Iterative Soft Decoding of Reed-Solomon Convolutional Concatenated Codes

Li Chen, *Member, IEEE*

***Abstract*—Reed-Solomon convolutional concatenated (RSCC) code is a popular coding scheme whose application can be found in wireless and space communications. However, iterative soft decoding of the concatenated code is yet to be developed. This paper proposes a novel iterative soft decoding algorithm for the concatenated code, aiming to better exploit its error-correction potential. The maximum *a posteriori* (MAP) algorithm is used to decode the inner convolutional code. Its soft output will be deinterleaved and then given to the soft-in-soft-out (SISO) decoder of the outer Reed-Solomon (RS) code. The RS SISO decoder integrates the adaptive belief propagation (ABP) algorithm and the Koetter-Vardy (KV) list decoding algorithm, attempting to find out the transmitted message. It feeds back both the deterministic and the extrinsic probabilities of RS coded bits, enabling the soft information to be exchanged between the inner and outer decoders. An extrinsic information transfer (EXIT) analysis of the proposed algorithm is presented, analyzing its iterative decoding behavior for RSCC codes. The EXIT analysis also leads to the design insight of inner code in the concatenation. Computational complexity of the proposed algorithm is also analyzed. Finally, the iterative decoding performance is shown and its advantage over the existing decoding algorithms is demonstrated.**

***Index Terms*—Concatenated codes, convolutional codes, iterative soft decoding, Reed-Solomon codes.**

I. INTRODUCTION

CONCATENATED codes were first introduced by Forney in [1]. It has been shown that concatenating a nonbinary outer code and a binary inner code could constitute a capacity approaching error-correction code with a polynomial-time decoding complexity. The legacy Reed-Solomon convolutional concatenated (RSCC) code is a popular example, in which the Reed-Solomon (RS) code and the convolutional code are the outer code and inner code, respectively. The inner code is good at correcting spread bit errors, while the outer code is good at correcting burst errors, enabling the RSCC codes' strong error-correction capability and their application can be widely found in wireless and space communications [2] - [4].

The classical decoding scheme for RSCC codes employs the Viterbi algorithm [5] and the Berlekamp-Massey (BM) algorithm [6] to decode the inner and outer codes, respectively. A block interleaver (deinterleaver) is employed between the inner and outer encoders (decoders) in order to spread the burst errors that are resulted from the Viterbi decoding. An improved decoding algorithm that performs repeated decoding trials for RSCC codes was proposed in [7]. It is a primitive attempt to decode the concatenated codes iteratively. However, since the BM algorithm is used to decode the outer code, preventing the soft information being given as the feedback. Another attempt to improve the error-correction performance is to utilize a stronger RS decoding algorithm, e.g., the Guruswami-Sudan (GS) algorithm [8] [9] and Koetter-Vardy (KV) algorithm [10]. Utilizing the KV algorithm in the repeated decoding mechanism of [7] was considered in [11]. Meanwhile, utilizing the RS decoding output statistics to form the soft feedback information for iterative decoding the concatenated codes was proposed in [12]. Finally, collaborative decoding of RS codes has also been considered for the concatenated codes [13]. It enables different RS codewords to be decoded jointly, allowing the BM algorithm to correct symbol errors beyond the half distance bound for each RS code.

The development of turbo codes [14] showed that allowing two decoders to exchange soft information iteratively may yield a capacity approaching performance. However, a truly iterative soft decoding algorithm for RSCC codes is yet to be developed. This is due to the challenge in designing a soft-in-soft-out (SISO) decoder for RS codes. The earlier RS SISO decoding attempt was the maximum likelihood (ML) decoding that utilizes the code's binary image [15] [16]. But its complexity grows exponentially with the length of the code. Recently, SISO decoding of RS codes utilizing the adaptive belief propagation (ABP) algorithm was proposed in [17] [18]. The ABP algorithm enhances the reliability of the soft received information and passes it to the following algebraic decoding, i.e., the BM or KV algorithm. Such an approach was later extended to decode the general algebraic-geometric codes in [19].

The profound consequence of the ABP algorithm is the extrinsic probabilities of RS coded bits can be calculated based on the adapted Tanner graph of the code with a moderate complexity. As a result, the soft information of RS coded bits can be iterated in a turbo decoding mechanism for RSCC codes. This paper proposes a novel iterative soft decoding algorithm for RSCC codes. The maximum *a posteriori* (MAP) [20] algorithm is used to decode the inner code, delivering the extrinsic probabilities for the interleaved RS coded bits. They are deinterleaved and mapped to the *a priori* probabilities of the RS coded bits. The RS SISO

Manuscript received December 10, 2012; revised July 4, 2013. The editor coordinating the review of this paper and approving it for publication was L. Szczecinski.

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

This paper was presented in part at the IEEE International Symposium on Information Theory, Istanbul, Turkey, July 2013.

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

Digital Object Identifier 10.1109/TCOMM.2013.082813.120943

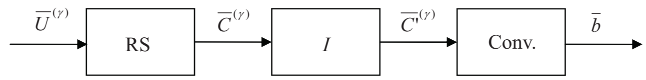

Fig. 1. Block diagram of the RSCC encoder.

decoding has two successive stages. The first stage is the bit reliability oriented ABP algorithm that improves the reliability of the received information. It delivers both the extrinsic probabilities and the *a posteriori* probabilities for the RS coded bits. The *a posteriori* probabilities will be utilized by the second RS decoding stage, i.e., KV algorithm. If the KV decoding is successful, deterministic probabilities of each RS coded bit will be given as the feedback. Otherwise, extrinsic probabilities that are yielded by the ABP algorithm will be fed back. They are then interleaved and mapped to the *a priori* probabilities of the interleaved RS coded bits for the next round MAP decoding. The proposed algorithm allows the extrinsic information of RS coded bits to be iterated between the inner and outer SISO decoders efficiently. Hence, it can well exploit the error-correction potential of RSCC codes. An extrinsic information transfer (EXIT) characteristics of the proposed algorithm is analyzed, leading to the insights of its iterative decoding behavior and design criteria of the inner and outer codes. The decoding complexity of the algorithm is also analyzed. Our simulation results show that it can outperform the classical Viterbi-BM algorithm with up to 2dB gain. Since RSCC codes are widely employed in different communication systems, it is important to design a decoding algorithm that can exploit their error-correction potential. It has been aware that upgrading a communication system can be expensive, e.g., to replace the RSCC codes by another coding scheme in space communications would involve launching a new satellite. It will be of economic interest to achieve a more reliable communication through upgrading the existing code's decoding system. The proposed work meets this end.

The rest of the paper is organized as the follows: Section II presents the background knowledge of RSCC codes. Section III presents the iterative soft decoding algorithm. Section IV presents the EXIT analysis of the proposed algorithm. Sections V and VI analyze its decoding complexity and performance, respectively. Finally, Section VII concludes the paper.

II. The RSCC Codes

Let $\mathbb{F}_q = \{\rho_1, \rho_2, \ldots, \rho_q\}$ denote the finite field of size q. In this paper, it is assumed that $\mathbb{F}_q$ is an extension field of $\mathbb{F}_2$ as $q = 2^\omega$, where ω is a positive integer. Let $\mathbb{F}_q[x]$ and $\mathbb{F}_q[x, y]$ denote the rings of univariate and bivariate polynomials defined over $\mathbb{F}_q$, respectively. The encoder block diagram of RSCC codes is shown by Fig. 1. There is a block interleaver between the inner and outer encoders, which has a vertical read-in and horizontal read-out interleaving pattern. Let D denote the depth of the block interleaver indicating there are D RS codewords being interleaved, and γ denote the index of the RS codeword where $1 \leq \gamma \leq D$. Without specifically mentioning, D is set as 10 in the paper.

The message vector of an (n, k) RS code can be written as

$$\overline{U}^{(\gamma)} = [U_1^{(\gamma)}\ U_2^{(\gamma)}\ \ldots\ U_k^{(\gamma)}] \in \mathbb{F}_q^k, \tag{1}$$

where n and k are the length and dimension of the code, respectively, and $n = q - 1$. The superscript (γ) denotes the variable belongs to the γth RS codeword. The generator matrix $\mathbf{G}$ of the RS code is

$$\mathbf{G} = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & \alpha & \cdots & \alpha^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{k-1} & \cdots & \alpha^{(k-1)(n-1)} \end{pmatrix}, \tag{2}$$

where α is a primitive element of $\mathbb{F}_q$ and $\mathbb{F}_q = \{0, 1, \alpha, \ldots, \alpha^{q-2}\}$. The γth RS codeword is generated by

$$\overline{C}^{(\gamma)} = \overline{U}^{(\gamma)} \cdot \mathbf{G} = [C_1^{(\gamma)}\ C_2^{(\gamma)}\ \ldots\ C_n^{(\gamma)}] \in \mathbb{F}_q^n. \tag{3}$$

In order to perform SISO decoding for an RS code, its parity-check matrix $\mathbf{H}$ needs to be known. For an (n, k) RS code, it is defined as

$$\mathbf{H} = \begin{pmatrix} 1 & \alpha & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \cdots & \alpha^{2(n-1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{n-k} & \cdots & \alpha^{(n-k)(n-1)} \end{pmatrix}. \tag{4}$$

Let $\sigma(x) \in \mathbb{F}_2[x]$ denote a primitive polynomial of $\mathbb{F}_q$, its companion matrix $\mathbf{A}$ is an $\omega \times \omega$ binary matrix [21]. The binary image of $\mathbf{H}$ can be generated by mapping its entries $\alpha^i \mapsto \mathbf{A}^i$ and $i = 0, 1, \ldots, q-2$, resulting in the binary parity-check matrix $\mathbf{H}_b$ with size $(n-k)\omega \times n\omega$.

After the D RS codewords have been generated, they will be interleaved by the block interleaver. The γth interleaved RS codeword is

$$\overline{C'}^{(\gamma)} = [C_1'^{(\gamma)}\ C_2'^{(\gamma)}\ \ldots\ C_n'^{(\gamma)}] \in \mathbb{F}_q^n. \tag{5}$$

Note that $\overline{C'}^{(\gamma)}$ may not be a valid RS codeword. All the D interleaved RS codewords $\overline{C'}^{(1)}, \overline{C'}^{(2)}, \ldots, \overline{C'}^{(D)}$ are then converted into a binary interleaved coded bit sequence to form the input to the inner encoder

$$\begin{aligned} &c_1', c_2', \ldots, c_{n\omega}', c_{n\omega+1}', c_{n\omega+2}', \ldots, c_{2n\omega}', \ldots, \\ &\qquad c_{(D-1)n\omega+1}', c_{(D-1)n\omega+2}', \ldots, c_{Dn\omega}'. \end{aligned} \tag{6}$$

The transfer functions [22] $G(x) \in \mathbb{F}_2[x]$ of the convolutional code can be written as

$$G(x) = \sum_{j=0}^{\varpi} g_j x^j, \tag{7}$$

where ϖ denotes the number of shift registers in the convolutional encoder and the code has $\Omega = 2^\varpi$ states. Note that ϖ zero padding bits $c_{Dn\omega+1}', \ldots, c_{Dn\omega+\varpi}'$ will be appended to the end of sequence (6). In this paper, the rate 1/2 convolutional code will be considered. For simplicity, we will now denote the transfer functions of the inner code in octal form. For example, a rate 1/2 4-state nonsystematic inner code with transfer functions $(G_1(x) = 1 + x^2, G_2(x) = 1 + x + x^2)$ is

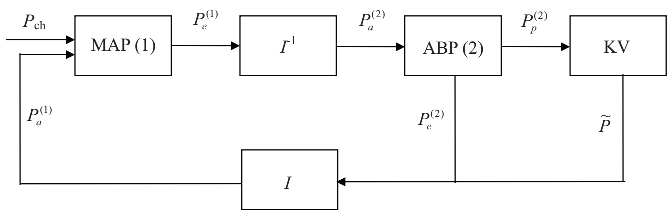

Fig. 2. Block diagram of the iterative soft decoding algorithm.

denoted as $(5,7)_8$. Similarly, its systematic feedback counterpart with transfer functions $(1, G_1(x)/G_2(x) = (1+x^2)/(1+x+x^2))$ is denoted as $(1, 5/7)_8$. With the above mentioned input, the convolutional codeword is

$$\overline{b} = [b_1 \ b_2 \ \cdots \ b_N] \in \mathbb{F}_2^N, \tag{8}$$

where $N = 2(Dn\omega + \varpi)$ and the rate of the RSCC code is $k/2n$ [1]. Note that variable rates of the concatenated codes can be realized by puncturing the output of the inner code.

III. Iterative Soft Decoding

The block diagram of the proposed iterative soft decoding algorithm is shown by Fig. 2. We use P_a, P_e and P_p to denote the *a priori* probability, the extrinsic probability and the *a posteriori* probability, respectively, and

$$P_p = P_a \cdot P_e. \tag{9}$$

For the probabilities, superscripts (1) and (2) are used to indicate they are associated with the MAP algorithm and the ABP algorithm, respectively. Moreover, P_{ch} denotes the channel observations which will be left unchanged during the iterations. $\tilde{P}$ denotes the deterministic probability that is estimated by KV decoding and $\tilde{P} \in \{0, 1\}$. With the channel observations P_{ch} and the *a priori* probabilities $P_a^{(1)}$ of the interleaved RS coded bits, the MAP algorithm is performed to determine the extrinsic probabilities $P_e^{(1)}$ of the interleaved RS coded bits. They are then deinterleaved and mapped to the *a priori* probabilities $P_a^{(2)}$ of RS coded bits. For each RS codeword, the ABP algorithm is performed, delivering the extrinsic probabilities $P_e^{(2)}$ and the *a posteriori* probabilities $P_p^{(2)}$ of RS coded bits. With $P_p^{(2)}$, KV algorithm is performed to retrieve the transmitted message. Once it is found, the deterministic probabilities $\tilde{P}$ of the corresponding RS coded bits are fed back. Otherwise, the system feeds back the extrinsic probabilities $P_e^{(2)}$. Both $\tilde{P}$ and $P_e^{(2)}$ are then interleaved and mapped back to the *a priori* probabilities $P_a^{(1)}$ for the next round MAP decoding. The decoding terminates once all the RS codewords have been decoded or the maximal iteration number N_{ITER} is reached.

A. SISO Decoding of the Inner Code

SISO decoding of the inner code is realized by the MAP algorithm [20]. Let $\mathcal{S} = \{1, 2, \ldots, \Omega\}$ denote the set of states of the code's trellis and $\theta \in \{0, 1\}$. Given $\overline{\mathcal{Y}} \in \mathbb{R}$ as the received vector, the channel observations can be obtained as

[1] Since $\varpi \ll Dn\omega$, rate loss caused by the zero padding bits is ignorable.

$$P_{\text{ch},j}(\theta) = \Pr[b_j = \theta \mid \overline{\mathcal{Y}}], \tag{10}$$

where $j = 1, 2, \ldots, N$. The *a priori* probabilities of the interleaved RS coded bits c'_j are

$$P_{a,j}^{(1)}(\theta) = \Pr[c'_j = \theta], \tag{11}$$

where $j = 1, 2, \ldots, Dn\omega + \varpi$. At the beginning, the *a priori* probabilities are initialized as $P_{a,j}^{(1)}(0) = P_{a,j}^{(1)}(1) = 0.5$ for $j = 1, 2, \ldots, Dn\omega$, and $P_{a,j}^{(1)}(0) = 1$ and $P_{a,j}^{(1)}(1) = 0$ for $j = Dn\omega + 1, \ldots, Dn\omega + \varpi$. Let us assume that at the time instant j, an input of $c'_j = \theta$ corresponds to two coded bits of $b_{2j-1}b_{2j} = \theta_1\theta_2$, where $(\theta_1, \theta_2) \in \{0, 1\}$. It triggers a trellis state transition from state $\mathcal{X}_j$ to $\mathcal{X}_{j+1}$, where $(\mathcal{X}_j, \mathcal{X}_{j+1}) \in \mathcal{S}$. With the knowledge of $P_{\text{ch},j}$ and $P_{a,j}^{(1)}$, the state transition probability can be determined by

$$\Gamma_j(\mathcal{X}_j, \mathcal{X}_{j+1}) = P_{a,j}^{(1)}(\theta) \cdot P_{\text{ch},2j-1}(\theta_1) \cdot P_{\text{ch},2j}(\theta_2). \tag{12}$$

The MAP algorithm will then perform the forward and backward traces to determine the *a posteriori* probabilities of interleaved RS coded bits c'_j, which are defined as

$$P_{p,j}^{(1)}(\theta) = \Pr[c'_j = \theta \mid \overline{\mathcal{Y}}]. \tag{13}$$

Based on (9), the extrinsic probabilities of interleaved RS coded bits c'_j can be determined by

$$P_{e,j}^{(1)}(\theta) = \mathcal{N}_E \frac{P_{p,j}^{(1)}(\theta)}{P_{a,j}^{(1)}(\theta)} \tag{14}$$

for $j = 1, 2, \ldots, Dn\omega$, and $\mathcal{N}_E = (P_{e,j}^{(1)}(0) + P_{e,j}^{(1)}(1))^{-1}$ is a normalization factor.

Probabilities $P_{e,j}^{(1)}(\theta)$ are then deinterleaved. Since each RS codeword symbol can be decomposed into ω bits, every ω consecutive pairs of probability values $(P_{e,j}^{(1)}(0), P_{e,j}^{(1)}(1))$ are grouped to represent an RS codeword symbol during the deinterleaving. Note that the inner SISO decoding can also be executed using the log-MAP algorithm. But since we would later show the outer SISO decoding is feeding back the deterministic probabilities of 0 and 1, it is more convenient to describe the inner SISO decoding as the MAP algorithm.

B. SISO Decoding of the Outer Code

SISO decoding of the outer code is realized by the ABP-KV algorithm [17] [18]. By reading out each row of the deinterleaver and mapping

$$P_e^{(1)} \mapsto P_a^{(2)}, \tag{15}$$

we can obtain the *a priori* probability for each RS coded bit c_j. For simplicity, we will now describe the SISO decoding of an RS codeword and hence drop the codeword index γ. Let

$$P_{a,j}^{(2)}(\theta) = \Pr[c_j = \theta] \tag{16}$$

denote the *a priori* probabilities of the RS coded bits c_j, where $j = 1, 2, \ldots, n\omega$. The *a priori* log-likelihood ratio (LLR) value of c_j can be determined by

$$L_{a,j} = \ln\Big(\frac{P_{a,j}^{(2)}(0)}{P_{a,j}^{(2)}(1)}\Big). \tag{17}$$

Consequently, the *a priori* LLR vector of an RS codeword can be formed as

$$\overline{L}_a = [L_{a,1}\ L_{a,2}\ \ldots\ L_{a,(n-k)\omega}\ \ldots\ L_{a,n\omega}]. \tag{18}$$

The ABP algorithm will first sort the *a priori* LLR values based on their magnitudes $|L_{a,j}|$, yielding a refreshed bit index sequence $\delta_1, \delta_2, \ldots, \delta_{(n-k)\omega}, \ldots, \delta_{n\omega}$ which implies $|L_{a,\delta_1}| < |L_{a,\delta_2}| < \cdots < |L_{a,\delta_{(n-k)\omega}}| < \cdots < |L_{a,\delta_{n\omega}}|$. Since a higher magnitude implies the bit is more reliable, we know $c_{\delta_1}, c_{\delta_2}, \ldots, c_{\delta_{(n-k)\omega}}$ are the $(n-k)\omega$ least reliable bits. Let $UR = \{\delta_1, \delta_2, \ldots, \delta_{(n-k)\omega}\}$ denote the set of the unreliable bit indices and $|UR| = (n-k)\omega$. Its complementary set is $UR^c = \{\delta_{(n-k)\omega+1}, \delta_{(n-k)\omega+2}, \ldots, \delta_{n\omega}\}$. Based on the set UR, the sorted *a priori* LLR vector is

$$\overline{L}_a^{UR} = [L_{a,\delta_1}\ L_{a,\delta_2}\ \cdots\ L_{a,\delta_{(n-k)\omega}}\ \cdots\ L_{a,\delta_{n\omega}}]. \tag{19}$$

The ABP algorithm will then perform Gaussian elimination on matrix $\mathbf{H}_b$, reducing the columns that correspond to the unreliable bits to weight-1 columns. Let Υ_δ denote the weight-1 column vector with 1 at its δth entry. For matrix $\mathbf{H}_b$, Gaussian elimination reduces column δ_1 to Υ_1, then reduces column δ_2 to Υ_2, and etc. Gaussian elimination reduces the first $(n-k)\omega$ independent columns implied by UR to weight-1 columns, resulting in an adapted parity-check matrix $\mathbf{H}'_b$. This is to prevent the propagation of the unreliable information during the following BP process [17] [19].

Let $h_{uj} \in \{0,1\}$ denote the entry of matrix $\mathbf{H}'_b$ and let

$$\mathbf{U}(j) = \{u \mid h_{uj} = 1,\ \forall\ 1 \le u \le (n-k)\omega\}, \tag{20}$$

$$\mathbf{J}(u) = \{j \mid h_{uj} = 1,\ \forall\ 1 \le j \le n\omega\}. \tag{21}$$

The iterative BP process is performed based on the Tanner graph that is associated with matrix $\mathbf{H}'_b$, yielding the extrinsic LLR value for each RS coded bit by

$$L_{e,j} = \sum_{u\in\mathbf{U}(j)} 2\tanh^{-1}\Big(\prod_{\tau\in\mathbf{J}(u)\setminus j} \tanh\Big(\frac{L_{a,\tau}}{2}\Big)\Big). \tag{22}$$

After a number of BP iterations, the *a posteriori* LLR of each RS coded bit is determined by

$$L_{p,j} = L_{a,j} + \eta L_{e,j}, \tag{23}$$

where $\eta \in (0,1]$ is the damping factor [18]. Therefore, the *a posteriori* LLR vector of an RS codeword can be formed as

$$\overline{L}_p = [L_{p,1}\ L_{p,2}\ \ldots\ L_{p,(n-k)\omega}\ \ldots\ L_{p,n\omega}]. \tag{24}$$

The extrinsic LLR calculation of (22) can be simplified to

$$L_{e,j} = \sum_{u\in\mathbf{U}(j)} \Big(\prod_{\tau\in\mathbf{J}(u)\setminus j} \mathrm{sign}(L_{a,\tau}) \cdot \min_{\tau\in\mathbf{J}(u)\setminus j}\{|L_{a,\tau}|\}\Big). \tag{25}$$

Given a random variable ψ, $\mathrm{sign}(\psi) = 0$ if $\psi \ge 0$, or $\mathrm{sign}(\psi) = 1$ otherwise. It is useful for decoding the long RS codes, e.g., those with $n = 255$. For long RS codes, their parity-check matrices still have a large number of short cycles which will affect the credit of the extrinsic LLR calculation (22).

The ABP algorithm itself is an iterative process. If there are multiple Gaussian eliminations, the *a posteriori* LLR vector will be mapped back to the *a priori* LLR vector by

$$\overline{L}_p \mapsto \overline{L}_a. \tag{26}$$

Based on the updated $\overline{L}_a$ vector, the next round bit reliability sorting and Gaussian elimination will be performed. Based on each adapted matrix $\mathbf{H}'_b$, a number of BP iterations will be carried out, delivering both the extrinsic and *a posteriori* LLR values. Hence, the extrinsic probabilities and the *a posteriori* probabilities of RS coded bits can be determined by

$$P_{e,j}^{(2)}(0) = \frac{1}{1+e^{-L_{e,j}}},\ P_{e,j}^{(2)}(1) = \frac{1}{1+e^{L_{e,j}}}, \tag{27}$$

$$P_{p,j}^{(2)}(0) = \frac{1}{1+e^{-L_{p,j}}},\ P_{p,j}^{(2)}(1) = \frac{1}{1+e^{L_{p,j}}}. \tag{28}$$

With the knowledge of the *a posteriori* probabilities, the reliability matrix $\mathbf{\Pi} \in \mathbb{R}^{q\times n}$ w.r.t. an RS codeword $\overline{C}$ can be formed. Its entry $\pi_{\mu\nu}$ is the *a posteriori* probability of an RS codeword symbol C_ν being the field symbol ρ_μ where $\mu = 1, 2, \ldots, q$ and $\nu = 1, 2, \ldots, n$. Let Ξ_μ denote the binary representation of the field symbol ρ_μ and

$$\Xi_\mu = [\theta_1\theta_2\cdots\theta_\omega \mid \rho_\mu = \sum_{\kappa=1}^{\omega} \theta_\kappa \alpha^{\omega-\kappa} \text{ and } \theta_\kappa \in \{0,1\}]. \tag{29}$$

The symbol wise *a posteriori* probability $\pi_{\mu\nu}$ can be determined by

$$\pi_{\mu\nu} = \Pr[C_\nu = \rho_\mu \mid \overline{\mathcal{Y}}] = \prod_{\theta_\kappa\in\Xi_\mu} P_{p,(\nu-1)\omega+\kappa}^{(2)}(\theta_\kappa). \tag{30}$$

Matrix $\mathbf{\Pi}$ will then be transformed into a multiplicity matrix $\mathbf{M} \in \mathbb{N}^{q\times n}$ [10] with entries $m_{\mu\nu}$. The cost of $\mathbf{M}$ is

$$\Lambda(\mathbf{M}) = \frac{1}{2}\sum_{\mu,\nu} m_{\mu\nu}(m_{\mu\nu}+1), \tag{31}$$

which indicates the number of interpolation constraints [9] [10]. It is also an important parameter for defining the KV decoding complexity. Interpolation will then be carried out, yielding an interpolated polynomial $Q \in \mathbb{F}_q[x,y]$ [8] [9]. Finally, factorization will be performed to find the y-roots of Q [23] [24]. Those y-roots are polynomials of $\mathbb{F}_q[x]$ and their coefficients form the decoded message candidates. Let $\mathcal{L}$ denote the factorization output list and $\max\{|\mathcal{L}|\} = l$, meaning that there are at most l decoding output candidates. By increasing the designed output list size l, i.e., the y-degree of Q, KV algorithm will have a better error-correction capability.

The KV decoding is considered to be successful if the transmitted message is found. The decoding output validation can either be realized by the ML criterion [25] or the cyclic redundant check (CRC) code while the latter one will result in a slight rate loss. The ML validation is performed with the knowledge of the hard-decision received word, the decoded RS codeword and matrix $\mathbf{\Pi}$. It is presented as Lemma 1 in [26]. Having an accurate output validation is important for the iterative soft decoding system, since the decoded RS bits will become the known *a priori* information in the next round MAP decoding. With a successful KV decoding, we can obtain the deterministic probabilities of the decoded bits. Let $\hat{c}_j \in \{0,1\}$

denote the decoded RS bit, where $j = 1, 2, \ldots, n\omega$. The deterministic probability of $\hat{c}_j$ is

$$\begin{cases} \tilde{P}_j(0) = 1,\ \tilde{P}_j(1) = 0,\ \text{if } \hat{c}_j = 0; \\ \tilde{P}_j(0) = 0,\ \tilde{P}_j(1) = 1,\ \text{if } \hat{c}_j = 1. \end{cases} \tag{32}$$

The deterministic probabilities will be left unchanged in the rest of the iterations and those decoded RS codewords will not be processed again.

If the KV decoding is not successful, the system will give the extrinsic probabilities of (27) as feedback. Hence, after SISO decodings of all the D RS codewords, the extrinsic probabilities $P_{e,j}^{(2)}$ of the undecoded bits and the deterministic probabilities $\tilde{P}_j$ of the decoded bits will be fed back. They are then interleaved and mapped to the *a priori* probabilities of the interleaved RS coded bits c'_j as

$$P_e^{(2)},\ \tilde{P} \mapsto P_a^{(1)}. \tag{33}$$

With the deterministic probabilities, the next round MAP decoding is functioning with a portion of known *a priori* information. It is important to mention that the identities of the decoded bits need to be recognized by the iterative system. In the next round MAP decoding, only the extrinsic probabilities of the undecoded bits will be determined. This can be done by using a binary indicator of size $Dn\omega$ memorizing each RS coded bit's decoding status. The binary information will also be interleaved alongside their corresponding probabilities $P_e^{(2)}$ or $\tilde{P}$, so that the MAP algorithm knows identities of the undecoded bits. Alternatively, the system can also identify each bit's decoding status by assessing its *a priori* probability $P_a^{(1)}$. If $P_a^{(1)} = \{0, 1\}$, it implies the bit has been decoded. Otherwise, it implies the bit's extrinsic probability needs to be calculated. The decoding terminates when either all the D RS codewords have been decoded or the maximal iteration number N_{ITER} is reached.

Summarizing this section, the iterative soft decoding algorithm is presented as in Algorithm 1.

Notice that iterations w.r.t. different parity-check matrix adaptations will be terminated once the intended RS codeword has been found. Then, the iterative algorithm will start to decode the next undecoded RS codeword.

C. Performance Improving Approaches

Since the ABP-KV decoding of the outer code is not optimal, the iterative decoding performance can be improved by strengthening this outer SISO decoding. With achieving more successful RS decoding events during each iteration, more deterministic probabilities can be supplied to the next round MAP decoding. To improve the decoding performance of the outer code, we can improve the KV decoding by increasing its factorization output list size l [10] [28], or improve the ABP decoding by restructuring the original sorted LLR vector $\overline{L}_a^{UR}$ [17] [18]. The latter approach is to spin out a number of independent ABP-KV decoding processes to find the transmitted message. By restructuring $\overline{L}_a^{UR}$, more coded bits' corresponding columns can be reduced to weight-1, warranting them the possibility to be corrected by the BP process. Let N_{GR} denote the designed number of unreliable bit index groups and $z = \lfloor n\omega / N_{\text{GR}} \rfloor$. The original sorted LLR vector $\overline{L}_a^{UR}$ can be expressed as:

$$\begin{aligned} \overline{L}_a^{UR} = [&L_{a,\delta_1} \ \ldots \ L_{a,\delta_z} \ L_{a,\delta_{z+1}} \ \ldots \ L_{a,\delta_{2z}} \ \ldots \\ &L_{a,\delta_{(t-1)z+1}} \ \ldots \ L_{a,\delta_{tz}} \ \ldots \ L_{a,\delta_{n\omega}}], \end{aligned} \tag{34}$$

where t is the group index and $1 \leq t \leq N_{\text{GR}}$. If $z < (n-k)\omega$, $UR^{(t)} = \{\delta_{(t-1)z+1}, \ldots, \delta_{tz}, \delta_1, \ldots, \ \delta_{(n-k)\omega - z}\}$ and the restructured sorted LLR vector becomes

$$\begin{aligned} \overline{L}_a^{UR^{(t)}} = &[\underbrace{L_{a,\delta_{(t-1)z+1}} \ \ldots \ L_{a,\delta_{(n-k)\omega - z}}}_{UR^{(t)}} \ L_{a,\delta_{(n-k)\omega - z+1}} \\ &\ldots \ L_{a,\delta_{(t-1)z}} \ L_{a,\delta_{tz+1}} \ \ldots \ L_{a,\delta_{n\omega}}]. \end{aligned} \tag{35}$$

Gaussian elimination will be performed on the first $(n-k)\omega$ independent columns that are implied by $UR^{(t)}$. As a result, the wrongly estimated bits in the original set UR^c will also have the opportunity to be corrected by the BP decoding.

IV. The EXIT Analysis

This section utilizes the EXIT chart [27] to analyze the iterative decoding behavior of the proposed algorithm, revealing the EXIT characteristics of the ABP-KV algorithm and the interplay between the two SISO decoders. The analysis will also lead to the design insight of inner code. In the following analysis, we use $\mathcal{I}_a^{(1)}$ ($\mathcal{I}_a^{(2)}$) and $\mathcal{I}_e^{(1)}$ ($\mathcal{I}_e^{(2)}$) to denote the mutual information of the *a priori* probabilities and the extrinsic probabilities of the MAP algorithm (the ABP-KV algorithm), respectively.

In analyzing the EXIT characteristics of the ABP-KV algorithm, we take the decoding output of the D RS codewords as an entity to measure the mutual information of the feedback probabilities, including $P_e^{(2)}$ and $\tilde{P}$. That says $\mathcal{I}_e^{(2)}$ is determined by

$$\mathcal{I}_e^{(2)} = 1 - \frac{1}{Dn\omega} \sum_{j=1}^{Dnw} \mathcal{H}_b(P_j^{(2)}), \tag{36}$$

where $P_j^{(2)} = \tilde{P}_j$ if bit c_j is decoded, or $P_j^{(2)} = P_{e,j}^{(2)}$ otherwise, and $\mathcal{H}_b(P_j^{(2)})$ is the binary entropy function that is defined as

$$\mathcal{H}_b(P_j^{(2)}) = -P_j^{(2)}(0) \log_2 P_j^{(2)}(0) - P_j^{(2)}(1) \log_2 P_j^{(2)}(1). \tag{37}$$

Note that the ABP algorithm has a dynamic decoding structure since its adapted matrix $\mathbf{H}'_b$ varies according to the sorted LLR vector $\overline{L}_a^{UR}$. In order to show a more statistical insight of the ABP-KV algorithm's EXIT characteristics, the EXIT curve of the ABP-KV algorithm is obtained by running 1000 decoding trails and calculating the average of $\mathcal{I}_e^{(2)}$. The input of the ABP-KV algorithm is constituted by D *a priori* LLR vectors, each of which is generated randomly according to the same $\mathcal{I}_a^{(2)}$ value.

Fig. 3 shows the EXIT characteristics of the ABP-KV algorithm in decoding the (63, 50) RS code with different ABP parameters $(N_{\text{ADP}}, N_{\text{BP}})$, where N_{ADP} denotes the number of matrix adaptations and N_{BP} denotes the number of BP

Algorithm 1 Iterative Soft Decoding of RSCC Codes

Initializations: Determine the channel observations P_{ch} of (10), initialize the *a priori* probabilities $P_a^{(1)}$ of (11) and the iteration index $v = 1$;
1: Perform the MAP algorithm to determine the *a posteriori* probabilities $P_p^{(1)}$;
2: For the undecoded bits, determine their extrinsic probabilities $P_e^{(1)}$ as in (14);
3: Perform deinterleaving and map the extrinsic probabilities $P_e^{(1)}$ to the *a priori* probabilities $P_a^{(2)}$ as in (15);
4: For each undecoded RS codeword {
5: For each parity-check matrix adaptation {
6: Form the *a priori* LLR vector $\overline{L}_a$ of (18);
7: Sort vector $\overline{L}_a$, yielding a sorted *a priori* LLR vector $\overline{L}_a^{UR}$ of (19);
8: Perform Gaussian elimination on matrix $\mathbf{H}_b$, yielding an adapted matrix $\mathbf{H}_b'$;
9: Determine the extrinsic LLR values as in (22) (or (25)) by a number of BP iterations;
10: Determine the *a posteriori* LLR values as in (23) and form vector $\overline{L}_p$ of (24);
11: Update the *a priori* LLR vector $\overline{L}_a$ as in (26);
12: Determine the *a posteriori* probabilities of RS coded bits as in (28);
13: Determine the reliability matrix $\mathbf{\Pi}$ of an RS codeword as in (30);
14: Perform KV decoding to find out the transmitted RS codeword; }}
15: If all the D RS codewords have been found, terminate the decoding; Otherwise, determine the extrinsic probabilities of the undecoded bits as in (27);
16: Perform interleaving and map both the extrinsic probabilities $P_e^{(2)}$ and the deterministic probabilities $\tilde{P}$ to the *a priori* probabilities $P_a^{(1)}$ as in (33);
17: Update the iteration index as $v = v + 1$;
18: If $v > N_{\text{ITER}}$, terminate decoding; Otherwise, go to **1**.

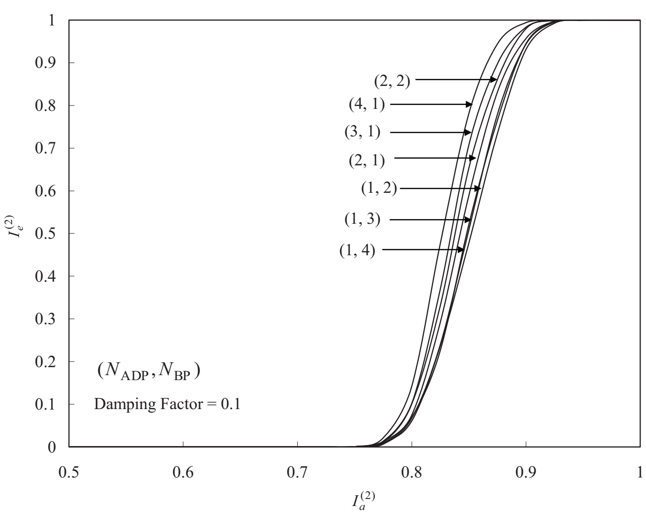

Fig. 3. EXIT characteristics of the ABP-KV algorithm.

iterations w.r.t. each adapted matrix $\mathbf{H}_b'$. The algorithm will perform at most N_{ADP} Gaussian eliminations and $N_{\text{ADP}}N_{\text{BP}}$ BP iterations. The KV algorithm is functioning with $l = 10$. The shown decoding parameters are chosen to ensure both the numbers of Gaussian eliminations and BP iterations are not greater than 4, so that they have a similar complexity. It can be seen that the ABP-KV algorithm is only competent in yielding an improved extrinsic information with $\mathcal{I}_e^{(2)} > \mathcal{I}_a^{(2)}$ when it has a sufficiently good *a priori* input, e.g., with $\mathcal{I}_a^{(2)} > 0.88$. This would inevitable push the pinch-off signal-to-noise ratio (SNR) limit [2] (denoted as SNR_{off}) of the iterative decoder to a higher value. On the other hand, it also shows that with the same *a priori* input, the ABP-KV decoding that runs multiple Gaussian eliminations yields a better output. However, the EXIT characteristics does not predict the performance comparison among those parameters. Table I shows the codeword error rate (CER) of iterative decoding of the RS (63, 50) - conv. $(15, 17)_8$ code with different $(N_{\text{ADP}}, N_{\text{BP}})$ permutations in the additive white Gaussian noise (AWGN) channel. It can be seen that with 2 iterations, the (4, 1) permutation performs the best. But with 10 iterations, it is the (2, 1) permutation that prevails. Overall, the (2, 2) permutation yields a compromised performance and therefore it is chosen as the ABP decoding parameter in this paper.

Fig. 4 shows the EXIT chart of the proposed algorithm in decoding the RS (63, 50) - conv. $(15, 17)_8$ code. Puncturing with rates of $3/4$ and $3/5$ [3] are utilized to adjust the code rate to 0.529 and 0.661, respectively. It shows without puncturing, at 1.7dB, an exit tunnel starts to exist between the MAP and the ABP-KV algorithms' EXIT curves. Therefore, the decoder's bit error rate (BER) cliff is predicted to start at 1.7dB. While for puncture rates of $3/4$ and $3/5$, the exit tunnel starts to exist at 2.5dB and 3.3dB, respectively. The BER performance of the concatenated code will be further validated in Section VI.

Based on the above analysis, the EXIT chart can be further utilized to design the inner code. With a chosen outer code, the ABP-KV algorithm's EXIT characteristics is determined. The inner code should be chosen such that it can result in a smaller SNR_{off} value. Table II shows with the (63, 50) RS code as the outer code, the SNR_{off} values that are associated with

[2]The SNR value at which the algorithm's BER cliff starts to happen.

[3]If the first 4 consecutive convolutional coded bits are $b_1\ b_2\ b_3\ b_4$, the rate 3/4 punctured output are $b_1\ b_2\ b_4$. If the first 10 consecutive convolutional coded bits are $b_1\ b_2\ b_3\ b_4\ b_5\ b_6\ b_7\ b_8\ b_9\ b_{10}$, the rate 3/5 punctured output are $b_1\ b_2\ b_4\ b_5\ b_8\ b_9$.

TABLE I
CER PERFORMANCE COMPARISON OF ITERATIVE DECODING OF THE RS (63, 50) - CONV. $(15, 17)_8$ CODE, SNR = 3DB.

N_{ITER} \ $(N_{\text{ADP}}, N_{\text{BP}})$	(2, 2)	(2, 1)	(3, 1)	(4, 1)
2	5.67×10^{-4}	6.70×10^{-4}	2.86×10^{-4}	2.10×10^{-4}
10	3.10×10^{-5}	3.00×10^{-5}	7.33×10^{-5}	1.90×10^{-4}

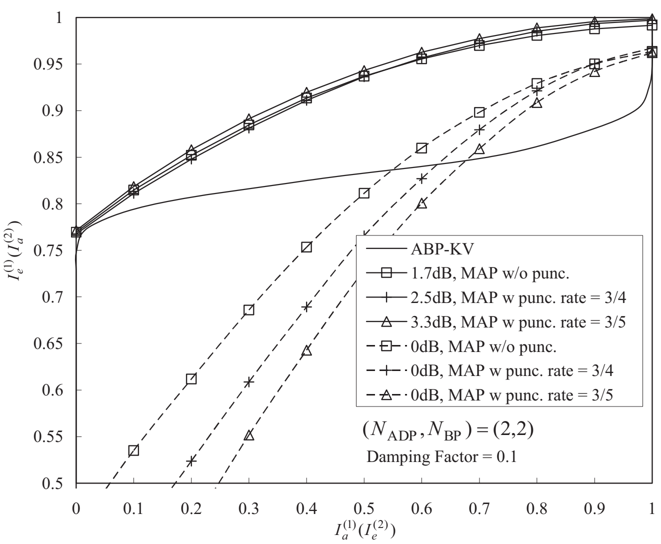

Fig. 4. EXIT chart of iterative soft decoding of the RS (63, 50) - conv. $(15, 17)_8$ code with different puncturing rates.

TABLE II
PINCH-OFF SNR LIMITS FOR DIFFERENT RS (63, 50) - CONV. CODES.

Inner codes	$(15, 17)_8$	$(1, 37/23)_8$	$(1, 5/7)_8$	$(1, 21/37)_8$
SNR_{off}	1.7dB	1.4dB	1.5dB	1.3dB

different choice of inner codes. It can be seen that utilizing a systematic feedback inner code allows a lower SNR_{off} value for the iterative decoder. However, more than solely looking at the SNR_{off} value, the concatenated code's performance is also determined by its distance. An inner code with less states will affect the concatenated code's distance. Hence, although the $(1, 5/7)_8$ inner code yields a lower SNR_{off} value than the $(15, 17)_8$ inner code, the RS (63, 50) - conv. $(1, 5/7)_8$ code may not prevail in performance. However, if the inner codes have the same number of states, the one that yields a lower SNR_{off} value should enable the concatenated code to achieve a better performance. Again, this inner code design analysis will be further validated in Section VI.

V. Complexity Analysis

The iterative soft decoding algorithm requires three types of computations. They are the floating point operations that are required by the MAP and BP decodings, the binary operations that are required by Gaussian elimination and the finite field arithmetic operations that are required by KV decoding. We will first analyze the computational cost for each of the decoding processes, then analyze the overall decoding complexity.

With a chosen inner code that has Ω states and an input message length of $Dn\omega$ bits, SISO decoding of the inner code requires $O(12\Omega Dn\omega)$ floating point operations. For SISO decoding of RS codes, Gaussian elimination requires $O(n\omega(n\omega - k\omega)^2)$ binary operations. Let Θ denote the average row weight of matrix $\mathbf{H}_b'$, each BP iteration requires $O((n-k)\omega\Theta^2)$ floating point operations. Finally, performing KV decoding with a factorization output list size of l requires $O(\Lambda^2(\mathbf{M})(l+1))$ finite filed arithmetic operations.

With the above mentioned knowledge, we can now analyze the worst case decoding complexity of the proposed algorithm by considering all the predefined decoding parameters are reached. To decode an RS codeword, there are at most $N_{\text{ADP}}N_{\text{BP}}$ BP iterations. Therefore, together with the MAP decoding, the proposed algorithm requires at most

$$O(N_{\text{ITER}}(12\Omega Dn\omega + N_{\text{ADP}}N_{\text{BP}}D(n-k)\omega\Theta^2)) \tag{38}$$

floating point operations. During each iteration in decoding the concatenated code, there are at most N_{ADP} Gaussian eliminations and KV decodings in decoding an RS codeword. Therefore, the proposed algorithm requires at most

$$O(N_{\text{ITER}}N_{\text{ADP}}Dn\omega(n\omega - k\omega)^2) \tag{39}$$

binary operations, and at most

$$O(N_{\text{ITER}}N_{\text{ADP}}D\Lambda^2(\mathbf{M})(l+1)) \tag{40}$$

finite field arithmetic operations. If we improve the outer SISO decoding by increasing the N_{GR} value, the algorithm will then require at most $O(N_{\text{ITER}}(12\Omega Dn\omega + N_{\text{GR}}N_{\text{ADP}}N_{\text{BP}}D(n-k)\omega\Theta^2))$ floating point operations. Meanwhile, the binary operations (39) and finite field arithmetic operations (40) will also be scaled up by a factor of N_{GR}. However, notice that the actual decoding complexity is less than the above mentioned computational scales. First, not all the outer SISO decodings would require N_{ADP} matrix adaptations. The iterative ABP-KV decoding terminates once the transmitted RS codeword has been found. Second, since the decoded RS codeword will not be processed again, the number of RS decoding events decreases as the iteration progresses and it is quite often that all the D RS codewords would have been decoded before the maximal iteration number N_{ITER} is reached.

It can be realized that the proposed algorithm's complexity is far greater than the existing decoding algorithms. For example, the classical one-shot Viterbi-BM algorithm requires only $O(2\Omega Dn\omega)$ floating point operations and $O(Dn^2)$ finite filed arithmetic operations. While the iterative MAP-KV algorithm [11] requires at most $O(12N_{\text{ITER}}\Omega Dn\omega)$ floating point operations and $O(N_{\text{ITER}}\, D\Lambda^2(\mathbf{M})(l+1))$ finite field arithmetic operations. Addressing the complexity issue to facilitate the iterative decoding will be the author's future work.

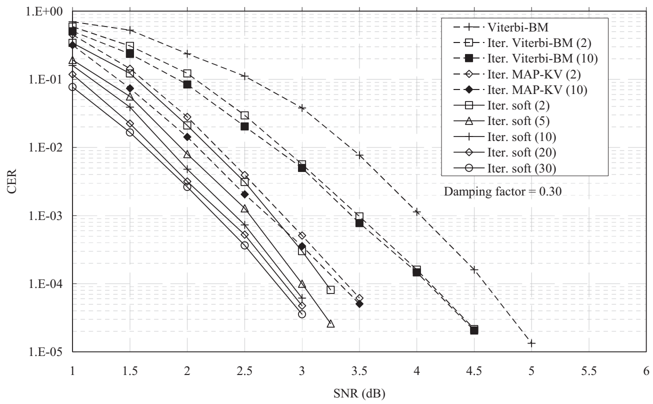

Fig. 5. Iterative soft decoding performance of the RS (15, 11) - conv. $(5,7)_8$ code.

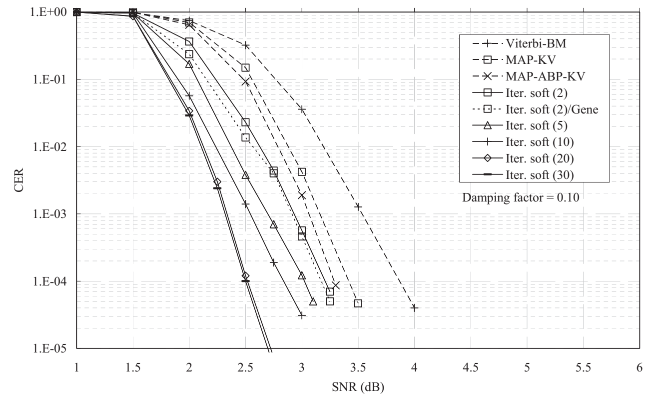

Fig. 7. Iterative soft decoding performance of the RS (63, 50) - conv. $(15,17)_8$ code.

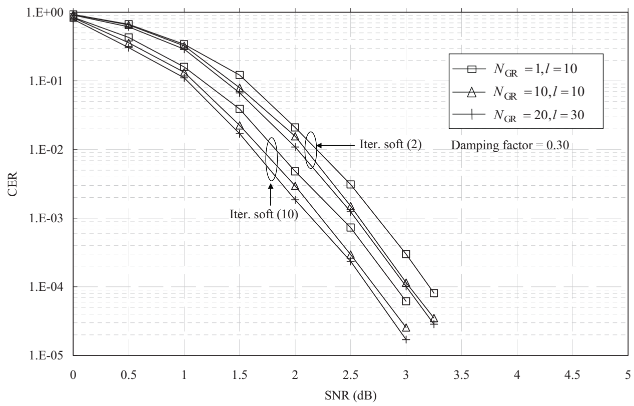

Fig. 6. Improved iterative soft decoding performance of the RS (15, 11) - conv. $(5,7)_8$ code.

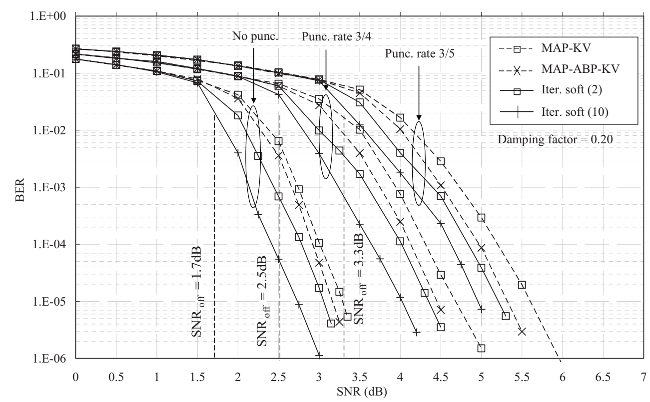

Fig. 8. Iterative soft decoding performance of the punctured RS (63, 50) - conv. $(15,17)_8$ code.

VI. Performance Analysis

This section analyzes the proposed algorithm's error-correction performance in the AWGN channel with using the binary phase shift keying (BPSK) modulation. As mentioned in Section V, the ABP decoding parameters are set as $(N_{\text{ADP}}, N_{\text{BP}}) = (2,2)$. Without specifically mentioning, the KV decoding parameter is set as $l = 10$ and the ABP decoding is running with $N_{\text{GR}} = 1$.

Fig. 5 shows the CER performance of iterative soft decoding of the RS (15, 11) - conv. $(5,7)_8$ code. The RS decoding output is validated by the ML criterion [25]. The performance comparison benchmarks include the classical one-shot Viterbi-BM algorithm, the iterative Viterbi-BM algorithm of [7] and the iterative MAP-KV algorithm of [11]. It can be observed that significant performance improvements can be made over the benchmark schemes. For the existing iterative decoding approaches, due to the lack of extrinsic information being given as the RS decoding feedback, the performance improvements made by iterations is limited. In contrast, the proposed algorithm's performance can be improved significantly by increasing the number of iterations. Asymptotically, iterative soft decoding with 30 iterations achieves almost 2dB gain over the one-shot Viterbi-BM algorithm. However, as the number of iterations increases, the achievable performance improvement becomes marginal. Since increasing the iteration number results in a more complex decoding, our result shows 10 is a compromised iteration number by considering the performance-complexity tradeoff. It is important to point out that this RSCC code has a particularly short length. Its powerful error-correction performance makes it attractive to some communication scenarios in which the short packet length is common, e.g., the wireless sensor networks. As mentioned in Section III.C, the decoding performance can be improved by strengthening the RS decoding. Fig. 6 shows the performance improvement made by having stronger ABP and KV decodings. For example, with 2 iterations, increasing N_{GR} from 1 to 20 and l from 10 to 30 allows a 0.25dB decoding gain at CER of 10^{-4}. However, it is also noticed that the limited performance improvements are achieved by an expensive computational cost. Hence, for RSCC codes, performing the iterative soft decoding with $N_{\text{GR}} = 1$ and a moderate l value would be recommended.

Fig. 7 shows the CER performance of the RS (63, 50) - conv. $(15,17)_8$ code. The CRC-4 code is utilized for the RS decoding output validation. The comparison benchmarks include the Viterbi-BM, MAP-KV and MAP-ABP-KV algorithms, all of which are one-shot decodings. Note that the MAP-ABP-KV algorithm corresponds to the proposed algorithm with $N_{\text{ITER}} = 1$. Again, it shows significant performance improvements over the existing schemes. Since utilizing the CRC-4 code causes a rate loss for the concatenated code from

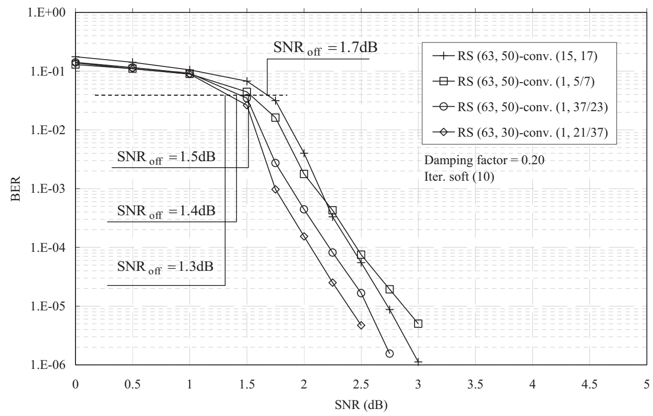

Fig. 9. Iterative soft decoding performance of the RS (63, 50) outer code concatenating with different inner codes.

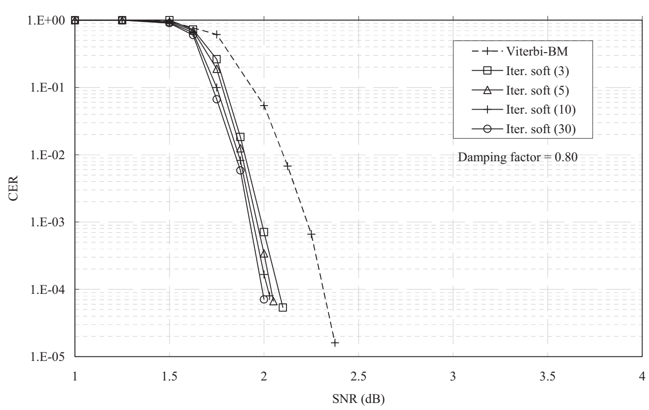

Fig. 10. Iterative soft decoding performance of the RS (255, 223) - conv. $(133, 171)_8$ code.

0.397 to 0.392, Fig. 7 also shows the performance difference between using an aided gene and the CRC code. The rate loss further causes a 0.05dB performance loss which can be tolerated in practice. Fig. 8 shows the BER performance of the punctured RS (63, 50) - conv. $(15, 17)_8$ code with puncturing rates of $3/4$ and $3/5$. It validates the pinch-off SNR limits that were predicted by the EXIT analysis of Fig. 4. Fig. 9 further shows the BER performance of the concatenated code with different choice of inner codes. Similarly, it validates the analysis of Table II by showing the BER cliff starts at the predicted SNR_{off} values. It can be seen that with the same number states, the inner code that exhibits a lower SNR_{off} value yields a better BER performance. While comparing the RS (63, 50) - conv. $(1, 5/7)_8$ code and the RS (63, 50) - conv. $(15, 17)_8$ code, the former one outperforms in the low to medium SNR region as it has a lower SNR_{off} value. However, with SNR > 2.2dB, the latter one outperforms. This is due to the fact that with SNR being increased, most of the decodings do not need iterations. Consequently, the code's distance becomes a more dominant factor in decoding.

The RS (255, 223) - conv. $(133, 171)_8$ code has been widely used in the space communications [3] [4]. Fig. 10 shows the CER performance of this concatenated code. In iterative soft decoding, the RS decoding output is validated by the CRC-16 code. Following the standard, the depth of the block interleaver is 8. The BP process for decoding the outer code utilizes the extrinsic LLR calculation of (25). It can be seen that error-correction performance improvements can be made over the currently used Viterbi-BM algorithm. Iterative soft decoding with $N_{\text{ITER}} = 5$ achieves a 0.3dB gain over the Viterbi-BM algorithm at CER of 10^{-4}. However, marginal performance improvement can be further made with an iteration number greater than 5. Overall, our results show that the proposed iterative soft decoding algorithm can also be applied to improve the existing space communication systems.

VII. Conclusions

This paper has proposed an iterative soft decoding algorithm for the popular RSCC codes. SISO decodings of the inner and outer codes are accomplished by the MAP algorithm and the ABP-KV algorithm, respectively. The ABP algorithm enhances the reliability of the soft information from the inner decoder. At the meantime, it delivers the extrinsic probabilities of RS coded bits in an efficient manner. With the improved soft information, the algebraic KV algorithm will be performed to retrieve the transmitted message. After the ABP-KV decoding of multiple RS codewords, the deterministic probabilities of the decoded bits and the extrinsic probabilities of the undecoded bits are fed back for the next round MAP decoding. The proposed algorithm enables the extrinsic probabilities of RS coded bits to be iterated in a soft information exchange decoding mechanism, well exploiting the error-correction potential of RSCC codes. The EXIT analysis of the proposed algorithm is presented, analyzing its iterative decoding behavior. The analysis has shown the EXIT characteristics of the ABP-KV algorithm and predicted the error-correction performance of the concatenated codes. Moreover, it leads to the insight of choosing the inner codes. Our complexity analysis has further revealed the implementation cost of the proposed algorithm. Finally, our simulation results have demonstrated that significant performance improvements can be achieved over the existing decoding algorithms, and validated the EXIT analysis. Therefore, the proposed decoding algorithm can be considered for implementation towards improving the performance of RSCC codes.

References

[1] G. D. Forney, *Concatenated Codes*. MIT Press, 1966.

[2] R. D. Cideciyan, E. Eleftheriou, and M. Rupf, "Concatenated Reed-Solomon/convolutional coding for data transmission in CDMA-based cellular systems," *IEEE Trans. Commun.*, vol. 45, no. 10, pp. 1291–1303, Oct. 1997.

[3] Consultative Committee for Space Data Systems, "Recommendation for space data systems standard, telemetry channel coding," CCSDS 101.0-B-2, *Blue Book*, issue 2, Jan. 1987.

[4] Consultative Committee for Space Data Systems, "Recommendation for space data systems standard, telemetry channel coding," CCSDS 101.0-B-6, *Blue Book*, issue 6, Oct. 2002.

[5] A. Viterbi, "Error bounds for convolutional codes and assymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, pp. 260–269, 1967.

[6] J. L. Massey, "Shift register synthesis and BCH decoding," *IEEE Trans. Inf. Theory*, vol. 15, pp. 122–127, 1969.

[7] E. Paaske, "Improved decoding for a concatenated coding system recommended by CCSDS," *IEEE Trans. Commun.*, vol. 38, no. 8, pp. 1138–1144, Aug. 1990.

[8] 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.

[9] L. Chen, R. A. Carrasco, and E. G. Chester, "Performance of Reed-Solomon codes using the Guruswami-Sudan algorithm with improving interpolation efficiency," *IET Commun.*, vol. 1, no. 2, pp. 241–250, 2007.
[10] 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.
[11] J. Zhang, J. You, L. Lu, and X. Ma, "An iterative soft-decision decoding algorithm for conventional concatenated codes," in *Proc. 2009 Int. Conf. Commun. Netw. China*.
[12] M. Lamarca, J. Sala, and A. Martinez, "Iterative decoding algorithms for RS-convolutional concatenated codes," in *Proc. 2005 IEEE Int. Symp. Turbo Codes*.
[13] G. Schmidt, V. Sidorenko, and M. Bossert, "Collaborative decoding of intereleaved Reed-Solomon codes and concatenated code designs," *IEEE Trans. Inf. Theory*, vol. 55, no. 7, pp. 2991–3012, July 2009.
[14] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: turbo-codes (1)," in *Proc. 1993 IEEE Int. Conf. Commun.*
[15] 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.
[16] V. Ponnampalam and B. Vucetic, "Soft decision decoding for Reed-Solomon codes," *IEEE Trans. Commun.*, vol. 50, no. 11, pp. 1758–1768, Nov. 2002.
[17] 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.
[18] 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.
[19] L. Chen, "Iterative soft-decision decoding of Hermitian codes," *IEEE Trans. Commun.*, vol. 61, no. 1, pp. 33–42, Jan. 2013.
[20] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, no. 2, pp. 284–287, Mar. 1974.
[21] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge University Press, 1985.
[22] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*. Pearson Prentice Hall, 2004.
[23] 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.
[24] L. Chen, R. A. Carrasco, M. Johnston, and E. G. Chester, "Efficient factorisation algorithm for list decoding algebraic-geometric and Reed-Solomon codes," *Proc. 2007 IEEE Int. Conf. Commun.*, pp. 851–856.
[25] T. Kaneko, T. Nishijima, H. Inazumi, and S. Hirasawa, "An efficient maximum-likelihood-decoding algorithm for linear block codes with algebraic decoder," *IEEE Trans. Inf. Theory*, vol. 40, no. 2, pp. 320–327, Mar. 1994.
[26] S. Tang, L. Chen, and X. Ma, "Progressive list-enlarged algebraic soft decoding of Reed-Solomon codes," *IEEE Commun. Lett.*, vol. 16, no. 6, pp. 901–904, June 2012.
[27] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.
[28] 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.

Li Chen (S'07-M'08) received his B.Sc. degree in applied physics from Jinan University, China, in 2003, M.Sc. degree in communications and signal processing, and Ph.D. degree in mobile communications in 2004 and 2008, respectively, both from Newcastle University of the United Kingdom. In 2010, he joined the School of Information Science and Technology, Sun Yat-sen University of China, where he is now an Associate Professor and Associate Head of the Department of Electronic and Communication Engineering. From 2007 to 2010, he was a Research Associate with Newcastle University. During 2011–2012, he was 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). He is a principle investigator of two National Natural Science Foundation of China (NSFC) projects 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.