Improved Soft-Decision Decoding of RSCC Codes

Li Chen
School of Information Science and Technology, Sun Yat-sen University
Guangzhou, China. Email: chenli55@mail.sysu.edu.cn

***Abstract*—Reed-Solomon convolutional concatenated (RSCC) codes are a popular coding scheme for wireless communications. However, the current decoding algorithm for the outer code, i.e., the Reed-Solomon (RS) code, employs hard-decision decoding and cannot make full use of the soft information provided by the decoder of the inner code. Consequently, the concatenated code's error-correction potential is not fully exploited. This paper proposes an improved soft-decision decoding algorithm for the RSCC codes. The maximum *a posteriori* (MAP) algorithm is applied to decode the inner code, providing soft information for the outer code. The iterative decoding algorithm that can approach the maximum likelihood (ML) decoding performance for RS codes is applied to decode the outer code, exploiting the benefits of the soft output of the inner decoder. The iterative decoding of RS codes integrates the adaptive belief propagation (ABP) algorithm and the Koetter-Vardy (KV) list decoding algorithm, namely the ABPKV algorithm. Our performance analysis shows that sizable error-correction performance gains can be achieved over the conventional decoding scheme. The complexity of the proposed decoding scheme will also be presented, discussing the implementation cost for achieving the performance improvement.**

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

I. Introduction

Concatenated codes were first introduced by Forney [1] in the 1960s. It showed that the concatenation of a nonbinary outer code and a binary inner code could constitute a capacity approaching error-correction code with a polynomial-time decoding complexity. One popular example of the concatenated codes is the legacy Reed-Solomon convolutional concatenated (RSCC) codes, with the Reed-Solomon (RS) code and the convolutional code being the outer code and the inner code, respectively. The inner code is good at correcting spread bit errors, while the outer code is good at correcting burst errors. Such combinatorial functions ensure that RSCC codes are a popular coding scheme whose application can be found in wireless and space communications [2] [3].

The conventional decoding scheme for RSCC codes employs the Viterbi algorithm [4] for the inner code and the Berlekamp-Massey (BM) algorithm [5] for the outer code, respectively. In order to strengthen the error-correction performance, a block interleaver is usually employed between the inner and outer codes. In [2], the RSCC code is used in code division multiple access (CDMA) systems. Specifically, interleaver design was proposed to cope with the multiple access interference. Moreover, the error-correction performance upper bound for the concatenated coding scheme was derived considering the presence of multi-path fading and multiple access interference. In [3], an improved decoding algorithm that performs repeated decoding trials for RSCC codes was proposed. Information on the outcome of BM decoding is given as the constraints for the next round Viterbi decoding. However, existing efforts have not fully exploited the error-correction potential of the concatenated code. On one hand, the maximum *a posteriori* (MAP) algorithm [6] can be used to decode the inner code, providing the *a posteriori* probability (APP) values for the RS coded bits. On the other hand, soft decoding of RS codes has been well developed in recent years. They include the Koetter-Vardy list decoding algorithm [7] and the iterative soft decoding algorithms [8] [9], both of which are capable of making a better use of the MAP outputs.

This paper proposes an improved soft-decision decoding algorithm for RSCC codes. The MAP algorithm is employed to decode the inner code, providing the APP values for the following RS decoding. The APP values are then deinterleaved symbol wise. The deinterleaved APP values will be utilized by the iterative soft decoding of the outer code. The iterative soft decoding first performs the belief propagation (BP) algorithm based on an adaptive parity-check matrix whose density has been reduced according to the bit reliabilities. It is called the adaptive belief propagation (ABP) algorithm. It provides improved symbol reliabilities for the following Koetter-Vardy (KV) list decoding algorithm. The KV algorithm finalizes the decoding of each RS codeword frame. The iterative soft decoding algorithm is therefore named the ABPKV algorithm. It has been shown that the ABPKV algorithm can approach the maximum likelihood (ML) error-correction performance bound for the RS codes [9]. Hence, the decoding algorithm for the inner code is optimal, while the decoding algorithm for the outer code approaches the optimal performance. The proposed MAP-ABPKV soft-decision decoding scheme is capable of better exploiting the error-correction potential of the concatenated codes. To the best of the author's knowledge, this is the first time a bit level soft-decision decoding algorithm for RS codes is utilized in the concatenated coding scheme. It can therefore make better use of the RS coded bits' APP which are inherited from the MAP algorithm. Our performance analysis shows significant performance improvements can be obtained over the conventional Viterbi-BM decoding algorithm. In order to give an insight into the implementation cost of the proposed algorithm, a discussion on the decoding complexity will also be provided.

The rest of the paper is organized as the follows. Section II presents the preliminary knowledge of the RSCC code. Section III presents the proposed soft-decision decoding scheme for the concatenated code. Sections IV and V present the com-

978-1-4673-5939-9/13/$31.00 ©2013 IEEE

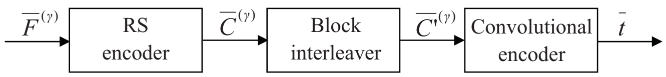

Fig. 1. Block diagram of the RSCC encoder.

plexity and performance analyses of the proposed decoding scheme, respectively. Finally, Section VI concludes the paper.

II. RSCC Codes

The encoder block diagram of an RSCC code is shown by Fig. 1. Let $\mathcal{D}$ denote the depth of the block interleaver, indicating there are $\mathcal{D}$ RS codeword being interleaved. Let γ denote the index of the RS codeword and $1 \leq \gamma \leq \mathcal{D}$. The code rates of the RS code and the convolutional code are r_1 and r_2, respectively. As a result, the code rate of the concatenated code is $r_1 r_2$. Let $\mathbb{F}_q$ denote the finite field of size q and $\mathbb{F}_q = \{\rho_1, \rho_2, \ldots, \rho_q\}$. In this paper, it is assumed that $q = 2^\omega$ and ω is an integer that is greater than 1.

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

$$\overline{F}^{(\gamma)} = [F_1^{(\gamma)}, F_2^{(\gamma)}, \ldots, F_k^{(\gamma)}] \in \mathbb{F}_q^k, \tag{1}$$

where n and k are the length and dimension of the code, respectively. The generator matrix $\mathbf{G}$ of the RS code is defined as

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

where α is a primitive element of $\mathbb{F}_q$. The codeword can be generated by

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

Note that in the remainder of the paper, the superscript (γ) denotes the variable belongs to the γth RS codeword.

In order to perform the ABP decoding for RS codes, it is necessary for the parity-check matrix to be known. It is defined as

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

With knowledge of the companion matrix $\mathbf{A}$ [10] of the primitive polynomial for $\mathbb{F}_q$, the equivalent binary image of matrix $\mathbf{H}$ can be generated by replacing its entries α^i by $\mathbf{A}^i$, where $i = 0, 1, \ldots, q-2$. It is called the binary parity-check matrix $\mathbf{H}_b$ with dimensions $(n-k)\omega \times n\omega$.

The $\mathcal{D}$ codeword are then interleaved by a block interleaver, yielding the interleaved codeword denoted as

$$\overline{C'}^{(\gamma)} = [C_1'^{(\gamma)}, C_2'^{(\gamma)}, \ldots, C_n'^{(\gamma)}]. \tag{5}$$

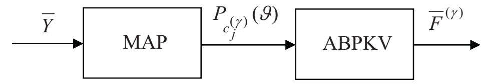

Fig. 2. Block diagram of the MAP-ABPKV algorithm

Note that the interleaved codeword $\overline{C'}^{(\gamma)}$ may not be a valid RS codeword. The $\mathcal{D}$ interleaved codeword are then read out horizontally, giving an interleaved codeword sequence

$$\overline{C'}^{(1)}, \overline{C'}^{(2)}, \ldots, \overline{C'}^{(\mathcal{D})}.$$

It is then converted to a binary interleaved codeword sequence

$$c_1'^{(1)}, c_2'^{(1)}, \ldots, c_{n\omega}'^{(1)}, \ldots, c_1'^{(\mathcal{D})}, c_2'^{(\mathcal{D})}, \ldots, c_{n\omega}'^{(\mathcal{D})},$$

which is the input to the convolutional encoder. In this paper, it is assumed that the convolutional code is a nonsystematic nonrecursive code with $r_2 = \frac{1}{2}$. Its generator polynomials $G_1(D)$ and $G_2(D)$ are represented in an octal form [10]. Notice that variable rates of the RSCC codes can be realized by puncturing the convolutional codeword. We denote $\bar{t}$ as the convolutional codeword defined by

$$\bar{t} = [t_1, t_2, \ldots, t_N] \in \mathbb{F}_2^N, \tag{6}$$

where $N = 2n\omega\mathcal{D}$.

III. The MAP-ABPKV Decoding

This section presents the MAP-ABPKV soft-decision decoding algorithm for RSCC codes. Its block diagram is shown in Fig. 2.

A. The MAP Algorithm

Given $\overline{Y} \in \mathbb{R}$ as the received vector observed from the channel. We are able to obtain the APP values for each convolutional coded bit t_i as

$$P_{t_i}(\vartheta) = \Pr[t_i = \vartheta \mid \overline{Y}], \tag{7}$$

where $i = 1, 2, \ldots, N$ and $\vartheta \in \{0, 1\}$. The MAP algorithm [6] will then be applied based on the trellis of the convolutional code. The APP values of (7) are used to determine the trellis transition probabilities during the *forward trace* and *backward trace* of the MAP algorithm [6]. It yields the APP values for the interleaved RS coded bits $c_j'^{(\gamma)}$ as

$$P_{c_j'^{(\gamma)}}(\vartheta) = \Pr[c_j'^{(\gamma)} = \vartheta \mid \overline{Y}], \tag{8}$$

where $j = 1, 2, \ldots, n\omega$. They will then be deinterleaved, yielding the APP values for the RS coded bits $c_j^{(\gamma)}$ as

$$P_{c_j^{(\gamma)}}(\vartheta) = \Pr[c_j^{(\gamma)} = \vartheta \mid \overline{Y}]. \tag{9}$$

Notice that since every ω consecutive pairs of bit APP values $(P_{c_j'^{(\gamma)}}(0), P_{c_j'^{(\gamma)}}(1))$ constitute the APP values for an RS codeword symbol, they are grouped together to participate in the deinterleaving process.

B. The ABPKV Algorithm

By reading out the deinterleaved APP values horizontally, we can obtain the bit APP values for the $\mathcal{D}$ RS codeword. For the γth RS codeword, we have

$$P_{c_1^{(\gamma)}}(\vartheta), P_{c_2^{(\gamma)}}(\vartheta), \ldots, P_{c_{n\omega}^{(\gamma)}}(\vartheta). \quad (10)$$

The ABPKV algorithm is now applied to decode each of the $\mathcal{D}$ RS codewords. Based on the bit reliability information of (10), the ABPKV algorithm will first perform Gaussian elimination on the binary parity-check matrix $\mathbf{H}_b$, reducing its density and eliminating some of its short cycles. Then, BP decoding will be performed to enhance the bit reliabilities. Finally, KV list decoding will be performed based on the enhanced reliability, to find the message vector $\overline{F}^{(\gamma)}$.

The log-likelihood ratio (LLR) value of bit $c_j^{(\gamma)}$ can be determined by

$$L(c_j^{(\gamma)}) = \ln\left(\frac{P_{c_j^{(\gamma)}}(0)}{P_{c_j^{(\gamma)}}(1)}\right). \quad (11)$$

The LLR vector $\overline{L}^{(\gamma)}$ for the γth RS codeword is

$$\overline{L}^{(\gamma)} = [L(c_1^{(\gamma)}), L(c_2^{(\gamma)}), \ldots, L(c_{n\omega}^{(\gamma)})]. \quad (12)$$

A larger magnitude $|L(c_j^{(\gamma)})|$ implies bit $c_j^{(\gamma)}$ is more reliable. Hence, all the magnitudes $|L(c_j^{(\gamma)})|$ will be sorted in an ascending order. It yields a refreshed bit index sequence $\delta_1, \delta_2, \ldots, \delta_{(n-k)\omega}, \ldots, \delta_{n\omega}$, indicating

$$|L(c_{\delta_1}^{(\gamma)})| < |L(c_{\delta_2}^{(\gamma)})| < \cdots < |L(c_{\delta_{(n-k)\omega}}^{(\gamma)})| < \cdots < |L(c_{\delta_{n\omega}}^{(\gamma)})|. \quad (13)$$

Let $B \subseteq \{1, 2, \ldots, n\omega\}$ be a set of the bit indices and $|B| = (n-k)\omega$. With $B = \{\delta_1, \delta_2, \ldots, \delta_{(n-k)\omega}\}$ collecting the indices of the $(n-k)\omega$ least reliable bits, the sorted LLR vector becomes

$$\overline{L}_B^{(\gamma)} = [L(c_{\delta_1}^{(\gamma)}), L(c_{\delta_2}^{(\gamma)}), \ldots, L(c_{\delta_{(n-k)\omega}}^{(\gamma)}), \cdots, L(c_{\delta_{n\omega}}^{(\gamma)})]. \quad (14)$$

Notice that the complementary set $B^c = \{1, 2, \ldots, n\omega\} \setminus B$. For matrix $\mathbf{H}_b$, Gaussian elimination will be performed on the columns that correspond to the bits of B. Let Υ_δ denote the weight-1 column vector with 1 at its δth entry and 0 elsewhere. Gaussian elimination reduces column δ_1 to Υ_1, then reduces column δ_2 to Υ_2 etc. It attempts to reduce the first $(n-k)\omega$ independent columns defined by B to weight-1 columns. However, it is not guaranteed all the columns that are indicated by B can be reduced. In that case, the columns w.r.t. the bordering bits between sets B and B^c will be reduced. The above mentioned process is called matrix adaptation. It results in an updated binary parity-check matrix $\mathbf{H}'_b$.

Let $h_{uj} \in \{0, 1\}$ denote the entry of $\mathbf{H}'_b$ with $1 \leq u \leq (n-k)\omega$ and $1 \leq j \leq n\omega$. Let

$$\mathbf{U}(j) = \{u \mid h_{uj} = 1,\ \forall\ 1 \leq u \leq (n-k)\omega\}, \quad (15)$$

$$\mathbf{J}(u) = \{j \mid h_{uj} = 1,\ \forall\ 1 \leq j \leq n\omega\}. \quad (16)$$

Iterative BP decoding will be performed, yielding the extrinsic bit LLR values

$$L_{ext}(c_j^{(\gamma)}) = \sum_{u \in \mathbf{U}(j)} 2\tanh^{-1}\Big(\prod_{\tau \in \mathbf{J}(u)\setminus j} \tanh\Big(\frac{L(c_\tau^{(\gamma)})}{2}\Big)\Big). \quad (17)$$

After a number of BP iterations, the LLR value for the coded bit $c_j^{(\gamma)}$ is updated by

$$L^*(c_j^{(\gamma)}) = L(c_j^{(\gamma)}) + \eta \cdot L_{ext}(c_j^{(\gamma)}), \quad (18)$$

where $0 < \eta \leq 1$ is the damping factor [9]. They form an updated LLR vector, i.e.,

$$\overline{L^*}^{(\gamma)} = [L^*(c_1^{(\gamma)}), L^*(c_2^{(\gamma)}), \ldots, L^*(c_{n\omega}^{(\gamma)})]. \quad (19)$$

We can then transform the updated LLR value into the enhanced APP values for bit $c_j^{(\gamma)}$ by

$$P^*_{c_j^{(\gamma)}}(0) = \frac{1}{1 + e^{-L^*(c_j^{(\gamma)})}}, \quad (20)$$

$$P^*_{c_j^{(\gamma)}}(1) = \frac{1}{1 + e^{L^*(c_j^{(\gamma)})}}. \quad (21)$$

With knowledge of each RS coded bit's enhanced APP values, the reliability matrix $\mathbf{\Pi}$ w.r.t. an RS codeword $\overline{C}^{(\gamma)}$ can be formed. Based on the binary decomposition of each field element ρ_μ ($\mu = 1, 2, \ldots, q$), every ω consecutive pairs of bit APP values will be multiplied in q different permutations, generating a column of the reliability matrix $\mathbf{\Pi}$. Its entry $\pi_{\mu\nu}$ is the APP value for the RS codeword symbol $C_\nu^{(\gamma)}$ ($\nu = 1, 2, \ldots, n$) and defined as

$$\pi_{\mu\nu} = \Pr[C_\nu^{(\gamma)} = \rho_\mu \mid \overline{Y}]. \quad (22)$$

Matrix $\mathbf{\Pi}$ will then be transformed into a multiplicity matrix $\mathbf{M}$ [7] with entries $m_{\mu\nu}$. The cost of matrix $\mathbf{M}$ is

$$\Lambda(\mathbf{M}) = 0.5 \sum_{\mu,\nu} m_{\mu\nu}(m_{\mu\nu} + 1), \quad (23)$$

which indicates the number of the interpolation constraints [7]. Interpolation will be carried out based on the instruction of $\mathbf{M}$, yielding an interpolated polynomial $Q(x, y)$ [11]:

$$Q(x, y) = \sum_{a,b \in \mathbb{N}} Q_{ab} x^a y^b, \quad (24)$$

where $\mathbb{N}$ denotes the set of nonnegative integers and $Q_{ab} \in \mathbb{F}_q$. Factorization will then be carried out [12] [13], finding the y roots of the interpolated polynomial by

$$\{p(x) \mid Q(x, p(x)) = 0 \text{ and } \deg p(x) \leq k - 1\}. \quad (25)$$

The coefficients of $p(x)$ form a decoded message vector. Observe that by increasing the factorization output list size l, i.e., the y-degree of Q, the KV algorithm will have a better error-correction capability.

It is important to mention that the ABPKV algorithm is an iterative decoding process. Let $\mathcal{N}_{\text{ADP}}$ and $\mathcal{N}_{\text{BP}}$ denote the number of the matrix adaptations and the BP iterations, respectively. Based on each adapted binary parity-check matrix $\mathbf{H}'_b$, $\mathcal{N}_{\text{BP}}$ BP iterations will be performed, yielding an improved reliability matrix $\mathbf{\Pi}$. The KV algorithm will then be deployed to find the decoded message candidates, which are stored in the global decoding output list $\mathcal{L}$. If there are multiple matrix adaptations, i.e., $\mathcal{N}_{\text{ADP}} > 1$, the updated LLR vector $\overline{L^*}^{(\gamma)}$ will be fed back, and mapped to the original LLR vector as $\overline{L^*}^{(\gamma)} \mapsto \overline{L}^{(\gamma)}$. Based the newly updated LLR vector $\overline{L}^{(\gamma)}$, the next round bit reliability sorting process will be performed. It is then followed by Gaussian elimination which produces another updated binary parity-check matrix $\mathbf{H}'_b$. Based on $\mathbf{H}'_b$, the BP decoding and KV list decoding will be performed. If a new message candidate is produced, it will be added to the global decoding output list $\mathcal{L}$.

The incentive of reducing the columns defined by B to weight-1 is to prevent the propagation of unreliable information during the BP process in (17) and (18). At the same time, their reliabilities are likely to be improved. However, it is possible that the bit LLR values of B^c are incorrectly estimated. If we can ensure their corresponding columns will be reduced to weight-1, they will also have the opportunity to be corrected. Therefore, after the initial sorting process, we can exchange the bit indices between B and B^c, generating a different set of bit indices B [9]. For example, with $z < \min\{k\omega, (n-k)\omega\}$, we can regenerate B as

$$B = \{\delta_{(n-k)\omega+1}, \ldots, \delta_{(n-k)\omega+z}, \delta_1, \ldots, \delta_{(n-k)\omega-z}\}. \quad (26)$$

The proceeding Gaussian elimination will reduce columns that are defiend by B to weight-1. Based on each regenerated bit indices set B, the above mentioned APBKV decoding algorithm will be performed. Hence, given $\mathcal{N}_{\text{B}}$ as the number of bit indices set B, the ABPKV algorithm will produce at most $l\mathcal{N}_{\text{B}}\mathcal{N}_{\text{ADP}}$ decoded message vectors in the output list $\mathcal{L}$. The one whose corresponding codeword has the minimal Euclidean distance to the received vector $\overline{Y}$ will be chosen as the decoded message.

IV. Complexity Analysis

This section analyzes the computational complexity of the above mentioned MAP-ABPKV algorithm. Recall $\mathcal{N}_{\text{B}}$, $\mathcal{N}_{\text{ADP}}$ and $\mathcal{N}_{\text{BP}}$ are the numbers of bit indices set, matrix adaptations and BP iterations, respectively. To decode each of the $\mathcal{D}$ RS codeword, the ABPKV algorithm will deploy $\mathcal{N}_{\text{B}}\mathcal{N}_{\text{ADP}}$ Gaussian eliminations and KV decodings, and $\mathcal{N}_{\text{B}}\mathcal{N}_{\text{ADP}}\mathcal{N}_{\text{BP}}$ BP iterations. Therefore, the MAP-ABPKV algorithm has a high decoding complexity. This section will give a numerical insight of the decoding complexity, revealing the computational cost. At the same time, a facilitated decoding strategy will also be suggested.

If the trellis of the convolutional code has Ω states, performing the MAP algorithm requires $O(8\mathcal{D}\Omega n\omega)$ floating point operations. The ABPKV decoding consists of Gaussian elimination, BP iteration and KV list decoding. Gaussian elimination requires $O(n\omega(n\omega - k\omega)^2)$ binary operations. While the BP iteration and KV decoding require $O((n\omega)^2)$ floating point operations and $O(\frac{2}{3}\Lambda^3(\mathbf{M}))$ finite field arithmetic operations, respectively. Hence, to decode an RSCC code with the interleaver depth of $\mathcal{D}$, the MAP-ABPKV algorithm requires

$$O(8\mathcal{D}\Omega n\omega + \mathcal{D}\mathcal{N}_{\text{B}}\mathcal{N}_{\text{ADP}}\mathcal{N}_{\text{BP}}(n\omega)^2) \quad (27)$$

floating point operations,

$$O(\mathcal{D}\mathcal{N}_{\text{B}}\mathcal{N}_{\text{ADP}}n\omega(n\omega - k\omega)^2) \quad (28)$$

binary operations, and

$$O(\mathcal{D}\mathcal{N}_{\text{B}}\mathcal{N}_{\text{ADP}}\frac{2}{3}\Lambda^3(\mathbf{M})) \quad (29)$$

finite field arithmetic operations, respectively.

Although the MAP-ABPKV algorithm inherits a high decoding complexity, the decoding process can be made more efficient by notifying the decoder once the intended message vector $\overline{F}^{(\gamma)}$ has been found, and the decoding process will be terminated. Practically, such an output validation function can be realized by utilizing a cyclic redundant check (CRC) code. In fact, when the channel condition is sufficiently good, most message vectors can be found with $\mathcal{N}_{\text{B}} = 1$ and $\mathcal{N}_{\text{ADP}} = 1$, and the decoding complexity can be scaled down significantly.

V. Performance Analysis

This section analyzes the error-correction performance of the proposed MAP-ABPKV algorithm. It is measured on the additive white Gaussian noise (AWGN) channel using binary phase shift keying (BPSK) modulation scheme. The MAP-ABPKV algorithm is parameterized by the KV decoding output list size (l) and the ABP decoding parameters ($\mathcal{N}_{\text{B}}, \mathcal{N}_{\text{ADP}}, \mathcal{N}_{\text{BP}}$). In the presented results, the damping factor is set as $\eta = 0.20$, which optimizes the MAP-ABPKV decoding performance. All the shown RSCC codes employ an interleaver of depth $\mathcal{D} = 10$.

Fig. 3 shows the bit error rate (BER) performance of the RSCC code with the (15, 11) RS code as an outer code and the $(5, 7)_8$ convolutional code as an inner code. The inner code is a 4 states trellis code. It can be seen that the MAP-ABPKV algorithm achieves significant performance gains over the conventional Viterbi-BM algorithm. For example, with decoding parameters of $l = 10$ and $(\mathcal{N}_{\text{B}}, \mathcal{N}_{\text{ADP}}, \mathcal{N}_{\text{BP}}) = (1, 3, 3)$, 0.8dB performance gain can be achieved at a BER of 10^{-5}. Compared to the MAP-KV (optimal) algorithm, it has a 0.3dB performance gain. This is due to the fact that the iterative ABPKV algorithm has a stronger error-correction capability than the KV list decoding algorithm. It can make full use of the soft information provided by the MAP algorithm. Notice

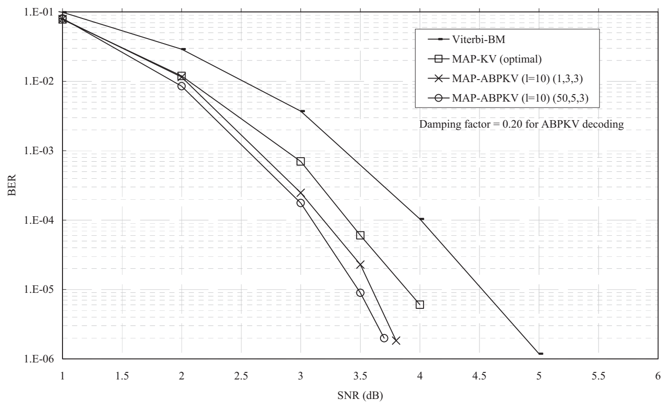

Fig. 3. Performance of the RS(15, 11)-conv$(5, 7)_8$ concatenated code over the AWGN channel

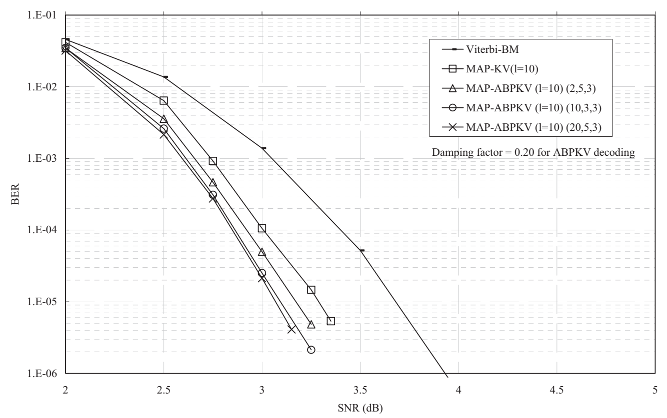

Fig. 4. Performance of the RS(63, 50)-conv$(15, 17)_8$ concatenated code over the AWGN channel

that the optimal KV decoding performance can be achieved with knowledge of the reliability matrix $\mathbf{\Pi}$ [7]. It shows the asymptotic optimal error-correction performance of the KV algorithm as $l \to \infty$. Further performance improvement can be achieved by enhancing the decoding parameters, e.g., with $(\mathcal{N}_{\text{B}}, \mathcal{N}_{\text{ADP}}, \mathcal{N}_{\text{BP}}) = (50, 5, 3)$. However, based on the analysis of Section IV, such a performance improvement is achieved with a larger computational cost.

Fig. 4 shows the BER performance of the RSCC code with the (63, 50) RS code and the 16-state $(15, 17)_8$ convolutional code as the outer code and inner code, respectively. Again, we can see significant performance gains can be achieved over the conventional Viterbi-BM algorithm. The above two figures show that significant performance gains can be achieved if the RS decoding algorithm can utilize the soft output of the inner code decoding algorithm, i.e., by deploying the KV list decoding algorithm or the iterative ABPKV algorithm.

Variable rates of the RSCC codes can be realized by puncturing the output of the inner code. Fig. 5 shows the performance of the above mentioned RSCC code with puncture

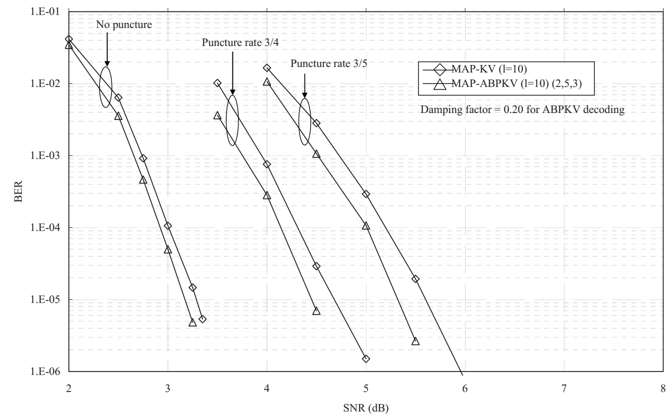

Fig. 5. Performance of the punctured RS(63, 50)-conv$(15, 17)_8$ concatenated code over the AWGN channel

rates of $3/4$ and $3/5$. If the first 4 consecutive bits of the convolutional codeword are:

$$t_1,\ t_2,\ t_3,\ t_4.$$

The rate $3/4$ puncture output is:

$$t_1,\ t_2,\ t_4.$$

It results in the code rate of the RSCC code being 0.529. If the first 10 consecutive bits of the convolutional codeword are:

$$t_1,\ t_2,\ t_3,\ t_4,\ t_5,\ t_6,\ t_7,\ t_8,\ t_9,\ t_{10}.$$

The rate $3/5$ puncture output is:

$$t_1,\ t_2,\ t_4,\ t_5,\ t_8,\ t_9.$$

It results in the code rate of the RSCC code being 0.661. Fig. 5 shows that by increasing the code rate, the MAP-ABPKV algorithm achieves a larger performance gain over the MAP-KV algorithm. It demonstrates when the code rate of the RSCC code increases, it is more beneficial to deploy the ABPKV algorithm to decode the outer code.

VI. Conclusion

This paper has proposed an improved soft-decision decoding algorithm for the RSCC codes, called the MAP-ABPKV algorithm. The MAP algorithm is used to decode the inner code, while the ABPKV algorithm is used to decode the outer code. Since the MAP algorithm delivers the bit APP values for the interleaved RS codeword, it is desirable to have an RS decoder that can fully utilize those bit APP values. The ABPKV algorithm that integrates the ABP algorithm and the KV list decoding algorithm is the best candidate. The ABP algorithm is bit reliability oriented, providing the improved symbol wise reliability information for the following KV algorithm. A complexity analysis of the proposed MAP-ABPKV algorithm has also been presented to demonstrate the computational cost of the improved soft-decision decoding approach. Our performance analysis showed that the MAP-ABPKV algorithm achieves sizable performance gains over the conventional Viterbi-BM algorithm. Variable rates of RSCC codes have been realized by puncturing the output of the convolutional code. It has been also shown that by increasing the code rate, the MAP-ABPKV algorithm achieves more significant performance gain over the MAP-KV algorithm, demonstrating its advantage in high rate scenarios. Therefore, the proposed MAP-ABPKV algorithm is a more advanced decoding approach for the RSCC codes and can be considered for more practical applications.

ACKNOWLEDGMENT

This work is supported 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.

REFERENCES

[1] G. D. Forney, *Concatenated Codes.* MIT Press, Cambridge, MA, 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 (10), pp. 1291-1303, Oct. 1997.

[3] E. Paaske, "Improved decoding for a concatenated coding system recommended by CCSDS," *IEEE Trans. Commun.*, vol. 38 (8), pp. 1138-1144, Aug. 1990.

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

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

[6] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, pp. 284-287, Mar. 1974.

[7] R. Koetter and A. Vardy, "Algebraic soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Inform. Theory*, vol. 49 (11), pp. 2809-2825, Nov. 2003.

[8] J. Jiang and K. Narayanan, "Iterative soft-input-soft-output decoding of Reed-Solomon codes by adapting the parity check matrix," *IEEE Trans. Inform. Theory*, vol. 52 (8), pp. 3746-3756, Aug. 2006.

[9] M. El-Khamy and R. McEliece, "Iterative algebraic soft-decision list decoding of Reed-Solomon codes," *IEEE J. Sel. Area in Commun.*, vol.24 (3), pp. 481-490, Mar. 2006.

[10] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications,* 1et editon, Prentice Hall, 1983.

[11] 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 (2), pp.241-250, 2007.

[12] R. Roth and G. Ruckenstein, "Efficient decoding of Reed-Solomon codes beyond half the minimum distance," *IEEE Trans. Inform. Theory*, vol. 46 (1), pp. 246-257, Jan. 2000.

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