

# An Efficient SISO Algorithm for Reed-Solomon Codes

Vishakan Ponnampalam and Alex Grant

**Abstract**—There has been renewed interest in iterative decoding algorithms for concatenated codes since the introduction of Turbo codes. Such iterative methods are built on top of soft-in-soft-out algorithms. Powerful concatenated codes may be constructed with linear block codes as constituent codes. Such codes can be better alternatives to Turbo Codes either when high coding rates or when short interleaver lengths are required. This paper presents a computationally efficient maximum a-posteriori (MAP) soft-in-soft-out (SISO) algorithm for RS codes and related codes.

**Index Terms**—Reed-Solomon codes, soft decoding, iterative decoding, concatenated codes, SISO algorithms.

## I. INTRODUCTION

Since the introduction of Turbo codes [1] there has been a strong interest in concatenated coding schemes and associated iterative decoding algorithms. RS codes have long been recognized as a powerful class of linear block codes and have been widely used as constituent codes for concatenated codes. The high coding rates offered by RS codes, make them good candidates for constituent codes.

A number of SISO algorithms for linear block codes have appeared in the literature [2–4]. The trellis based algorithm proposed by Hagenauer et al [2] gives good performance for binary BCH codes. Hagenauer et al [5] later showed that such a scheme can achieve within 0.27 dB of Shannon Capacity when applied on high rate product Hamming codes. However the complexity of the trellis inhibits the extension of this method to RS codes. The same problem is faced when applying Fossorier and Lin's ordered statistics based algorithm [3] to these codes. Pyndiah [4, 6] proposed a low complexity SISO algorithm for RS codes based on the well known Chase Type 2 algorithm. However simulation results show that the performance of this algorithm is significantly lower than that of MAP decoding [4]. This can be attributed to the poor performance of the Chase Type 2 algorithm. A sub-optimal SISO algorithm, related to the one proposed in this paper was earlier presented by the Ponnampalam and Vucetic [7]. However this algorithm was of significantly higher complexity.

This paper presents a SISO algorithm of moderate complexity for RS codes and a set of its sub-field sub-codes. The proposed algorithm is related to MLD algorithms for RS codes proposed by Vardy and Beery [8] and subsequently by Ponnampalam and Vucetic [9]. Vardy and Beery [8] showed that

binary images of RS codes can be represented as linear combinations of two sub-field sub-codes which results in an efficient decoding algorithm. We exploit this property to derive an efficient soft-output MAP decoding algorithm for RS codes. The algorithm can also be applied to some sub-codes of RS codes, which as we will show later have good performance. The algorithm is also well suited for hardware implementation, as a significant number of computations may be performed in parallel. We also show that the proposed algorithm is many orders lower in complexity compared to MAP algorithm applied on the Wolf trellis.

The remainder of this paper is organized as follows. Section II gives a brief overview of algebraic properties of RS codes. The proposed decoding algorithm is presented in Section III. Simulation results for the proposed algorithm applied on product RS codes are given in Section IV. Finally, Section V, gives conclusions and directions for future work.

## II. BACKGROUND

Let  $\mathbf{g}_{\text{RS}}(X)$  be the generator polynomial of an  $(n, k, d_{\min}^H)$  RS code,  $\mathcal{C}_{\text{RS}}$ , over  $\text{GF}(2^m)$ . If  $\alpha$  is a primitive element of  $\text{GF}(2^m)$ ,  $\mathbf{g}_{\text{RS}}(X)$  is given by

$$\mathbf{g}_{\text{RS}}(X) = \prod_{i=1}^{2t} (X + \alpha^i) \quad (1)$$

where  $d_{\min}^H = 2t + 1 = n - k + 1$ . For every  $\mathcal{C}_{\text{RS}}$ , there exists a  $(n, k', d_{\min}^{H'})$  binary BCH code,  $\mathcal{C}_{\text{BCH}}$ , generated by polynomial  $\mathbf{g}_{\text{BCH}}(X)$  with roots  $\{\alpha, \alpha^2, \alpha^3, \dots, \alpha^{2t}\}$  and their cyclotomic conjugates over  $\text{GF}(2^m)$ . The message length  $k'$ , is less than or equal to  $k$  and the minimum distance  $d_{\min}^{H'}$  is greater than or equal to  $d_{\min}^H$ . Define a transformation  $\phi: \text{GF}(2^m) \rightarrow \text{GF}(2^m)$  with basis  $\{\gamma_1, \gamma_2, \dots, \gamma_m\}$ , where  $\gamma_i \in \text{GF}(2^m)$ . Using this transformation,  $\phi$ , any element  $c_i \in \text{GF}(2^m)$  can be written as,

$$c_i = \sum_{j=1}^m c_i^{(j)} \gamma_j \text{ where } c_i^{(j)} \in \text{GF}(2) \quad (2)$$

Let  $\mathcal{E}$  represent the set of coset leader terms of  $\mathcal{C}_{\text{BCH}}$  over  $\text{GF}(2)^n$ . Note that we can arbitrarily choose a member of a coset as its leader. Now rewrite (2) as

$$\begin{aligned} \mathbf{c}_{\text{RS}}(X) &= \sum_{j=1}^m \gamma_j \left[ \mathbf{c}_{\text{BCH}}^{(j)}(X) + \mathbf{1}^{(j)}(X) \right] \\ &= \sum_{j=1}^m \gamma_j \mathbf{c}_{\text{BCH}}^{(j)}(X) + \sum_{j=1}^m \gamma_j \mathbf{1}^{(j)}(X) \end{aligned} \quad (3)$$

V. Ponnampalam is with Cambridge Positioning Systems, Level 9, 123 Epping Rd, North Ryde, NSW 2113, Australia

A. Grant is with Institute for Telecommunications Research, Mawson Lakes, SA 5095, Australia, University of South Australia.

The work of A. Grant was supported in part by the Australian Government under ARC grant DP0344856.

where  $\mathbf{c}_{\text{BCH}}^{(j)}(X) \in \mathcal{C}_{\text{BCH}}$  and  $\mathbf{l}^{(j)}(X) \in \mathcal{E}$ .

Define codes  $\mathcal{B}$  and  $\mathcal{L}$  as

$$\mathcal{B} = \left\{ \mathbf{b}(X) \mid \mathbf{b}(X) = \sum_{j=1}^m \gamma_j \mathbf{c}_{\text{BCH}}^{(j)}(X), \mathbf{c}_{\text{BCH}}^{(j)}(X) \in \mathcal{C}_{\text{BCH}} \right\} \quad (4)$$

and

$$\mathcal{L} = \left\{ \mathbf{l}(X) \mid \mathbf{l}(X) = \sum_{j=1}^m \gamma_j \mathbf{l}^{(j)}(X), \mathbf{l}^{(j)}(X) \in \mathcal{E} \text{ and } \mathbf{l}(\beta) = 0 \right. \\ \left. \text{for } \beta \in \{\alpha, \alpha^2, \dots, \alpha^{2t}\} \right\}. \quad (5)$$

Observe that the codewords belonging to  $\mathcal{B}$  are formed by *interleaving* BCH codewords. Similarly codewords belonging to  $\mathcal{L}$  are formed by interleaving coset leader terms. Also note that the RS code,  $\mathcal{C}_{\text{RS}}$  may be considered a linear combination of  $\mathcal{B}$  and  $\mathcal{L}$ . These observations have been made by Vardy and Be'ery [8]. In the above paper the generator matrix of the RS code is represented as a sum of the generator matrices of codes  $\mathcal{B}$  and  $\mathcal{L}$ . Since  $\mathcal{B}$  is a permutation of  $m$  BCH codewords,  $|\mathcal{B}| = 2^{m(k')}$ . Therefore the cardinality of  $\mathcal{L}$  is given by  $|\mathcal{L}| = 2^{m(k-k')}$ .

Re-call that the set of codewords of  $\mathcal{C}_{\text{RS}}$  are linear combination of  $\mathcal{B}$  and  $\mathcal{L}$ . A family of sub-field sub-codes of  $\mathcal{C}_{\text{RS}}$  may be generated by linear combination of  $\mathcal{B}$  and some subsets of  $\mathcal{L}$ . It can be shown that such sub-field sub-codes have the same minimum distance,  $d_{\min}^H$  as  $\mathcal{C}_{\text{RS}}$  and dimensions between  $K-1$  and  $k$ . The proposed algorithm can also be applied to such sub-field sub-codes.

### III. SISO ALGORITHM

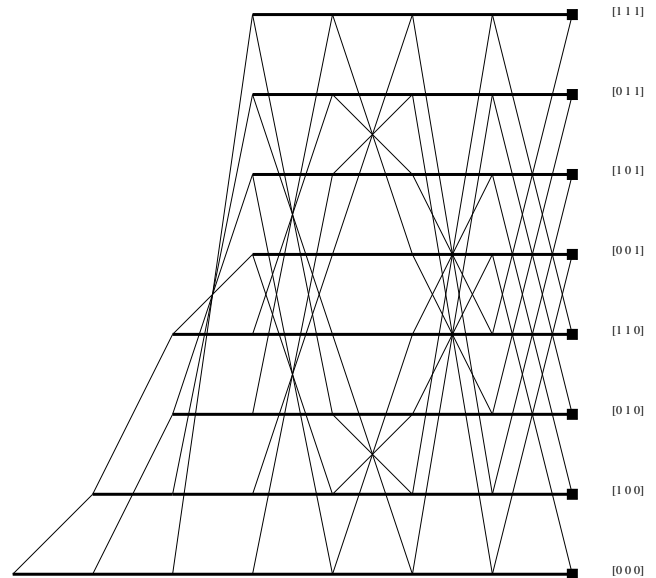
This section introduces the proposed SISO algorithm. We will attempt to describe the general idea without going into the mathematical details.

In the previous section we saw that the  $mN$ -bit binary image of a codeword of  $\mathcal{C}_{\text{RS}}$  can be decomposed into  $m$  segments corresponding to the basis of the transformation  $\phi$ . Now each of these segments are of length  $N$ . We also saw that any  $N$ -ary binary word can be represented as a sum of  $\mathcal{C}_{\text{BCH}}$  codeword and a coset leader term.

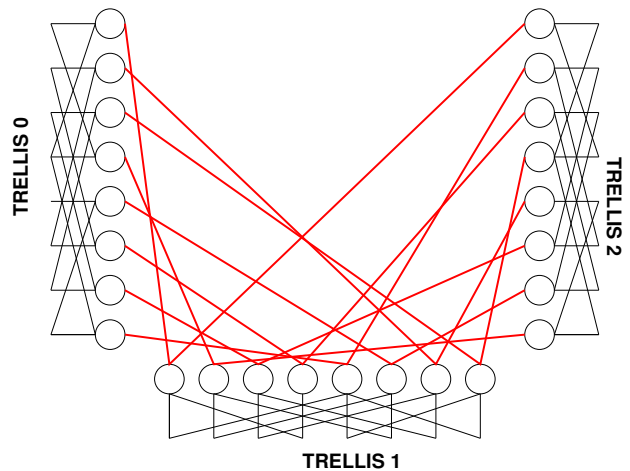
We can represent each of the segments using a binary trellis. The trellis is constructed by the Wolf method [10] using the parity check matrix of  $\mathcal{C}_{\text{BCH}}$ . Such a trellis will contain paths corresponding to all possible binary  $N$ -tuples. Furthermore paths corresponding to the  $N$ -tuples of the same coset will terminate at a common terminal node. Fig. 1(a) shows such a trellis corresponding to the  $(7, 5, 3)$  RS constructed using the parity check matrix of the  $(7, 4, 3)$  BCH code.

We construct  $m$  such identical trellises which correspond to the  $m$  segments of the codeword. We also saw in the previous section that RS codewords correspond to only certain coset leader *permutations*. This is governed by the elements of  $\mathcal{L}$ . Each member of  $\mathcal{L}$  corresponds to a *permutation* of  $m$  cosets of  $\mathcal{C}_{\text{BCH}}$ . Since each terminal node corresponds to a coset of  $\mathcal{C}_{\text{BCH}}$ ,

we can represent  $\mathcal{L}$  as *connections* between the terminal nodes of each segment. Note that the same process can be applied to a subset of  $\mathcal{L}$  associated with a sub-field sub-code of  $\mathcal{C}_{\text{RS}}$ . Fig. 1(b) shows the *connections* between the terminal nodes of the three trellises for the  $(7, 5, 3)$  RS code. There are a total of eight connections between the terminal nodes corresponding to the eight members of  $\mathcal{L}_{(7,5,3)}$ .



(a) Segment Trellis



(b) Terminal Node Connections

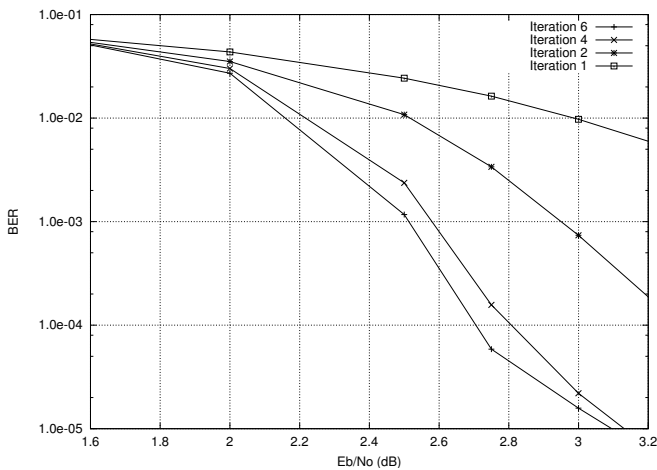
Fig. 1: Trellis and terminal node connections for the  $(7, 5, 3)$  RS code.

We have now constructed  $m$  binary trellises of depth  $N$  and formed connections between their terminal nodes. The proposed MAP SISO algorithm is executed as follows. First we do the forward recursion independently on each trellis to compute the forward recursion metric denoted  $\alpha$  for each node. This step can be performed in parallel for the  $m$  trellises. Now we have reached the terminal nodes in each trellis. The probability of being at the a particular terminal node of a particular trellis

lis is dependent on the probabilities of being at terminal nodes connected to it. This is taken into account in the backward recursion. We set the backward propagation metric,  $\beta$  of the terminal nodes using the forward metrics of the connected nodes. Once we do this we can perform the backward recursion in the traditional way. This step can also be performed in parallel for the  $m$  trellises.

#### IV. SIMULATION

In this section we present bit error rate performances of some product codes decoded using the proposed algorithm. A product code is constructed by parallel concatenation of two identical RS codes or sub-codes. Traditionally, the interleaver length of the product code is set to the squared of the message length of the component codes. However the interleaver length can be adjusted as required. It is assumed that the code is transmitted over an AWGN channel using antipodal signaling. We use a block interleaver for all codes presented in this section. It should be noted that block interleaver may not be optimal for such codes and that it might be possible to improve the performance by using better interleavers.

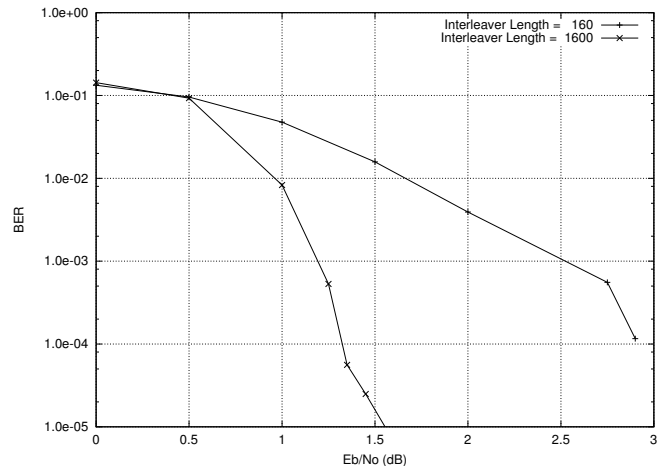


**Fig. 2:** BER performance of the  $(15, 13, 3) \times (15, 13, 3)$  product code with interleaver length 2704 bits.  $R = 0.76$

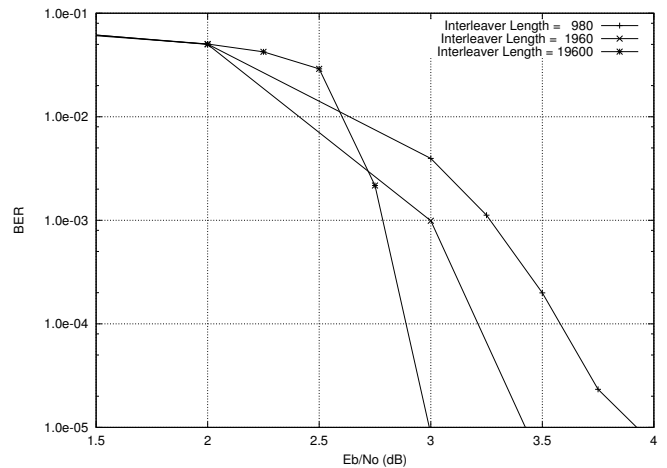
Fig. 2 shows the bit error rate (BER) performance a product code constructed using the  $(15, 13, 3)$  RS code. The interleaver size is 2704 bits and the code rate is approximately 0.76. Observe that a BER of  $10^{-5}$  is achieved around 3.1dB after just six iterations. Out of the three codes used in this section, this code has the least decoding complexity.

Fig. 3 shows the performance of a product code constructed using a  $(15, 10, 5)$  sub-field sub-code of the  $(15, 11, 5)$  RS code. The performance using interleaver sizes of 160 and 1600 are shown. Observe that a BER of  $10^{-5}$  can be achieved after six iterations at approximately 3 dB for interleaver size of 160 and at approximately 1.5 dB for the case when the interleaver length is set to 1600 bits. Observe that the code rate is equal to 0.5 for this product code.

Finally, Fig. 4 presents the performance of product code with a relatively high coding rate of 0.82. The component code is a  $(31, 28, 3)$  sub-field sub-code of the  $(31, 29, 3)$  RS code. The



**Fig. 3:** BER performance of the  $(15, 10, 5) \times (15, 10, 5)$  product code after six iterations.  $R = 0.50$



**Fig. 4:** BER performance of the  $(31, 28, 3) \times (31, 28, 3)$  product code after six iterations.  $R = 0.82$

figure shows performance of the code for three different interleaver lengths. For an interleaver length of 19600, a BER of  $10^{-5}$  is achieved around 3dB after six iterations.

As we saw in the previous section, the proposed algorithm uses a  $m$  trellises of depth  $N$  to compute the forward and backward metrics instead of a single  $mN$  depth trellis used in conventional trellis based SISO decoding. However there are some additional computations that need to be done to initialize the backward recursion. Table I gives the maximum number of nodes at each depth, depth and the additional operations needed for the proposed algorithm compared to the Wolf trellis. It can be seen from the table that the reduction in the trellis complexity outweighs the number of additional computations required for the proposed algorithm. The reduction in complexity is more

TABLE I: Decoding Complexity

Code	Proposed Algorithm			Wolf Trellis Based MAP	
	Max Nodes	Depth $\times m$	Extra Comps	Max Nodes	Depth
$(15, 13, 3) \times (15, 13, 3)$	16	60	64	256	60
$(15, 10, 5) \times (15, 10, 5)$	256	60	16348	$2^{20}$	60
$(31, 28, 3) \times (31, 28, 3)$	32	155	5120	32768	155

profound for longer codes and larger minimum distances.

## V. CONCLUSIONS

We have presented a computationally efficient soft-output MAP algorithm for RS codes and some of its sub-field sub-codes. The algorithm is used to iteratively decode concatenated coding schemes constructed using these codes. Simulation results show that good bit error rate performance can be achieved with relatively low interleaver length and high coding rates. We have not attempted to design appropriate interleavers for concatenated RS codes in this paper. Using well designed interleavers may further enhance the error performance of the proposed algorithm.

## REFERENCES

- [1] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error correcting coding and decoding: turbo codes," *Proceedings of the IEEE ICC*, pp. 1064–1070, May 1993.
- [2] E. O. J. Hagenauer and L. Papke, "Iterative decoding of binary block codes and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar 1996.
- [3] M. Fossorier and S. Lin, "Soft-input-soft-output decoding of linear block codes based on ordered statistics," *Proceedings of IEEE GLOBECOM*, vol. 5, pp. 2828–2833, Nov 1998.
- [4] R. Pyndiah, A. Glavieux, A. Picard, and S. Jacq, "Near optimal decoding of product codes," *Proceedings of the IEEE GLOBECOM*, pp. 339–343, Nov 1994.
- [5] J. H. H. Nickl and F. Burkert, "Approaching Shannon's capacity limit by 0.27 db using simple Hamming codes," *Comm. Lett.*, vol. 1, pp. 130–132, Sep 1997.
- [6] R. Pyndiah, "Near-optimum decoding of product codes: Block turbo codes," *IEEE Trans. Commun*, vol. 46, pp. 1003–1010, Aug 1998.
- [7] V. Ponnampalam and B. Vucetic, "A soft-in-soft-out algorithm for reed-solomon codes," in *Proc. IEEE Personal, Indoor, and Mobile Radio Conference*, (Osaka, Japan), September 1999.
- [8] A. Vardy and Y. Be'ery, "Bit level soft-decision decoding of Reed-Solomon codes," *IEEE Trans. Commun*, vol. 39, pp. 440–444, Mar 1991.
- [9] V. Ponnampalam and B. Vucetic, "Soft decision decoding of Reed-Solomon codes," *IEEE Trans. Commun*, Nov 2002.
- [10] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 76–80, Jan 1978.