Code Design and Performance Evaluation for Distributed Classification Fusion Based on Algebraic Codes

Eline Alves Santos[1], Francisco M. de Assis[2], Edmar Candeia Gurjão[3], Raimundo C. S. Freire [4]

[1] Universidade Federal de Campina Grande, Campina Grande, PB, Brasil, eline.santos@ee.ufcg.edu.br
[2] Universidade Federal de Campina Grande, Campina Grande, PB, Brasil, fmarcos@dee.ufcg.edu.br
[3] Universidade Federal de Campina Grande, Campina Grande, PB, Brasil, ecandeia@dee.ufcg.edu.br
[4] Universidade Federal de Campina Grande, Campina Grande, PB, Brasil, freire@dee.ufcg.edu.br

Abstract: Distributed classification fusion using error correcting codes has been proposed for wireless sensor networks to incorporate fault-tolerance capability [2]. Usually, the codewords are obtained by random search in the set of binary strings of length N, where N is the number of sensors. In this work it is proposed the use of classical block codes, more specifically BCH codes, to obtain these codewords. The proposed approach allows tailoring decoding algorithms supported by well known algebraic decoding algorithms. In particular, with the new approach it is possible to avoid a massive table look-up-based decoding for a large number of hypotheses, what cannot be achieved with random selected codewords. It is showed that algebraic code-based classification performance is similar to the previous random search-based classification.

Keywords: Distributed classification, wireless sensor networks, coding.

1. INTRODUCTION

Wireless sensor networks (WSNs) are usually composed of a large number of sensor nodes densely deployed to monitor an environment. Sensor nodes are able to sense, process data, and communicate with a fusion center or other node. This kind of network has a wide range of application, such as environment monitoring, medical care, and military surveillance [1]. In this work, we are concern with the problem of multiple target or event classification based on observations from distributed sensor nodes in a noisy environment.

Usually, sensor nodes have limited power and communication capability, thus it is important employ a local data compression on the raw observation at each sensor. Fault-tolerance capability is also an important issue in WSN, since the nodes are prone to failures and replace them may be impossible [1].

Under the constraints above mentioned, in [2] and [3] a distributed classification fusion approach using error correcting codes was proposed to provide a good fault-tolerance capability. In this approach, the sensor nodes only send out binary decisions to the fusion center, but the fusion center or cluster head produces an M-ary decision, where *M* is the number of target or event classes to be distinguished.

The idea is to assigning a block of *N* binary digits to each class in such way that each sensor node produces exactly one of these bits. In [2] and [3] is utilized a completely random search method (Simulated Annealing or Cyclic Column Replacement) in order to build an MxN so-called code matrix. Each one of *M* possible hypotheses or classes is associated with a codeword (defined by one line of the code matrix) and each column describes the decision rule employed at the corresponding sensor node. The decision rule of the fusion center is the minimum Hamming distance between the received word, formed by the bits estimated by the receiver, and codewords.

We propose an approach based on linear block codes. The codewords are a sub-code of a linear block code. This sub-code is not linear, but keeps the minimum Hamming distance between codewords, allows an algebraic decoding and the results indicate that has a similar performance than the previous approach.

The remainder of this paper is organized as follows. In Section 2, we formulated the problem. In Section 3 we expose an upper bound on the probability of error that permits to analyze the performance of system. In Section 4, we describe the code matrix design. Section 5 presents the performance evaluation of the proposed approach. Finally, the Section 6 contains some concluding remarks.

2. PROBLEM STATEMENT

Consider the classification problem of a target or an event with *M* hypotheses, all sensor nodes observe the same phenomenon. We assume that the sensor nodes do not communicate with each other and there is no feedback from fusion center to any sensor node.

Let y_j be the observation of j-th sensor node, where $j = 1, ..., N$. Independent interferences are assumed present at the sensors, and then its observations $\{y_j\}_{j=1}^{N}$ are conditionally independent given their hypotheses.

Based on its observation, the *j*-th sensor makes a decision between one of *M* possible hypotheses and transmits an output bit u_j that is the element $c_{\ell,j}$ of the code matrix **C**, if the hypothesis H_ℓ is locally considered true. The probability of *j*-th sensor to classify H_ℓ given that H_ℓ is the true hypothesis is denoted by $h_{\ell|i}^{(j)}$.

Due to possible channel transmission errors, the received word $u^* = (u_1^*, u_2^*, ..., u_N^*)$ may be different of the transmitted word $u = (u_1, u_2, ..., u_N)$. We assume that the event of link error is independent for all the communication links between sensors and the fusion center, and is also independent of the observations $\{y_j\}_{j=1}^N$ as well as the true hypothesis H_i, and its probability $\Pr[u_j^* \neq u_j]$ is denoted by ε_j.

Based on the received word $u^* = (u_1^*, u_2^*, ..., u_N^*)$, the fusion center makes a multiclass decision by performing minimum Hamming distance decoding. The final decision is H_w if $w = \arg \min_{0 \le \ell \le M-1} d(u, c_\ell)$, where $d(x, y)$ is the Hamming distance between **x** and **y**, and c_ℓ is the row of **C** corresponding to the hypothesis H_ℓ. The code matrix **C** is an MxN matrix with elements $c_{\ell,j} \in \{0,1\}$, $\ell = 0, ..., M-1$, and $j = 1, ..., N$. Each row corresponds to a codeword assigned for one hypothesis $H_\ell \in \Omega = \{H_0, ..., H_{M-1}\}$ and the columns represent the binary classifiers employed at each corresponding sensor node.

If there are more than one hypothesis with the same smallest Hamming distance to the received word, one of them is randomly selected. The model of the system describes in this section is shown in Fig. 1, this topology is widely used [4][5]][6].

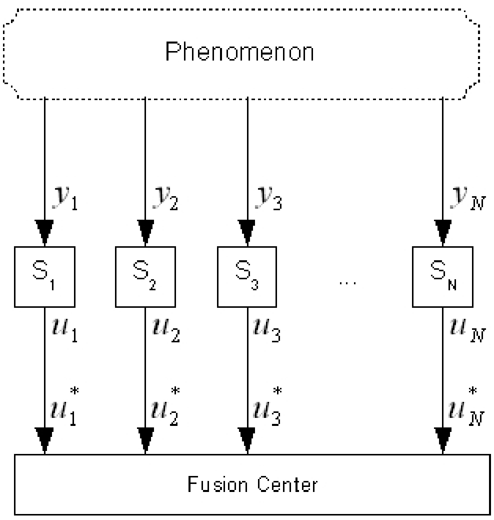

Fig. 1. System Model

3. PERFORMANCE ANALYSI S

In this section we will expose an upper bound on the probability of error that was obtained in [3], this bound permits to analyze the performance of the system given a code matrix and it can be a criterion for a search algorithm to select the code matrix.

Let P_e be the average probability of minimum Hamming distance fusion error defined as:

$$P_e \triangleq \frac{1}{M} \sum_{i=0}^{M-1} \Pr\left(\text{fusion decision } \neq H_i \mid H_i\right) \qquad (1)$$

Consider $\varepsilon_j = \varepsilon$ for $1 \le j \le N$, where $0 \le \varepsilon \le 1/2$, and $h_{k|i}^{(j)} = h_{k|i}$ is the same for all sensors. Then P_e can be bounded above by:

$$P_e \le \frac{1}{M} \sum_{i=0}^{M-1} \sum_{0 \le \ell \le M-1, \ell \neq i} \left(1 - (1 - 2\varepsilon)^2 \times \left(\frac{\sum_{k=0}^{M-1} h_{k|i} \left[d(c_i, c_k) - d(c_\ell, c_k) \right]}{d(c_\ell, c_i)} \right)^2 \right)^{d(c_\ell, c_i)/2} \qquad (2)$$

Note that the bound (2) is function of pair-wise Hamming distances what simplifies its evaluation. Also, the effects of local accuracy is represented by the term $h_{k|i}$ and the effects of link noise is represented by the term ε. In the next section we will describe the code design methodology, where this bound is the criterion for the search algorithm used to select the code matrix.

4. CODE MATRIX DESIGN

The performance of the approach proposed is related with the selected code matrix. It should have a large minimum Hamming distance between codewords (matrix lines) and simultaneously to result in good local binary classifiers, this makes an analytical approach quite difficult.

In [3] the code matrix is selected by simulated annealing, the energy function is set to the probability bound (2). The codewords can be any block of N binary digits. In this work it is proposed that the codewords are in a BCH code. A BCH(n,k,d) is a linear vector sub-space of $\mathbb{Z}_2^n$ with dimension k and minimum Hamming distance d.

For a BCH(n,k,d) there are 2^k codewords that can be combined in groups of M to form a matrix code MxN. In generally the number of possible codedwords is greater than the number of the hypotheses, thus, there are different ways to form the code matrix. Then it is necessary to select the most adequate code matrix, in other words, search a subcode in a BCH code. To select the code matrix we propose a search algorithm called genetic algorithm guided by algebraic code proposed in [7].

Genetic Algorithm (GA) is a particular class of evolutionary algorithms inspired by natural selection. In GA a population of individual solutions is repeatedly modified by genetic operators (selection, crossover and mutation) until the population evolves to an optimal solution.

In a genetic algorithm guided by algebraic code, the classical genetic operators are applied on binary strings that are coded in codewords by multiplication for generator matrix of the code $G_{k \times n}$, the fitness evaluation is made in the codewords.

For the algorithm used here the search is made in a BCH(n,k,d) code. For this, we take $n=N$, each individual is a binary string of length M x k that corresponds to a matrix with ***M*** rows and ***k*** columns, like illustrate in Fig 2.

$$\begin{pmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 \end{pmatrix} \Rightarrow \boxed{0}\boxed{1}\boxed{0}\boxed{1}\boxed{0}\boxed{1}\boxed{0}\boxed{1}\boxed{0}\boxed{1}\boxed{0}\boxed{1}\boxed{1}\boxed{1}\boxed{1}$$

Fig. 2. Example of an individual

The solution is the result of this matrix multiplied by a generator matrix $G_{k \times n}$ of a BCH(n,k,d) code, that is, a code matrix MxN, this ensures that all words of the code matrix are in the BCH code. The objective function is the bound (2) exposed in the previous section. The algorithm pseudo-code is displayed in box 1, in this algorithm *P(t)* represents the population with a number *Q* of individuals in the ***t*** generation and *P'(t)* is generated by classical genetic operators of selection, crossover and mutation. The rule of stop can be maximum number of generations, time limit or sufficient fitness achieved.

Input: objective-function, bound(2), generator code matrix, $G_{k\times n}$
Output: optimum or sub-optimum code matrix $C_{M\times N}$
Initialization:

- $t \leftarrow 0$;
- initialize $p(t) \subset \{0,1\}^{M\times k}$; evaluate $P(t)G$;

Iteration:
WHILE STOP = FALSE **DO**

- $P'(t) \leftarrow$ variation $P(t)$;
- evaluate $P'(t)G$;
- $P(t+1) \leftarrow$ selection $P'(t)$;
- $t \leftarrow t+1$;

END

Box.1. Genetic Algorithm guide by BCH

For comparison purpose, we also selected code matrices by GA without restrictions for the codewords, that is, the individuals are binary strings of length MxN that corresponds a matrix with *M* rows and *N* columns. Box 2 displays the algorithm pseudo-code for this case.

5. RESULTS

In this section, we present some simulations and numerical results to compare the performances between algebraic code-based classification and random search-based classification. For these simulations we assume that:

Input: objective-function, bound(2)
Output: optimum or sub-optimum code matrix $C_{M\times N}$
Initialization:

- $t \leftarrow 0$;
- initialize $p(t) \subset \{0,1\}^{M\times N}$; evaluate $P(t)$;

Iteration:
WHILE STOP = FALSE **DO**

- $P'(t) \leftarrow$ variation $P(t)$;
- evaluate $P'(t)$;
- $P(t+1) \leftarrow$ selection $P'(t)$;
- $t \leftarrow t+1$;

END

Box. 2. Genetic Algorithm for a random search

- The observations $\{y_j\}_{j=1}^{N}$ have Gaussian distribution with mean ℓ and variance $1/\gamma_0$ given that hypothesis H_ℓ is true. Define the local classification rule as H_i is declared true if $(y_j - i)^2 \leq \min_{0\leq \ell \leq M-1, \ell\neq i} (y_j - \ell)^2$.
- Each communication link employs binary antipodal signaling.
- The communication channel between each local sensor to the fusion center is an additive white Gaussian channel.
- The probability of the event of link error ε_j is the same for all sensor nodes. Hence, $\varepsilon_j = \varepsilon = \frac{1}{2} erfc\left(\sqrt{\gamma_s}\right)$, where **erfc(.)** is the complementary error function, and γ_s is the signal-to-noise ratio of the communication link.

First, consider a system with fifteen sensor nodes (N=15) to identify eight hypotheses (M=8). We choose γ_s = 0 dB and γ_0 = 6 dB as target signal-to-noise ratios during the code search. Then according to the code matrix design methodology shown in previous section, we search a code matrix based on the BCH(15,5,7), and compare it with a code matrix selected without restriction in the search space. We also calculate the limit (2) for both code matrix.

In Fig 3, we can see that the code matrix based on BCH(15,5,7) and the code matrix with random selected words have the same performance for γ_0 = 0 dB to about γ_0 = 6 dB, and the first is better from γ_0 = 6 dB onwards. The bounds curves of both code matrix have a behavior similar to the simulation curves, except for the transition point that is at about γ_0 = 8 dB.

We also evaluate large sensor networks, again we choose γ_s = 0 dB and γ_0 = 6 dB as target signal-to-noise ratios during the code search. The code matrices are based on BCH(511,10,223). The networks size was an M = 8 and N = 511 network and an M = 16 and N = 511 network. The results are shown in Fig 4 and Fig 5.

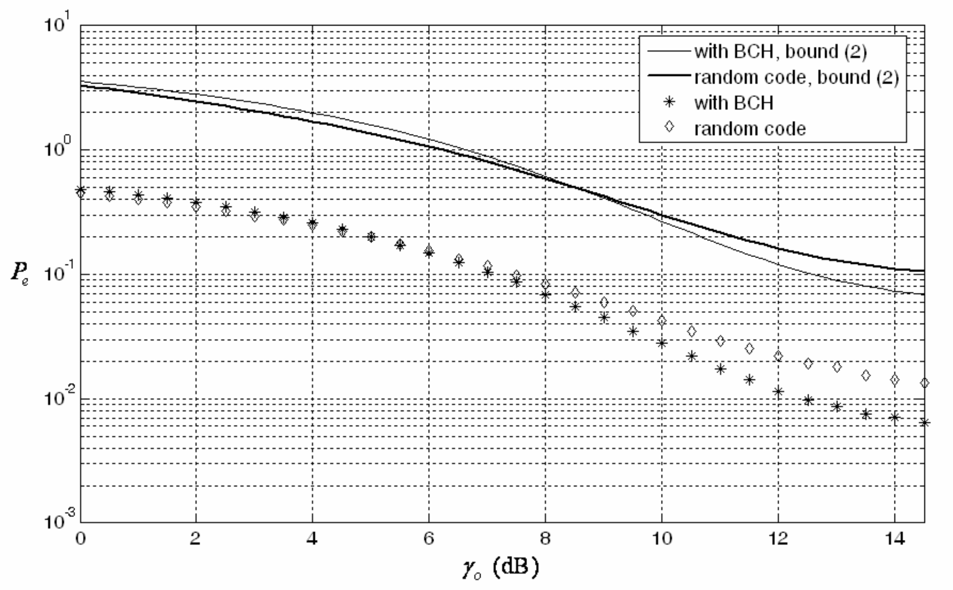

Fig. 3. Simulated performance and bound (2) for two code matrix 8x15 at $\gamma_s = 0$ dB.

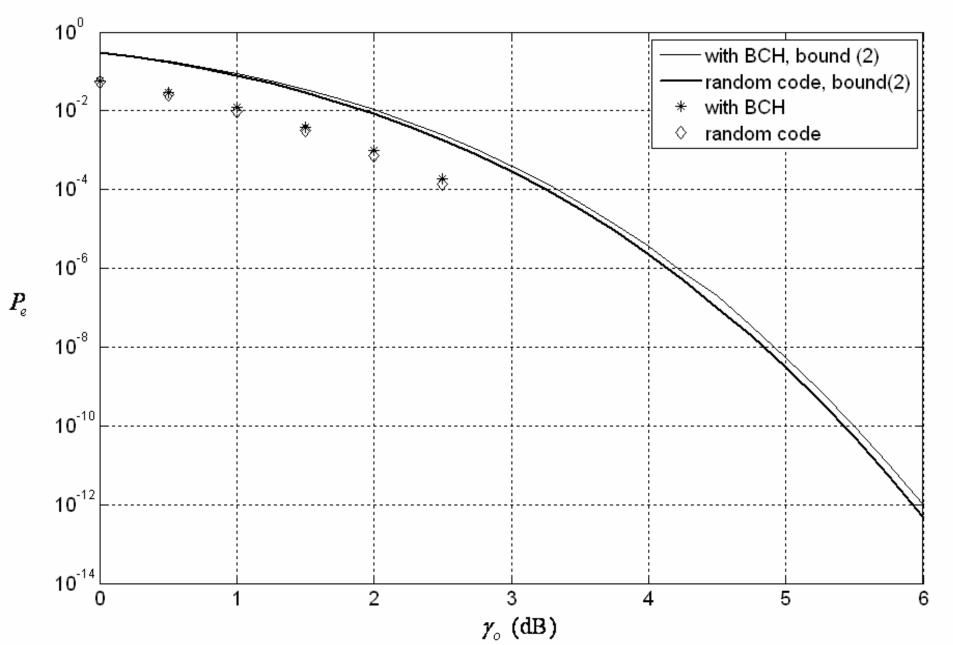

Fig. 4. Simulated performance and bound (2) for two code matrix 8x511 at $\gamma_s = 0$ dB.

Oberving Fig 4 and Fig 5 it possible to see that the code matrices based on BCH(511,10,223) has performance as good as the one obtained with code matrices with random selected words. However, the code matrix based on BCH is advantageous to the decoding processing, since it permits algebraic decoding and the random codes only permits table lookup decoding. To a large number of sensors table lookup decoding is inefficient and in these case blocs codes must be used.

6. CONCLUSIONS

In this work, we propose an approach where the codewords are obtained from classical block codes like BCH, we also propose a code design by genetic algorithm guided by code. The results show that there is no significant loss of performance when the approach based on algebraic codes are used, with the advantage of a more structured code search and decoding processes.

REFERENCES

[1] Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci; "A Survey on Sensor Networks", Communications Magazine, IEEE, August 2002.

[2] T. Y. Wang, Y. Han, P. Varshney, P. N. Chen; "Distributed Fault-Tolerant Classification in Wireless Sensor Networks", Selected Areas in Communications, IEEE journal, April 2005.

[3] C. Yao, P. N. Chen, T. Y. Wang, Y. S. Han, P. K. Varshney; "Performance Analysis and Code Design for Minimum

Hamming Distance Fusion in Wireless Sensor Networks", Information Theory, IEEE Transactions, May 2007.

[4] J. J. Xiao, Z. Q. Luo. "Decentralized Estimation in an Inhomogeneous Sensing Enviroment", Information Theory, IEEE Transactions, Octuber 2005.

[5] J. F. Chamberland, V. Veeravalli; "Asymptotic Results for Decentralized Detection in Power Constrained Wireless sensor Networks", Selected Areas in Communications, IEEE Journal, August 2004.

[6] J. F. Chamberland, V. Veeravalli; "Descentralized Detection in Sensor Networks", Signal Processing, IEEE Transactions, February 2003.

[7] ASSIS, F. M. Genetic algorithms and packing of block codes. In: International Conference on Telecomunications 97, 1997, Melbourne. Proceedings of the ICT97. Melbourne, Australia: Office of Continuing Education, Monash University, 1997. v. 3. p. 1045-1048.

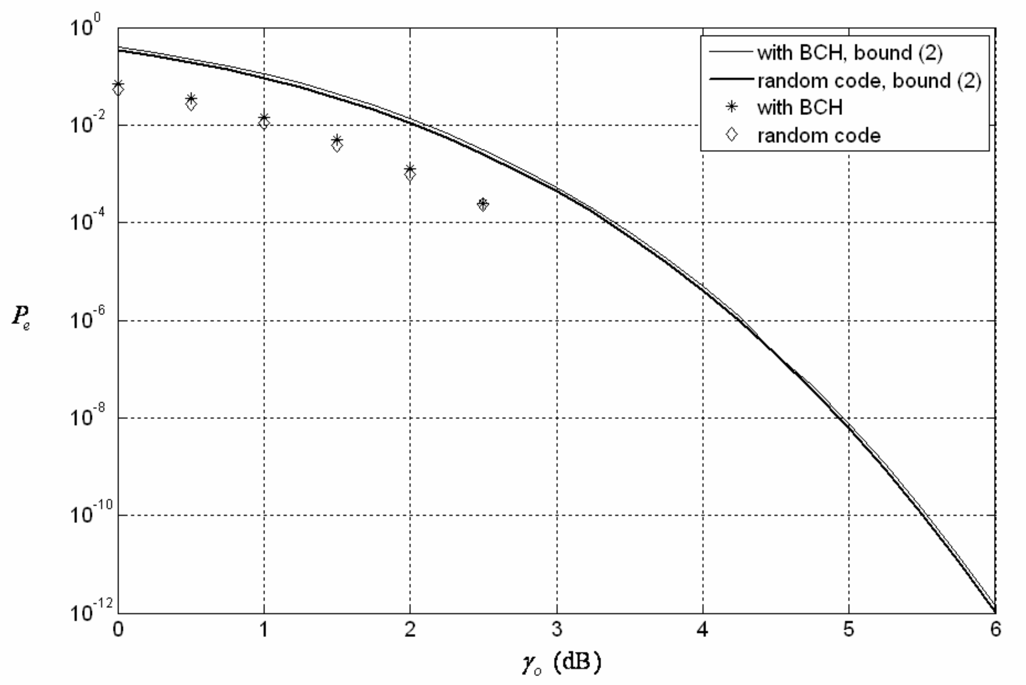

Fig. 5. Simulated performance and bound (2) for two code matrix 16x511 at γ_s = 0 dB.