SOFT DECISION DECODING OF RS CODES UNDER HIGHER DATA RATE

V. Angayarkanni

Department of electronics and communication engineering, Jeppiaar Institute of Technology, Sriperumbudur, Chennai, Tamilnadu

ABSTRACT

Two new iterative soft decision decoding methods for Reed-Solomon (RS) codes are proposed. These methods are based on bit level belief propagation (BP) decoding. In order to make BP decoding effective for RS codes, we use an extended binary parity check matrix with a lower density and reduced number of 4-cycles compared to the original binary parity check matrix of the code. In the first proposed method, we take advantage of the cyclic structure of RS codes. Based on this property, we can apply the belief propagation algorithm on any cyclically shifted version of the received symbols with the same binary parity check matrix. For each shifted version of received symbols, the distribution of reliability values will change and deterministic errors can be avoided. This method results in considerable performance improvement of RS codes compared to hard decision decoding. The performance is also superior to some popular soft decision decoding methods. The second method is based on information correction in BP decoding. It means that we determine least reliable bits and by changing their channel information, the convergence of the decoder is improved. Compared to the first method, this method needs less BP iterations (less complexity) but its performance is not as good.

KEYWORDS: Reed-Solomon codes, belief propagation, extended binary parity check matrix, cyclic codes, information
I.INTRODUCTION

Reed-Solomon (RS) codes are non-binary linear block codes with a wide range of applications in wireless communications and storage systems. RS codes concatenated with convolutional codes are the standard channel codes for satellite transmission and deep-space communications. They are also outer codes in the third generation (3G) wireless standard and expected to be used as outer codes in concatenated coding schemes for fourth generation wireless standard. RS codes are maximum distance separable which is a very attractive property for error-correcting codes. Hard decision decoders with low complexity exist for RS codes, but they don’t use the channel reliability information which causes considerable performance loss that may not be acceptable in many applications. Guruswami and Vardy have shown that the maximum likelihood decoding of RS codes is NP-hard. Based on this fact, many different soft decision decoding methods have been introduced for RS codes in an effort to have moderate complexity and a performance close to the optimal maximum likelihood (ML) decoding. Algebraic list decoding algorithm (the Koetter-Vardy (KV) algorithm) provides significant coding gain over hard decision decoding for low rate RS codes. However, in order to have large coding gain, its complexity might increase prohibitively. ML decoding of RS codes using their binary image expansion was first proposed by decomposing RS codes into BCH subfield subcodes. The complexity of this method grows exponentially with the code length.

Using the binary image of RS codes, reliability based ordered statistics decoding (OSD) and its variations can also be used for RS codes. Belief Propagation (BP) iterative decoding was first proposed for decoding low-density parity check (LDPC) codes. It has been shown that iterative decoders for very long codes can achieve performances close to Shannon capacity. Therefore, it would be ideal if Reed-Solomon codes are suitable for this class of decoders. Standard BP iterative decoding is not suitable for high density parity check codes like RS codes, because for these codes, there are a large number of short cycles in the factor graph which cause correlation between the messages and error propagation. In, the cyclic structure of RS codes is used and BP decoding is applied to a random shift of the received vector at each iteration to avoid error propagation. For short RS codes, the coding gain is significant but it diminishes for long
codes. Another bit-level decoding method based on belief propagation was proposed, but it was only efficient for very low rate RS code. An algorithm for removing all the 4-cycles in the factor graph of linear block codes is introduced. This method improves the suitability of iterative decoders for short low rate RS codes. The first successful bit-level iterative decoding method for RS codes was adaptive parity-check (ADP) algorithm.

In ADP, in order to make BP decoding effective for dense RS parity check matrix, Gaussian Elimination is performed on the binary parity check matrix before each iteration such that the variables of lowest reliability connect into the graph only once. It has been shown that this algorithm has very good performance compared to other soft-decision decoding algorithms for RS codes. The problem with ADP algorithm is that the required adaptation of the parity check matrix at each iteration is very complex. Recently more decoding methods based on BP algorithm have been proposed for general linear block codes. In multiple-bases belief-propagation for linear block codes with dense parity check matrices has been proposed. It makes use of the fact that a code has many structurally diverse parity-check matrices, capable of detecting different error patterns.

In this paper, for effective iterative decoding of RS codes, we use a fixed binary parity check matrix. This matrix is the extended version of the original parity check matrix with lower density and less number of 4-cycles. This parity check matrix representation is better suited for iterative decoders. We should mention that, parity check matrix extension is only applied in the receiver for the decoding without affecting the RS code itself (i.e. the transmitter). Even using the extended parity check matrix, the performance of normal BP decoding is not very good. In this paper, in order to improve the performance of normal BP decoding, we propose two new bit level iterative soft decision decoding methods for RS codes using the fixed extended binary parity check matrix.

II.SYSTEM MODEL

We use a Reed-Solomon (\(N\)) code over Galois Field (\(2^m\)) where \(N=2^m-1\). The parity check matrix of the code can be represented by:

\[
H = \begin{pmatrix}
1 & \alpha & \alpha^2 & \ldots & \alpha^{(N-1)} \\
1 & \alpha^2 & \alpha^4 & \ldots & \alpha^{2(N-1)} \\
1 & \alpha^{(N-K)} & \alpha^{2(N-K)} & \ldots & \alpha^{(N-K)(N-1)}
\end{pmatrix}
\] (1)
For any codeword $c$ of the RS code, $HcT=0$. Since any element $\beta \in (2m)$ has a $m$-tuple representation, we can show any codeword of length $Mn$ binary form:

$$c_0 = [c_1, c_2, \ldots, c_m, c'_{1}, c'_{2}, \ldots, c'_{m}, c_N, c_{N+1}, \ldots, c_N, c_{N+m}]$$

(2)

We assume BPSK modulation $x = -2cb + 1$ over AWGN channel. So, the received signal is:

$$y = x + n$$

Where $n$ is the AWGN vector with variance $\sigma^2$. The reliability of the received vector can be expressed in terms of the log likelihood ratios (LLR’s) that are given by $\rho = 2y/\sigma^2$. Here for the decoding, we consider RS codes over an extension field of $\mathbb{F}_2$. We denote a primitive polynomial of $(2m)$ over $(2)$ by $g(x) = x^{m} + \ldots + x^{2} + 1$. We also suppose that $a$ is a root of $(x)$ and therefore a primitive element in $\mathbb{F}_{2m}$. For $(x)$, there is a $m \times m$ companion matrix which is given as follows:

$$C_p = \begin{pmatrix}
0 & \ldots & 0 & a_0 \\
I_{m-1} & \ldots & I_{m-1} & a_1 \\
& \ldots & \vdots & \ddots & \ddots & \ddots \\
P_{m-1} & \ldots & P_{m-1} & \ldots & \ldots & \ldots & I_{m-1} & a_{m-1} \\
& & & & & & & & & & a_{m-1}
\end{pmatrix}$$

(3)

Where $I_{m-1}$ is an $(m - 1) \times (m - 1)$ identity matrix. A field isomorphism can be defined by the mapping $ai \rightarrow cip, i \in \{0, 1, \ldots\}$. Based on this mapping, each element of the parity check matrix of the code is replaced with a $m \times m$ binary matrix resulting a binary parity check matrix $H_{bc}$ of size $(N-K) \times Nm$. Such a mapping results in $H_{bc}Tb = 0$. This equation specifies the set of linear constraints satisfied by the codeword bits. Every linear code can also be represented by a bipartite graph where the set of variable nodes represents the codeword bits and the set of check nodes represents the set of parity-check constraints satisfied by the codeword bits.

There is also a set of edges that connect every check node with all the variables nodes involved in its check equation. We denote the number of the check equations that a variable node $i \in 1, 2, \ldots, m$ is involved with by $d_{vi}$ and refer to it as the degree of that variable node. Also, we denote the number of the variable nodes that are involved in a check equation $i \in 1, 2, \ldots, (N-K) \times m$ by $d_{vi}$ and refer to it as the degree of that check node. The (edge) degree distributions of the code are defined as:

$$\lambda(x) = \sum_{r=0}^{d_{va}} \lambda_{x} x^{r-1}$$

(4)

$$\rho(x) = \sum_{r=0}^{d_{kv}} \rho_{x} x^{r-1}$$

(5)
Here $A(\rho)$ is equal to the fraction of edges that connect to variable (check) nodes of degree $L$.

III. EFFICIENT ITERATIVE DECODING OF RS CODES

Using the extended parity check matrix discussed previously, we can perform phantom decoding of RS codes using normal BP iterations. However, the performance will be far away from the performance of the ML decoder because there are still large number of cycles in the factor graph of the code. Therefore, in this section we present two methods to improve the performance of phantom decoding.

Method A: Iterative Decoding of RS Codes Based on Their Cyclic Properties

Since RS codes are cyclic, any cyclically shifted version of a codeword is also a valid codeword. Therefore, we can consider a cyclically shifted version of the received signal by symbols $\phi\in\{0, -1\}$ as the received signal when a shifted version of symbols which is also a valid codeword was transmitted. Based on this fact, we can apply the BP algorithm on any cyclically shifted version of the received reliabilities with the same binary parity check matrix. At the end of each iteration, the updated reliabilities are shifted back to their original positions. Since for each cyclically shifted version of the reliabilities, their values are differently distributed, some deterministic errors can be avoided. So, the idea is to have outer rounds during the decoding. During each outer round, a different cyclically shifted version of the received reliabilities is generated and then used as the input to the BP decoding algorithm.

Denoting the low density extended parity check matrix by $H$, the proposed algorithm is described as follows:

Step 1) Inputs: $H$, $r_{\text{ch}} = 2y/\sigma^2$ and $\theta$.

Step 2) Initialization step: $c_{it} = 0$, $BP_{it} = 0$.

Step 3) $\phi = c_{it}$.

Step 4) cyclically shifting the channel LLR’s by symbols $\phi\in\{0, -1\}$ as the received signal when a shifted version of symbols which is also a valid codeword was transmitted.

Step 5) $\rho_{BP_{it}}(\cdot) = [00...0]_{1 \times (N_m + (N - K)m/2)}$ and $\rho_{BP_{it}}(1: N_m) = \rho_{\text{ch-shift}}(\cdot)$.

Step 6) Belief Propagation: feed $\rho_{BP_{it}}$ and the parity check matrix $H$ into the BP decoder and generate extrinsic LLR’s for each bit: $\rho_{x}$.

Step 7) Update the LLR of each bit: $\rho_{BP_{i+1}}(c_i) = \rho_{BP_{it}}(c_i) + \theta \rho_{x}(c_i)$

where $0 < \theta \leq 1$ is the damping coefficient.
Step 8) Hard Decisions, for \( i = 1, 2, \ldots \):
\[
\hat{c}_i = \begin{cases} 
0 & \rho_{\text{BP}, t+1}(c_i) > 0 \\
1 & \rho_{\text{BP}, t+1}(c_i) < 0 
\end{cases}
\]  
(6)

Step 9) If \( \hat{c} \) satisfies all the check equations, shift the decoded bits back to their original position: \( \hat{c} = \hat{c}(\varphi) \), save \( \hat{c} \), terminate the algorithm and go to step 12.

Else: Berlekamp-Massey (BM) decoding, \( c_B M = BM(\hat{c}) \) and if a decoding success was signaled, \( c_B M = c_B M(\varphi) \) and save \( c_B M \).

Step 10) Inner round (BP iterations):
if \( BPit = BPit_{\text{max}} \), go to step 11.
Else \( BPit = BPit + 1 \) and go to step 6.

Step 11) Outer round: if \( cit = cit_{\text{max}} \), terminate the algorithm and go to step 12.
Else \( cit = cit + 1 \), \( BPit = 0 \) and go to step 3.

Step 12) among all the codewords that were saved during the algorithm, choose the one with minimum Euclidean distance from the received vector.

The maximum number of outer rounds, \( cit_{\text{max}} \), can be chosen to be equal \( N - 1 \) such that all the possible cyclically shifted versions of the received reliabilities are considered as the input of BP decoding. It should be mentioned that most of the time; our algorithm finds the response very soon (during the first or the second outer round) and is terminated. So, there is no need to complete all of the outer rounds.

In the proposed algorithm we use both BP decoding and hard decision BM decoding. At the end of each BP iteration, the hard decision version of updated reliabilities is used as the input to the BM decoding algorithm. Even in the situations that BM algorithm can recover a codeword, we do not stop the algorithm just save that codeword. In the end, if the BP decoding was not successful, we choose one of the saved codewords with minimum Euclidean distance from the received vector as the response of the proposed iterative decoding method. Therefore, the block error rate of the method can be written as:

\[
P_B = P_B - BM \times P_B - BP
\]

Where \( P_B - BM \)s the block error rate of BM decoding method and \( P_B - BP \)s the block error rate of BP decoding with cyclic shifting of reliabilities.

In order to reduce the number of required BP iterations before finding the response of the algorithm, we define a radius \( r = \sqrt{N \sigma^2} \). Each time there is a decoding
success at the output of BM algorithm at the end of step (9), we check the distance of that codeword from the received vector $d = \| x^T \mathbf{M} - y \|$, where $x^T \mathbf{M} = -2c^T \mathbf{M} + 1$. If $d < r$, we terminate the algorithm and go to step (12). We refer to this process as extra stopping criterion ($d \leq r$). Of course by doing so, the performance of our proposed algorithm will be slightly worse than before. Therefore, we have a trade-off between the performance and the complexity.

As we will see in simulation results, the performance of iterative decoding using cyclic property of RS codes is very good and even for short RS codes close to the ML performance.

However, the disadvantage is that for low signal to noise ratios we still need to perform so many iterations of belief propagation. In the next section, we will present another method for efficient decoding of RS codes. Although the performance of this method will not be as good but it will require much less BP iterations.

**Analysis of Cyclic Shifting of RS Codes: Geometric Interpretation of the Proposed Algorithm**

The belief propagation based algorithms were shown to be special cases of the gradient descent algorithm. We define the potential function $\mathcal{J}$ as:

$$\mathcal{J}(H_b, T) = -\sum_{j=1}^{n-k} \gamma_j = -\sum_{j=1}^{n-k} \prod_{p=1}^{r} T_p$$

(7)

Where $\mathcal{J}$ is a function of both the $(n - k) \times n$ parity checks matrix $H_b$ and the received soft information $T$:

$$T = [T_1, T_2, ..., T_n] = [v(\rho(c_1)), ..., v(\rho(c_n))].$$

(8)

Here, the operator: $[-\infty, \infty] \rightarrow [-1, 1]$ is a mapping from the LLR domain to tanh domain:

$$v(\rho) = tanh(\frac{\rho}{2}) = \frac{1}{1 - e^{\rho/2}}.$$  

(9)

When all the checks are satisfied, we have reached a valid codeword and the potential function $\mathcal{J}$ is minimized. In this case, $|T_j| = 1$ for $j = 1$, and $\min |T_j| = -(n - k)$. In the case of RS codes, even using the extended parity check matrix, the density is still high and applying the iterative decoding might lead to some local minimum points called pseudo-equilibrium points. These points are not corresponded to valid codewords and the iterative algorithm gets stuck at them.

Actually, there are a few unreliable bits which do not let $\mathcal{J}$ to be minimum. Because $\mathcal{J}$ is a function of both $H_b$ and $T$ (bit reliabilities), different arrangements of the reliabilities with
the same $H_b$, result in different potential functions $J$. The proposed algorithm uses this fact and when a pseudo-equilibrium point is reached, we change the arrangement of bit reliabilities using cyclic shifting by $\varphi$ symbols ($\varphi = 0, -1$), such that it allows the update to proceed rather than getting stuck at the pseudo equilibrium point. In Fig. 2, we have shown the potential function $J$ versus the number of iterations for three outer rounds during the iterative decoding of RS(15,11) code over AWGN channel with $E_b/N_0=4$ dB. As it can be seen in this figure, during the first outer round, BP decoding gets stuck in a pseudo-equilibrium point and can not converge to a codeword. However, by cyclically shifting the reliabilities by one symbol during the second round, BP decoding converges to the correct codeword. Here, for RS(15,11) code, the extended parity check matrix is an $r \times n$ matrix with $r = (N - K)m + (N - K)m_2$ and therefore $j_{\text{min}} = -r = -24$ that has been reached during the second round.

Method B: Iterative Decoding of RS Codes Based on Modified Belief Propagation

In this section, we introduce another iterative method based on improving the normal belief propagation technique. As we mentioned before, even using the extended binary parity check matrix, there will be a big gap between normal belief propagation decoding of RS codes and ML decoding. This is because of the large number of 4-cycles in the factor graph of the code. The new method is based on information correction in the BP decoding algorithm [20]. In this method, first normal BP decoding is performed for $B\text{Pit} - \max$ iterations. During these iterations, the average decoded LLRs are recorded:

$$r_{\text{sum}}^{B\text{Pit} - \max}(c_j) = \frac{1}{B\text{Pit} - \max} \sum_{t=1}^{B\text{Pit} - \max} r_t(c_j),$$

$$j = 1, ..., Nm + \frac{(N-K)m}{2} \quad (10)$$

Where $(c_j)$ is the reliability of the $j$th bit after $\text{iter}$ iterations. If a codeword is reached during these iterations, we terminate the algorithm. However, if decoding was not successful at the end of $B\text{Pit} - \max$ iterations, $\text{ICmax}$ steps of information correction are performed:

Step 1) Based on the average LLRs, we select the least reliable bit:

$$c_p = \arg \min_{c_j} \left| r_{\text{sum}}^{B\text{Pit} - \max}(c_j) \right|. \quad (11)$$

Changing the channel information of the selected bit very likely improves the convergence of the decoder to a codeword. Therefore, we perform two tests by setting $\rho_{ch}$
If a valid codeword is reached at any step, the decoding is terminated. Here, like the previous algorithm, at the end of each BP iteration, we perform BM decoding using the updated reliabilities. However, if the BM decoding is successful, we do not stop the algorithm just save that codeword. In the end, we choose one of the saved codewords with minimum Euclidean distance from the received vector as the decoding response. Using this strategy we can enforce the correct values on the selected bits and eliminate pseudo-codewords. Therefore, the chance of BP iterations to converge to a codeword is improved significantly. It should be noted that by increasing the signal to noise ratio, the information correction technique is more effective, because at higher $E_b/\sigma^2$, there are less bit errors at the channel output.

As we mentioned before, there are considerable short cycles even in the factor graph of the extended parity check matrix.

Therefore, there will be correlation between the messages and the values of high reliable bits may be affected significantly by the values of low reliable bits. To avoid this, we can select the highest reliable bits using a threshold value $\tau$.
We then fix the values of these selected bits using hard decision and do not update them during the BP iterations:

\[ \rho^{ch}(c_j) = \text{sign}(\rho^{ch}(c_j)) \times \inf, \quad \forall c_j \in C_h. \]

If the threshold is selected appropriately, the selected bits will have correct channel information with high probability. This way, high reliable correct bits are safe and their values are not affected in any ways. In method B, we also add the extra stopping criterion used in the first method in order to reduce the number of required BP iterations.

**Complexity Analysis of the Proposed Algorithms**

For both algorithms, the extended parity check matrix is used. The complexity of belief propagation in each iteration is proportional to the number of nonzero elements in the parity check matrix of the code. Because the density of the extended binary parity check matrix is around 15%, the time complexity of each BP iteration is:

\[ O(0.15 \times re \times ne) \]

Where \( re = (N - K) + (N - K/2) \) and \( ne = Nm + (N - K)m/2 \).

In the ADP method, each iteration involves \( (N2m2) \) floating point operations for sorting and BP, and \( (Nm \times (K2m2), (N - K)/2m2) \) binary operations for Gaussian Elimination. So, the complexity of each iteration of our methods is less than ADP method. The maximum number of BP iterations in the first method based on cyclic shifting of RS codes is:

\[ c_{it-min} \times BP_{it-min}. \]  

(13)

However, most of the times, we find the response very soon (during the first or the second outer round) and we terminate the algorithm. So, the actual complexity will be much less than the maximum complexity. In the next section that we provide simulation results for different RS codes, we will also give a table of the average number of required BP iterations for different signal to noise ratios.

The maximum number of BP iterations in the second algorithm based on information correction is:

\[ BP_{it-min} + BP_{it-add} \times \left( \sum_{j=1}^{Kmax} 2^j \right) \]

(14)

Here also most of the times, we find the response very soon and the actual complexity is
much less than the maximum complexity. During the simulation results for different RS codes, the average number of required BP iterations for different signal to noise ratios is provided.

SIMULATION RESULTS AND DISCUSSIONS

In this section, simulation results for decoding of RS codes based on the proposed algorithms are presented. For all the simulations, we suppose BPSK transmission over AWGN channel. We set the maximum number of BP iterations during the inner round of the first algorithm to be equal 60. So, the maximum number of iterations will be $60N$. For the second algorithm based on information correction, the maximum number of initial BP iterations is set to $BP_{\text{it-max}} = 60$. We will have 4 steps of information correction and $BP_{\text{add}} = 10$. So, the maximum number of iterations using equation will be 360. It is clear that the maximum number of iterations in the second algorithm is much less than the first one. We compare our algorithms with Berlekamp-Massey (BM) hard decision decoding method [1] and also the algebraic soft decision decoding proposed by Koetter and Vardy [3] [22] that will be mentioned by KV algorithm. We will also compare our results with phantom decoding [23] and ADP method [13]. It is also important to compare our algorithms with the best performance possible, which is that of the ML decoder.

The weight enumerator of an RS code under a specific binary image expansion is not known. The averaged ensemble of the RS code can be found by averaging over all possible binary expansions [26]. The averaged binary weight enumerator can then be used by the Divsalar simple bound [27] to bound the ML error probability [13].

In Fig. 1, we have shown the performances of our proposed methods. From this figure, at codeword error rate of $10^{-3}$, decoding using method A will provide more than 1.5 dB coding gain compared to the asymptotic performance of KV algorithm, 2.7 dB compared to the BM algorithm and 1.7 dB compared to phantom decoding. This method
has also about 0.5 dB coding gain compared to the performance of ADPBM (5,1) [13]. The performance of method B is also shown in this figure which is almost the same as method A with extra stopping criterion ($d \leq r$). It has also 1.4 dB coding gain compared to phantom decoding. In order to consider the complexity, Table I shows the average number of required BP iterations during the decoding for both methods. By comparing the complexity and the performance of method A with and without extra stopping criterion, we see that adding the extra stopping criterion ($d \leq r$) causes about 0.2 dB performance loss and at the same time reduces the average number of required BP iterations considerably. Therefore, for longer RS codes, we always add this extra stopping criterion.

From Fig. 2, decoding using algorithm A with the extra stopping criterion ($d \leq r$) will provide more than 1.5 Db coding gain compared to asymptotic performance of KV algorithm, about 2.5 dB compared to the BM algorithm and 1 dB compared to phantom decoding at codeword error rate of $10^{-4}$. Also, method A has about 0.5 dB coding gain compared to ADP-BM (5,1) and 0.2 dB compared to ADP-BM (20,1). From this figure, at codeword error rate of $10^{-4}$, method B has 0.5 dB coding gain compared to phantom decoding. The performance of method B is not as good as method A and it is close to the performance of ADP-BM (5,1). However, as we can see in Table II, the complexity of method B is much less than method A especially at low signal to noise ratios.

Fig 2: Performance of the proposed algorithms for RS (31,25)

Fig 3: Performance of the proposed algorithms for RS (63,55)
From Fig.3, at codeword error rate of $10^{-3}$, method A provides better performance (about 0.2 dB coding gain) compared to ADP-BM (5,1) algorithm. Also it provides a coding gain of about 1.75 dB compared to the BM algorithm and 0.75 dB compared to phantom decoding. The performance of method A is about 0.25 dB away from the performance of ADP-BM (20,3). Method B has about 1.35 dB coding gain over BM algorithm and 0.3 dB over phantom decoding.

From Fig. 4, at codeword error rate of $10^{-4}$, method A provides better performance (about 0.15 dB coding gain) compared to ADP-BM (5,1) algorithm. Also it provides a coding gain of about 1.05 dB compared to the BM algorithm, 0.5 dB compared to KV algorithm and 0.7 dB compared to phantom decoding. Method B has about 0.65 dB coding gain over BM algorithm, 0.2 dB over phantom decoding and its performance is very close to the performance of KV algorithm. The performance of method A is about 0.25 dB away from the performance of ADP-BM (20,3) at codeword error rate of $10^{-3}$. However, as discussed in the complexity analysis section, the complexity of our methods is much less than ADP method mainly because we do not perform Gaussian eliminations at every iteration.

### IV. CONCLUSION

We proposed two soft decision decoding algorithms based on bit level belief propagation decoding for RS codes. The advantage of our methods over ADP method [13] is that they work with the fixed parity check matrix. In both methods, we used an extended binary parity check matrix with lower density and reduced number of 4-cycles compared to the original binary parity check matrix of the code. We investigated the performance of normal BP decoding with different binary parity check matrices over binary erasure channel and we realized that the extended binary parity check matrix provides better performance. The first method was based on cyclic property of RS codes and we also...
presented geometric interpretation for that. Simulation results showed that our method has significant gain over hard decision decoding. The performance was also superior to some other popular soft decision decoding methods including KV method and ADP method. The second method was based on information correction in BP decoding. Compared to the first method, method B needed less BP iterations but its performance was not as good. We also presented complexity analysis for both methods.

REFERENCES


