Correspondence

Weighted Nonbinary Repeat–Accumulate Codes

Kyeongcheol Yang, *Member, IEEE*

***Abstract*—Repeat–accumulate (RA) codes are random-like codes having remarkably good performance over an additive white Gaussian noise (AWGN) channel, like turbo and low-density parity-check (LDPC) codes. In this correspondence, we introduce an ensemble of random codes called "weighted nonbinary repeat–accumulate (WNRA) codes" whose encoder consists of a nonbinary repeater, a weighter, a pseudorandom symbol interleaver, and an accumulator over a finite field GF (q). They can be decoded in a simple way by applying the sum–product algorithm to their factor graphs over GF (q). Simulation results show that WNRA codes with proper weighting values over GF (4) or GF (8) are superior to binary RA codes on AWGN channels.**

***Index Terms*—Factor graph, iterative decoding, repeat–accumulate (RA) codes, sum–product algorithm.**

I. Introduction

Since Shannon's work in 1948, coding theorists have attempted to design error-correcting codes having performance close to the Shannon limit. Recently, spectacular progress has been made on capacity-approaching coding schemes by developing codes that are naturally defined on graphs.

In 1962, Gallager introduced low-density parity-check (LDPC) codes and their iterative decoding (probabilistic decoding) [6]. With a notable exception of Tanner [13], iterative coding systems were all but forgotten until the introduction of turbo codes by Berrou *et al.* [1]. LDPC codes were rediscovered recently by MacKay and Neal [11], and their experiments with LDPC codes showed that they exhibit very low bit-error rates (BER) at low signal-to-noise ratios (SNR) like turbo codes.

Remarkably good performances have also been achieved with repeat–accumulate (RA) and irregular repeat–accumulate (IRA) codes using the sum–product algorithm [5], [8]. RA (or IRA) codes are composed of a simple binary regular (or irregular) repetition code, a pseudorandom interleaver, and a trivial rate-1 two-state "accumulate" code. Binary RA codes can be extended to codes over GF (q) in a natural way, where GF (q) is the finite field of q elements.

In this correspondence, we introduce an ensemble of codes called "weighted nonbinary repeat–accumulate (WNRA) codes." They can be easily decoded by applying the sum–product algorithm to their factor graph. In order to investigate their performance under binary decoding, we first transform their factor graph over GF (q) into a "binary" factor graph and derive the degree distributions under some assumption. Simulation results show that WNRA codes with proper weighting values over GF (4) or GF (8) have better performance than binary RA codes over additive white Gaussian noise (AWGN) channels.

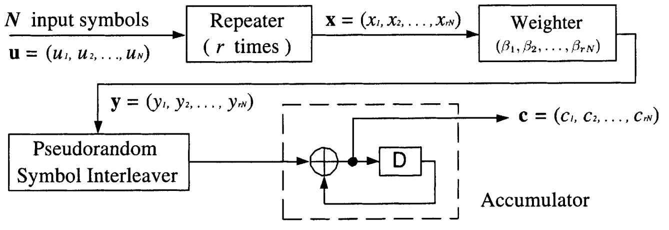

Fig. 1. Encoder of a rate-$1/r$ WNRA code over GF(q). Here $\oplus$ denotes addition over GF (q).

The outline of the correspondence is as follows. In Section II, we describe the structure and factor graph of WNRA codes. In Section III, we discuss decoding of WNRA codes, based on the sum–product algorithm. In Section IV, we transform the factor graph of WNRA codes into a binary form and derive their degree distributions. Some numerical results are given in Section V. Finally, we give some concluding remarks in Section VI.

II. The Structure of WNRA Codes

The encoder of WNRA codes can be divided into four parts: 1) a repeater over GF (q) as an outer code; 2) a random weighter; 3) a pseudorandom symbol interleaver; 4) an accumulator over GF (q) as an inner code. In fact, WNRA codes are similar to nonbinary RA codes except for the random weighter. The WNRA encoder is depicted in Fig. 1. As a first step, input data are divided into the frames of N symbols and are encoded frame by frame, where a symbol corresponds to an element of GF (q).

Let $\boldsymbol{u} = (u_1, u_2, \ldots, u_N)$ be a frame of N symbols to be encoded. The frame $\boldsymbol{u}$ is an input to the repeater over GF (q), whose output is denoted by $\boldsymbol{x} = (x_1, x_2, \ldots, x_{rN})$, where r is the number of repetitions and $x_{ir+j} = u_{i+1}$ for any integer i and j, $0 \leq i \leq N-1, 1 \leq j \leq r$. In general, any nonzero element in GF (q) can be chosen as the weighting value for a specific input. Let $\beta_1, \beta_2, \ldots, \beta_{rN}$ be the rN weighting values of the weighter. Then the output $\boldsymbol{y} = (y_1, y_2, \ldots, y_{rN})$ of the weighter corresponding to the input $\boldsymbol{x}$ is given by $y_i = \beta_i x_i$ for any $i = 1, 2, \ldots, rN$. The symbols in the frame $\boldsymbol{y}$ are next interleaved by a pseudorandom symbol interleaver and its corresponding output is denoted by $\boldsymbol{z} = (z_1, z_2, \ldots, z_{rN})$. Finally, $\boldsymbol{z}$ is fed into the accumulator, giving the output $\boldsymbol{c} = (c_1, c_2, \ldots, c_{rN})$ with

$$c_i = z_1 + z_2 + \cdots + z_i, \qquad i = 1, 2, \ldots, rN$$

where the addition is performed in GF (q). The vector $\boldsymbol{c}$ is the codeword of the WNRA code corresponding to the input $\boldsymbol{u}$ and the overall rate of the WNRA code is $1/r$.

Factor graphs are a basic tool for decoding of turbo, LDPC, and RA codes, based on the sum–product algorithm [10]. Similarly, they are also an efficient tool in decoding and analyzing WNRA codes. As an example, the factor graph of a rate-$1/3$ WNRA code over GF(4) is shown in Fig. 2, where α is a primitive element of GF(4) and $\alpha, \alpha^2, \alpha^3 = 1$ are chosen repeatedly as the weighting values. In Fig. 2, the empty circles ($\circ$), the black circles ($\bullet$), and the check boxes (⊞) represent the information nodes, the parity nodes, and the check nodes, respectively. The performance of WNRA codes is heavily dependent on the choice of weighting values, which will be discussed

Manuscript received February 1, 2002; revised September 9, 2003. This work was supported in part by the Brain Korea 21 Project, the University IT Research Center Project, and the Com2 Mac-KOSEF, and by the Norwegian Research Council.

The author is with the Department of Electronic and Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Kyungbuk 790-784, Korea (e-mail: kcyang@postech.ac.kr).

Communicated by R. Urbanke, Associate Editor for Coding Techniques.
Digital Object Identifier 10.1109/TIT.2004.825255

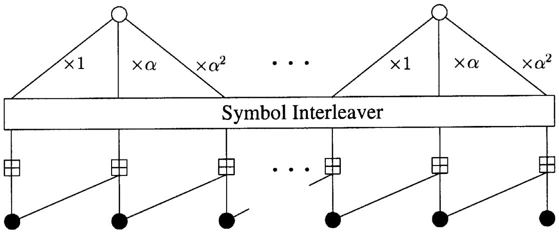

Fig. 2. Factor graph of WNRA codes over GF(4).

in Section V. Therefore, it is critical to choose the weighting values properly in order to get good WNRA codes.

III. Decoding of WNRA Codes

The decoding of WNRA codes is similar to that of the binary RA codes, based on their factor graph and the sum–product algorithm. In order to apply the sum–product algorithm to the factor graph of WNRA codes, it may be necessary to find nonbinary updating rules for the variable and check nodes. Binary updating rules for the variable and check nodes are well described in [10]. They can be easily extended to nonbinary updating rules by using the message-passing algorithm.

Let α be a primitive element of GF(q) and $M = q - 1$. For simple notation, let $[M] \triangleq \{1, 2, \ldots, M\}$. For each variable v, we associate it with the probability vector $\boldsymbol{P} = (p_0, p_1, p_2, \ldots, p_M)$ as a message for the sum–product algorithm, where p_0 is the probability that v takes the symbol 0 and p_i is the probability that v takes the symbol α^i for $i \in [M]$. Note that the log-likelihood ratios may be employed as a message instead of the probability vectors.

According to the generic updating rules in [10], when two probability vectors $\boldsymbol{P} = (p_0, p_1, p_2, \ldots, p_M)$ and $\boldsymbol{Q} = (q_0, q_1, q_2, \ldots, q_M)$ arrive at a variable node of degree three, the resulting output message is the probability vector given by

$$\mathrm{VAR}\,(\boldsymbol{P}, \boldsymbol{Q}) = \delta(p_0 q_0, p_1 q_1, \ldots, p_M q_M)$$

where δ is the normalizing factor given by $\delta = 1/\sum_{i=0}^{M} p_i q_i$. Similarly, at a check node of degree three, the output message is the probability vector $\boldsymbol{S} = (s_0, s_1, \ldots, s_M)$ given by

$$\boldsymbol{S} = \mathrm{CHK}\,(\boldsymbol{P}, \boldsymbol{Q})$$

where

$$s_i = \begin{cases} \sum\limits_{j=0}^{M} p_j q_j, & \text{if } i = 0 \\ p_0 q_i + \sum\limits_{j=1}^{M} p_j q_{i \boxplus j}, & \text{if } i \in [M]. \end{cases}$$

Here, $i \boxplus j \in [M]$ denotes the exponent of $\alpha^i + \alpha^j$ with respect to the base α.

The weighter by α^j transforms the input probability vector $\boldsymbol{P}$ into the output probability vector $\boldsymbol{Q}$, where $q_0 = p_0$, $q_i = p_{((M-j+i))}$ for $i \in [M]$, and $((l))$ denotes $l \bmod M$ in $[M]$. In other words, the probability vector for the output of the weighter by α^j is simply a cyclic shift of the input probability vector to the left by $M - j$, except the probability for the symbol 0. For example, if α is chosen as the weighting value in the weighter for a message with probability vector (p_0, p_1, p_2, p_3), then the probability vector for the corresponding output message of the weighter is (p_0, p_3, p_1, p_2). This fact implies that the multiplication by the weighting values in WNRA codes does not increase the complexity of decoding, compared with nonbinary RA codes.

These rules can be easily extended to the more general cases where variable or check nodes have degree larger than three by transforming the variable and check nodes of higher degree into multiple nodes of degree three [10]. For example, the VAR and CHK functions may be extended to more than two arguments via the relations

$$\begin{aligned} \mathrm{VAR}(\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_n) &= \mathrm{VAR}(\boldsymbol{P}_1, \mathrm{VAR}(\boldsymbol{P}_2, \ldots, \boldsymbol{P}_n)) \\ \mathrm{CHK}(\boldsymbol{P}_1, \boldsymbol{P}_2, \ldots, \boldsymbol{P}_n) &= \mathrm{CHK}(\boldsymbol{P}_1, \mathrm{CHK}(\boldsymbol{P}_2, \ldots, \boldsymbol{P}_n)) \end{aligned}$$

where $\boldsymbol{P}_i$ is a probability vector.

Given the rN probability vectors $\boldsymbol{R}(1), \boldsymbol{R}(2), \ldots, \boldsymbol{R}(rN)$ at the initial stage, WNRA codes over GF(q) can be decoded iteratively by applying the above updating rules to their factor graph. For an efficient decoding, the same message scheduling as that of binary RA codes described in [9] may be applied.

IV. Binary Factor Graph of WNRA Codes

It may be very interesting to investigate the performance of WNRA codes over GF(2^m) when binary decoding is applied to them. For this purpose, we transform the factor graph of WNRA codes into a structure of binary form, based on the binary representation of GF(2^m). This will be referred to as the "binary" factor graph of WNRA codes. Similar transformation was dealt with for decoding of nonbinary LDPC codes in [3], [4].

Let α be a primitive element of GF(2^m) satisfying $p(\alpha) = 0$, where

$$p(x) = x^m + p_{m-1}x^{m-1} + \cdots + p_0$$

is a primitive polynomial of degree m over GF(2). Every element a in GF(2^m) is associated to the vector $\boldsymbol{a} = (a_1, a_2, \ldots, a_m)^T$ by the relation $a = a_1 + a_2\alpha + \cdots + a_m\alpha^{m-1}$, where $a_i \in$ GF(2). Let b be the output of the weighter with α as a weighting value. Then $b = \alpha a$ and the corresponding vector $\boldsymbol{b} = (b_1, b_2, \ldots, b_m)^T$ is given by $\boldsymbol{b} = M\boldsymbol{a}$, where M is the companion matrix of $p(x)$ given by

$$M = \begin{bmatrix} 0 & 0 & 0 & & 0 & p_0 \\ 1 & 0 & 0 & \cdots & 0 & p_1 \\ 0 & 1 & 0 & & 0 & p_2 \\ & \vdots & & & \vdots & \\ 0 & 0 & 0 & \cdots & 1 & p_{m-1} \end{bmatrix}.$$

Applying it repeatedly, $\alpha^i a$ is associated to $M^i\boldsymbol{a}$. This is exactly the input–output relation of the weighter in the binary factor graph of WNRA codes.

As an example, Fig. 3 shows the transformation of the weighter over GF(4) into a binary structure according to the weighting values. Assuming that $1, \alpha, \alpha^2$ are chosen repeatedly as the weighting values, the binary factor graph of a rate-$1/3$ WNRA code over GF(4) is shown in Fig. 4. Here, the identity interleaver is assumed to be chosen as the symbol interleaver only for simple analysis.

In general, the binary factor graph of WNRA codes has an irregular structure in the information and check nodes, even though the factor graph of WNRA codes has a regular structure. Let $\lambda(x)$ be the generating function of the degree distribution for the information nodes, given by

$$\lambda(x) \triangleq \sum_{i=2}^{d_\lambda} \lambda_i x^{i-1}$$

where λ_i is the fraction of edges between information and check nodes that are adjacent to the information nodes of degree i and d_λ is the maximum degree such that $\lambda_i \neq 0$. Similarly, let

$$\rho(x) \triangleq \sum_{i=2}^{d_\rho} \rho_i x^{i-1} \quad \text{and} \quad \tau(x) \triangleq \sum_{i=2}^{d_\tau} \tau_i x^{i-1}$$

be the generating functions of the degree distributions for the check nodes, where ρ_i is the fraction of edges between information and

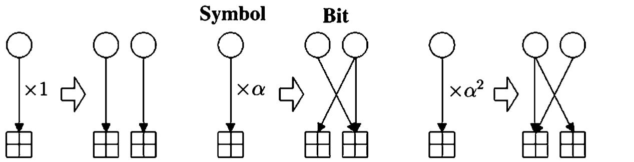

Fig. 3. Transformation of the weighter over GF(4) into a binary structure according to the weighting values over GF(4).

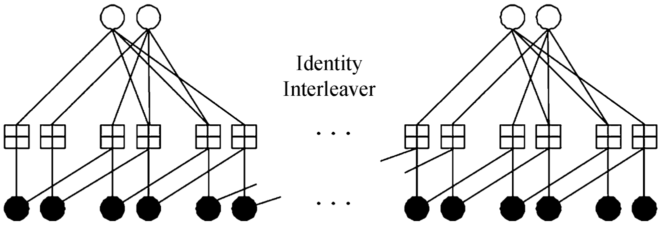

Fig. 4. Binary factor graph of a rate-$1/3$ WNRA code over GF(4) in Fig. 2.

check nodes that are adjacent to the check nodes of degree i, and τ_i is thefraction of edges between parity and check nodes that are adjacent to the check nodes of degree i, respectively.

Assume that α^i, $i = 1, 2, \ldots, rN$, are chosen repeatedly as the weighting values for WNRA codes. When the information frame length N is sufficiently large (or approaches to the infinity), each nonzero element of GF(2^m) is chosen equally likely as a weighting value. The r elements $\alpha^{jr+1}, \alpha^{jr+2}, \ldots, \alpha^{jr+r}$ for each integer j appear in the same information node of the factor graph of WNRA codes. When these elements are regarded as a weighting block for the jth information node, the sequence of the weighting blocks has a period of $\frac{2^m-1}{\gcd(2^m-1,r)}$. Therefore, every nonzero element of GF(2^m) appears $\frac{r}{\gcd(2^m-1,r)}$ times during the period.

Theorem 1: Assume that α^i, $i = 1, 2, \ldots, rN$, are chosen as the weighting values for a rate-$1/r$ WNRA code over GF(2^m) and the information frame length N is sufficiently large (or approaches to the infinity). Then the corresponding binary factor graph has

$$\rho(x) = \frac{1}{m \cdot 2^{m-1}} \sum_{i=1}^{m} i \binom{m}{i} x^{i+1}$$

and

$$\tau(x) = \frac{1}{2^m - 1} \sum_{i=1}^{m} \binom{m}{i} x^{i+1}.$$

In particular, $\rho(x)$ and $\tau(x)$ are independent of r.

Proof: From the factor graph of WNRA codes and the given assumption, it suffices to compute the degree distributions $\rho(x)$ and $\tau(x)$ in the binary structures of any $2^m - 1$ check nodes with weighting values $\alpha, \alpha^2, \ldots, \alpha^{2^m-1}$. Furthermore, it suffices to consider the first binary check nodes in the binary factor graph of the above check nodes, since every nonzero element in GF(2^m) appears equally likely. The number of the first binary check nodes of degree $i+2$ (i.e., i branches are adjacent to the information nodes and two branches are connected to the parity nodes) is $\binom{m}{i}$. Thus, the number of edges between information and check nodes that are adjacent to the first binary check nodes of degree $i+2$ is $i\binom{m}{i}$. Similarly, the number of edges between parity and check nodes that are adjacent to the first binary check nodes of degree $i+2$ is $2\binom{m}{i}$. Therefore, $\rho(x)$ and $\tau(x)$ are given by

$$\rho(x) = \frac{1}{N_\rho} \sum_{i=1}^{m} i \binom{m}{i} x^{i+1}$$

and

$$\tau(x) = \frac{1}{N_\tau} \sum_{i=1}^{m} \binom{m}{i} x^{i+1}$$

where N_ρ and N_τ are the normalizing constants given by $N_\rho = m \cdot 2^{m-1}$ and $N_\tau = 2^m - 1$, respectively. □

Compared with $\rho(x)$ and $\tau(x)$, the computation of $\lambda(x)$ has more complicated situations, since $\lambda(x)$ depends on the number r of repetition times. In the cases of GF(4) and GF(8), $\lambda(x)$ is easily computed depending on r and summarized in the following two theorems.

Theorem 2: Assume that α^i, $i = 1, 2, \ldots, rN$, are chosen as the weighting values for a rate-$1/r$ WNRA code over GF(4) and the information frame length N is sufficiently large (or approaches infinity). Then the corresponding binary factor graph has

$$\lambda(x) = \begin{cases} \frac{4k+1}{2(3k+1)} x^{4k} + \frac{2k+1}{2(3k+1)} x^{4k+1}, & \text{if } r = 3k+1 \\ \frac{2k+1}{2(3k+2)} x^{4k+1} + \frac{4k+3}{2(3k+2)} x^{4k+2}, & \text{if } r = 3k+2 \\ x^{4k+3}, & \text{if } r = 3k+3 \end{cases}$$

for any nonnegative integer k.

Proof: From the factor graph of WNRA codes and the given assumption, it suffices to compute $\lambda(x)$ in the binary structures of any three consecutive information nodes, where $\alpha, \alpha^2, \alpha^3$ appear exactly r times as the weighting values. Therefore, it suffices to consider the first binary information nodes in the binary factor graph of the above information nodes. The degree sequence of the first binary information nodes in the binary factor graph can be computed from the basis integer sequence 121 of period 3, where the ith value is exactly the number of edges at the check node corresponding to the first component of α^i, as shown in Fig. 4. Note that the degree sequence of the first binary information nodes in the binary factor graph can be computed from a cyclic shift of the sequence 121. Combining r consecutive numbers in the basis sequence results in the sequence of degrees in the first information nodes in the binary factor graph. Based on these results, $\lambda(x)$ can be easily computed. □

Theorem 3: Assume that α^i, $i = 1, 2, \ldots, rN$, are chosen as the weighting values for a rate-$1/r$ WNRA code over GF(8) and the information frame length N is sufficiently large (or approaches infinity). Then the corresponding binary factor graph has the following values, shown at the bottom of the page, for any nonnegative integer k.

$$\lambda(x) = \begin{cases} \frac{12k+1}{4(7k+1)} x^{12k} + \frac{6k+1}{2(7k+1)} x^{12k+1} + \frac{4k+1}{4(7k+1)} x^{12k+2}, & \text{if } r = 7k+1 \\ \frac{6k+1}{3(7k+2)} x^{12k+1} + \frac{4k+1}{2(7k+2)} x^{12k+2} + \frac{3k+1}{3(7k+2)} x^{12k+3} + \frac{12k+5}{6(7k+2)} x^{12k+4}, & \text{if } r = 7k+2 \\ \frac{4k+1}{4(7k+3)} x^{12k+2} + \frac{2(3k+1)}{3(7k+3)} x^{12k+3} + \frac{12k+5}{12(7k+3)} x^{12k+4} + \frac{2k+1}{2(7k+3)} x^{12k+5} + \frac{12k+7}{6(7k+3)} x^{12k+6}, & \text{if } r = 7k+3 \\ \frac{12k+5}{6(7k+4)} x^{12k+4} + \frac{2k+1}{2(7k+4)} x^{12k+5} + \frac{12k+7}{12(7k+4)} x^{12k+6} + \frac{2(3k+2)}{3(7k+4)} x^{12k+7} + \frac{4k+3}{4(7k+4)} x^{12k+8}, & \text{if } r = 7k+4 \\ \frac{12k+7}{6(7k+5)} x^{12k+6} + \frac{3k+2}{3(7k+5)} x^{12k+7} + \frac{4k+3}{2(7k+5)} x^{12k+8} + \frac{6k+5}{3(7k+5)} x^{12k+9}, & \text{if } r = 7k+5 \\ \frac{4k+3}{4(7k+6)} x^{12k+8} + \frac{6k+5}{2(7k+6)} x^{12k+9} + \frac{12k+11}{4(7k+6)} x^{12k+10}, & \text{if } r = 7k+6 \\ x^{12k+11}, & \text{if } r = 7k+7 \end{cases}$$

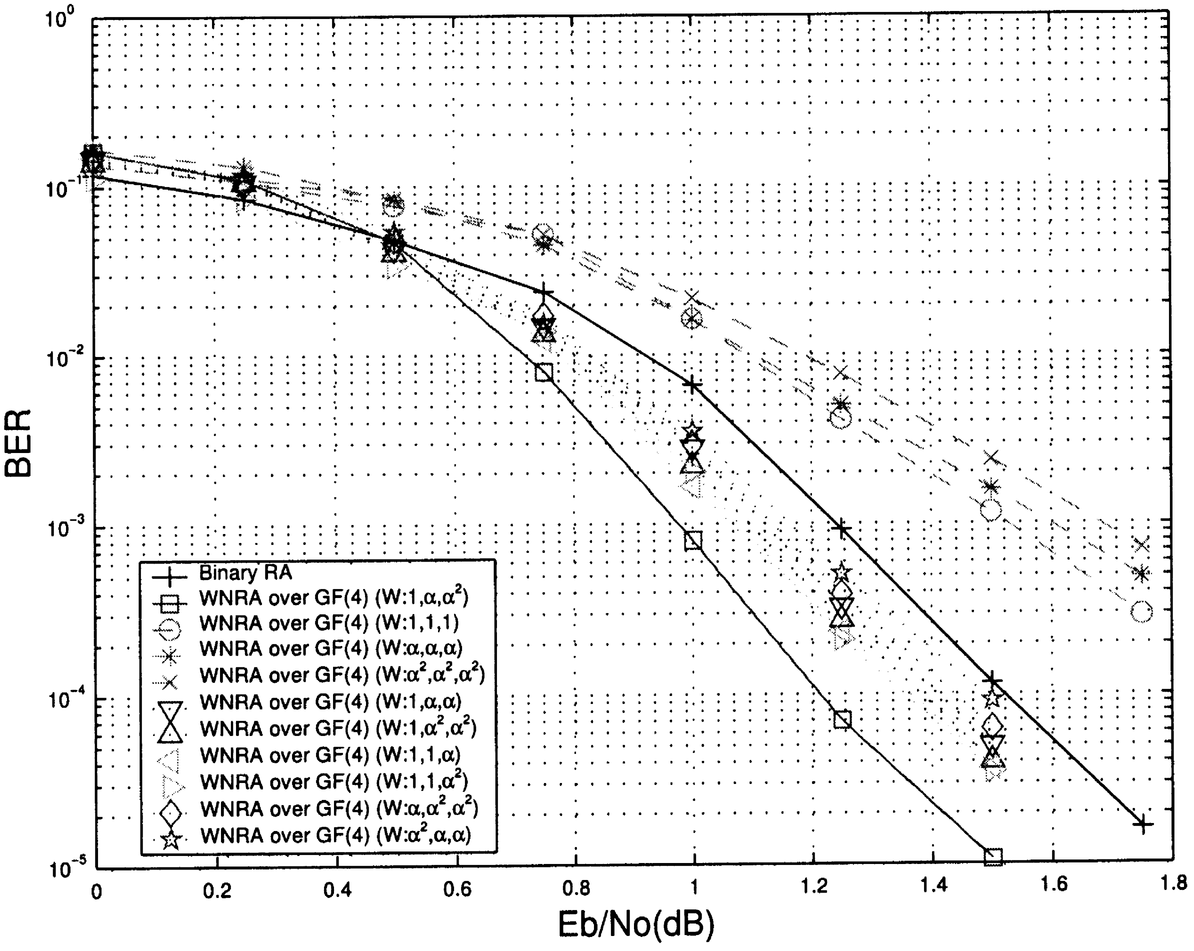

Fig. 5. Performance of a rate-$1/3$ binary RA code ("$+$" solid curve) and rate-$1/3$ WNRA codes over GF(4) with various weighting values.

Proof: In a similar way as in the Proof of Theorem 2, the results come from a basis sequence of degrees in the first binary information nodes when the weighting values are $\alpha, \alpha^2, \dots, \alpha^7$. It is easily checked that the basis sequence is given by 1122321 (or the reverse sequence depending on the choice of the primitive polynomials) in this case. □

In a straightforward way, similar ideas can be easily extended to the larger field GF(2^m) in order to get $\lambda(x)$ for the binary factor graph of a rate-$1/r$ WNRA code.

V. Numerical Results

Consider the situation where WNRA codes over GF(2^m) are used to communicate digital data over a memoryless binary-input continuous-output AWGN channel by representing each symbol of WNRA codewords into m bits and then transforming them into binary phase-shift keying (BPSK) modulated signals.

For a received real m-tuple vector $y = (y_1, y_2, \dots, y_m)$ corresponding to the transmitted symbol $c = (c_1, c_2, \dots, c_m)$, the probability vector $\boldsymbol{R} = (r_0, r_1, r_2, \dots, r_M)$ is computed by

$$r_i \triangleq \Pr(c = a|y) = \prod_{j=1}^{m} \Pr(c_j = a_j|y_j).$$

Here, c_j and a_j are the jth bits in the binary representation of the symbols c and a in GF(2^m), respectively.

Computer simulations are done to analyze the performance of WNRA codes over an AWGN channel. Numerical results show that the performance of WNRA codes is heavily dependent on the choice of weighting values. As an example, the performance of rate-$1/3$ WNRA codes over GF(4) for various weighting values is shown in Fig. 5 when the information symbol length is 512. The nonbinary decoding based on the sum–product algorithm is applied with 20 iterations. It is also compared with the performance of the rate-$1/3$ binary RA code with the same information length. Simulation results show that the rate-$1/3$ WNRA code over GF(4) with $1, \alpha, \alpha^2$ as the weighting values has the best performance. This fact implies that all nonzero symbols should appear equally likely as the weighting values in order to get better performance, as discussed in [3].

Fig. 6 shows the performance of rate-$1/3$ WNRA codes over GF(4) or GF(8) with $1, \alpha, \alpha^2 \dots$ as the weighting values for various information symbol lengths, when the number of iterations is 20. For comparisons, the performance of the rate-$1/3$ binary RA code for various information lengths is also shown in Fig. 6. Simulation results show that WNRA codes over GF(4) or GF(8) have better performance than binary RA codes of comparable complexity over an AWGN channel.

When the binary decoding is applied to the binary factor graph of WNRA codes in Section IV, they show worse performance than that in the case of nonbinary decoding. This fact implies that it seems more natural that nonbinary decoding should be applied to WNRA codes because they are nonbinary.

VI. Conclusion

We introduced an ensemble of random codes called WNRA codes over GF(q) which can be encoded in linear time. Under the iterative sum–product decoding, the multiplication of weighting values over

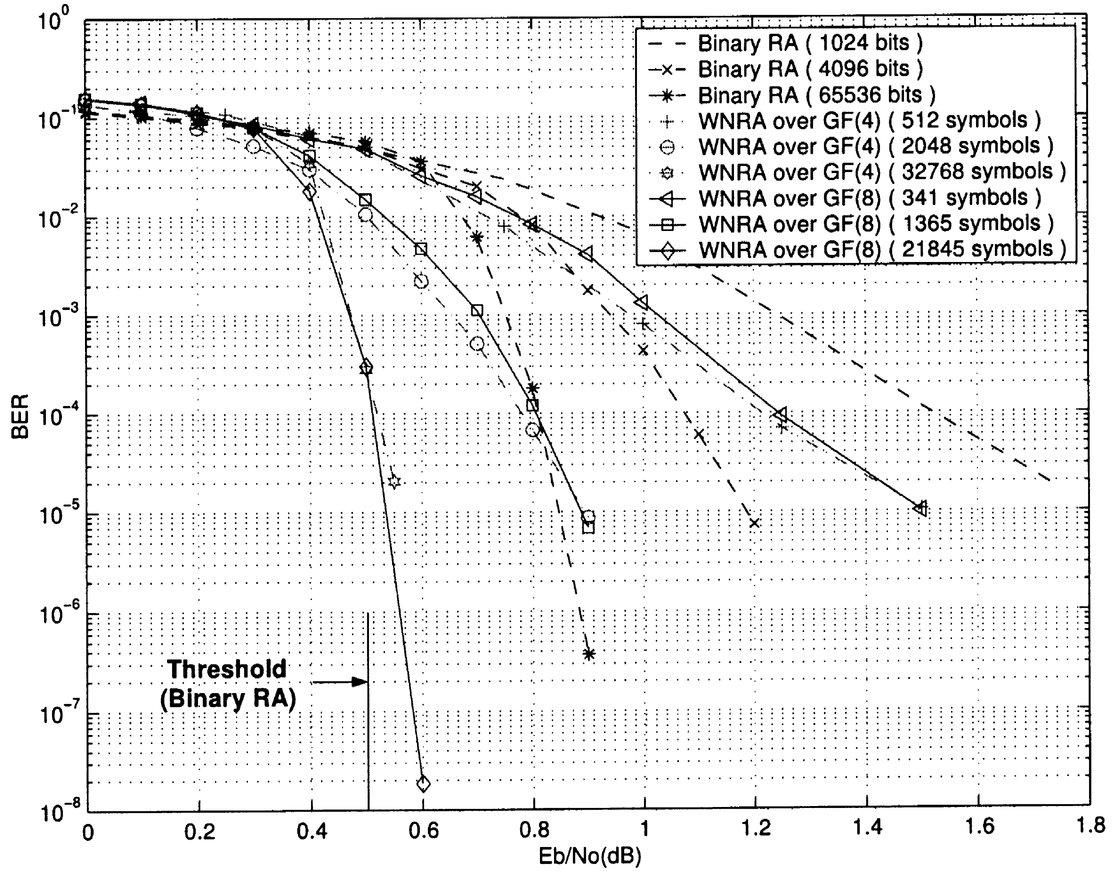

Fig. 6. Performance of a rate-$1/3$ binary RA code (dashed curves) and rate-$1/3$ WNRA codes over GF(4) or GF(8), where $1, \alpha, \alpha^2, \ldots$ are chosen as the weighting values.

GF(q) do not increase the decoding complexity, since the message probability vector from multiplication by a weighting value is just a shift of the given message probability vector except the probability for the symbol 0. One interesting point is that the binary factor graph of WNRA codes over GF(2^m) has an irregular structure in the information and check nodes, even though their factor graph has a regular structure. Numerical results show that all nonzero symbols should appear equally likely as the weighting values and nonbinary decoding should be applied to WNRA codes to get better performance. It may be very interesting to investigate whether WNRA codes achieve the Shannon limit or not.

Acknowledgment

The author would like to thank the reviewers for their valuable comments. He is also grateful to Seho Myung for his help with the preparation of some of the figures.

References

[1] C. Berrou, A. Glavieux, and P. Thitimajshima, “Near Shannon limit error-correcting coding and decoding: Turbo codes,” in *Proc. IEEE Int. Conf. Commununications*, Geneva, Switzerland, May 1993, pp. 1064–1070.

[2] S.-Y. Chung, T. J. Richardson, and R. L. Urbanke, “Analysis of sum-product decoding of low-density parity-check codes using a Gaussian approximation,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 657–670, Feb. 2001.

[3] M. C. Davey, “Error-correction using low-density parity-check codes,” Ph.D. dissertation, Univ. Cambridge, Cambridge, U.K, 1999.

[4] M. C. Davey and D. J. C. MacKay, “Low-density parity check codes over GF(q),” *IEEE Commun. Lett.*, vol. 2, pp. 165–167, June 1998.

[5] D. Divsalar, H. Jin, and R. J. McEliece, “Coding theorems for ’turbo-like’ codes,” in *Proc. 36th Annu. Allerton Conf. Communication, Control, and Computing*, Monticello, IL, Sept. 1998, pp. 201–210.

[6] R. G. Gallager, “Low-density parity-check codes,” *IRE Trans. Inform. Theory*, vol. IT-8, pp. 21–28, Jan. 1962.

[7] J. Hagenauer, E. Offer, and L. Papke, “Iterative decoding of binary block and convolutional codes,” *IEEE Trans. Inform. Theory*, vol. 42, pp. 429–445, Mar. 1996.

[8] H. Jin, A. Khandekar, and R. J. McEliece, “Irregular repeat-accumulate codes,” in *Proc. 2nd Int. Symp. Turbo Codes and Related Topics*, Sept. 2000, pp. 1–9.

[9] F. R. Kschischang, “Codes on graphs and iterative decoding,” Tutorials of IEEE Int. Symp. Information Theory, Sorrento, Italy, Jun. 2000. [Online]. Available: http://www.comm.utoronto.ca/frank/isit2000/.

[10] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 498–519, Feb. 2001.

[11] D. J. C. MacKay and R. M. Neal, “Near Shannon limit performance of low density parity check codes,” *Electron. Lett.*, vol. 32, pp. 1645–1646, Aug. 1996.

[12] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity check codes,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[13] R. M. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inform. Theory*, vol. IT-27, pp. 533–547, Sept. 1981.

[14] M. Xu, “Iterative decoding and graphical representations,” Ph.D. dissertation, Calif. Inst. Technol., Pasadeba, CA, 1999.