Iterative Soft-Decision Decoding of Hermitian Codes

Li Chen, *Member, IEEE*

***Abstract*—This paper proposes an iterative soft-decision decoding algorithm for one of the most popular algebraic-geometric (AG) codes – Hermitian codes. The algorithm is designed by integrating the two most powerful soft-decision decoding algorithms, the adaptive belief propagation (ABP) algorithm and the Koetter-Vardy (KV) list decoding algorithm. The ABP algorithm performs iterative decoding based on an adapted parity-check matrix of a Hermitian code to enhance the reliability of the soft received information. With the enhanced reliability, the KV algorithm performs soft-decision list decoding to obtain the original message. Since the matrix adaptation relies on bit reliabilities, regrouping of the unreliable bits is introduced to assist the ABP decoding. A complexity reducing ABP-KV decoding approach is proposed based on assessing the soft information provided by the ABP algorithm and determining whether the following KV decoding steps should be carried out. Geometric interpretation of the ABP algorithm is presented, demonstrating the necessity of performing matrix adaptation. Our performance analysis shows the proposed iterative decoding algorithm outperforms both the existing decoding approaches for Hermitian codes and the ABP-KV decoding of Reed-Solomon (RS) codes.**

***Index Terms*—Adaptive belief propagation, algebraic-geometric codes, complexity reduction, Hermitian codes, iterative decoding, Koetter-Vardy algorithm, list decoding, Reed-Solomon Codes.**

I. Introduction

REED-Solomon (RS) codes are widely used for error-correction in modern communication and data storage systems. However, the length of a RS code cannot exceed the size of the finite field over which it is defined, limiting the number of codes and its error-correction capability. Being able to design a larger code from a moderate size finite field is of great interest to both academia and industry. Algebraic-geometric (AG) codes [1] are the obvious candidates to fulfill such a demand, among which Hermitian codes are the most celebrated AG codes. Recent work [2] - [8] showed that thanks to their longer code length, the Hermitian codes can outperform the RS codes that are defined over the same finite field.

Paper approved by A. Graell i Amat, the Editor for Coding and Communication Theory of the IEEE Communications Society. Manuscript received December 25, 2011; revised June 17, 2012.

This paper was presented in part at the IEEE Information Theory Workshop, Lausanne, Switzerland, Sep. 2012.

The author is with the School of Information Science and Technology, Sun Yat-sen University, Guangzhou, China, 510006 (e-mail: chenli55@mail.sysu.edu.cn); website: sist.sysu.edu.cn/∼chenli.

This work is sponsored by the National Natural Science Foundation of China (NSFC) with project ID: 61001094, the National Basic Research Program of China (973 program) with project ID: 2012CB316100, and the Guangdong Natural Science Foundation (GDNSF) with project ID: 10451027501005078.

Digital Object Identifier 10.1109/TCOMM.2012.100512.110871

The theoretical framework on the efficient decoding of Hermitian codes was proposed by Sakata *et al.* [9]. Combined with the majority voting [10], the Sakata algorithm can correct symbol errors up to half of the code's designed minimum distance. Guruswami and Sudan [11] [12] later proposed a polynomial-time list decoding algorithm (or the so called GS algorithm) for both the RS and AG codes, correcting errors beyond the half distance bound. By defining the interpolation property of a trivariate polynomial that is defined over the pole basis of a Hermitian curve, Hoholdt and Nielsen [13] [14] presented a list decoding algorithm for Hermitian codes. Their performance was first evaluated by Chen *et al.* [4], who later presented a more efficient list decoding algorithm for Hermitian codes [5]. Recently, soft-decision list decoding of Hermitian codes was introduced by Chen *et al.* [6] [7] and Lee *et al.* [8] independently.

At the same time, soft-decision decoding of RS codes is a very active area of research. Earlier attempts to soft decode RS codes include the generalized minimum distance (GMD) algorithm [15] and the Chase algorithm [16]. The maximal likelihood (ML) decoding of RS codes was proposed by Vardy and Be'ery [17] which utilizes the binary image expansion of the RS code. A reduced complexity variant was later proposed by Ponnampalam and Vucetic [18]. However, the complexity of such a decoding approach grows exponentially with the length of the code, preventing its practical application to large RS codes. Using the code's binary image, another type of soft decoding approach called the ordered statistics decoding was proposed by Fossorier and Lin [19]. It later evolved to include the reliability based hybrid decoding algorithm [20] and the box and match algorithm [21]. The well known Koetter and Vardy's soft-decision list decoding algorithm [22] has a polynomial-time complexity and offers a significant performance gain over the hard-decision decoding. Iterative soft decoding of RS codes using the popular belief propagation (BP) algorithm was proposed by Jiang and Narayanan [23] [24]. The BP algorithm [25] is performed based on an adapted parity-check matrix, known as the adaptive BP (ABP) algorithm. Its output will be utilized by the hard-decision decoding. Further improvement was proposed by El-Khamy and McEliece [26], in which the ABP soft output is utilized by the Koetter-Vardy (KV) list decoding algorithm. Results of [24] [26] showed the ML performance bound of RS codes is approached with a moderate decoding complexity.

As a possible candidate to replace RS codes, iterative decoding of Hermitian codes is still unknown in the literature. So far, their best error-correction performance was achieved by using the KV algorithm [6] [8]. Therefore, a more sophisticated

0090-6778/13$31.00 © 2013 IEEE

decoding approach that can fully utilize the soft information would be desirable. This paper proposes the first iterative soft-decision decoding algorithm for Hermitian codes. Similar to the iterative soft decoding of RS codes [26], it is designed by combining the ABP and KV algorithms. The ABP algorithm performs the first stage decoding to enhance the reliability of the received information. It will then be passed to the second stage decoding that is the KV decoding algorithm. The parity-check matrix of the Hermitian code is defined and Gaussian-Jordan (GJ) elimination is employed according to the bit reliability to reduce its density and eliminate part of its short cycles. To improve the performance, regrouping of the unreliable bits is also introduced. Since the KV algorithm is computationally expensive, this paper introduces a complexity reducing ABP-KV algorithm that uses a couple of successive criteria to assess the quality of the soft information provided by the ABP algorithm. It excludes the deployment of the unnecessary KV decoding steps. Geometric interpretation of the ABP algorithm is presented, demonstrating the necessity of performing matrix adaptation. The performance analysis of the proposed algorithm will be provided, comparing it with the existing decoding approaches for Hermitian codes and the ABP-KV decoding of RS codes.

The rest of the paper is organized as follows. Section II presents some background knowledge on this work. Section III presents the iterative soft-decision decoding algorithm for Hermitian codes. A complexity reducing ABP-KV algorithm will be presented in Section IV. Section V presents the geometric interpretation of the ABP algorithm. Section VI provides the performance analysis and Section VII concludes the paper.

II. Background Knowledge

This section presents some background knowledge on the Hermitian codes and an overview of the KV algorithm.

A. Hermitian Codes

Let $\mathbb{F}_q$ denote the finite field of size q and $\mathbb{F}_q = \{0, 1, \alpha, \alpha^2, \ldots, \alpha^{q-2}\}$, where α is a primitive element. In this paper, it is assumed $\mathbb{F}_q$ is an extension field of $\mathbb{F}_2$ such that $q = 2^{\varpi}$ where ϖ is an even number. Let $\mathbb{F}_q[x]$, $\mathbb{F}_q[x, y]$ and $\mathbb{F}_q[x, y, z]$ denote the rings of univariate, bivariate and trivariate polynomials defined over $\mathbb{F}_q$, respectively. The Hermitian curve that is defined in $\mathbb{F}_q$ can be written as [1]:

$$H_w(x, y, z) = x^{w+1} + y^w z + y z^w, \tag{1}$$

where $w = \sqrt{q}$. The construction of a Hermitian code can be elaborated from one of its affine components $H_w(x, y, 1)$. There are $n = w^3$ affine points $p_j = (x_j, y_j, 1)$ $(1 \le j \le n)$ and a point at infinity $p_\infty = (0, 1, 0)$ [1] [14] [27]. Pole basis Φ_w consists of bivariate monomials $\phi_a = x^\delta y^\lambda$ $(0 \le \delta \le w, \lambda \ge 0)$ with an increasing pole order that is defined as $\upsilon_{p_\infty}(\phi_a^{-1}) = \upsilon_{p_\infty}((x^\delta y^\lambda)^{-1}) = w\delta + (w+1)\lambda$. Consequently, pole basis Φ_w can be defined as [5] [27]:

$$\Phi_w = \{\phi_a \mid \upsilon_{p_\infty}(\phi_a^{-1}) < \upsilon_{p_\infty}(\phi_{a+1}^{-1}),\ a \in \mathbb{N}\}, \tag{2}$$

where $\mathbb{N}$ denotes the set of nonnegative integers. E.g., $\Phi_2 = \{1, x, y, x^2, xy, y^2, x^2y, xy^2, y^3, \ldots\}$.

Since all the affine points can be distinguished by their x and y components, they can be simplified as $p_j = (x_j, y_j)$. With knowledge of the affine points p_j and the pole basis Φ_w, the generator matrix $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ of an (n, k) Hermitian code can be defined as:

$$\mathbf{G} = \begin{pmatrix} \phi_0(p_1) & \phi_0(p_2) & \cdots & \phi_0(p_n) \\ \phi_1(p_1) & \phi_1(p_2) & \cdots & \phi_1(p_n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{k-1}(p_1) & \phi_{k-1}(p_2) & \cdots & \phi_{k-1}(p_n) \end{pmatrix}, \tag{3}$$

where n and k are the length and dimension of the code, respectively. Given a message vector $\overline{F} = [F_1, F_2, \ldots, F_k] \in \mathbb{F}_q^k$, the codeword $\overline{C}$ can be generated by:

$$\overline{C} = [C_1, C_2, \ldots, C_n] = \overline{F} \cdot \mathbf{G}, \tag{4}$$

where $\overline{C} \in \mathbb{F}_q^n$. Note that vector $\overline{F}$ can be represented by a polynomial $F(x, y) = \sum_{j=1}^{k} F_j \phi_{j-1}$. The encoding process can be interpreted as evaluating the n affine points over the message polynomial. Note that the length of the Hermitian code is $n = q^{3/2}$ which is larger than that of the RS code defined over the same finite field. Its parity-check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k) \times n}$ is defined as:

$$\mathbf{H} = \begin{pmatrix} \phi_0(p_1) & \phi_0(p_2) & \cdots & \phi_0(p_n) \\ \phi_1(p_1) & \phi_1(p_2) & \cdots & \phi_1(p_n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n-k-1}(p_1) & \phi_{n-k-1}(p_2) & \cdots & \phi_{n-k-1}(p_n) \end{pmatrix}. \tag{5}$$

With a valid codeword $\overline{C}$, we have $\overline{C} \cdot \mathbf{H}^T = \mathbf{0}$ where $\mathbf{0}$ represents the all-zero matrix.

In order to perform ABP decoding of Hermitian codes, the binary image of its parity-check matrix is required. Let $\sigma(x) \in \mathbb{F}_2[x]$ be a primitive polynomial of $\mathbb{F}_q$ and $\mathbf{A} \in \mathbb{F}_2^{\varpi \times \varpi}$ be its companion matrix [28], for a field element α^t $(t = 0, 1, 2, \ldots, q-2)$, mapping $\alpha^t \mapsto \mathbf{A}^t$ is applied to obtain its binary image. Consequently, the binary image of a parity-check matrix can be generated by replacing its entries α^t by their corresponding matrices $\mathbf{A}^t$. We use $\mathbf{H}_b$ to denote such a binary parity-check matrix. Let $N = n\varpi$ and $K = k\varpi$, $\mathbf{H}_b \in \mathbb{F}_2^{(N-K) \times N}$. Let $\bar{c}$ denote the binary representation of codeword $\overline{C}$ as $\bar{c} = [c_1, c_2, \ldots, c_N]$, where we also have $\bar{c} \cdot \mathbf{H}_b^T = \mathbf{0}$.

B. The Koetter-Vardy Algorithm

The KV algorithm utilizes the soft information provided by the channel for decoding. The channel observations form a reliability matrix $\mathbf{\Pi} \in \mathbb{R}^{q \times n}$ whose entries π_{ij} represent the *a posteriori* probabilities (APP) of the transmitted codeword symbol C_j being $\rho_i \in \mathbb{F}_q$, where $\mathbb{R}$ denotes real numbers. Matrix $\mathbf{\Pi}$ is then transformed into a multiplicity matrix $\mathbf{M} \in \mathbb{N}^{q \times n}$ whose entries m_{ij} represent the interpolation multiplicity w.r.t. a unit (p_j, ρ_i). Interpolation is then carried out based on $\mathbf{M}$, delivering the interpolated polynomial $Q \in \mathbb{F}_q[x, y, z]$. Factorization will then be performed to determine the z-roots of Q, which are the decoded output candidates.

Definition I: Let $i_j = \text{index}\{\rho_i \mid \rho_i = C_j\}$ such that $\overline{C} = [\rho_{i_1}, \rho_{i_2}, \ldots, \rho_{i_n}]$. The reliability-based codeword score

is defined as [6]:

$$S_{\mathbf{\Pi}}(\overline{C}) = \sum_{j=1}^{n} \pi_{i_j j}. \tag{6}$$

Similarly, the multiplicity-based codeword score is defined as:

$$S_{\mathbf{M}}(\overline{C}) = \sum_{j=1}^{n} m_{i_j j}. \tag{7}$$

For list decoding of an (n, k) Hermitian code, the weight of variable z is defined as $w_z = \upsilon_{p_\infty}(\phi_{k-1}^{-1})$. Consequently, the $(1, w_z)$-weighted degree of monomial $\phi_a z^b$ is: $\deg_{1,w_z}(\phi_a z^b) = \upsilon_{p_\infty}(\phi_a^{-1}) + b \cdot w_z$. With the $(1, w_z)$-lexicographic order [6], if $\phi_{a'} z^{b'}$ is the maximal monomial of polynomial Q with coefficient $Q_{a'b'} \neq 0$, the $(1, w_z)$-weighted degree of Q is $\deg_{1,w_z}(Q) = \deg_{1,w_z}(\phi_{a'} z^{b'})$. The following two theorems define the successful KV decoding conditions.

Theorem 1: If the multiplicity-based codeword score is large enough such that:

$$S_{\mathbf{M}}(\overline{C}) > \deg_{1,w_z}(Q), \tag{8}$$

the message polynomial $F(x, y)$ can be found by determining the z-roots of Q [6].

By increasing the interpolation multiplicity, the KV decoding performance can be enhanced. Let $\mathcal{C}(\mathbf{M}) = \frac{1}{2}\sum_{i,j} m_{ij}(m_{ij} + 1)$ denote the number of interpolation constraints, the asymptotically optimal performance of the KV algorithm can be achieved as $\mathcal{C}(\mathbf{M}) \to \infty$.

Theorem 2: If the reliability-based codeword score is large enough such that:

$$S_{\mathbf{\Pi}}(\overline{C}) > \sqrt{w_z \sum_{i,j} \pi_{ij}^2}, \tag{9}$$

the message polynomial can be found with a sufficiently large interpolation cost $\mathcal{C}(\mathbf{M})$ [6].

The above theorems show that the optimal soft-decision list decoding performance is dictated by the reliability matrix $\mathbf{\Pi}$. Its performance can be improved by enhancing $\mathbf{\Pi}$.

III. Iterative Soft-Decision Decoding

This section presents the iterative soft-decision decoding algorithm for Hermitian codes, including the proposed ABP-KV decoding algorithm and the regrouping of unreliable bits that assists the ABP-KV decoding. The proposed iterative decoding approach consists of two decoding stages. The first stage is the ABP decoding, supplying a number of updated reliability matrices $\mathbf{\Pi}'$ for the following KV decoding process. The updated matrix $\mathbf{\Pi}'$ will also be given as feedback for the next round of ABP decoding. With each updated matrix $\mathbf{\Pi}'$, the KV algorithm is performed to determine a list of output candidates $P(x, y)$ that are in the form of $F(x, y)$. They are stored in the global output list $\mathcal{L}$. After the predefined decoding parameters are reached, the decoding will be terminated. The ML selection criterion is applied to $\mathcal{L}$ and picks out the output candidate whose codeword has the minimal Euclidean distance to the received vector.

A. The ABP-KV Decoding

It is assumed the binary phase shift keying (BPSK) scheme is used to map the coded bits c_j to the modulated symbols $s_j \in \{+1, -1\}$ for $j = 1, 2, \ldots, N$. After the channel, a received vector $[y_1, y_2, \ldots, y_N] \in \mathbb{R}^N$ is obtained. The log-likelihood ratio (LLR) value of c_j is determined by:

$$L(c_j) = \ln \frac{\Pr[c_j = 0 | y_j]}{\Pr[c_j = 1 | y_j]}, \tag{10}$$

where $\Pr[c_j = 0|y_j]$ and $\Pr[c_j = 1|y_j]$ are the APP values. The LLR vector $\overline{L}$ that collects all the LLR values of the coded bits is:

$$\overline{L} = [L(c_1), L(c_2), \ldots, L(c_{N-K}), \ldots, L(c_N)]. \tag{11}$$

The magnitude $|L(c_j)|$ represents the reliability of bit c_j, where a higher magnitude implies the bit is more reliable. Hence, all the magnitudes $|L(c_j)|$ will be sorted in an ascending order. This yields a new bit index sequence $j_1, j_2, \ldots, j_{N-K}, \ldots, j_N$ with

$$|L(c_{j_1})| < |L(c_{j_2})| < \ldots < |L(c_{j_{N-K}})| < \cdots < |L(c_{j_N})|. \tag{12}$$

Let $B \subseteq \{1, 2, \ldots, N\}$ be a set of the bit indices and $|B| = N-K$. With $B = \{j_1, j_2, \ldots, j_{N-K}\}$ that collects the indices of the $N-K$ least reliable bits, the sorted LLR vector becomes

$$\overline{L}_B = [\underbrace{L(c_{j_1}), L(c_{j_2}), \ldots, L(c_{j_{N-K}})}_{B}, \underbrace{L(c_{j_{N-K+1}}), \cdots, L(c_{j_N})}_{B^c}]. \tag{13}$$

Note that the complementary set $B^c = \{1, 2, \ldots, N\} \setminus B$. For matrix $\mathbf{H}_b$, GJ elimination will be performed on the columns that correspond to the bits of B. Let Υ_j denote the weight-1 column vector with 1 at its jth entry and 0 elsewhere. GJ elimination reduces column j_1 to Υ_1, then reduces column j_2 to Υ_2 and etc. It attempts to reduce the first $N-K$ independent columns implied by B to the weight-1 columns. But it is not guaranteed all the columns w.r.t. B can be reduced. In that case, the columns w.r.t. the border bits between sets B and B^c will be reduced. This process is called matrix adaptation, resulting in an updated binary parity-check matrix $\mathbf{H}'_b$.

Let h_{ij} denote the entry of matrix $\mathbf{H}'_b$. The conventional BP algorithm will now be applied to $\mathbf{H}'_b$. Let us define $I(j)$ and $J(i)$ as:

$$I(j) \triangleq \{i \mid h_{ij} = 1, \ \forall \ h_{ij} \in \mathbf{H}'_b\}, \tag{14}$$

$$J(i) \triangleq \{j \mid h_{ij} = 1, \ \forall \ h_{ij} \in \mathbf{H}'_b\}. \tag{15}$$

Let matrices $\mathbf{V}$, $\mathbf{U} \in \mathbb{R}^{(N-K)\times N}$ with entries v_{ij} and u_{ij}, respectively. At the beginning of the BP decoding, matrix $\mathbf{V}$ is initialized as:

$$v_{ij} = L(c_j) \cdot h_{ij}, \forall \ 1 \leq i \leq N-K, 1 \leq j \leq N. \tag{16}$$

First, the horizontal step will be performed to update matrix $\mathbf{U}$ as:

$$u_{ij} = 2 \tanh^{-1}\Big(\prod_{\tau \in J(i)\setminus j} \tanh\Big(\frac{v_{i\tau}}{2}\Big)\Big). \tag{17}$$

Afterwards, the vertical step will be performed to update matrix $\mathbf{V}$ as:

$$v_{ij} = L(c_j) + \eta \sum_{\tau \in I(j) \backslash i} u_{\tau j}, \quad (18)$$

where $0 < \eta \leq 1$ is the damping factor [26]. The extrinsic information of bit c_j is given by:

$$L_{ext}(c_j) = \sum_{\tau \in I(j)} u_{\tau j}. \quad (19)$$

Calculations of (17)-(18) define one iteration of BP decoding. Let $\mathcal{N}_{\text{BP}}$ denote the pre-determined number of BP iterations. Once $\mathcal{N}_{\text{BP}}$ is reached, the LLR value of bit c_j is updated by:

$$L'(c_j) = L(c_j) + \eta L_{ext}(c_j). \quad (20)$$

As a result, the updated LLR vector $\overline{L'}$ can be formed as follows:

$$\overline{L'} = [L'(c_1), L'(c_2), \ldots, L'(c_{N-K}), \ldots, L'(c_N)]. \quad (21)$$

A hard-decision for bit c_j can be made based on $\hat{c}_j = \text{sign}(L'(c_j))$. Note that given any random variable ψ, $\text{sign}(\psi) = 0$ if $\psi \geq 0$, or $\text{sign}(\psi) = 1$ otherwise.

Multiple matrix adaptations can be performed and each of them is followed by a number of BP iterations. If the next round of matrix adaptation is to be carried out, the LLR sorting process will be performed based on the updated LLR vector $\overline{L'}$. Given $\mathcal{N}_{\text{ADP}}$ as the number of matrix adaptations, the total number of BP iterations becomes $\mathcal{N}_{\text{ADP}}\mathcal{N}_{\text{BP}}$. The motivation of performing matrix adaptation is two-fold. On one hand, the density[1] of the original parity-check matrix $\mathbf{H}_b$ is reduced and part of its short cycles are eliminated. For example, the density of $\mathbf{H}_b$ is about 50%, while the density of $\mathbf{H}'_b$ is about 37%. On the other hand, it prevents the propagation of the unreliable information during the BP decoding process. For example, the horizontal update of the reliable bits will only involve few unreliable bits. At the same time, the LLR update for the unreliable bits will take the LLR values of most of the reliable bits into account and their reliabilities are more likely to be enhanced.

One can make a hard-decision on each coded bit c_j. If $[\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_N] \cdot \mathbf{H}'^T_b = \mathbf{0}$, the decoding will be terminated. However, this decoding approach does not provide a good error-correction performance [24]. In order to fully utilize the soft outputs of the ABP algorithm, they will be passed to another soft decoding algorithm, i.e., the KV algorithm. Each updated LLR value $L'(c_j)$ will now be converted back to a pair of APP values using:

$$\Pr[c_j = 0|y_j] = \frac{1}{1 + e^{-L'(c_j)}}, \ \Pr[c_j = 1|y_j] = \frac{1}{1 + e^{L'(c_j)}}. \quad (22)$$

Based on the binary decomposition of each field element ρ_i, every ϖ consecutive pairs of APP values will be multiplied in q different permutations, generating a column of reliability values π'_{ij} of matrix $\mathbf{\Pi}'$. Matrix $\mathbf{\Pi}'$ will be transformed into a multiplicity matrix $\mathbf{M}$ [6]. Interpolation will be carried out, yielding the interpolated polynomial $Q \in \mathbb{F}_q[x, y, z]$ [5] [14]:

$$Q(x, y, z) = \sum_{a,b \in \mathbb{N}} Q_{ab} \phi_a(x, y) z^b. \quad (23)$$

Factorization will then be carried out [29] [30] [31]. By increasing the factorization output list size l, i.e., the z-degree of Q, the KV algorithm will have a better error-correction capability.

The KV algorithm will be performed after every $\mathcal{N}_{\text{BP}}$ BP iterations. With $\mathcal{N}_{\text{ADP}}$ matrix adaptations, the ABP-KV algorithm produces at most $l\mathcal{N}_{\text{ADP}}$ output candidates in the global list $\mathcal{L}$.

B. Regrouping of Unreliable Bits

The above description shows that those bits whose corresponding columns fall into the identity submatrix of $\mathbf{H}'_b$ are more likely to be corrected by the BP decoding. The ABP algorithm enables bits of B to have the priority to be corrected. However, it is possible that bits of B^c are wrongly estimated by their LLR values. If their corresponding columns can be reduced to weight-1, they are also likely to be corrected. Therefore, after the initial sorting process, we can restructure the sorted LLR vector and enable bits of B^c to fall into its first $N - K$ positions. So that, their corresponding columns will be reduced by the following GJ elimination. Such a process creates different groups of bits whose corresponding columns will be reduced. They are indicated by set B and the ABP-KV algorithm will be performed based on different patterns of $\overline{L}_B$ [24] [26].

Let $\mathcal{N}_{\text{GR}}$ denote the designed number of unreliable groups after restructuring vector $\overline{L}_B$ and $r = \lfloor N/\mathcal{N}_{\text{GR}} \rfloor$. The original sorted LLR vector $\overline{L}_B$ can be expressed as:

$$\begin{aligned} \overline{L}_B \ = \ & [L(c_{j_1}), \ldots, L(c_{j_r}), L(c_{j_{r+1}}), \ldots, L(c_{j_{2r}}), \ldots, \\ & L(c_{j_{(g-1)r+1}}), \ldots, L(c_{j_{gr}}), \ldots, L(c_{j_N})], \end{aligned} \quad (24)$$

where g $(1 \leq g \leq \mathcal{N}_{\text{GR}})$ is the group index. Let $\overline{L}_{B^{(g)}}$ denote the restructured LLR vector of group g. For group 1, $\overline{L}_{B^{(1)}} = \overline{L}_B$ since no restructuring is needed. For group g $(g > 1)$, we will restructure $\overline{L}_B$. If $r < N - K$, $B^{(g)} = \{j_{(g-1)r+1}, \ldots, j_{gr}, j_1, \ldots, j_{N-K-r}\}$ and

$$\begin{aligned} \overline{L}_{B^{(g)}} \ = \ & [\underbrace{L(c_{j_{(g-1)r+1}}), \ldots, L(c_{j_{N-K-r}})}_{B^{(g)}}, L(c_{j_{N-K-r+1}}), \\ & \ldots, L(c_{j_{(g-1)r}}), L(c_{j_{gr+1}}), \ldots, L(c_{j_N})]. \end{aligned} \quad (25)$$

If $r \geq N - K$, $B^{(g)} = \{j_{(g-1)r+1}, j_{(g-1)r+2}, \ldots, j_{(g-1)r+N-K}\}$ and

$$\begin{aligned} \overline{L}_{B^{(g)}} \ = \ & [\underbrace{L(c_{j_{(g-1)r+1}}), \ldots, L(c_{j_{(g-1)r+N-K}})}_{B^{(g)}}, \ldots, L(c_{j_{gr}}), \\ & L(c_{j_1}), \ldots, L(c_{j_{(g-1)r}}), L(c_{j_{gr+1}}), \ldots, L(c_{j_N})]. \end{aligned} \quad (26)$$

With the knowledge of $B^{(g)}$, GJ elimination will be performed on the first $N-K$ independent columns implied by $B^{(g)}$, i.e., column $j_{(g-1)r+1}$, column $j_{(g-1)r+2}$ and so on.

Generalizing this section, Fig.1 illustrates the decoding parameters exchange between different steps of the ABP-KV algorithm. Notice that with $\mathcal{N}_{\text{GR}}$ unreliable groups, the

[1]The density is measured as the percentage of 1 in the binary parity-check matrix.

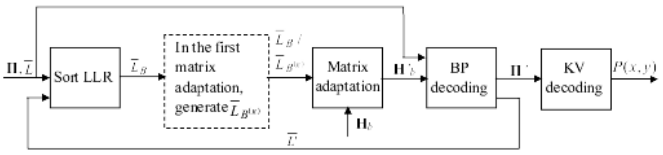

Fig. 1. Decoding parameters exchange of the ABP-KV algorithm.

decoding process will be deployed $\mathcal{N}_{\text{GR}}$ times, each of which inherits a specific pattern of $\overline{L}_{B^{(g)}}$. For each group, the sorted LLR vector restructuring process should only be performed prior to the first matrix adaptation of the ABP algorithm. Hence, the first adaptation is performed based on $\overline{L}_{B^{(g)}}$. After $\mathcal{N}_{\text{BP}}$ iterations, the updated LLR vector $\overline{L'}$ is formed and the following matrix adaptation will be performed based on $\overline{L'}$. Running the ABP-KV decoding based on $\mathcal{N}_{\text{GR}}$ different unreliable groups produces at most $l\mathcal{N}_{\text{GR}}\mathcal{N}_{\text{ADP}}$ output candidates in the global list $\mathcal{L}$.

IV. Reduced Complexity ABP-KV Decoding

The above section shows that given the decoding parameters ($\mathcal{N}_{\text{GR}}$, $\mathcal{N}_{\text{ADP}}$, $\mathcal{N}_{\text{BP}}$), the decoder would deploy $\mathcal{N}_{\text{GR}}\mathcal{N}_{\text{ADP}}$ matrix adaptations and KV decodings, and $\mathcal{N}_{\text{GR}}\mathcal{N}_{\text{ADP}}\mathcal{N}_{\text{BP}}$ BP iterations. Hence, the ABP-KV algorithm has a high decoding complexity. Ref [24] and [26] proposed a number of complexity reduction approaches for iterative decoding of RS codes.

This section proposes another complexity reduction approach based on reducing the deployment of the KV decoding steps. We introduce a couple of successive criteria to assess the quality of the processed reliability matrix $\mathbf{\Pi}'$ and the multiplicity matrix $\mathbf{M}$. Such assessments further determine whether the following KV decoding steps should be performed.

Definition II: In matrix $\mathbf{\Pi}'$, let i_j^* denote the row index of the maximal entry of column j as $i_j^* = \{i' \mid \pi'_{i'j} > \pi'_{ij},\ \forall\ (i,i') \in [1,q]$ and $i' \neq i\}$. The hard-decision received word is $\overline{R} = [\rho_{i_1^*}, \rho_{i_2^*}, \ldots, \rho_{i_n^*}] \in \mathbb{F}_q^n$. The reliability-based received word score is defined as:

$$S_{\mathbf{\Pi}'}(\overline{R}) = \sum_{j=1}^{n} \pi'_{i_j^* j}, \tag{27}$$

and the multiplicity-based received word score is defined as:

$$S_{\mathbf{M}}(\overline{R}) = \sum_{j=1}^{n} m_{i_j^* j}. \tag{28}$$

Hence, the following lemma is introduced to assess the quality of matrix $\mathbf{\Pi}'$.

Lemma 3: Given the processed reliability matrix $\mathbf{\Pi}'$, if

$$S_{\mathbf{\Pi}'}(\overline{R}) \leq \sqrt{w_z \sum_{i,j} \pi'^2_{ij}}, \tag{29}$$

the intended message polynomial cannot be found by the KV algorithm.

Proof: Given a matrix $\mathbf{\Pi}'$, $S_{\mathbf{\Pi}'}(\overline{C})$ is determined by $S_{\mathbf{\Pi}'}(\overline{C}) = \sum_{j=1}^{n} \pi'_{i_j j}$. Based on Theorem 2, the message polynomial can be found if $S_{\mathbf{\Pi}'}(\overline{C}) > \sqrt{w_z \sum_{i,j} \pi'^2_{ij}}$. Since $\pi'_{i_j j} \leq \pi'_{i_j^* j}$ for $1 \leq j \leq n$, we have $S_{\mathbf{\Pi}'}(\overline{C}) \leq S_{\mathbf{\Pi}'}(\overline{R})$. Therefore, if $S_{\mathbf{\Pi}'}(\overline{R}) \leq \sqrt{w_z \sum_{i,j} \pi'^2_{ij}}$, then $S_{\mathbf{\Pi}'}(\overline{C}) \leq \sqrt{w_z \sum_{i,j} \pi'^2_{ij}}$ and the message polynomial cannot be found by the KV algorithm. ■

Lemma 3 provides a criterion to assess the quality of $\mathbf{\Pi}'$. If the inequality of (29) is held, it implies the provided matrix $\mathbf{\Pi}'$ cannot be used to achieve a successful KV decoding. The KV decoding steps do not need to be applied. Otherwise, it is possible to find the message polynomial $F(x,y)$. The reliability transform should be performed to generate matrix $\mathbf{M}$. Then, we can determine whether the following interpolation and factorization should be performed.

Lemma 4: Given the multiplicity matrix $\mathbf{M}$, if

$$S_{\mathbf{M}}(\overline{R}) \leq \deg_{1,w_z}(Q), \tag{30}$$

the intended message polynomial cannot be found by the KV algorithm.

Proof: The reliability values are proportionally transformed into the multiplicity values [22]. With $\pi'_{i_j j} \leq \pi'_{i_j^* j}$, we have $m_{i_j j} \leq m_{i_j^* j}$ and hence $S_{\mathbf{M}}(\overline{C}) \leq S_{\mathbf{M}}(\overline{R})$. If $S_{\mathbf{M}}(\overline{R}) \leq \deg_{1,w_z}(Q)$, then $S_{\mathbf{M}}(\overline{C}) \leq \deg_{1,w_z}(Q)$. Recall Theorem 1, we know the intended message polynomial cannot be found by the KV algorithm. ■

Lemma 4 provides another criterion to assess the quality of matrix $\mathbf{M}$. With (30) being held, the successful decoding condition of (8) cannot be achieved. The following interpolation and factorization processes do not need to be applied. Note that without performing the interpolation, the $(1, w_z)$-weighted degree of Q can be predicted by knowing the interpolation cost $\mathcal{C}(\mathbf{M})$ [6].

Summarizing the above two sections, the reduced complexity ABP-KV decoding algorithm for Hermitian codes is presented in Algorithm 1.

In order to facilitate the decoding process, the outer most loop regarding the creation of different unreliable groups can be implemented in parallel. To validate the proposed complexity reduction approach, the average number[2] of KV decoding steps are measured against the channel signal-to-noise ratio (SNR). Tables I and II show the complexity reductions for ABP-KV decoding of the (64, 39) and the (64, 52) Hermitian codes, respectively. The ABP-KV decoding parameters are set as $(l = 5)$ and $(\mathcal{N}_{\text{GR}}, \mathcal{N}_{\text{ADP}}, \mathcal{N}_{\text{BP}}) = (10, 5, 2)$. The damping factor $\eta = 0.1$. Without the two successive criteria, the KV decoding steps will be performed 50 times for each codeword frame. But with the criteria, the average number of KV decoding steps are reduced. The reduction is more significant for the high rate code, implying equations (29) and (30) are more effective to assess the quality of $\mathbf{\Pi}'$ and $\mathbf{M}$ for high rate codes. However, by increasing the SNR value, the complexity reduction becomes less significant.

[2]The average numbers are obtained by running the ABP-KV algorithm for 10000 codeword frames at each SNR. The average values are quantized to the their closest integers.

TABLE I
COMPLEXITY REDUCTION FOR ABP-KV DECODING OF THE (64, 39) HERMITIAN CODE

SNR (dB) / Measurements	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
$\mathbf{\Pi}' \rightarrow \mathbf{M}$ transform	49	49	50	50	50	50	50	50	50	50	50
Interpolation/factorization	11	31	45	49	50	50	50	50	50	50	50

TABLE II
COMPLEXITY REDUCTION FOR ABP-KV DECODING OF THE (64, 52) HERMITIAN CODE

SNR (dB) / Measurements	0	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5
$\mathbf{\Pi}' \rightarrow \mathbf{M}$ transform	0	0	1	12	32	46	50	50	50	50	50
Interpolation/factorization	0	0	0	0	1	9	27	42	48	50	50

Algorithm 1 Reduced Complexity ABP-KV Decoding of Hermitian Codes

for each group g **do**
  Let $\overline{L'} = \overline{L}$;
  for each parity-check matrix adaptation **do**
    Generate a sorted LLR vector $\overline{L}_B$ based on $\overline{L'}$;
    if it is the first matrix adaptation **then**
      Restructure the sorted LLR vector to $\overline{L}_{B^{(g)}}$;
      Perform GJ elimination for $\mathbf{H}'_b$ based on $\overline{L}_{B^{(g)}}$.
    else
      Perform GJ elimination for $\mathbf{H}'_b$ based on $\overline{L}_B$;
    end if
    Initialize matrix $\mathbf{V}$ as in (16);
    for each BP iteration **do**
      Perform the horizontal step as in (17);
      Perform the vertical step as in (18);
    end for
    Determine the extrinsic information as in (19) and update its LLR value as in (20);
    Form the updated LLR vector $\overline{L'}$ as in (21);
    Generate N pairs of bit APP values as in (22);
    Determine the processed reliability matrix $\mathbf{\Pi}'$;
    while $S_{\mathbf{\Pi}'}(\overline{R}) > \sqrt{w_z \sum_{i,j} \pi'^2_{ij}}$ **do**
      Transform matrix $\mathbf{\Pi}'$ into matrix $\mathbf{M}$;
      while $S_{\mathbf{M}}(\overline{R}) > \deg_{1,w_z}(Q)$ **do**
        Perform interpolation to determine Q of (23);
        Perform factorization to find out $P(x, y)$;
      end while
    end while
  end for
end for

V. GEOMETRIC INTERPRETATION OF ABP DECODING

This section presents the geometric interpretation for the ABP algorithm and demonstrates the necessity of performing matrix adaptation prior to the BP decoding.

The conventional BP decoding can be seen as a gradient descent decoding problem [32]. The coded bit LLR values $L(c_j)$ with $L(c_j) \in [-\infty, +\infty]$ can be normalized to the region of $[-1, +1]$ by the following mapping function:

$$\xi(L(c_j)) = \tanh\Big(\frac{L(c_j)}{2}\Big) = \frac{e^{L(c_j)} - 1}{e^{L(c_j)} + 1}. \tag{31}$$

The reliability of bit c_j can again be reflected by its magnitude $|\xi(L(c_j))|$. Given two distinct bits c_{j_1} and c_{j_2}, bit c_{j_1} is more reliable if $|\xi(L(c_{j_1}))| > |\xi(L(c_{j_2}))|$. By normalizing all the LLR values of a codeword using the above mapping function, we can form vector $\overline{T}$ as:

$$\overline{T} = [T_1, T_2, \ldots, T_N] = [\xi(L(c_1)), \xi(L(c_2)), \ldots, \xi(L(c_N))]. \tag{32}$$

With the knowledge of T_j, the estimated coded bit $\hat{c}_j$ can be determined by:

$$\hat{c}_j = \text{sign}\Big(\ln \frac{1 + T_j}{1 - T_j}\Big). \tag{33}$$

Vector $\overline{T}$ corresponds to an estimated codeword that satisfies all the checks. With matrix $\mathbf{H}'_b$ and vector $\overline{T}$, the potential function $\mathcal{P}(\mathbf{H}'_b, \overline{T})$ of a Hermitian code is defined as [24] [32]:

$$\mathcal{P}(\mathbf{H}'_b, \overline{T}) = -\sum_{i=1}^{N-K} \prod_{j \in J(i)} T_j. \tag{34}$$

The quantization of $\mathcal{P}(\mathbf{H}'_b, \overline{T})$ describes the reliability of vector $\overline{T}$. Consequently, the LLR updates of (20) can be seen as the gradient descent update as follows:

$$T'_j = T_j - \eta \frac{\partial \mathcal{P}(\mathbf{H}'_b, \overline{T})}{\partial T_j} = T_j + \eta\Big(\sum_{i=1}^{N-K} \prod_{\tau \in J(i) \backslash j} T_\tau\Big). \tag{35}$$

In order to confine T'_j in the region of $[-1, +1]$, the above update should be modified as:

$$T'_j = \xi\Big[\xi^{-1}(T_j) + \eta\Big(\sum_{i=1}^{N-K} \xi^{-1}\Big(\prod_{\tau \in J(i) \backslash j} T_\tau\Big)\Big)\Big], \tag{36}$$

where $\xi^{-1}(\psi) = 2\tanh^{-1}(\psi)$. With all the checks from matrix $\mathbf{H}'_b$ being satisfied, a valid codeword is reached if $|T_j| = 1$ for $j = 1, 2, \ldots, N$. Function $\mathcal{P}(\mathbf{H}'_b, \overline{T})$ is minimized as $\min\{\mathcal{P}(\mathbf{H}'_b, \overline{T})\} = -(N - K)$. Therefore, finding an

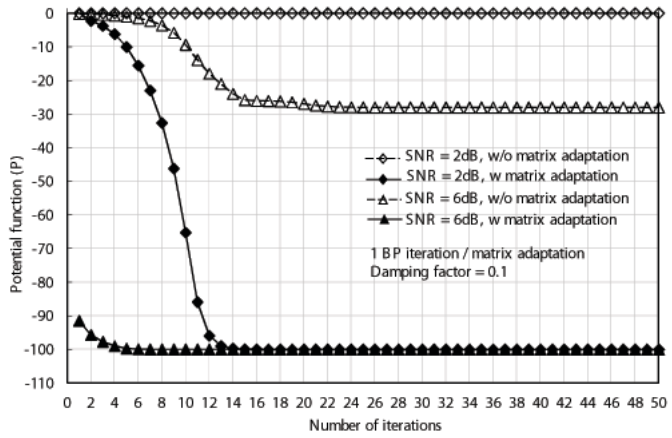

Fig. 2. Convergence of the potential function of the (64, 39) Hermitian code.

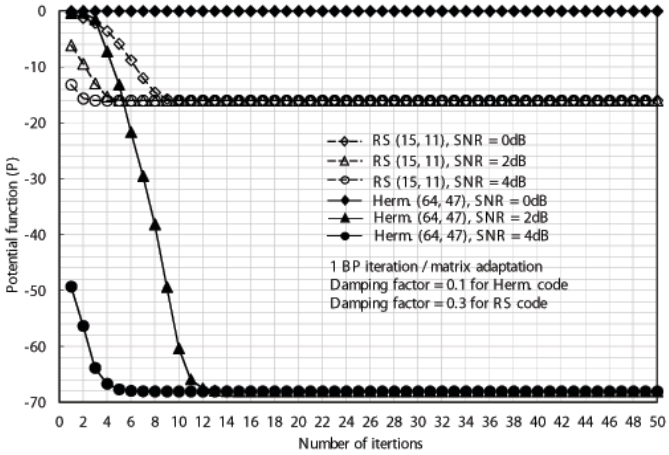

Fig. 3. Convergence of the potential functions of the (64, 47) Hermitian code and the (15, 11) RS code.

estimated codeword using the BP algorithm can be interpreted as identifying the vertex at which the potential function is minimized.

Without performing matrix adaptation, the density of the parity-check matrix of a Hermitian code remains high. The gradient descent decoding is easily hindered at some pseudo-equilibrium points which prevent the potential function from reaching its minimum. Fig.2 shows the convergence behavior of the potential function of the (64, 39) Hermitian code. It is measured on the additive white Gaussian noise (AWGN) channel. It shows without the matrix adaptation, the potential function cannot converge to its minimum. By increasing the SNR, a valid codeword is reached with fewer iterations. Fig.3 compares the convergence behavior of the potential functions of the (64, 47) Hermitian code and the (15, 11) RS code. They are defined over the same finite field and have a similar code rate. It shows under the same channel condition, the potential function of the RS code converges to its minimum faster than the Hermitian code. It implies the ABP algorithm provides a better treatment for enhancing the reliability matrix of the RS code. The ABP-KV decoding performance comparison of these two codes will be discussed in Section VI.

VI. Performance Analysis

This section presents the error-correction performance for the ABP-KV decoding algorithm. Comparisons with the existing decoding algorithms for Hermitian codes and with the ABP-KV decoding of RS codes are made. In the simulations, BPSK modulation is used. The ABP-KV decoding parameters are represented by the ternary tuple $(\mathcal{N}_{\text{GR}}, \mathcal{N}_{\text{ADP}}, \mathcal{N}_{\text{BP}})$. Error-correction performances of the Sakata algorithm, the optimal GS algorithm and the optimal KV algorithm[3] are presented as the comparison benchmarks. Please note that in order to facilitate the decoding process, the simulation results were obtained by assuming the use of an aided gene. It can notify the ABP-KV decoder to terminate once the intended message polynomial has been found. Such an assumption yields a mildly improved error-correction performance compared to the more practical scenario where the ML selection criterion is used. This is because the ML criterion cannot always guarantee an accurate selection from the output list. In the following discussions, coding gains are measured at the bit error rate (BER) of 10^{-5}.

A. Over the AWGN Channel

Figs.4, 5 and 6 show the BER performance of the (64, 32), (64, 39) and (64, 47) Hermitian codes over the AWGN channel, respectively. The ABP-Sakata algorithm is also shown as a comparison benchmark. It can be noticed that with an improved second stage decoding, the ABP-KV algorithm outperforms the ABP-Sakata algorithm. Its performance can be improved by increasing the KV algorithm's error-correction capability, i.e., the factorization output list size l. Comparing the three codes, we can notice that with the code rate being increased, more significant performance improvement can be made by the ABP-KV algorithm over the conventional decoding approaches. For example, with the decoding parameters $(2, 5, 2)$, ABP-KV $(l = 20)$ decoding of the (64, 47) Hermitian code has 1.1dB coding gain over the optimal KV decoding. While for the (64, 32) Hermitian code, only 0.1dB coding gain is observed.

Fig.5 shows with the same number of matrix adaptations and BP iterations, increasing the number of unreliable groups can enhance the error-correction performance significantly. But it is at the cost of decoding complexity. The decoding complexity of the ABP-KV algorithm is mainly caused by the GJ elimination, the BP iteration and the KV decoding process. They require at most $O(N(N-K)^2)$ binary operations, $O(N^2)$ floating point operations and $O(\frac{2}{3}\mathcal{C}^3(\mathbf{M}))$ finite field arithmetic operations, respectively. Given the decoding parameters $(\mathcal{N}_{\text{GR}}, \mathcal{N}_{\text{ADP}}, \mathcal{N}_{\text{BP}})$, the ABP-KV decoding process consists of $\mathcal{N}_{\text{GR}}\mathcal{N}_{\text{ADP}}$ GJ eliminations and KV decoding processes, and $\mathcal{N}_{\text{GR}}\mathcal{N}_{\text{ADP}}\mathcal{N}_{\text{BP}}$ BP iterations. In order to give a better insight of the complexity difference by having different decoding parameters, Table III shows the amount of different operations w.r.t. the three sets of ABP-KV decoding

[3]The optimal GS decoding performance is obtained by measuring the Hamming distance between the hard-decision received word $\overline{R}$ and the codeword $\overline{C}$. If it is greater than $n - \lfloor \sqrt{n(n-d)} \rfloor - 1$, where d is the code's designed minimum distance, a decoding failure is declared. Similarly, with the knowledge of matrix $\mathbf{\Pi}$ (or $\mathbf{\Pi}'$), the optimal KV decoding performance can be obtained by assessing the inequality of (9).

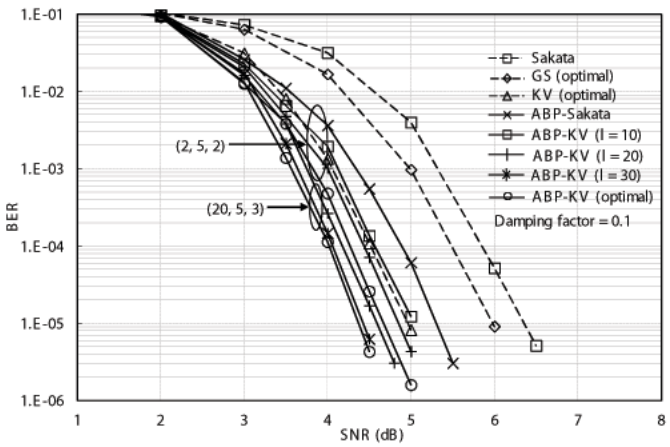

Fig. 4. BER performance of the (64, 32) Hermitian code over the AWGN channel.

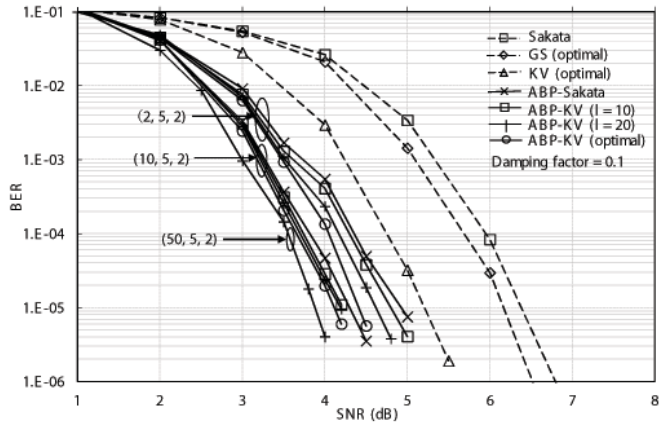

Fig. 5. BER performance of the (64, 39) Hermitian code over the AWGN channel.

parameters shown in Fig.5. The KV algorithm is operating with $l = 20$, which yields an average interpolation cost of 9130. It demonstrates the computational cost that is required to achieve the respective performance improvement.

From Figs.4, 5 and 6, we can notice that by having a better first stage decoding, performance difference generated by having a different second stage decoding is less significant. For example in Fig.5, with decoding parameters (2, 5, 2), the ABP-KV $(l = 20)$ decoding has 0.3dB coding gain over the ABP-Sakata decoding. However, with (10, 5, 2), the coding gain becomes 0.1dB. Therefore, if the system can afford a large number of matrix adaptations and BP iterations, it can simply use the Sakata algorithm for the second stage decoding. In Fig.6, it is interesting to compare the ABP-KV decodings with parameters (1, 10, 2) and (2, 5, 2). They imply a similar decoding complexity, while ABP-KV decoding with (2, 5, 2) prevails in performance. Such a comparison shows that given a budget on the number of matrix adaptations, they should be spread into a number of unreliable groups to achieve a better error-correction performance.

It is worthwhile to point out that the theoretical ML decoding performance bound for Hermitian codes is yet to be developed. This is due to the lack of knowledge of the codes' binary weight distribution. Therefore, the author cannot claim the optimality of the proposed ABP-KV algorithm. Developing the ML decoding performance bound for Hermitian codes is an open problem and the presented results offer a reference for future endeavors in this direction.

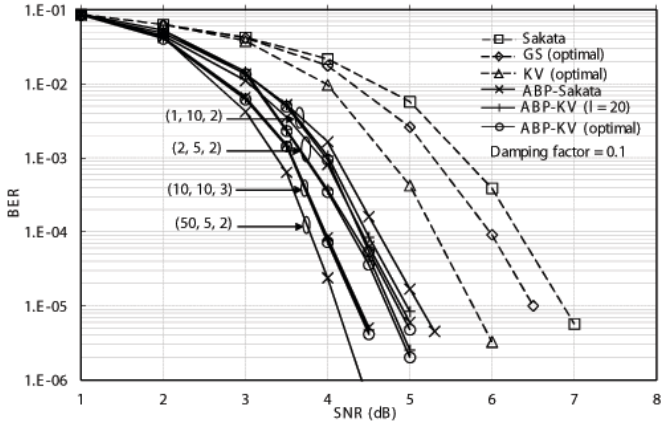

Fig. 6. BER performance of the (64, 47) Hermitian code over the AWGN channel.

TABLE III
COMPLEXITY COMPARISON OF ABP-KV DECODING OF THE (64, 39) HERMITIAN CODE

Oper. \ Para.	(2, 5, 2)	(10, 5, 2)	(50, 5, 2)
Binary	2.56×10^7	1.28×10^8	6.40×10^8
Floating point	1.31×10^6	6.55×10^6	3.28×10^7
Finite field	5.07×10^{12}	2.54×10^{13}	1.27×10^{14}

B. Over the Rayleigh Fading Channel

In order to evaluate the proposed algorithm in a more realistic scenario, Fig.7 shows the BER performance of the proposed algorithm over the fast Rayleigh fading channel in which each transmitted symbol experiences an independent fading. The channel state information (CSI) is assumed to be known at the decoder. The second stage decoding is carried out by the KV algorithm with $l = 20$. It can be seen that over the fast fading channel, more significant performance improvements can be achieved. Note that the optimal KV decoding performance is prohibitive in practice due to its high decoding complexity. While the decoding complexities required by the ABP-KV $(l = 20)$ algorithm with parameters (1, 1, 1), (1, 2, 2) and (1, 5, 2) are acceptable in practice. Hence, the ABP-KV decoding approach is suitable to be applied in practical wireless communications where the running time of the decoding algorithm is an important issue.

C. Comparison With RS Codes

Fig.8 compares the (64, 47) Hermitian code with the (15, 11) RS code. To ensure a fair comparison, ABP-KV decoding of the RS code is also functioning with an aided gene. It can be observed that with the same decoding parameters, the Hermitian code outperforms the RS code. This mainly thanks to its longer codeword length allowing more symbol

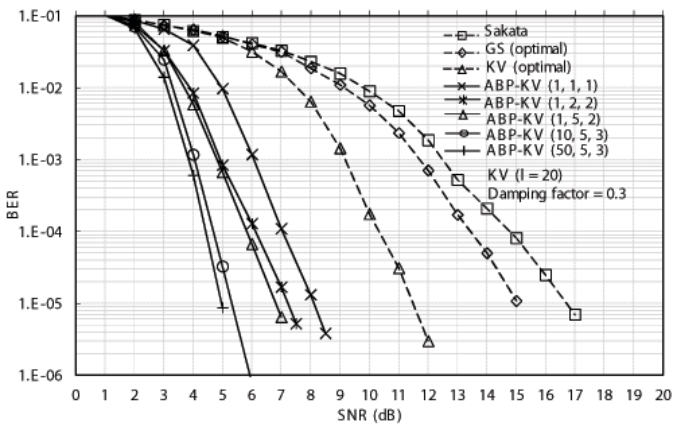

Fig. 7. BER performance of the (64, 47) Hermitian code over the Rayleigh fading channel.

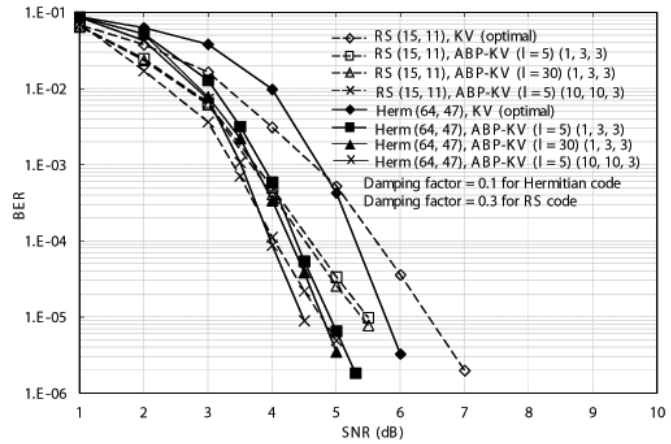

Fig. 8. Comparison of the (64, 47) Hermitian code and the (15, 11) RS code over the AWGN channel.

errors to be corrected. Since the length of the code defines the scale of the decoding complexity, ABP-KV decoding of the Hermitian code is more complex. On the other hand, it is observed that ABP-KV decoding achieves larger coding gains over the optimal KV decoding for the RS code. Recall our earlier analysis shown in Fig.3 that the ABP algorithm offers a better treatment to enhance the reliability of soft received information for the RS code. It should result in a larger coding gain over the optimal KV decoding. It is interesting to compare ABP-KV decoding of the Hermitian code with $l = 5$ and $(1, 3, 3)$ with ABP-KV decoding of the RS code wth $l = 5$ and $(10, 10, 3)$. Table IV shows the decoding complexity of these two scenarios. It can be seen that they are comparable. ABP-KV decoding of the Hermitian code requires more binary operations and finite field operations, but less floating point operations. Fig.8 shows the Hermitian code will prevail asymptotically. Therefore, by considering the performance-complexity tradeoff, Hermitian codes still have the potential to replace RS codes for better error-correction.

VII. Conclusion

This paper has proposed an iterative soft-decision decoding algorithm for the most popular AG codes – Hermitian codes.

TABLE IV
COMPLEXITY COMPARISON OF ABP-KV DECODING OF THE $(64, 47)$ HERMITIAN CODE WITH $(l = 5)$ AND $(1, 3, 3)$ AND THE $(15, 11)$ RS CODE WITH $(l = 5)$ AND $(10, 10, 3)$

Oper. \ Codes	Herm (64, 47)	RS (15, 11)
Binary	3.55×10^6	1.54×10^6
Floating point	5.90×10^5	1.08×10^6
Finite field	8.44×10^8	2.25×10^8

The iterative approach consists of two successive decoding stages: the ABP algorithm for enhancing the reliability of the received information and the KV algorithm for determining the decoding outputs. Parity-check matrix adaptation that is performed based on the bit reliabilities has been introduced as an a priori process for the BP decoding algorithm. Regrouping of unreliable bits has also been introduced as an important decoding strategy to enhance the error-correction performance. In order to reduce the decoding complexity, a couple of successive criteria are proposed to assess the quality of the reliability matrix $\mathbf{\Pi}'$ and the multiplicity matrix $\mathbf{M}$. It eliminates any redundant KV decoding steps for the unqualified matrices. Geometric interpretation of the ABP algorithm was presented. It analyzed the convergence behavior of the potential function of Hermitian codes, through which the necessity of performing matrix adaptation was demonstrated. Our performance analysis shows the proposed ABP-KV decoding algorithm outperforms the existing decoding algorithms for Hermitian codes. Specifically, the ABP-KV decoding performance advantage is more significant over the fast Rayleigh fading channel. A performance comparison with RS codes has also been made, showing ABP-KV decoding of Hermitian codes still outperforms its counterpart. Therefore, the proposed iterative soft decoding approach for Hermitian codes can be considered for a wide range of applications.

References

[1] V. D. Goppa, "Codes on algebraic curves," *Soviet Math*, vol. 24, pp. 75–91, 1981.
[2] M. Johnston and R. A. Carrasco, "Construction and performance of algebraic-geometric codes over AWGN and fading channels," *IEE Proc. Commun.*, vol. 152, pp. 713–722, 2005.
[3] M. Johnston and R. A. Carrasco, "Performance of Hermitian codes using combined error and erasure decoding," *IEE Proc. Commun.*, vol. 153, pp. 21–30, 2006.
[4] L. Chen, R. A. Carrasco, and M. Johnston, "List decoding performance of algebraic geometric codes," *IET Elect. Lett.*, vol. 42, 2006.
[5] L. Chen, R. A. Carrasco, and M. Johnston, "Reduced complexity for list decoding Hermitian codes," *IEEE Trans. Wireless Commun.*, vol. 7, no. 11, pp. 4353–4361, Nov. 2008.
[6] L. Chen, R. A. Carrasco, and M. Johnston, "Soft-decision list decoding of Hermitian codes," *IEEE Trans. Commun.*, vol. 57, no. 8, pp. 2169–2176, Aug. 2009.
[7] L. Chen and R. A. Carrasco, "Soft-decoding of algebraic-geometric codes using the Koetter-Vardy algorithm," *IET Elect. Lett.*, vol. 45, no. 25, 2009.
[8] K. Lee and M. E. O'Sullivan, "Algebraic soft-decision decoding of Hermitian codes," *IEEE Trans. Inf. Theory*, vol. 56, no. 6, pp. 2587–2600, June 2010.
[9] S. Sakata, J. Justesen, Y. Madelung, H. E. Jensen, and T. Hoholdt, "Fast decoding of algebraic-geometric codes up to the designed minimum distance," *IEEE Trans. Inf. Theory*, vol. 41, pp. 1672–1677, Nov. 1995.
[10] G. L. Feng and T. R. N. Rao, "Decoding algebraic-geometric codes up to the designed minimum distance," *IEEE Trans. Inf. Theory*, vol. 39, pp. 37-46, Jan. 1993.

[11] M. Sudan, "Decoding of Reed-Solomon codes beyond the error-correction bound," *J. Complexity*, vol. 13, pp. 180–193, 1997.

[12] V. Guruswami and M. Sudan, "Improved decoding of Reed-Solomon and algebraic-geometric codes," *IEEE Trans. Inf. Theory*, vol. 45, pp. 1757–1767, Sep. 1999.

[13] T. Hoholdt and R. R. Nielsen, "Decoding Hermitian codes with Sudan's algorithm," in *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes (Lecture Notes in Computer Science)*, vol. 1719, H. I. N. Fossorier, S. Lin, and A. Pole, editors. Springer-Verlag, pp. 260–270, 1999.

[14] R. R. Nielsen, "List decoding of linear block codes," Lyngby, Denmark, Ph.D. thesis, Tech. Univ. Denmark, 2001.

[15] G. D. Forney, "Generalized minimum distance decoding," *IEEE Trans. Inf. Theory*, vol. 12, no. 2, pp. 125–131, Apr. 1966.

[16] D. Chase, "A class of algorithms for decoding block codes with channel measurement information," *IEEE Trans. Inf. Theory*, vol. 18, no. 1, pp. 170–182, Jan. 1972.

[17] A. Vardy and Y. Be'ery, "Bit-level soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Commun.*, vol. 39, no. 3, pp. 440–444, Mar. 1991.

[18] V. Ponnampalam and B. Vucetic, "Soft decision decoding for Reed-Solomon codes," *IEEE Trans. Commun.*, vol. 50, no. 11, pp. 1758–1768, Nov. 2002.

[19] M. Fossorier and S. Lin, "Soft-decision decoding of linear block codes based on ordered statistics," *IEEE Trans. Inf. Theory*, vol. 41, no. 5, pp. 1379–1396, Sep. 1995.

[20] T. Hu and S. Lin, "An efficient hybrid decoding algorithm for Reed-Solomon codes based on bit reliability," *IEEE Trans. Commun.*, vol. 51, no. 7, pp. 1073–1081, July 2003.

[21] M. Fossorier and A. Valembois, "Reliability-based decoding of Reed-Solomon codes using their binary image," *IEEE Commun. Lett.*, vol. 7, pp. 452–454, July 2004.

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

[23] J. Jiang and K. Narayanan, "Iterative soft decision decoding of Reed-Solomon codes," *IEEE Commun. Lett.*, vol. 8, pp. 244–246, Apr. 2004.

[24] J. Jiang and K. Narayanan, "Iterative soft-input-soft-output decoding of Reed-Solomon codes by adapting the parity check matrix," *IEEE Trans. Inf. Theory*, vol. 52, no. 8, pp. 3746–3756, Aug. 2006.

[25] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Pearson Prentice Hall, 2004.

[26] M. El-Khamy and R. McEliece, "Iterative algebraic soft-decision list decoding of Reed-Solomon codes," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 481–490, Mar. 2006.

[27] L. Chen, "Design of an efficient list decoding system for Reed-Solomon and algebraic-geometric codes," Ph.D. thesis, Newcastle University, UK, 2008.

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

[29] R. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Trans. Inf. Theory*, vol. 46, no. 1, pp. 246–257, Jan. 2000.

[30] X. W. Wu and P. Siegel, "Efficient root-finding algorithm with application to list decoding of algebraic-geometric codes," *IEEE Trans. Inf. Theory*, vol. 47. no. 6, pp. 2579–2587, Sep. 2001.

[31] L. Chen, R. A. Carrasco, M. Johnston, and E. G. Chester, "Efficient factorisation algorithm for list decoding algebraic-geometric and Reed-Solomon codes," in *Proc. 2007 IEEE Int. Conf. Commun.*, pp. 851–856.

[32] R. Lucas, M. Bossert, and M. Breitbach, "On iterative soft-decision decoding of linear binary block codes and product codes," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 2, pp. 276–296, Feb. 1998.

Li Chen (S'07-M'08) received his BSc degree in applied physics from Jinan University, China in 2003, MSc degree in communications and signal processing and PhD degree in mobile communications in 2004 and 2008, respectively, both from Newcastle University of United Kingdom. From 2010, he joined the School of Information Science and Technology, Sun Yat-sen University of China, where he is now an Associate Professor. He is also a Visiting Lecturer with Newcastle University. From 2007 to 2010, he was a Research Associate with Newcastle University. During 2011-12, he is a Visiting Scholar with the Institute of Network Coding, the Chinese University of Hong Kong. He was a recipient of the British Overseas Research Scholarship (ORS). Currently, he is a principle investigator for a National Natural Science Foundation of China (NSFC) project and a co-investigator of the National Basic Research Program (973 program) project. His primary research interests include: information theory, channel coding and wireless communications.