\{y_i|u_i = \omega\}}. \qquad (10)$$

However, from the MDS property of RS codes any k symbols (only) are k-wise independent and could be treated as information symbols and thus uniformly distributed. The independence assumption is thus more valid for higher rate codes and for memoryless channels. It is well known that the extrinsic information $\boldsymbol{\Lambda^x}$ generated by a BP algorithm takes into account the geometry of the code and the correlation between symbols (see for example [13]). Thus adding $\boldsymbol{\Lambda^x}$ to the (approximate) channel LLRs $\boldsymbol{\Lambda^{ch}}$, tends to transform approximate channel reliabilities to more appropriate a-posteriori reliabilities. Thus, for a codeword $\boldsymbol{u}$, an MA algorithm $\mathcal{A}$ we have the following Markov chain,

$$\boldsymbol{u} \rightarrow \Pi^{ch} \overset{ABP}{\longrightarrow} \hat{\Pi} \underbrace{\overset{\mathcal{A}}{\longrightarrow} M \rightarrow}_{ASD} \hat{\boldsymbol{u}}, \qquad (11)$$

where M is the multiplicity matrix and $\hat{\boldsymbol{u}}$ is the decoder output.

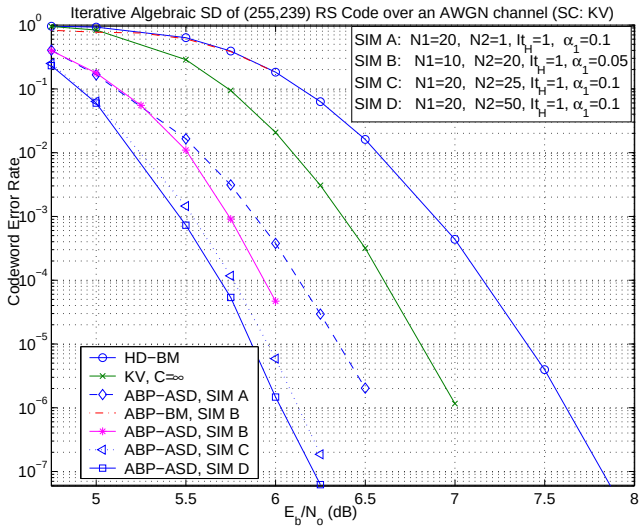

Figure 4: Iterative ASD of (255,239) RS code BPSK modulated over an AWGN channel

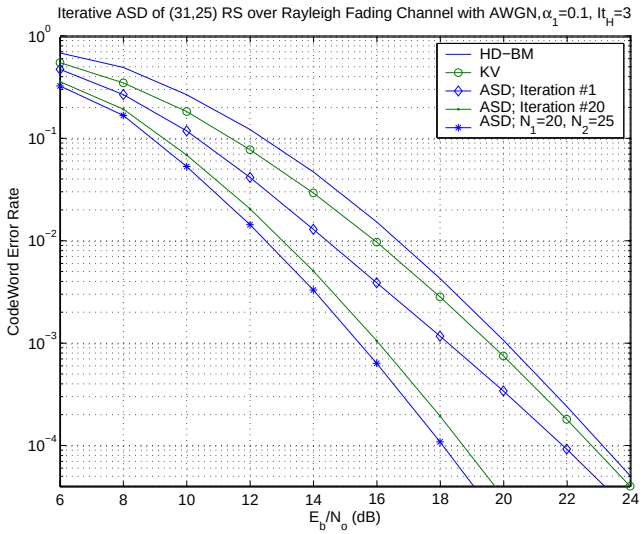

Figure 5: Iterative ASD of (31,25) RS code BPSK modulated over a Rayleigh fading channel , $\alpha_1 = 0.1$

8. Numerical Results

An (n, k) RS code is BPSK modulated and transmitted over an AWGN channel. The stopping criteria compared here are 'SC:BM' and 'SC:KV'. To demonstrate the convergence speed of the proposed algorithms, if the stopping criteria is 'BM' then 'KV' (ASD) could be checked after the loop stops using the modified LLRs. In this case, the performance of BM and KV will be denoted by 'ABP-BM, SC:BM' and 'ABP-ASD, SC:BM' respectively. If the SC is KV, then the performance of KV (checked at each iteration) and that of BM when checked after the loop exits will be denoted by 'ABP-ASD, SC:KV' and 'ABP-BM, SC:KV' respectively. We could also check if hard decisions only on the resulting LLRs after the loop exits will result in a codeword and this is referred to as 'ABP'. The performance depends largely on which SC is used. The performance is also compared with other ASD algorithms. A tight upper bound on the ML performance, based on an averaged binary weight enumerator and the tangential sphere bound, is shown and labelled 'ML TSB' [14].

In simulating 'KV', the sufficient condition (4) is used to check if the transmitted codeword is on the GS generated list [4]. This is partially justified by the fact that the average list size is 1 [15]. In a real implementation, if the list is empty then the condition will not be satisfied. However, if the condition (4) is not satisfied and a codeword other than the transmitted one is on the list, then this is an ML error.

Figure 1 shows simulation results for the $(15, 11)$ RS code over GF(16). All ASD and MA algorithms shown here are asymptotic in the interpolation cost, C, to show their potential. Using SC:KV with only $N1 = 5$ inner iterations and $It_H = 3$ iterations on each PC matrix (SIM II), the performance of ABP-ASD is close to ML decoding. ML decoding was simulated using the BCJR algorithm on the trellis associated with the binary image of the RS code [16]. The 'ABP-ASD, SC:KV' curve has a significant coding gain over 'ABP-BM , SC:BM'. With SC:BM, ABP alone, if checked when BM fails, does not improve over BM. The performance of 'ABP-BM, SC:KV' becomes very close to the BM curve (and thus is not plotted) due to the fast convergence of the KV algorithm and thus not enough iterations are carried out for BM to succeed. On the other hand, for 'SC:BM' (SIM I), the performance of KV checked after the loop exits, ABP-ASD, has a better performance than ABP-BM. This is due to fact that in most cases if 'BM' succeeds, the KV algorithm will also succeed but the opposite is not true. It is worth noting that with SC:BM, the performance with $N_1 = 5$ and $It_H = 3$ is better than that with $N1 = 10$ and It_H=1. These curves were omitted for the sake of figure clarity. It is also interesting to compare the performance with other MA schemes. It has about 2 dB coding gain over the KV algorithm at a codeword error rate (CER) of 10^{-6}. As expected, the Chernoff method has a comparable performance at the tail of the error probability. Near ML decoding for the same code is also achieved with a finite cost of 10^3 and SC:KV as shown in Fig.2. Comparisons are made between the possible coding gains if the maximum number of iterations is limited to $k = 1, 2, 5$ (denoted by 'Iteration #k'). Note

that KV decoding is done at each iteration.

The performance of the $(255, 239)$ code over an AWGN channel is shown in Fig.4. At an CER of 10^{-6} the coding gain of ABP-ASD (SC-KV) with N1=20 and N2=25 over BM is about 1.5 dB. Increasing the number of outer iterations N2 to 50 results in a coding gain of about 0.1 dB more. For the (31,25) RS code over AWGN, Fig.3, the performance of 'ABP-ASD, SC: KV' with 20 inner iterations (N1) and 10 outer iterations (N2) is better than the ML upper bound and has more than 3 dB coding gain over the BM algorithm at an CER of 10^{-4}. As expected from the discussion in Sec. 7, the coding gain of ABP-ASD is much more if the underlying channel model is not memoryless. This is demonstrated in Fig.5 where an $(31, 25)$ code is BPSK modulated over a relatively fast Rayleigh fading channel with AWGN. The coding gain of ABP-ASD over the BM algorithm at an CER of 10^{-4} is 5 dB when the channel is unknown to the decoder.

In general, it is noticed that the performance gain between iterations decreases with the number of iterations. It is to be noted that for the same simulation parameters, the performance of KV checked after the iteration loop with an BM stopping criteria exits is worse than that of ABP-ASD if the KV SC was employed in every iteration. Since on average BM requires more iterations for success, this implies that running a large number of iterations of ABP only may result in saturating the reliabilities at a non-desired solution. In other words, ASD should be employed after each iteration of ABP to make use of the improved reliabilities.

9. Conclusions

In this paper, we proposed a joint ABP-ASD algorithm based on the ABP algorithm proposed in [9] and the ASD algorithm in [4]. Our algorithm has faster convergence in terms of the number of iterations required. A small loss in performance results when using reasonable interpolation costs. The coding gain is larger for channels with memory. Our algorithm is also a multiplicity assignment algorithm for the GS algorithm.

References

[1] Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems," *IEEE Trans. Inform. Theory*, vol. 24, pp. 384–386, May 1978.

[2] V. Guruswami and A. Vardy, "Maximum likelihood decoding of Reed Solomon codes is NP-hard," submitted to *IEEE Trans. Inform. Theory.*

[3] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon codes and algebraic geometry codes," *IEEE Trans. Inform. Theory*, vol. 45, no. 6, pp. 1757–1767, Sept. 1999.

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

[5] F. Parvaresh and A. Vardy, "Multiplicity assignments for algebraic soft-decoding of Reed-Solomon codes," in *Proc. ISIT*, 2003.

[6] M. El-Khamy, R. McEliece, and J. Harel, "Performance enhancements for algebraic soft-decision decoding of Reed-Solomon codes." in *Proc. ISIT*, 2004.

[7] M. El-Khamy and R. J. McEliece, "Interpolation multiplicity assignment algorithms for algebraic soft-decision decoding of Reed-Solomon codes," submitted to *AMS-DIMACS volume, "Algebraic Coding Theory and Information Theory".*

[8] M. Fossorier, "Iterative reliability-based decoding of low-density parity check codes," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 908–917, May 2001.

[9] J. Jiang and K. Narayanan, "Iterative soft decision decoding of Reed Solomon codes based on adaptive parity check matrices," in *Proc. ISIT*, 2004.

[10] R. J. McEliece, *The Theory of Information and Coding*, 2nd ed. Cambridge: Cambridge U. Press, 2002.

[11] R. Horn and C. Johnson, *Matrix Analysis.* Cambridge University Press, 1985.

[12] R. Gallager, *Low Density Parity Check Codes.* MIT Press, 1963.

[13] R. McEliece, D. MacKay, and J. Cheng, "Turbo decoding as an instance of pearl's belief propagation algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 140–152, Feb 1998.

[14] M. El-Khamy and R. J. McEliece, "Bounds on the average binary minimum distance and the maximum likelihood performance of Reed Solomon codes," submitted to 42nd Allerton Conf. on Communication, Control and Computing.

[15] R. J. McEliece, "On the average list size for the Guruswami-Sudan decoder," in *ISCTA03*, 2003.

[16] M. Kan, private communication.