Impact of Bit-Flip Combinations on Successive Soft Input Decoding of Reed Solomon Codes

Obaid ur Rehman and Natasa Zivic
Institute for Data Communications Systems
University of Siegen
Hoelderlinstrasse 3, 57076 Siegen, Germany
email: obaid.ur-rehman@uni-siegen.de and natasa.zivic@uni-siegen.de

***Abstract*- Soft Input Decoding of Reed Solomon codes using successive iterative bit-flipping was introduced earlier. The proposed method uses a concatenation of Convolutional and Reed Solomon Codes. It was shown that the proposed method not only improves the coding gain, but also helps in circumventing the miscorrections, that occur with the given probability, using the hard decision Reed Solomon decoder. In this paper, the impact of bit-flip combinations on the soft decision Reed Solomon decoder is analyzed. Increase in the bit-flip combinations results in an increase in the error locator set of the iterative decoding algorithm. It is shown that by increasing the possible bit-flip combinations using a threshold, the coding gain of the concatenated codes increases while the miscorrections rate decreases. The impact of the increase in the threshold on the decoder performance is also investigated in this paper. Simulations are performed for different code rates and an improvement in the coding gain is obvious through the simulation results presented. Moreover, the simulation statistics show a decrease in the miscorrections with an increase in the size of the bit-flip combinations.**

Keywords*- *Concatenated Codes; Reed Solomon Codes; BCJR; Successive Decoding; Miscorrection Detection.

I. INTRODUCTION

An iterative method for the Soft Input Decoding of Reed Solomon (RS) codes with successive iterative bit-flip decoding was shown earlier [1]. The proposed method considered concatenated Convolutional/RS Codes with CRC for error detection. Code concatenation is a method to increase the decoding capability of the individual codes by concatenating them together in a serial or parallel manner. Code Concatenation was first introduced by G. David Forney [2]. It is a method of combining two (normally different) codes to obtain a better Bit Error Rate (BER). They have many practical applications such as in the Compact Disc technology, Digital Video Broadcast, WiMax and the Voyager program. A typical concatenated arrangement is to use the hard decision Reed Solomon codes as the outer code and the soft decision Viterbi code as the inner code with an interleaver in between to break the burst of errors.

In this work, concatenated codes with Reed Solomon code as the outer and a convolutional code with the corresponding BCJR/MAP [3] decoder as the inner code are considered. The Maximum a-posteriori Probability (MAP) decoder outputs the hard decoded data (the binary form of decoded data) along with the Log Likelihood Ratio (LLR) or the L-Value of each and every decoded bit. These LLRs (or L-Values) give an estimate of the confidence on the hard-decision bits. Thus lower the absolute value of the LLR (i.e., |LLR|) of a bit, the least reliable it is and vice versa. The hard decision RS decoding is done on the output of the MAP. The hard decision RS decoder will result in either a decoding success or a decoding failure. There is, however, a known probability of decoding error [4]. The decoding error is also termed as “miscorrection” in literature. Decoding error occurs when the decoder decodes the received word to a codeword other than the transmitted one. RS decoder “sees” the decoding error as successful decoding, so in this work a CRC code is used to detect and then avoid the decoding errors, with a high probability.

In case of a decoding failure and/or error, a successive iterative algebraic decoding on the Reed Solomon codes is performed with a different combination of bit-flips in each iteration. The number of the bit-flip combinations dictates the size of error locator set used in the successive iterative decoding. The size of the error locator set is directly proportional to the performance and the computational costs. In this work, the impact of an increase in the bit-flip combinations on the decoder, introduced in [1], is studied both analytically and via simulations.

It is shown that increasing the bit-flip combinations will increase the coding gain. The number of iterations, however, doubles with every next bit added to the bit-flip combination set. The idea of inversion of the least reliable bits was introduced earlier in Chase decoding algorithms [5] and the Generalized Minimum Distance (GMD) decoding [6] algorithms. Recently, the idea of bit-flipped decoding has developed much interest, mainly, due to the rediscovery of Low Density Parity Check Codes [7]. Similarly, soft decoding techniques for the RS codes were initiated in the pioneering work done by [8] and [9]. More recently, the author in [10] has treated the subject in more detail by introducing some soft decoding techniques using bit-level soft information.

The rest of this paper is organized as follows. In Section II, the Soft Input Decoding of Reed Solomon Codes with miscorrection detection and avoidance [2] is discussed briefly. In Section III, the impact of bit-flip combinations on the scheme is discussed. In Section IV, the simulation results are presented with different bit-flip combinations and different code rates. Finally, a conclusion is given in Section V.

II. SOFT DECODING OF REED SOLOMON CODES WITH MSCORRECTION DETECTION AND AVOIDANCE

In this section, the encoder and decoder for successive iterative decoding of RS codes with miscorrection detection and avoidance is introduced briefly.

A. The Encoding Process

The Encoding process used in this study is depicted in Fig. 1. Here, the encoder is a simple concatenated RS-Convolutional encoder. This code concatenation is widely used in many popular protocols such as the Digital Video Broadcast (DVB-S) and WiMax.

CRC of m-bits is computed on the data of length k-1 and the data along with the appended CRC are then encoded with a systematic RS encoder. In this case, the RS encoder produces a codeword (c) of n-symbols (each of m-bits over GF(q)), where $q=2^m$ and n = q-1 (or 2^m-1). The codeword c is represented as,

$$c = (c_1,c_2,c_3,\dots\dots\dots,c_n), \qquad c_i \in GF(q) \qquad (1)$$

This codeword (c) is then converted into its binary equivalent for encoding with a convolutional encoder. The binary equivalent of "c" can be formally expressed as,

$$c_b=(c_{1,1},c_{1,2},\dots,c_{1,m},c_{2,1},c_{2,2},\dots,c_{2,m},\dots\dots\dots,c_{n,1},c_{n,2},\dots\dots\dots\dots,c_{n,m}), \qquad c_{i,j} \in GF(2) \qquad (2)$$

This binary-vector c_b is then encoded by the convolutional encoder to produce the message u for transmission. The message u is then BPSK modulated and transmitted over the Additive White Gaussian Noise (AWGN) channel.

The role of the CRC code is to detect the decoding errors [4] made by the RS decoder. This detection helps in avoiding these decoding errors through a successive iterative decoding process.

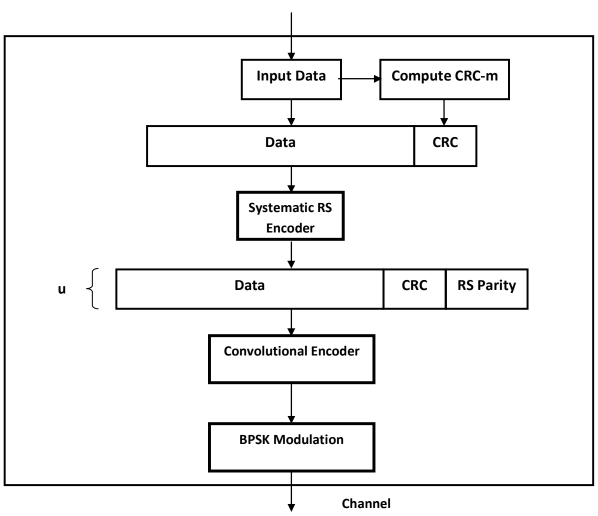

Figure 1. The Encoder

If the received word (r=u+e, where e is the error pattern) has a smaller Hamming distance to another valid codeword, the RS decoder will "miscorrect" it to this closest codeword. Such a miscorrection is called a decoding error [3] and in the decoder introduced in this work a CRC code is used to detect such a decoding error. CRC gives the ability of detecting the decoding errors at the cost of additional redundancy, thereby effectively reducing the code rate. This is catered-for in the simulation results by making comparisons with the corresponding low rate encoders (for fairness). In this work, CRC codes of different lengths for different RS code rates are considered. The CRC code, however, always occupy one symbol in the codeword in order to have a minimum impact on the overall code rate, as discussed in the section on simulation results. CRC is placed in the last m-bits of the RS data part, so that it gets protected from channel noise by the RS parity, allowing for the detection of RS decoder errors

B. The Soft Input Decoder

The successive iterative soft input decoding of the concatenated Convolutional/RS codes is illustrated in Fig. 2. The decoding starts with the Soft Input Soft Output (SISO) Log MAP decoding. The output of the Log MAP decoder is the soft estimate of the decoded data along with their reliability values (so called L-Values or the Log Likelihood Ratios (LLRs)). Hard decisions are made on the output of the MAP decoder and the binary data is fed to the hard decision RS decoder. The RS decoder will either successfully decode the data make a decoding error or fail to decode it, resulting in a decoding failure. These situations are summarized by the three cases given below,

1. The RS decoder succeeds to decode the given word. The CRC is recomputed on the k-1 decoded data symbols and compared with the CRC present in the k^{th} data symbol.
   a. If both the CRCs match, the decoding is successful and the iterative process stops.
   b. If the CRCs do not match, a decoding error is made by the RS decoder and some predefined number of attempt are made to correct the "miscorrected codeword".
2. The RS decoder fails to decode the given word. This happens when the RS decoder is neither successful in rightly decoding the given word, nor in miscorrecting it. In this case the "erroneous word" is not discarded, rather subjected to the iterative bit-flipping technique in order to try and recover the transmitted codeword (c).

In cases 1b and 2, the successive bit-flip decoding of RS codes is performed iteratively with a different combination of bit-flips in each iteration. The bit-flipping is done on the basis of |LLR| values obtained from the MAP decoder. A combination of bits from this "wrongly decoded" codeword is flipped followed by the hard decision RS decoding and CRC comparison. If the CRCs match, there is a success and the successive iterative decoding stops. If not, then the next combination of bits is flipped from the original "erroneous word".

 ISBN: 978-1-61208-126-7

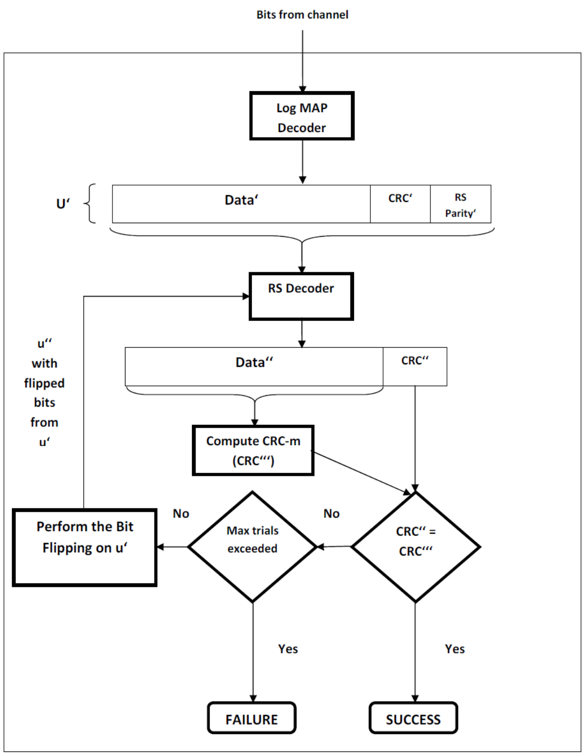

Figure 2. The soft input decoding of RS codes based on Bit-Flipping [1]

This process of bit-flipping and CRC comparison is repeated until either an RS decoding success followed by CRC comparison success or a threshold number of iterations. When the threshold is reached, the decoding fails.

The combination of the bits to be flipped is decided as follows. In the first iteration the least reliable bit is flipped. In the second iteration, the second least reliable bit is flipped. In the third iteration the first and the second least reliable bits are flipped and so on. This is limited by the number of bit-flip combinations chosen in advance, e.g., for 8-bit bit-flip combination a total of 2^8-1 combinations will be considered. The decoding error of the successive iterative bit-flip decoder is bounded by the following equation (for explanation and derivation of the following please refer to [1]),

$$P_E = P_{ERS} * P_{ECRC} \qquad (3)$$

where P_{ERS} is the probability of decoding error with the hard decision RS decoder and P_{ECRC} is the probability that the CRC collision will go undetected.

III. Impact of Bit-Flip Combinations On the Decoder

Increase in the size of the bit-flip combinations results in an increase in the decoding capability, by considering a larger set of errors. For bf = 8 bits, a set of 8 bit-flip errors are evaluated by the successive iterative decoder. Thus the error correction capability of the simple error-correction only RS decoder is increased up to an additional 8 symbols (if all the least reliable 8-bits belong to different symbols).

The error correction capability (E_{cc}) of the hard decision RS decoder is "t" i.e.,

$$E_{cc} = t \qquad (4)$$

where,

$$t=(n-k)/2 \qquad (5)$$

n is the codeword length and k is the dimension of the RS code, thus n-k is the number of parity symbols in the codeword. "t" is therefore half the number of parity symbols.

If the value of bf is increased from 8 to 16, this means the least reliable 16 bits will be considered in different combinations resulting in an increased decoding capability. If all the bits happen to be in different symbols then the decoding capability of RS codes can be extended up to bf-bits in total. With the successive iterative bit-flip decoding using a bit-flip combination of "bf" bits, the error correction capability (E_{ccbf}) can be given by the following equation,

$$Ecc_{bf} \leq t+bf, \quad 1 \leq bf \leq n*m \qquad (6)$$

e.g., for bf = 8-bits an additional 8 symbol errors are correctable if all the 8 least reliable bits belong to different symbols. Similarly for bf = 16 bits, the additional error correctional capability is increased by 16 symbols (again if all the least reliable bits belong to different symbols). In the worst case the error correction capability is enhanced by one symbol. This happens when all the least reliable bits belong to the same symbol. However by increasing the bf, the chances of all the bits belonging to the same symbol decreases.

It is, however, infeasible to increase the value of bf up to n*m. Due to practical limitations, a smaller threshold value needs to be chosen.

It is to be noted that the increase in the error correction capability is beyond the error correction capability of the Chase 2 decoding algorithm (which is 2t). The reason is the presence of the CRC code in the data part, which will prevent the decoding error (miscorrection) and bring the received word from the Hamming sphere of another valid codeword back to its original position (or in the worst case at least identify it as a decoding error).

This, however, comes at the cost of extra computations. By increasing the bit-flip combinations by one bit, i.e., from bf to bf+1, we are effectively raising the size of error location set from 2^{bf} to $2^{(bf+1)}$. This means, the number of iterations of the successive iterative decoder is doubled i.e.,

$$2^{(bf+1)} = 2*2^{bf} \qquad (7)$$

Increasing the value of bf, also results in an increased miscorrection detection and avoidance capability. This is because by increasing the value of bf, there are more error locations to be tried and corrected. If a decoding error occurs, it is detected with the help of CRC and tried in the next iterations for correction. Due to increased iterations, there remains a good probability that the errors in the received word will be corrected.

IV. SIMULATION RESULTS

Simulations are performed over an Additive White Gaussian Noise (AWGN) channel. A convolutional encoder of rate-½ is used with BPSK modulation. RS codes of different block lengths (n) over the corresponding Galois Field (GF (2^m)) are considered in the simulations. For each of the symbol size (m), a corresponding size of CRC code is chosen so as to occupy exactly one symbol, e.g., for m=8 the CRC-8 (CCITT) is used and for m=6, the CRC-6 (ITU) is used. Use of the CRC effectively reduces the code rate to (k-1)/n instead of k/n. To compensate for the reduced rate and give a fair comparison, the shortened RS codes are plotted for comparisons, e.g., RS (15, 11) with a 4-bit CRC is plotted against the same rate RS (12, 8) code. Symbol Error Rate (SER) vs. E_b/N_0 curves are shown in Figs. 3-5 to demonstrate the results.

Convolutional codes are good at converting the random channel errors to burst errors and RS codes have the excellent ability of correcting the burst errors. Burst errors may belong to a group of bits making one symbol or a group of symbols. Because of the excellent ability of the RS codes, to correct the symbol errors and erasures, SER (instead of Bit Error Rate) is plotted vs. the SNR, for better comparisons. It is to be noted that an improvement in SER suggests an improvement in the Bit Error Rate as well. In each of the plotted curves, the value of bit-flip combinations (bfc) is kept at 8, 12 and 16. It can be noticed from the curves that the SER drops with the increase in the bit-flip combinations.

A. RS(15, 11) code

Fig. 3 shows the simulation comparison for iteratively bit-flipped decoded Convolutional/ RS code for RS (15, 11). The effective code rate is 10/15 (i.e., (k-1)/n) because of the CRC, so it is compared with the same rate RS (12, 8) code. The simulation results show a coding gain of 0.25 dB at SER of 10^{-4} using a bf of size 8, a gain of 0.4 dB at same SER using a bf of size 12 and a coding gain of 0.5 dB using a bf of size 16.

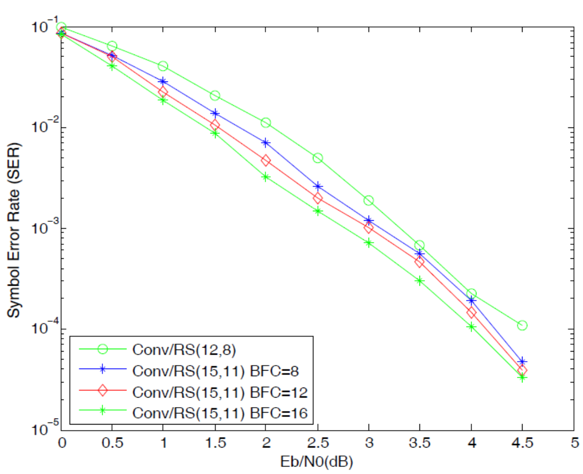

Figure 3. Convolutional/RS(15, 11) code with bit-flip combinations of 8, 12 and 16 bits compared to same rate RS(12, 8) code

B. RS(63, 55)

Fig. 4 shows the SER results for RS (63, 55) code compared with the same effective rate (i.e., 0.86) Reed Solomon code, i.e., RS (56, 48). A coding gain of 0.45 dB using a bfc of 8 bits is obtained at SER of 10^{-4}. A gain of 0.5 is obtained with a bf of size 12 and a gain of around 0.75 dB is obtained when 16-bit bit-flip combinations are considered.

C. RS(255, 223)

Fig. 5 shows an RS (255, 223) code with an 8-bit CRC, compared with the same effective rate RS(240, 208) code giving a coding gain of 0.2 dB using a bit-flip combination of 8 bits at SER of 10^{-3}. With 12-bit bit-flip combinations, the coding gain is 0.3dB and the coding gain increases to 0.4dB with a 16-bit bit-flip combinations.

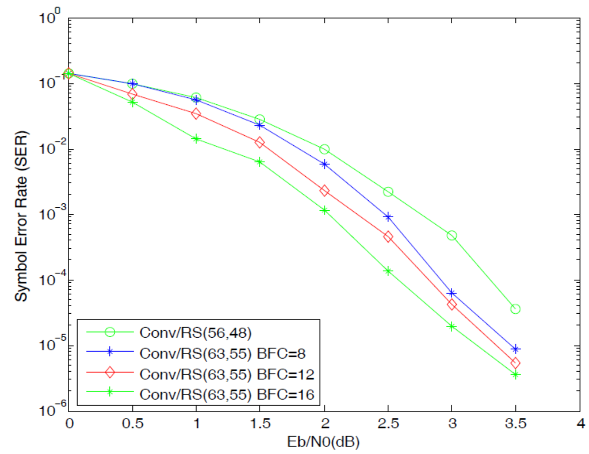

Figure 4. Convolutional/RS(63, 55) code with bit-flip combinations of 8, 12 and 16 bits compared to same rate RS(56, 48) code

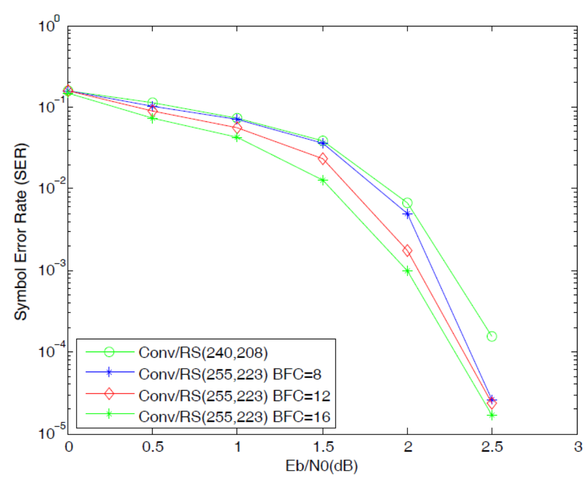

Figure 5. Convolutional/RS(255, 223) code with bit-flip combinations of 8, 12 and 16 bits compared to same rate RS(240, 208) code

In all of the simulations results, the curves for the different codes tend to meet with the curve for the hard decision Reed Solomon codes at the higher values of E_b/N_0. The reason for this is that at very high signal to noise ratio, the concatenated Convolutional/RS code is able to correct the errors in the received words with a high probability, thus giving almost the same SER as that of the bit-flipped scheme presented in this work. However the improvement in the SER is obvious for the smaller values of E_b/N_0 which is of more significance.

V. CONCLUSION AND FUTURE WORK

A scheme for the soft decoding of Reed Solomon codes is presented in this paper. The proposed scheme has the ability to correct errors beyond the decoding capability of Reed Solomon codes. The effect of the bit-flip combinations on the decoding scheme is analyzed. It is shown that by increasing the value of the threshold for the number of iterations to perform in the successive decoding of Reed Solomon codes, the error correcting capability increases and the probability of decoding error decreases. However, a very high threshold becomes practically infeasible.

The work presented in this paper is based on Error only Reed Solomon codes. In future it is planned to extend the work to Error and Erasure correcting Reed Solomon codes using symbol reliabilities in addition to the bit reliabilities. The error correction capability of the error and erasure RS codes is given by,

$$2*N_{err} + N_{era} <= 2t \quad (8)$$

where N_{err} is the number of errors and N_{era} is the number of erasures in the erroneous word. Using the symbol reliabilities, erasures can be introduced at known locations in the RS code and the decoding capability can be further enhanced by correcting up to 2t symbols. By being able to correct up to 2t erasures as well as the errors using the bit-flipping method discussed in this work, there is a likelihood of increasing the error correction capability beyond 2t.

REFERENCES

[1] O. U. Rehman and N. Zivic, Soft Input Decoding of Reed Solomon Codes with Miscorrection Detection and Avoidance, 4th International Conference on Signal Processing and Communication Systems, Gold Coast, Australia, pp. 1-5, Dec 13-15, 2010.

[2] G. David Forney, Concatenated Codes, MIT Press, Cambridge, MA, 1966.

[3] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, Optimal decoding of linear codes for minimizing symbol, IEEE Trans. Inform. Theory, vol. IT-20, pp. 284-287, Mar. 1974.

[4] R. J. McEliece and L. Swanson, On the decoder error probability for Reed-Solomon codes, IEEE Trans. Inform. Theory, vol. IT-32, pp. 701-703, 1986.

[5] D. Chase, A Class of Algorithms for Decoding Block Codes with Channel Measurement Information, IEEE Trans. Inform. Theory, IT-18, pp. 170-182, January 1972.

[6] G. David Forney, Generalized Minimum Distance decoding. IEEE Transactions on Information Theory, 12:125–131, 1966.

[7] D. J. C. MacKay and R. M. Neal, Good codes based on very sparse matrices, in Proc 5th IMA Conference Cryptography and Coding, (Lecture Notes in Computer Science), vol. 1025, pp. 100–111, 1995.

[8] M. Sudan, Decoding of Reed-Solomon beyond the error-correction bound, Journal of Complexity, vol. 13, pp. 180-193, Sep. 1997.

[9] R. Koetter and A. Vardy, Algebraic soft-decision decoding of Reed-Solomon codes, IEEE Trans. Information Theory, vol. 49, pp. 2809-2825, Nov. 2003.

[10] J. Jiang, Advanced Channel Coding Techniques using Bit Level Soft Information, PhD Thesis, Texas A & M University, Aug 2007.