Nonlinear Product Codes and Their Low Complexity Iterative Decoding

Haesik Kim, Garik Markarian, and Valdemar C. da Rocha, Jr.

This paper proposes encoding and decoding for nonlinear product codes and investigates the performance of nonlinear product codes. The proposed nonlinear product codes are constructed as *N*-dimensional product codes where the constituent codes are nonlinear binary codes derived from the linear codes over higher order alphabets, for example, Preparata or Kerdock codes. The performance and the complexity of the proposed construction are evaluated using the well-known nonlinear Nordstrom-Robinson code, which is presented in the generalized array code format with a low complexity trellis. The proposed construction shows the additional coding gain, reduced error floor, and lower implementation complexity. The (64, 24, 12) nonlinear binary product code has an effective gain of about 2.5 dB and 1 dB gain at a BER of 10^{-6} when compared to the (64, 15, 16) linear product code and the (64, 24, 10) linear product code, respectively. The (256, 64, 36) nonlinear binary product code composed of two Nordstrom-Robinson codes has an effective gain of about 0.7 dB at a BER of 10^{-5} when compared to the (256, 64, 25) linear product code composed of two (16, 8, 5) quasi-cyclic codes.

Keywords: Turbo product code, nonlinear error control coding.

I. Introduction

Turbo product codes represent a class of codes that allow near Shannon limit performance with reasonable complexity [1]. Because of these attractive properties, turbo codes are incorporated in a number of international standards such as the IEEE 802.16 [2], the ETSI HiperMAN [3], and the HomePlug [4]. Recently, a new IEEE 802.16m standard was initiated [5], aiming at developing an enhanced version of the IEEE 802.16e [6], which could lead to the second phase of WiMAX systems. One of the major requirements for these systems is the transmission of short packets of data with increased reliability due to higher mobility specifications. Conventional error-correcting coding schemes, such as the low-density parity check (LDPC) codes [7], turbo convolutional codes (TCC) [8], and linear turbo product codes [1], are not well-suited for such application and result in higher error floor levels [9].

The aim of this paper is to develop a new class of nonlinear binary product codes (NBPCs), propose their low-complexity iterative decoding, which will allow an overall performance improvement with reduced implementation complexity and lower error floor for short codes, and investigate their performance. To achieve this goal, nonlinear codes are used as constituent codes, which are binary images of linear codes over higher order alphabets. These constituent codes can be the well-known Preparata codes [10], Kerdock codes [11], or any other code with similar properties. For a given code rate, these codes allow higher minimum Hamming distance, which eventually improves overall code performance. To reduce the overall decoding complexity, the constituent codes are represented as generalized array codes (GAC) with low-complexity trellises, which are used for iterative decoding of the constituent codes. Recently, a paper analysing theoretical

Manuscript received Nov. 4, 2009; revised Feb. 4, 2010; accepted Feb. 18, 2010.

Haesik Kim (phone: +358 20 722 2060, email: haesik.kim@vtt.fi) was with the Department of Communications Systems, Infolab21, Lancaster University, Lancaster, UK, and is now with the Department of Communication Platforms, VTT Technical Research Center of Finland, Kaitoväylä 1, Oulu, Finland.

Garik Markarian (email: g.markarian@lancaster.ac.uk) is with the Department of Communications Systems, Infolab21, Lancaster University, Lancaster, UK.

Valdemar C. da Rocha, Jr. (email: vcr@ufpe.br) is with the Department of Electronics and Systems, Federal University of Pernambuco, Recife, Brazil.

doi:10.4218/etrij.10.0109.0643

bounds of nonlinear product codes was published in [12]. One month earlier, nonlinear product code construction and iterative decoding were published in [13]. The theoretical results presented in [12] and construction and decoding presented in [13] are expanded by developing low complexity iterative decoding for the nonlinear binary product code and describing their practical implementation. To evaluate the performance of the proposed construction, the well known Nordstrom-Robinson (NR) code [14] is used as a constituent code. However, these results are applicable to other known nonlinear codes with similar properties.

This paper is organized as follows. In section II, the NR code construction using a GAC is proposed and the associated trellis diagram is presented. In section III, a family of nonlinear product codes is designed, the first instance of which is composed of an NR code and a single parity check (SPC) code, and the second instance of which is composed of two NR codes. In addition, a low-complexity decoding algorithm suitable for nonlinear product codes using Max-log MAP and majority logic decoding (MLD) is shown. It is shown in section IV by means of a computer simulation that the bit error rate performance of this nonlinear product code construction is better than that of a comparable linear product code. Finally, in section V, the simulation results using nonlinear product codes are discussed.

II. NR Code Construction and Trellis Diagram

1. NR Code Derived from the Linear (8, 4, 4) Code over Z4

In [15], it was shown that binary nonlinear codes such as the NR, Kerdock, and Preparata codes can be derived from linear codes over Z4, the ring of integers modulo-4. For example, it is possible to express the generator matrix for the systematic (8, 4, 4) NR parent code over Z4 as follows [16]:

$$G_{\mathrm{NR}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 3 & 1 & 2 & 1 \\ 0 & 1 & 0 & 0 & 2 & 1 & 1 & 3 \\ 0 & 0 & 1 & 0 & 1 & 1 & 3 & 2 \\ 0 & 0 & 0 & 1 & 3 & 2 & 3 & 3 \end{bmatrix}.$$

In order to construct the systematic NR code, the first codeword of the parent code over Z4 is obtained by multiplying a Z4 message vector by the parent code generator matrix G_{NR}.

The codewords of the NR code are obtained by mapping the parent code codewords over Z4 into binary format, using Gray mapping as illustrated in Table 1.

This is a well-known construction that allows low complexity encoding. However, the trellis complexity of this construction is too high and not suitable for practical implementation.

2. NR Code as a Generalized Array Code

In [17], the (8, 4, 4) NR code was expressed as a generalized array code construction, and its trellis decoder was proposed. In this section, the binary NR code is represented using a GAC construction. The NR code, C_{NR}, can be represented as $C_{\mathrm{NR}}=C_1+C_2$, where C_1 is the (16, 5, 8) first-order Reed-Muller (RM) code and C_2 is a nonlinear code with 8 codewords [18]. The C_1 (16, 5, 8) code can be designed as a GAC with the following structure [19]:

$$C_1 = \begin{bmatrix} x_1 & x_4+x_1 & x_5+x_1 & x_4+x_5+x_1 \\ x_2 & x_4+x_2 & x_5+x_2 & x_4+x_5+x_2 \\ x_3 & x_4+x_3 & x_5+x_3 & x_4+x_5+x_3 \\ p_1 & x_4+p_1 & x_5+p_1 & x_4+x_5+p_1 \end{bmatrix},$$

where x_1, x_2, x_3, x_4, and x_5 are information bits, "+" denotes a modulo-2 addition (XOR operation), and p_1 represents the parity check symbol computed as

$$p_1 = x_1 + x_2 + x_3 .$$

The codewords are transmitted row by row: C_1= [x_1, x_4+x_1, x_5+x_1, x_4+x_5+x_1, x_2, x_4+x_2, x_5+x_2, x_4+x_5+x_2, x_3, x_4+x_3, x_5+x_3, x_4+x_5+x_3, p_1, x_4+p_1, x_5+p_1, x_4+x_5+p_1].

The nonlinear code C_2 with 8 codewords should have the following two-dimensional structure:

$$C_2 = \begin{bmatrix} p_2 + x_6 \times x_7 + x_6 \times x_8 + x_7 \times x_8 & x_6 & x_7 & x_8 \\ p_2 & x_8 + x_7 \times x_8 & x_6 + x_6 \times x_8 & x_7 + x_6 \times x_7 \\ p_2 & x_7 + x_7 \times x_8 & x_8 + x_6 \times x_8 & x_6 + x_6 \times x_7 \\ x_6 \times x_7 + x_6 \times x_8 + x_7 \times x_8 & 0 & 0 & 0 \end{bmatrix},$$

where x_6, x_7, and x_8 are information bits, "×" denotes a modulo-2 multiplication (AND operation), and p_2 represents the parity check symbol computed as

$$p_2 = x_6 + x_7 + x_8 .$$

Therefore, the NR code is constructed as

Table 1. Gray mapping.

Gray mapping	
Z4	Binary
0	00
1	01
2	11
3	10

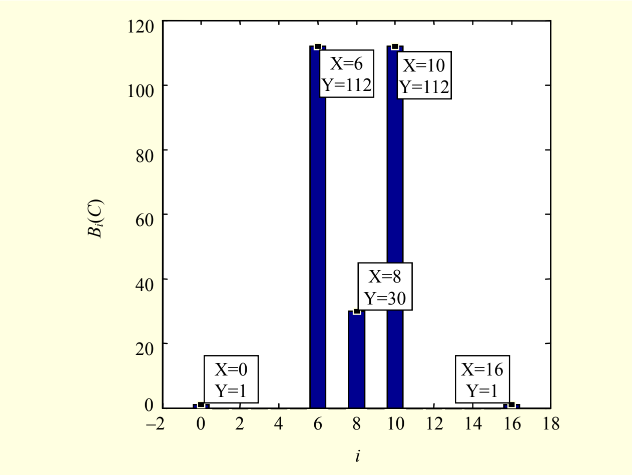

Fig. 1. Distance distribution of the NR code as GAC.

$$C_{\mathrm{NR}} = C_1 + C_2 = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 \\ c_5 & c_6 & c_7 & c_8 \\ c_9 & c_{10} & c_{11} & c_{12} \\ c_{13} & c_{14} & c_{15} & c_{16} \end{bmatrix}$$

$$= \begin{bmatrix} x_1 + p_2 + x_6 \times x_7 + x_6 \times x_8 + x_7 \times x_8 & x_4 + x_1 + x_6 & x_5 + x_1 + x_7 & x_4 + x_5 + x_1 + x_8 \\ x_2 + p_2 & x_4 + x_2 + x_8 + x_7 \times x_8 & x_5 + x_2 + x_6 + x_6 \times x_8 & x_4 + x_5 + x_2 + x_7 + x_6 \times x_7 \\ x_3 + p_2 & x_4 + x_3 + x_7 + x_7 \times x_8 & x_5 + x_3 + x_8 + x_6 \times x_8 & x_4 + x_5 + x_3 + x_6 + x_6 \times x_7 \\ p_1 + x_6 \times x_7 + x_6 \times x_8 + x_7 \times x_8 & x_4 + p_1 & x_5 + p_1 & x_4 + x_5 + p_1 \end{bmatrix},$$

and the transmission of a codeword is performed row by row.

When handling a nonlinear code, the code distance distribution is more significant than the corresponding weight distribution because the minimum distance can be obtained from the former, while the latter apparently cannot generally provide any relevant information [18]. The Hamming distance distribution of a code C of length n is the set $\{B_i(C) | 0 \le i \le n\}$, where

$$B_i(C) = \frac{1}{|C|} \sum_{c \in C} \left| \{ v \in C | d(v,c) = i \} \right|,$$

and $d(v,c)$ denotes the Hamming distance between c and v [18].

The distance distribution of the NR code is shown in Fig. 1 where x and y correspond to i and $B_i(C)$, respectively. The NR code has the property that its Hamming distance distribution coincides with its weight distribution [18].

3. Trellis Diagram of NR Code

Two trellis diagrams for the NR code are shown in Figs. 2(a) and 2(b), respectively. Figure 2(a) shows the trellis diagram of the systematic NR code derived from the code over Z4 with 2 sections, 256 states, and 8 coded bits for each branch [20]. Figure 2(b) shows the trellis diagram of the non-systematic NR code designed as a GAC with 4 sections, 64 states, and 4 coded bits for each branch. Furthermore, in Fig. 2(b), note that the trellis is a coset trellis [21], [22] composed of the first order RM code (16, 5, 8) and the codewords of the nonlinear code (16, 3, 6) as coset leaders.

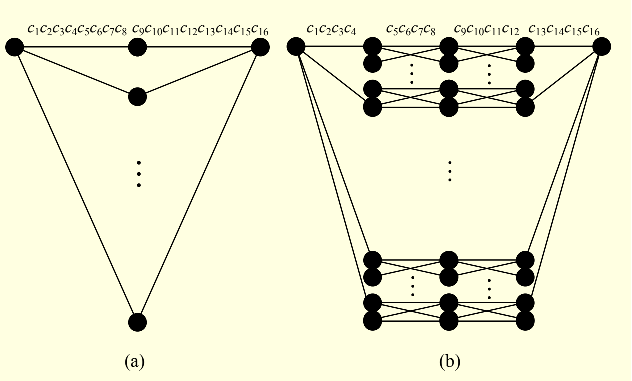
$c_1c_2c_3c_4c_5c_6c_7c_8$
$c_9c_{10}c_{11}c_{12}c_{13}c_{14}c_{15}c_{16}$
$c_1c_2c_3c_4$
$c_5c_6c_7c_8$
$c_9c_{10}c_{11}c_{12}$
$c_{13}c_{14}c_{15}c_{16}$
(a)
(b)

Fig. 2. Trellis diagram of NR code (a) derived from the code over Z4 and (b) NR code as GAC.

Table 2. State complexity for NR code trellises.

	Systematic NR code derived from a code over Z4	Non-systematic NR code as GAC
State-cardinality profile	{1, 256, 1}	{1, 64, 64, 64, 1}
Max number of states	256	64
Total number of states	258	194
Logarithmic state complexity profile	{0, 8, 0}	{0, 6, 6, 6, 0}
State complexity	8	6
Total span	8	18

4. Complexity Comparison of NR Code Trellises

The decoding complexity depends on the specific algorithm being used and can be often reduced by means of computational techniques, including gray coding, efficient processing of parallel transitions, and recursive decoding of sub-trellises [23]. In this section, the trellis complexity measure as defined in [18] is used. Tables 2 and 3 summarize the major parameters of the designed trellises and provide complexity estimation.

If a trellis is implemented in hardware, 2^s is the number of

Table 3. Edge (branch) complexity for NR code trellises.

	Systematic NR code derived from a code over Z4	Non-systematic NR code as GAC
Edge-cardinality profile	{2048, 2048}	{256, 1024, 1024, 256}
Max number of edges	2048	1024
Total number of edges	4096	2560
Logarithmic edge complexity profile	{11, 11}	{8, 10, 10, 8}
Edge complexity	11	10
Total edge span	22	36

Table 4. Expansion index and Viterbi decoding complexity for NR code trellises.

	Systematic NR code derived from a code over Z4	Non-systematic NR code as GAC
Expansion index $(E = \|E\| - \|V\| + 1)$	3841	2367
Viterbi decoding complexity $(D = 2\|E\| - \|V\| + 1)$	7937	4927

Table 5. State-complexity comparison of Nordstrom-Robinson code trellises.

Non-systematic NR code Trellis	{1 64 64 64 1}
Systematic NR code Trellis	{1 256 1}
NR Code Trellis by A. Lafourcade and A. Vardy [27]	{1 2 4 8 16 32 48 96 64 96 48 32 16 8 4 2 1}
NR Code Trellis by Forney [27]	{1 2 4 8 16 32 64 128 64 128 64 32 16 8 4 2 1}

add-compare-select (ACS) circuits that one has to put on a chip [24], where s denotes the state complexity. In [25], the total number of edges in the trellis is the most meaningful measure of Viterbi decoding complexity. Therefore, as illustrated in Table 4, the complexity of the developed non-systematic NR code is lower in comparison with that of the conventional systematic NR code derived from a code over Z4. In Table 4, $|E|$ and $|V|$ denote the total number of edges and the total number of states, respectively. Minimal trellises for nonlinear codes are generally computationally intractable, and the solutions leading to this minimization may result in improper or unobservable trellises [26].

State complexity comparison of NR code trellises is shown in Table 5. The maximum numbers of states for the non-systematic NR trellis, the systematic NR code trellis, the NR codes trellis by Lafourcade and Vardy, and the NR code by Forney are 64, 256, 96, and 128, respectively. The total numbers of states for them are 194, 258, 478, and 574, respectively. A trellis which has a minimum number of states (Vertices) is called a minimal trellis. Therefore, the non-systematic NR code trellis is close to the minimal trellis.

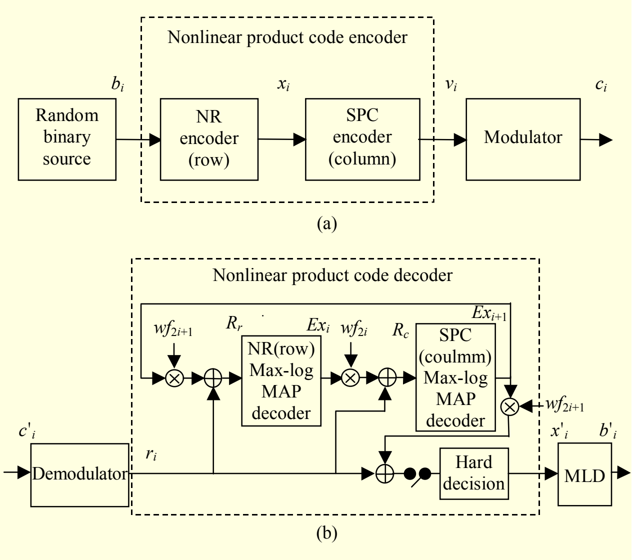

Fig. 3. (a) Transmitter and (b) receiver for the nonlinear product code composed of NR code as GAC and single parity check.

III. Nonlinear Product Code Design

1. Nonlinear Product Code Composed of NR Code and SPC Code

Firstly, a nonlinear product code composed by the (16, 8, 6) NR code as a row code and the (4, 3, 2) SPC code as a column code is designed for the sake of simplicity. In this nonlinear product code, the GAC representation of the NR code is used. Therefore, the first, second, and third rows are non-systematic NR codewords, and the last row is a parity (on columns) of the SPC code.

Figure 3(a) shows the block diagram of the transmitter which was used to simulate the designed nonlinear product code. In Fig. 3(a), a random binary vector b_i is encoded into the NR codeword x_i, represented as a GAC, and assigned as a row code. Then the x_i's are encoded into v_i using the SPC code as a column code. The corresponding codeword c_i belongs to a nonlinear product code with the parameters (64, 24, 12) = (16, 8, 6) × (4, 3, 2) the minimum distance of which is calculated in [12]. Figure 3(b) describes the receiver system.

Assuming that the channel is disturbed by the additive white Gaussian noise (AWGN), c_i' is the possibly corrupted received codeword which, after BPSK demodulation, is represented by r_i, that is, the soft value of the received sequence. At first the branch metric $b(s', s)$ is calculated in the row Max-log MAP decoder, where s' and s denote the previous state and the current state, respectively. This is expressed as

$$\log \gamma_{t_i}(b(s',s)) = \sum_{l=t_i}^{t_{i+1}-1} r_l \cdot (2v_l - 1),$$

where t_i denotes the time instant. Secondly, the forward state metric $\log \alpha_{t_i}(s)$ and the backward state metric $\log \beta_{t_i}(s)$ are calculated. They can be expressed as follows:

$$\log \alpha_{t_i}(s) = \max_{s' \in \Omega_{t_{i-1}}} \{\log \alpha_{t_{i-1}}(s') + \log \gamma_{t_{i-1}}(b(s',s))\},$$

$$\log \beta_{t_i}(s) = \max_{s' \in \Omega_{t_{i-1}}} \{\log \beta_{t_{i-1}}(s') + \log \gamma_{t_{i-1}}(b(s,s'))\},$$

where $\Omega_{t_{i-1}}$ denotes the set of the previous states s' connected with the current state s. Thirdly, the log likelihood ratio (LLR) $L_r(v_l)$ is calculated using the equation:

$$L_r(v_l) = \max_{\substack{(s',s)\\ v_l=1}} \left\{ \log \alpha_{t_i}(s') + \log \gamma_{t_i}(b(s',s)) + \log \beta_{t_{i+1}}(s) \right\}$$
$$- \max_{\substack{(s',s)\\ v_l=0}} \left\{ \log \alpha_{t_i}(s') + \log \gamma_{t_i}(b(s',s)) + \log \beta_{t_{i+1}}(s) \right\},$$

where $t_i \le l \le t_{i+1}$. For each coded bit v_l in a branch, $L_r(v_l)$ is calculated [28]. The output of the row Max-log MAP decoder, $L_r(v_l)$, is multiplied by wf_{1i}, which is the weighting factor to adjust the effect of the LLR value. The extrinsic information, Ex_i, is obtained by subtracting the soft input of Max-log MAP decoder from the soft output of Max-log MAP decoder:

$$Ex_i = wf_{1i} L_r(R_i) - R_i,$$

where R_i is the soft input of Max-log MAP decoder.

The input of the column Max-log MAP decoder is obtained by adding the received sequence r_i with the extrinsic information, that is,

$$R_c = r_i + wf_{2i} Ex_i,$$

where wf_{2i} is the weighting factor to adjust the extrinsic information. The experimental results indicate that the performance of turbo product codes is very sensitive to the weighting factors, and the evolution of the weighting factors is as

$$wf_{1i} = [0.2 \;\; 0.4 \;\; 0.6 \;\; 0.8 \;\; 1.0],$$

$$wf_{2i} = [0.4 \;\; 0.5 \;\; 0.6 \;\; 0.7 \;\; 0.8 \;\; 0.9 \;\; 1.0].$$

The column Max-log MAP decoder can be implemented in the same manner as the row Max-log MAP decoder.

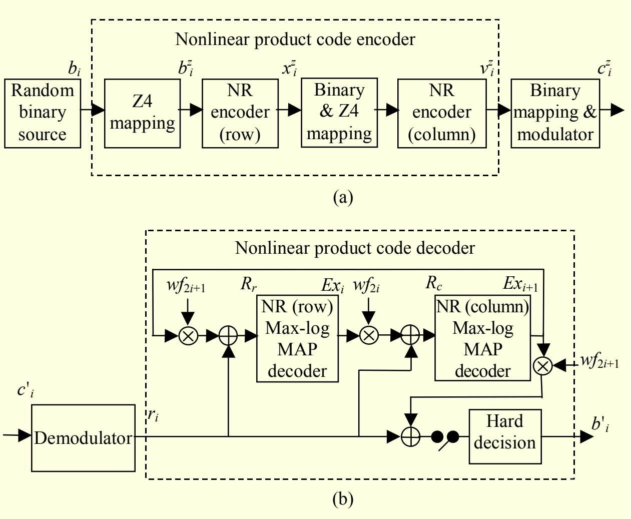

Fig. 4. (a) Transmitter and (b) receiver for nonlinear product code composed of two NR codes derived from the code over Z4.

The above decoding process is carried out for each row of code and then for each column of code and iterated several times to reduce the error probability. After the SISO decoding using the Max-log MAP algorithm is completed, the turbo product code decoder still needs a post-decoding algorithm because the output of the turbo product code decoder must be a codeword. The following describes how to employ a majority logic decoder at this stage. It is possible to carry out the MLD of the NR code, represented in GAC form. For example, the elements c_2, c_5, c_6, c_{10}, and c_{14} of an NR codeword in GAC form are chosen and then added:

$$\begin{aligned} &c_2 + c_5 + c_6 + c_{10} + c_{14} \\ &= (x_4 + x_1 + x_6) + (x_2 + p_2) + (x_4 + x_2 + x_8 + x_7 \times x_8) \\ &\quad + (x_4 + x_3 + x_7 + x_7 \times x_8) + (x_4 + p_1) = x_2. \end{aligned}$$

Likewise, several estimates of x_2 can be found using other combinations of digits, and x_2 can be extracted through MLD. In a similar way, the other information digits are extracted, and the decoded information bits can be obtained.

2. Nonlinear Product Code Composed of Two NR codes

One of two constituent codes must be a systematic code in order to construct the product code. Therefore, a systematic parent NR code derived from a code over Z4 is used. The generator matrix for the parent NR code derived from a code over Z4, which is a linear code, is in systematic form. Therefore, nonlinear product codes can be designed which are composed of two parent NR codes derived from a code over Z4. In Fig. 2(a), the trellis diagram of the parent NR code has two sections. The first section is the information part, and the second section is the parity part. Therefore, the Max-log MAP decoding algorithm can be applied with two sections.

In Fig. 4, c_i' is the received sequence which, after BPSK demodulation, produces r_i, the soft decision value of the received sequence. At first the branch metric is calculated as

$$B^{\text{info}}(s_i) = \sum_{l=1}^{8} r_l \cdot (2v_l - 1),$$

$$B^{\text{parity}}(s_i) = \sum_{l=9}^{16} r_l \cdot (2v_l - 1),$$

$$B(s_i) = B^{\text{info}}(s_i) + B^{\text{parity}}(s_i),$$

where B^{info} (s_i) and B^{parity} (s_i) are the first information section branch metric and the second parity section branch, respectively, and s_i denotes the state $0 \leq s_i \leq 255$. The forward state metric $\alpha_i(s_i)$ and the backward state metric $\beta_i(s_i)$ are calculated as follows:

$$\alpha_0(s_0) = 0$$
$$\alpha_1(s_i) = B^{\text{info}}(s_i)$$
$$\beta_0(s_0) = 0$$
$$\beta_1(s_i) = B^{\text{parity}}(s_i).$$

Finally, the LLR $L_{\text{r}}(v_l)$ is calculated as

$$L(v_l) = L^{+}(v_l) - L^{-}(v_l),$$

$$L^{+}(v_l)\begin{cases} \max\limits_{v_l=1, 0\leq s_i \leq 255} \{\alpha_0(s_0) + B^{\text{info}}(s_i) + \beta_1(s_i)\}, & 1 \leq v_l \leq 8, \\ \max\limits_{v_l=1, 0\leq s_i \leq 255} \{\alpha_1(s_i) + B^{\text{parity}}(s_i) + \beta_0(s_0)\}, & 9 \leq v_l \leq 16, \end{cases}$$

$$= \max_{v_l=1, 0\leq s_i \leq 255} \{B(s_i)\},$$

$$L^{-}(v_l) = \begin{cases} \max\limits_{v_l=0, 0\leq s_i \leq 255} \left\{\alpha_0(s_0) + B^{\text{info}}(s_i) + \beta_1(s_i)\right\}, & 1 \leq v_l \leq 8, \\ \max\limits_{v_l=0, 0\leq s_i \leq 255} \left\{\alpha_1(s_i) + B^{\text{parity}}(s_i) + \beta_0(s_0)\right\}, & 9 \leq v_l \leq 16, \end{cases}$$

$$= \max_{v_l=0, 0\leq s_i \leq 255} \left\{B(s_i)\right\}.$$

Therefore, the LLR can be simply expressed as

$$L(v_l) = \max_{v_l=1, 0\leq s_i \leq 255} \left\{B(s_i)\right\} - \max_{v_l=0, 0\leq s_i \leq 255} \left\{B(s_i)\right\}.$$

For each coded bit, $L(v_l)$ is calculated. The remaining decoding steps are the same as for the nonlinear product code composed of the NR code and the SPC code, except for the use of MLD. Although MLD is carried out for the NR code as a GAC caused by permuted codewords, MLD is not needed for the parent NR code derived from a code over Z4.

IV. Simulation

A computer simulation was carried out to compare the (64, 24, 12) nonlinear product code composed of the (16, 8, 6) NR code and the (4, 3, 2) SPC code, with the (64, 15, 16) linear product code composed of the (16, 5, 8) RM code and the (4, 3, 2) SPC code, and the (64, 24, 10) linear product code composed of the (16, 8, 5) quasi-cyclic (QC) code [29] and the (4, 3, 2) SPC code, under AWGN conditions.

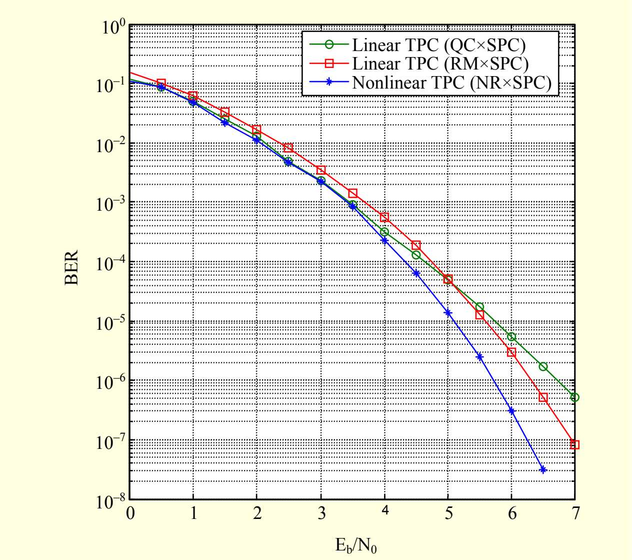

Fig. 5. Comparison of (64, 24, 12) nonlinear product code and (64, 15, 16) and (64, 24, 10) linear product codes after 3 iterations.

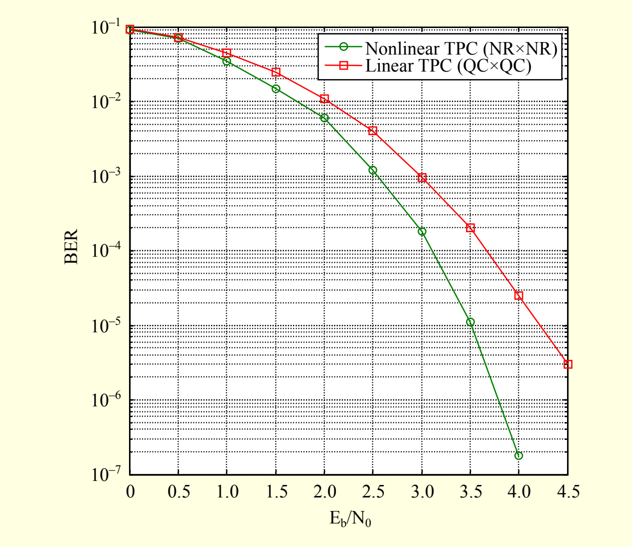

Fig. 6. Comparison of (256, 64, 36) nonlinear product code (NR×NR) and (256, 64, 25) linear product code (QC×QC) after 4 iterations.

Figure 5 shows the simulation results. At a bit error rate (BER) of 10^{-6}, the (64, 24, 12) nonlinear product code E_b/N_0 is 5.7 dB, and the (64, 15, 16) and (64, 24, 10) linear product codes for the corresponding E_b/N_0 are 6.2 dB and 6.7 dB, respectively. Besides, in order to compare the (64, 24, 12) nonlinear product code with the (64, 15, 16) linear product code at the same data rate, one has to consider the difference between the code rates [30]. Once this $10\log_{10}((24/64)/(15/64)) \approx 2$ dB is considered, it is evident that the nonlinear product code has about effective 2.5 dB gain (actual 0.5 dB gain) and 1 dB gain, respectively, and better low error rate performance.

The simulation results of the (256, 64, 36) nonlinear product code composed of two (16, 8, 6) NR codes and the (256, 64, 25) linear product code composed of two (16, 8, 5) QC codes are shown in Fig. 6. The nonlinear product code has a waterfall region between 3 dB and 4 dB but the waterfall region of the linear product code falls outside this range. At a BER of 10^{-5}, the nonlinear product code E_b/N_0 is 3.5 dB, and the linear product codes the corresponding E_b/N_0 is 4.2 dB. The nonlinear product code has about 0.7 dB gain due to better hamming distance.

V. Conclusion

A new technique for low-complexity encoding and decoding of nonlinear binary product codes has been proposed. The technique takes advantage of the well-known Preparata, Kerdock, Nordstrom-Robinson codes, or any other code with similar properties. In this paper, the attractiveness of these codes was enhanced by developing their low-complexity trellises and low-complexity decoders. To evaluate the performance of the proposed construction, the binary NR code construction using a GAC was proposed, and a number of low-complexity nonlinear product codes were developed. In addition, a low-complexity decoding algorithm was proposed for the developed nonlinear binary product codes, and their performance improvement was demonstrated upon comparison to the best known linear binary product codes with the same block length. In the first simulation, the (64, 24, 12) nonlinear binary product code has an effective gain of about effective 2.5 dB and 1 dB gain at BER 10^{-6} and lower error rate values. In the second simulation, the (256, 64, 36) nonlinear binary product code composed of two NR codes has about 0.7 dB gain at BER 10^{-5}. These additional gains, due to better minimum distance, can be transferred into the higher data rate or lower error rate of product codes which make them more attractive for practical application.

Acknowledgement

The authors would like to thank Prof. Shu Lin and Prof. Oyvind Ytrehus for their valuable comments and suggestions.

References

[1] R. Pyndiah, "Near-Optimum Decoding of Product Codes: Block Turbo Codes," *IEEE Trans. Commun.*, vol. 46, no. 8, Aug. 1998, pp. 1003-1010.

[2] D. Williams et al., "IEEE 802.16.3 PHY Utilizing Turbo Product Codes," IEEE 802.16 Broadband Wireless Access Working Group, Ottawa, Canada, IEEE 802.16.3p-01/05, Jan. 17, 2001.

[3] ESTI, "ETSI HiperMAN: Physical (PHY) layer," TS 102 177, v1.4.1, Nov. 2007.

[4] HomePlug Alliance, "HomePlug 1.0 Specifications," June 2001.

[5] IEEE 802.16 TGm, "IEEE 802.16m-07/001r1, Work Plan for Development of IEEE P802.16m Draft Standard & IMT-Advanced Submission," July 19, 2007.

[6] IEEE, "IEEE Standard for Local and Metropolitan Area Networks-Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems," IEEE std 802.16e-2005, Feb. 28, 2006.

[7] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding," *Proc. Int. Commun. Conf.*, Geneva, Switzerland, May 1993, pp. 1064-1070.

[8] R.G. Gallager, *Low Density Parity Check Codes*, Cambridge, USA: MIT Press, 1963.

[9] D. Divsalar et al., "Performance Analysis of Turbo Codes," *Military Commun. Conf.*, vol. 1, Nov. 1995, pp. 91-96.

[10] F.P. Preparata, "A Class of Optimum Nonlinear Double-Error-Correcting Code," *Information and Control*, vol. 13, issue 4, Oct. 1968, pp. 378-400.

[11] A.M. Kerdock, "A Class of Low-Rate Nonlinear Codes," *Information and Control*, vol. 20, no. 2, Mar. 1972, pp. 182-187.

[12] O. Amrani, "Nonlinear Codes: The Product Construction," *IEEE Trans. Comm.*, vol. 55, Oct. 2007, pp. 1845-1851.

[13] Haesik Kim, Garik Markarian, and Valdemar C. da Rocha Jr. "Nonlinear Turbo Product Codes," *Simpósio Brasileiro de Telecomunicações* (*SBrT'07*), Recife/PE, Brazil, Sept. 2007.

[14] A.W. Nordstrom and J.P. Robinson, "An Optimum Nonlinear Code," *Information and Control*, vol. 11, nos. 5-6, Nov.-Dec. 1967, pp. 613-616.

[15] A. Roger Hammon et al., "The Z4-Linearity of Kerdock, Preparata, Goethals, and Related Codes," *IEEE Trans. Inform. Theory*, vol. 40, no. 2, 1994, pp. 301-319.

[16] G.D. Forney, N. Sloane, and M.D. Trott, "The Nordstrom-Robinson Code is the Binary Image of the Octacode," *Coding and Quantization: DIMACS/IEEE Workshop*, Oct. 19-21, 1992, Amer. Math. Soc., edited by R. Calderbank, G. D. Forney, Jr., and N. Moayeri, 1993, pp. 19-26.

[17] G. Markarian and B. Honary, "The Nordstrom-Robinson Code is a (8, 4) GAC over Z4," *Proc. Third UK/Australian Int. Symp. DSP for Commun. Syst.*, Warwick, 1994.

[18] V.S. Pless, W.C. Huffman, and R.A. Brualdi, *The Handbook of Coding Theory*, Elsevier Science Publishers, 1998.

[19] B. Honary and G. Markarian, *Trellis Decoding of Block Codes: A Practical Approach*, Kluwer Academic Publishers, 1997.

[20] Y. Berger and Y. Be'ery, "The Twisted Squaring Construction, Trellis Complexity, and Generalized Weights of BCH and QR Codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 6, Nov. 1996, pp. 1817-1827.

[21] G.D. Forney, Jr., "Coset Codes–Part I: Introduction and Geometrical Classification," *IEEE Trans. Inform. Theory*, vol. 34, no. 5, Sept. 1988, pp. 1123-1151.

[22] G.D. Forney, Jr., "Coset Codes–Part II: Binary Lattices and Related Codes," *IEEE Trans. Inform. Theory*, vol. 34, no. 5, Sept. 1988, pp. 1152-1187.

[23] T. Fujiwara et al., "A Trellis-Based Recursive Maximum-Likelihood Decoding Algorithm for Binary Linear Block Codes," *IEEE Trans. Inform. Theory*, vol. 44, 1998, pp.714-729.

[24] H.T. Moorthy, S. Lin, and G.T. Uehara, "Good Trellises for IC Implementation of Viterbi Decoders for Linear Block Codes," *IEEE Trans. Comm.*, vol. 45, 1997, pp. 52-63.

[25] R.J. McEliece, "On the BCJR Trellis for Linear Block Codes," *IEEE Trans. Inform. Theory*, vol. 42, 1996, pp. 1072-1092.

[26] I. Reuven and Y. Be'ery, "Entropy/Length Profiles, Bounds on the Minimal Covering of Bipartite Graphs, and Trellis Complexity of Nonlinear Codes," *IEEE Trans. Information Theory*, vol. 44, no. 2, Mar. 1998, pp. 580-598.

[27] A. Lafourcade and A. Vardy, "Lower Bounds on Trellis Complexity of Block Codes," *IEEE Trans. Inf. Theory*, vol. 41, no. 6, Nov. 1995, pp. 1938-1954.

[28] S. Lin and D. J. Costello, Jr., *Error Control Coding*, Pearson Prentice Hall, 2004.

[29] T.A. Gulliver and V.K. Bhargava, "A Systematic (16, 8) Code for Correcting Double Errors and Detecting Triple-Adjacent Errors," *IEEE Trans. Computers*, vol. 42, no. 1, Jan. 1993, pp. 109-112.

[30] G.C. Clark and J.B. Cain, *Error-Correction Coding for Digital Communications*, New York: Plenum Press, 1981.

Haesik Kim received the PhD degree in communication systems from Lancaster University, UK in 2009. He is currently a research scientist in VTT Technical Research Center of Finland and is involved in cognitive radio and network projects. Previously, he was with Samsung Advanced Institute of Technology and NEC Laboratory Europe. His research interests include analysis, design, and optimization of wireless communications.

Garik Markarian holds a Chair in Communication Systems and acts as the Head of the Department of Communication Systems of Lancaster University. His reputation as a researcher was built during his involvement in the IEEE802.16, WiMAX, ETSI BRAN, ETSI HYPERACCESS, 3GPP, DVB-DSNG, DVB-RCS, and DVB-S2 standardization bodies, where he chaired a number of working groups and initiated the development of a number of international standards. This reputation was cemented by his publications in national and international professional journals (he has co-authored over 200 publications, including 4 text books, 39 national and international patents, and a great number of papers in international journals), and by his election to the number of national and international professional committees.

Valdemar C. da Rocha, Jr. (M'77, SM'04) received the BSc degree in electronics from the Escola Politecnica, Recife, Brazil, in 1970, and the PhD degree in electronics from the University of Kent at Canterbury, England, in 1976. For the past thirty years, he has been actively involved in teaching and research in the areas of telecommunications and information theory (IEEE Communications Society and Information Theory Society). In 1976, he joined the faculty of the Federal University of Pernambuco, Recife, Brazil, as an associate professor and founded its Electrical Engineering Postgraduate Program. From 1992 to 1996, he was Head of the Department of Electronics and Systems and in 1993 became a professor of Telecommunications. Since 1976, he has been a consultant to both the Brazilian Ministry of Education and the Ministry of Science and Technology on postgraduate education and research in electrical engineering. For two terms (1993 to 1995 and 1999 to 2001), he was the Chairman of the Electrical Engineering Committee in the Brazilian Ministry of Science and Technology. From 2005 to 2007, he was the Head of the Postgraduate Electrical Engineering Program for the Brazilian Ministry of Education. He has been a visiting professor at the Swiss Federal Institute of Technology-Zurich (1990 to 1992), Leeds University (2005 to 2006), and Lancaster University (2007). In 2002 he founded the Information Theory Society Chapter - Brazil. He has published many technical papers in both journals and conference proceedings in information theory, error-correcting codes, and cryptography. He is a founding member (1983) and past President (2004 to 2006 and 2006 to 2008) of the Brazilian Telecommunications Society. He is a Senior Member of the IEEE and a Fellow the Institute of Mathematics and its Applications. He has been involved in the organization of international conferences in Brazil and abroad.