Stochastic Chase Decoding of Reed-Solomon Codes

Camille Leroux, Saied Hemati, Shie Mannor, Warren J. Gross

Stochastic Chase Decoding of Reed-Solomon Codes

Camille Leroux, *Member, IEEE,* Saied Hemati, *Senior Member, IEEE,* Shie Mannor, *Senior Member, IEEE,*
and Warren J. Gross, *Senior Member, IEEE*

***Abstract*—In this letter, we propose a probabilistic approach to the generation of test patterns in the Chase Algorithm (CA) denoted as the Stochastic Chase Algorithm (SCA). We compare the performance of SCA with the regular CA for different Reed-Solomon codes. Simulation results show that the probabilistic nature of the SCA helps in providing a more efficient test pattern generation. SCA avoids the use of least reliable bits selection and reduces the number of candidate codewords up to 60% for the same decoding performance.**

***Index Terms*—Stochastic decoding, Chase algorithm, soft decision decoding, Reed-Solomon codes.**

I. Introduction

The Chase Algorithm (CA) [1] and Generalized Minimum Distance (GMD) decoding [2] are soft-decoding methods satisfying the property that a Hard Decision Decoder (HDD) is used to generate a set of candidate codewords that are compared by a likelihood measure. GMD and CA provide near-Maximum-Likelihood (ML) performance for high SNR values. It is important to efficiently generate a set of candidate codewords in such a way that the ML-codeword is included in the set. This set becomes rapidly large as the minimum distance of the code increases. We propose a probabilistic approach, denoted by Stochastic Chase Algorithm (SCA), to generate the candidate codewords for soft-decoding of Reed-Solomon (RS) codes.

II. Stochastic Test Pattern Generation in the Chase Algorithm

A. Model of the Communication System

Let $C(n,k,d)$ be a linear block code of size n, with dimension k and minimum distance d. An information vector $U = (u_1, u_2, ..., u_k)$ is encoded to a codeword $X = (x_1, x_2, ..., x_n) \in C$. Each x_i is transmitted over an Additive White Gaussian Noise (AWGN) channel using Binary Phase Shift Keying (BPSK) modulation. At the receiver side, the soft input of the decoder is computed from the received sequence $Y = (y_1, y_2, ..., y_n)$. The soft input can either be represented in the probability domain $P = (p_1, p_2, ..., p_n)$ with:

$$p_i = \Pr(y_i | x_i = 1) = \frac{1}{1 + e^{\frac{2y_i}{\sigma^2}}}$$

or in log-likelihood ratio domain $R = (r_1, r_2, ..., r_n)$ where

$$r_i = \frac{2y_i}{\sigma^2}, i = 1, ..., n.$$

Let $Y^H = (y_1^H, y_2^H, ..., y_n^H)$ be the bit-wise hard decision sequence such that:

$$y_i^H = \begin{cases} 0, & \text{if } r_i \geq 0 \text{ or } p_i \leq 0.5, \\ 1, & \text{otherwise.} \end{cases}$$

The reliability of the hard decision y_i^H is measured by the magnitude $|r_i|$ in log-likelihood ratio domain or equivalently by $|p_i - 0.5|$ in probability domain.

B. The Chase Algorithm

The CA [1] performs a hard decision decoding on a set of test patterns Y^m. A test pattern is devised by inverting at most λ least reliable bits in the received sequence Y. The decoded codeword is selected using the soft distance between Y and the corrected test pattern X^m (candidate codeword). Reference [1] proposed three different versions of this algorithm. They differ in the least reliable bit selection and in the test pattern generation method. The performance and the complexity of the algorithm increase with λ. In this work, we consider $\lambda \geq d$. The CA can be described using the following steps:

1) **For** $1 \leq i \leq n$, compute $r_i = \frac{2y_i}{\sigma^2}$.
2) Select the λ least reliable bits in R.
3) **For** $1 < m < 2^\lambda$,
   - Form the test pattern Y^m by inverting some of the least reliable bits,
   - Perform Berlekamp-Massey (BM) HDD on Y^m to get X^m,
   - Compute the soft weight of $Y^H \oplus X^m$:
     $$W(Y^H \oplus X^m) = \sum_{i=1}^{n} |r_i| (y_i^H \oplus x_i^m).$$
4) Select the decided word D such that:
   $$D = X^j,\ W(Y^H \oplus X^j) = \min_{m=1}^{2^\lambda} W(Y^H \oplus X^m).$$

For high-rate medium-size codes, the CA provides near-ML performance while having low complexity. But at lower rates (*i.e.*, larger d), the number of required test patterns grows exponentially with d.

C. The Stochastic Chase Algorithm

In the proposed Stochastic Chase Algorithm, the test pattern selection is modeled as a bit-wise stochastic experiment directed by the observation of the channel output. Each test pattern bit y_i^m is generated according to the reliability of the

Manuscript received April 12, 2010. The associate editor coordinating the review of this letter and approving it for publication was M. Lentmaier.

The authors are with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, H3A2A7, Canada (e-mail: {camille.leroux, saied.hemati, shie.mannor, warren.gross}@mcgill.ca). S. Mannor is also with the Department of Electrical Engineering, Technion, Haifa Israel (e-mail: shie@ee.technion.ac.il).

Digital Object Identifier 10.1109/LCOMM.2010.09.100594

received bit $|p_i - 0.5|$. The SCA consists of the following steps:

1) **For** $1 \leq i \leq n$,
   - **If** $p_i \leq 0.5 - \theta$ **then** $p_i = 0$, where $0 < \theta < 0.5$,
   - **Else If** $p_i \geq 0.5 + \theta$ **then** $p_i = 1$,
   - **Else** $p_i = \frac{1}{1+e^{\beta y_i}}$, where $\beta > 0$.
2) **For** $1 < m < \tau$
   - **For** $1 < i < n$
     - Generate a uniformly distributed random value: $s_i \in [0, 1]$.
     - Generate $y_i^m = \begin{cases} 0, & \text{if } s_i \leq p_i, \\ 1, & \text{otherwise.} \end{cases}$
   - Perform BM-HDD on Y^m to get X^m.
   - Compute the soft weight of $Y^H \oplus X^m$: $$W(Y^H \oplus X^m) = \sum_{i=1}^{n} |p_i - 0.5|(y_i^H \oplus x_i^m).$$
3) Select the decided word D such that: $$D = X^j, W(Y^H \oplus X^j) = \min_{m=1}^{\tau} W(Y^H \oplus X^m).$$

Step 1 saturates the most reliable input according to some threshold θ. It prevents the most reliable bits from being flipped. The parameter β is a positive constant that has to be optimized for the code. This factor is equivalent to the Noise Dependent Scaling (NDS) used in stochastic decoding of LDPC codes [3], which improves the decoding performance. This parameter reduces the average value of $|p_i - 0.5|$, and consequently the number of identical test patterns decreases. Furthermore, similar to the NDS, the soft input is independent from the noise power σ^2. Therefore, the noise power estimation is not required.

In step 2, τ test patterns are generated bit-wise in such a way that unreliable bits are more likely to be flipped. This probabilistic generation of test patterns is similar to stochastic computation used in LDPC decoding [4]. As opposed to the CA, the SCA does not limit the search to the subset defined by the least reliable bits. For instance, if the error pattern contains an error on the $(\lambda + 1)^{th}$ least reliable bit, the CA will not be able to correct it because the erroneous bit is out of the searched space. However, this particular bit may be flipped by the SCA and the correct codeword would be considered.
Furthermore, in the CA, in order to get close to ML-performance, the number of considered Least Reliable Bits (LRB) λ should be proportionnal to d and the LRB search complexity is $O(\lambda n)$. It means that the LRB search becomes rapidly complex for low rate codes (*i.e.*, large minimum distance). However, in the SCA, the LRB search is replaced by a stochastic generation of test patterns. This probabilistic process is a bit-wise operation and has a complexity $O(n)$. The SCA is then well adapted to large block length and low rate codes. In terms of hardware implementation, a high parallelism level can be reached by implementing n independent random generators. It is also possible to use several instanciations of the same SCA decoder (simply with different seeds) in order to reduce the latency. If we assume an infinite number of trials τ, $0 < p_i < 1$, and $\theta = 0.5$, then all the possible codewords will be generated which means that the SCA is an asymptotically ML-decoder.

III. Soft Decoding of RS Codes

An (n, k, d) RS code is defined over the Galois field $GF(2^M)$ with length $n = 2^M - 1$, dimension k and minimum distance d at a symbol level. Each symbol in $GF(2^M)$ can be mapped to a binary image according to a basis in $GF(2)$: $B = (b_1, b_2, ..., b_M)$. The resulting binary linear block code has a length $n_b = n \times M$, a dimension $k_b = k \times M$ and a binary minimum distance $d_b \geq d$.

Recently, several solutions have been proposed for soft decision decoding of RS codes. Some of these are based on polynomial interpolation [5], which provide good performance but require complex computation structures. Adaptive Belief Propagation (ABP) algorithm was proposed in [6], it provides even better performance but requires Gaussian eliminations on the parity check matrix which leads to a prohibitive complexity. In [7], it was shown that the association of ABP and the Koetter Vardy (KV) algorithm can achieve the ML lower bound for the RS(31,25). However, this solution also suffers from high complexity. In [8], a Stochastic Shifting based Iterative Decoding (SSID) provides reasonable coding gain but requires more than 10000 BP iterations for the RS(31,25) code and it only applies to cyclic codes. In [9] Gaussian noise is added to the received sequence in order to generate candidate codewords. After each iteration, the parameters of the Gaussian noise are updated according to the best candidate codeword, which requires a large amount of memory and complex computations. This method is called Stochastic Erasure-Only List Decoding (SEOLD). In [10], an Iterative Box and Match (IT-BMA) approach is used which provides good performance but requires complex computations. Finally, minimum distance based (Chase, GMD) algorithms are efficient solutions when a few test patterns have to be generated (*i.e.*, for high rate codes). However, they become complex when the minimum distance increases.

IV. Simulation Results

In this section, we simulate the performance of different RS codes on an AWGN with BPSK modulation scheme, such that soft inputs are computed bit-wise (see Section II.A.). We use the Berlekamp-Massey (BM) algorithm [11], [12] for HDD.
SCA performance is compared with CA and also with different reported soft decoding algorithms. We refer to Adaptive Belief Propagation algorithm [6] as ABP($A \times A'$) where A represents the number of BP iterations per outer round and A' refers to the number of outer rounds (Gaussian elimination on the parity check matrix). We refer to the Koetter-Vardy algorithm [5] as KV(μ) where μ represents the maximum multiplicity. SSID [8] with A BP-iterations per outer round and A' outer rounds (stochastic shift) is labeled SSID($A \times A'$). SEOLD [9] with A iterations is labeled SEOLD(A). SCA(τ) and CA(m) refer to SCA with τ trials and CA with $m = 2^\lambda$ test patterns, respectively.

We first present results for the RS(31,25) code which has a binary minimum distance $d_b = 7$. In order to have a fair comparison between the CA and the SCA, we set the number of trials of the SCA and the number of test patterns in the CA to the same value: $\tau = 2^\lambda = 1024$. As shown in Fig. 1, SCA provides a 0.15dB coding gain over CA, and approaches the

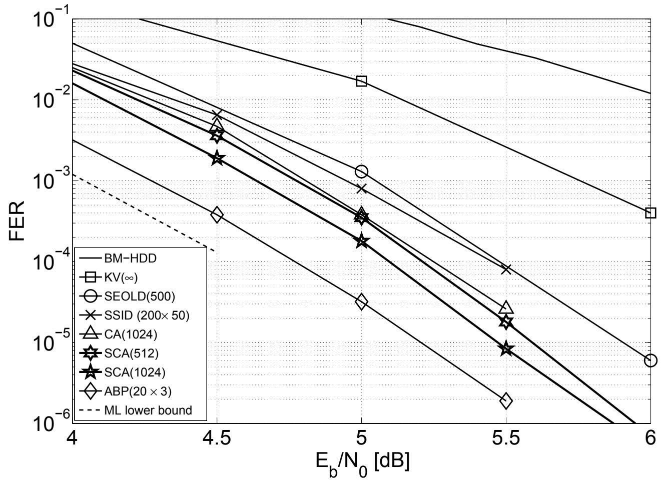

Fig. 1. RS(31,25) code on AWGN, $\beta = 6$, $\theta = 0.45$.

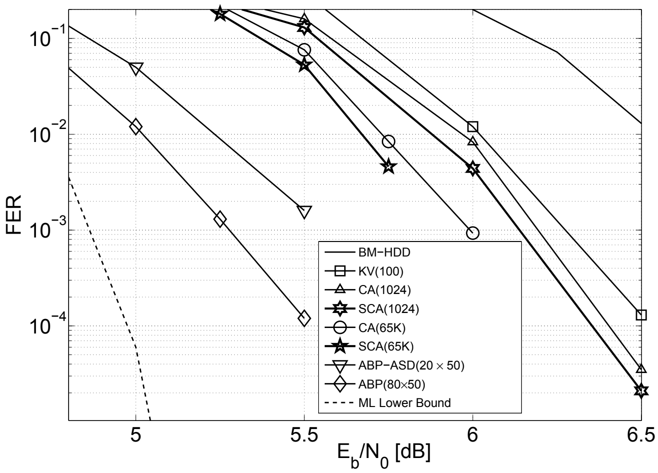

Fig. 2. RS(255,239) code on AWGN, $\beta = 22$, $\theta = 0.5$.

TABLE I
NUMBER OF TEST PATTERNS

Code	CA	SCA	SNR (dB) @ FER=10^{-4}
RS(31,25)	1024	402	5.2
RS(63,55)	1024	538	5.65
RS(255,239)	1024	549	6.4

ML lower bound within 0.5dB at a Frame Error Rate (FER) of 10^{-4}. It is also close to the ABP(20×3) performance which requires 60 BP iterations and 3 Gaussian eliminations on the parity check matrix. The SCA provides 0.4dB gain at FER= 10^{-4} compared to existing stochastic methods (SEOLD and SSID) that also have higher computational complexity. Performance of the widely used RS(255,239) code is considered in Fig. 2. SCA(1024) provides 0.15dB coding gain compared to KV(100). The performance can be traded off with complexity by increasing τ, the performance of SCA(65K) is 0.4dB away from IT-BMA [10]. It is also 0.6dB away from the ABP(80×50) which requires 50 Gaussian eliminations and 4K BP iterations. The ML lower bound in this figure is based on [13].

Table I compares the number of required test patterns in SCA with CA for different RS codes, at identical decoding performance. Results show that SCA reduces the number of test patterns by 46% to 60%.

V. CONCLUSION

In this letter, we proposed a modification of the Chase Algorithm in which a stochastic generation of test patterns is used. Simulations show that our modification provides a low cost solution for soft-decoding of RS codes. In the CA, the least reliable bits selection process requires at least $(n \times \lambda)$ comparisons, which becomes complex for large codes. The test pattern generation can be seen as an exhaustive search in a predetermined subset of the code C. On the contrary, in the SCA no least reliable bits selection is required. It is replaced by a simple stochastic generator which consists of a random number generator and a comparator. The complexity of SCA is tractable even for large-size and low-rate codes. It is possible to further improve the performance of SCA by preventing the generation of identical test patterns but it would increase the complexity. Just like the CA, our modification can be applied to decoding of any linear block code that has an HDD algorithm. It could, for example, be used for soft decoding of BCH codes. It is also possible to add a soft output computation stage in order to perform iterative decoding of product codes [14].

REFERENCES

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

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

[3] S. Sharifi Tehrani, W. Gross, and S. Mannor, "Stochastic decoding of LDPC codes," *IEEE Commun. Lett.*, vol. 10, pp. 716–718, Oct. 2006.

[4] V. Gaudet and A. Rapley, "Iterative decoding using stochastic computation," *Electron. Lett.*, vol. 39, no. 3, pp. 299–301, Feb. 2003.

[5] 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.

[6] 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.

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

[8] J. Jiang and K. Narayanan, "Iterative soft decoding of Reed-Solomon codes," *IEEE Commun. Lett.*, vol. 8, no. 4, pp. 244–246, Apr. 2004.

[9] C.-M. Lee and Y. Su, "Stochastic erasure-only list decoding algorithms for Reed-Solomon codes," *IEEE Signal Process. Lett.*, vol. 16, no. 8, pp. 691–694, Aug. 2009.

[10] M. Fossorier and A. Valembois, "Reliability-based decoding of Reed-Solomon codes using their binary image," *IEEE Commun. Lett.*, vol. 8, no. 7, pp. 452–454, July 2004.

[11] E. R. Berlekamp, *Algebraic Coding Theory*. Aegean, 1984.

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

[13] M. El-Khamy and R. McEliece, "Bounds on the minimum distance and the maximum likelihood performance of Reed-Solomon codes," in *Proc. 42nd Allerton Conf.*, 2004, pp. 290–299.

[14] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq, "Near optimum decoding of product codes," in *Proc. IEEE GLOBECOM*, 1994, pp. 339–343.