

Distributed Fault-Tolerant Classification in Wireless Sensor Networks

Tsang-Yi Wang, *Member, IEEE*, Yunghsiang S. Han, *Member, IEEE*, Pramod K. Varshney, *Fellow, IEEE*, and Po-Ning Chen, *Senior Member, IEEE*

Abstract—Fault-tolerance and data fusion have been considered as two fundamental functions in wireless sensor networks. In this paper, we propose a novel approach for distributed multiclass classification using a fault-tolerant fusion rule for wireless sensor networks. Binary decisions from local sensors, possibly in the presence of faults, are forwarded to the fusion center that determines the final classification result. Classification fusion in our approach is implemented via error correcting codes to incorporate fault-tolerance capability. This new approach not only provides an improved fault-tolerance capability but also reduces computation time and memory requirements at the fusion center. Code matrix design is essential for the design of such systems. Two efficient code matrix design algorithms are proposed in this paper. The relative merits of both algorithms are also studied. We also develop sufficient conditions for asymptotic detection of the correct hypothesis by the proposed approach. Performance evaluation of the proposed approach in the presence of faults is provided. These results show significant improvement in fault-tolerance capability as compared with conventional parallel fusion networks.

Index Terms—Data fusion, decision fusion, distributed classification, error correcting codes, fault-tolerance, multisensor systems, wireless sensor networks (WSNs).

I. INTRODUCTION

CLASSIFICATION based on observations from distributed sensor nodes is an important application of wireless sensor networks (WSNs) [1]–[6]. In WSN, the bandwidth of communication channels is limited, and each node has limited communication and computation ability. Thus, decision fusion instead of data fusion is generally preferable due to these constraints imposed by the network [4], [5]. Hence, in this paper, we only

consider the decision fusion approach. In a decentralized multiclass classification problem, each node (local detector) usually performs multiclass classification and transmits its decision to the fusion center (manager node or cluster head in WSN) [7], [8]. This decision is usually represented by $\log_2 M$ information bits, where M is the number of classes to be distinguished. However, in WSN, sensor nodes are driven by batteries and have very low-energy resources [1]. Energy consumption is a very important factor that determines the lifetime of a WSN. Thus, in order to conserve energy and increase WSN lifetime, it is necessary for each local detector to send fewer bits to the fusion center. This also reduces the bandwidth required. In this paper, we consider the case where the local detectors are only allowed to perform binary classification and send out a binary decision to the fusion center. These local decisions are, however, constrained through a designed code matrix that enables the fusion center to determine the final decision in favor of one of the M classes.

There are several related papers in which binary information is combined to make final multiclass decisions [9]–[11]. Unlike the previous approaches, in this paper, the design and performance analysis of the fusion rule in the presence of faults is considered. Fault-tolerance capability is an important issue, while designing classification systems in WSN. Several researchers have considered the design of fault-tolerant distributed detection systems [12]–[15]. However, they only designed the system based on a known *a priori* failure probability and considered the binary detection problem. Using error correcting codes to provide fault-tolerance capability has not been proposed. The extension from binary detections to multihypothesis detections also needs to be considered in WSN. Therefore, in this paper, we propose a novel fault-tolerant distributed multiclass classification fusion approach using error correcting codes (DCFEC) that provides excellent fault-tolerance capability in WSN. The DCFEC approach includes employing a fault-tolerant fusion rule to tolerate deterministic faults and incorporating *a priori* failure probabilities for random faults if these *a priori* failure probabilities can be obtained in advance. That is, the DCFEC approach can tolerate both deterministic or random faults. For deterministic faults such as stuck-at faults, and hardware or software damage, we employ the fault-tolerant fusion rule to achieve the fault-tolerance capability. When random faults are present according to a certain probability distribution function caused by continually changing environmental characteristics, such as channel transition errors, we design the fault-tolerant system by incorporating these *a priori* failure probabilities.

Since large scale wireless sensor networks often have many nodes, sensor nodes are usually aggregated into several groups

Manuscript received November 2003; revised October 2004. This work was supported in part by the W. M. Keck Foundation under Grant CK #1329, in part by the SUPRIA Program of the CASE Center, Syracuse University, Syracuse, NY, and in part by the NSC of Taiwan, R.O.C., under Grant NSC 90-2213-E-260-007. This work was presented in part at the 2003 SPIE, Orlando, FL, and the 2003 International Conference on Information Fusion, Cairns, Australia.

T.-Y. Wang was with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13210 USA. He is now with the Graduate Institute of Communication Engineering, National Chi Nan University, Nantou 545, Taiwan (e-mail: tswang@ncnu.edu.tw).

Y. S. Han was with the Department of Computer Science and Information Engineering, National Chi Nan University, Nantou 545, Taiwan. He is now with the Graduate Institute of Communication Engineering, National Taipei University, Taipei 237, Taiwan (e-mail: yshan@mail.ntpu.edu.tw).

P. K. Varshney is with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244 USA (e-mail: varshney@syr.edu).

P.-N. Chen is with the Department of Communication Engineering, National Chiao-Tung University, Hsinchu 30050, Taiwan (e-mail: qponing@mail.nctu.edu.tw).

Digital Object Identifier 10.1109/JSAC.2005.843541

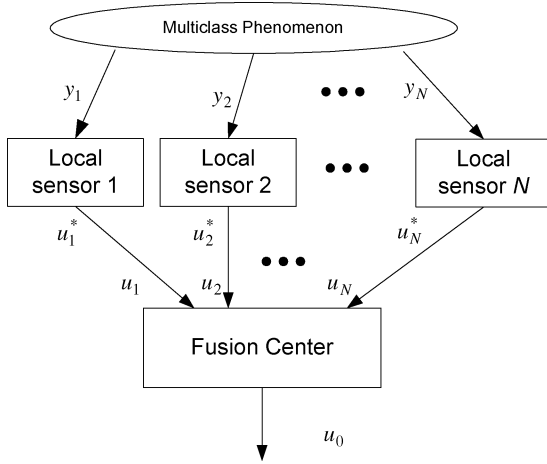


Fig. 1. Distributed classification system architecture.

(or clusters) to reduce the amount of power spent on long distance data transmissions [16]. Hence, the members of each group (or cluster) are within transmission range of each other and the number of members of each group (or cluster) is 10–40 [1]. It is possible for each group (or cluster) to run the distributed detection algorithm separately [16]. Collaborative detection processing is carried out among nodes within a group (or cluster) under the control of a manager node (or cluster head). At a node, the local decision is made according to the off-line optimized local decision rule based on the given fault-tolerant fusion rule in a particular cluster. The final decision is made at the manager node (or cluster head) by employing the fault-tolerant fusion rule. The general distributed classification architecture which is suitable for WSN is given in Fig. 1. A detailed discussion of Fig. 1 will be provided in Section II.

The proposed scheme is designed as follows. We first design an error-correcting code matrix. Each codeword forms a row in the code matrix and corresponds to one of the classes to be distinguished. Each column represents the binary decision rule employed at the corresponding local sensor. The local decision rule is designed off-line in advance by the system-wide optimization based on the code matrix.¹ During the on-line operation, each local sensor makes its decision by employing the off-line optimized decision rule. The fusion center decides on the class based on the binary inputs received (the received vector) from the local detectors. To provide fault-tolerance ability, the fusion center performs fault-tolerant fusion by minimum distance decoding, i.e., it decides on the codeword that is closest in Hamming distance to the received vector, where the Hamming distance between two binary vectors is defined as the number of distinct positions between these vectors. This decision on a codeword is equivalent to making the M -ary decision regarding the classes, i.e., to making a classification decision. The fault-tolerance or error correction capability of the system is determined by the minimum Hamming distance of the code employed.

¹In practice, this off-line local decision rule can be optimized at each local sensor or it can be determined by the base station of the WSN after the cluster is formed. Note that this optimization is performed only once as long as all members of the cluster remain the same.

Unlike the Chair–Varshney fusion rule [17], which is the optimal fusion rule when the local sensor decision rules are given, the proposed fault-tolerant fusion rule provides enough distance among all the decision regions corresponding to their hypotheses. The observed local decision vectors could still fall into correct decision regions even when several sensors fail. In addition to having good fault-tolerance ability, the DCFECC approach also reduces the memory requirement and speeds up the fusion process at the fusion center. The reduction in memory requirement is achieved due to the decoding operation based on Hamming distance operations employed in the fault-tolerant fusion rule. This fusion rule only needs a binary code matrix to be stored at the fusion center instead of real valued parameters required by the Chair–Varshney fusion rules. The speed up of the fusion process results from the fact that the fault-tolerant fusion process only requires computation involving integers while the Chair–Varshney fusion process requires computation involving real numbers. Thus, fewer bit computations are required in the DCFECC approach as compared with the system employing the Chair–Varshney fusion rule. The above benefits also imply economy in terms of the hardware cost. These potential advantages make the DCFECC approach quite suitable for use in WSN [1].

The classification performance of the DCFECC approach is strongly related to the chosen code matrix. How to design a good code matrix for use in this approach is clearly an important issue. It is quite difficult to obtain a good code matrix using an analytical approach, since the decision rules, i.e., the binary rules at the sensors specified by the columns of the code matrix, while keeping sufficient Hamming distance between rows of the code matrix to provide desired fault-tolerance, interact with each other in a very complicated manner. Exhaustive search for an optimal code matrix is computationally intensive and unaffordable even for a code matrix of small size. Therefore, we propose more computationally efficient code design algorithms based on a cyclic column replacement approach and simulated annealing in this paper. The results show that the cyclic column replacement approach is faster but may converge to a local optimum depending on the chosen initial code matrix. Simulated annealing is quite robust irrespective of the initial code matrix chosen, and is able to construct better code matrices.

The asymptotic performance analysis for the DCFECC approach is also provided. The results show that the probability of error for the DCFECC approaches zero asymptotically as long as the minimum Hamming distance of the employed code matrix satisfies the given conditions. The results reveal that the fault tolerance capability is related to the minimum Hamming distance of the code matrix employed by the DCFECC.

This paper is organized in the following manner. System description and a brief introduction to the DCFECC approach are given in Section II. Section III develops the optimal local decision rules when the fault-tolerant fusion rule is employed. In Section IV, we present the search algorithms for an optimal code matrix. Asymptotic performance analysis for the DCFECC is provided in Section V. Performance comparisons between the fault tolerance capability of DCFECC and the 1-bit conventional approach (CA) are provided in Section VI. Concluding remarks are provided in Section VII.

II. PRELIMINARIES

The multiclass classification problem is considered in this paper. System structure of a distributed M -ary classification system is shown in Fig. 1. Let H_ℓ , where $\ell = 0, 1, \dots, M-1$ and $M \geq 2$, denote the M hypotheses to be distinguished at each of the N sensors. The *a priori* probabilities of these M hypotheses are denoted by $P(H_\ell) = P_\ell$, respectively. The observation at each local sensor or detector is represented by y_j , where $j = 1, \dots, N$. After processing the observations locally, possibly in the presence of sensor faults, the local decisions u_j^* are transmitted to the fusion center. The fusion center then receives a word consisting of the binary decisions made by the local sensors, which can be expressed as $\mathbf{u} = (u_1, u_2, \dots, u_N)$, where $u_j, j = 1, \dots, N$, is the received local decision of sensor j at the fusion center. Note that, in general, u_j is not equal to u_j^* due to possible channel transmission errors. Furthermore, the following assumptions are made to the system.

Assumption 1: We assume that $N > M$, since sensor nodes in most sensor networks are densely deployed, and the number of sensors is much larger than the number of hypotheses.

Assumption 2: The distribution function of y_j under each hypothesis is known. Thus, the conditional probability density functions of these observations, when H_ℓ is given, can be represented by $P(y_j|H_\ell)$.

Assumption 3: The observations y_1, y_2, \dots, y_N are conditionally independent given their hypotheses.

Assumption 4: The output from each sensor j , u_j^* , and $1 \leq j \leq N$, is a binary value (0 or 1). The inputs to the fusion center, $u_j, 1 \leq j \leq N$, are also binary values.

Assumption 5: During the on-line operation, there is no communication among local sensors. Thus, each local sensor makes its decision by itself based only on its own observation and independently of the hypothesis present.

Assumption 6: The received local decision u_j at the fusion center depends only on the output decision u_j^* of local sensor j .

In our DCFECC approach, a code matrix \mathbf{T} is selected to perform both off-line local decision optimization and on-line fault-tolerant fusion at the fusion center. The code matrix is an $M \times N$ matrix with elements $t_{\ell j} \in \{0, 1\}$, $\ell = 0, \dots, M-1$, and $j = 1, \dots, N$. Each hypothesis $H_\ell \in \Omega = \{H_0, H_1, \dots, H_{M-1}\}$ is associated with a row in the code matrix \mathbf{T} . Each column of the matrix \mathbf{T} stands for a binary classification rule at each corresponding sensor. Each local decision rule $g_j(y_j)$ is off-line optimized in advance based on the code matrix \mathbf{T} to achieve a better system performance. During the on-line operation, each sensor employs the off-line designed local decision rule to make a binary decision u_j^* , where $u_j^* = 0$, if H_j^0 is declared, and $u_j^* = 1$, if H_j^1 is declared. $H_j^1 = \{H_\ell | t_{\ell j} = 1, H_\ell \in \Omega\}$ and $H_j^0 = \{H_\ell | t_{\ell j} = 0, H_\ell \in \Omega\}$ denote the two sets of classes with the properties $H_j^1 \cap H_j^0 = \emptyset$ and $H_j^1 \cup H_j^0 = \Omega$, which are determined according to the j th column of \mathbf{T} .

Fusion processing at the fusion center is carried out to obtain the multiclass final decision u_0 which is one of the possible hypotheses in Ω . Note that u_0 is not a binary decision as u_i is for $1 \leq i \leq N$. In this paper, the fault-tolerant fusion rule is proposed and is stated as follows:

Fault-tolerant fusion rule (minimum Hamming distance decoding rule): The multiclass final decision $u_0 \in \Omega$ is H_ω if $\omega = \arg \min_{0 \leq \ell \leq M-1} d_H(\mathbf{u}, \mathbf{t}_\ell)$, where $d_H(\mathbf{x}, \mathbf{y})$ is the Hamming distance between \mathbf{x} and \mathbf{y} , and $\mathbf{t}_\ell = (t_{\ell 1}, t_{\ell 2}, \dots, t_{\ell N})$ is the row of \mathbf{T} corresponding to the hypothesis H_ℓ . The tie-break rule is to randomly pick a codeword from those with the same smallest Hamming distance to the received vector.

Thus, by constraining the local decisions through the code matrix, binary local decisions suffice for an M -ary hypothesis testing problem without losing information regarding the hypotheses.

Let d_{\min} be the smallest Hamming distance between any pair of row vectors of the code matrix \mathbf{T} . In order to have a better fault-tolerance capability, it is reasonable to choose a code matrix such that its d_{\min} is as large as possible. In the case of binary hypotheses, there are only two codewords in the code matrix \mathbf{T} . Hence, the all zero and all one codewords have the largest Hamming distance and they expect to yield the best fault-tolerant performance. Furthermore, in this case, the proposed fault-tolerant fusion rule is equivalent to the majority rule. When $M > 2$ and given N , according to the Gilbert–Varshamov bound [18], there exists a code matrix whose minimum Hamming distance is at least d_{\min} provided

$$\sum_{i=0}^{d_{\min}-1} \binom{N-1}{i} < 2^{N-\lceil \log_2 M \rceil}.$$

Furthermore, for a given N and desired $d_{\min} > N/2$, the largest number of distinct hypotheses M is bounded by the Plotkin bound [18] as

$$M \leq 2 \left\lfloor \frac{d_{\min}}{2d_{\min} - N} \right\rfloor \quad \text{for } d_{\min} > \frac{N}{2}.$$

The above two bounds give a possible range of the largest d_{\min} that can be obtained when M and N are given. For example, if $M = 4$ and $N = 10$, the largest d_{\min} of matrices that can be found is between 3 and 6.

III. OPTIMAL LOCAL BINARY DECISION RULES

In the following, we design optimal local decision rules for the proposed multiclass distributed classification strategy based on the minimization of misclassification (or decision error) probability when the fault-tolerant fusion rule is employed.

Let us define $C_{i_1, i_2, \dots, i_N}^\ell$, where $i_1, i_2, \dots, i_N \in \{0, 1\}$, as the cost that the received word at the fusion center $\mathbf{u} = (u_1, u_2, \dots, u_N)$ equals (i_1, i_2, \dots, i_N) and the true hypothesis is H_ℓ . These costs $C_{i_1, i_2, \dots, i_N}^\ell$ can be determined by the decision regions of codewords. According to the fault-tolerant fusion rule at the fusion center, the decision region D of a codeword $\mathbf{t} \in \mathbf{T}$ is given as follows:

$$D(\mathbf{t}) = \{\mathbf{u} | d_H(\mathbf{u}, \mathbf{t}) \leq d_H(\mathbf{u}, \mathbf{t}') \text{ for all } \mathbf{t}' \in \mathbf{T}\}$$

where $\mathbf{T} = \{\mathbf{t}_\ell | \ell = 0, \dots, M-1\}$ is the set of all codewords, i.e., all rows of the code matrix. In order to minimize the probability of misclassification, set $C_{i_1, \dots, i_N}^\ell = 0$ if (i_1, \dots, i_N) is in the decision region of \mathbf{t}_ℓ that is the row of \mathbf{T} corresponding to the hypothesis H_ℓ ; otherwise, set $C_{i_1, \dots, i_N}^\ell = 1$. Whenever

a received vector (i_1, \dots, i_N) simultaneously belongs to decision regions of $\mathbf{t}_{\ell_0}, \mathbf{t}_{\ell_1}, \dots, \mathbf{t}_{\ell_{q-1}}$, where $q > 1$, for all $\ell = \ell_0, \dots, \ell_{q-1}$, set $C_{i_1, \dots, i_N}^\ell = (1 - 1/q)$, i.e., the tie-break rule. For instance, given the code matrix \mathbf{T} in

Code Matrix I : (1, 9, 5, 13, 3, 11, 7, 1, 2, 4)

we have $C_{0,0,1,1,0,0,1,0,1,1}^0 = 1$, $C_{0,0,1,1,0,0,1,0,1,1}^1 = 1$, $C_{0,0,1,1,0,0,1,0,1,1}^2 = 0$, and $C_{0,0,1,1,0,0,1,0,1,1}^3 = 1$, since the Hamming distance between $\mathbf{u} = (0, 0, 1, 1, 0, 0, 1, 0, 1, 1)$ and codeword \mathbf{t}_2 is the smallest. Throughout this paper, the referred code matrices are represented as a vector of M bit integers. Each integer r_j corresponding to column j in the code matrix represents a column vector in the code matrix, and can be expressed as $r_j = \sum_{\ell=0}^{M-1} t_{\ell j} \times 2^\ell$. For instance, the integer 5 in column 3 of Code Matrix I represents $t_{03} = 1$, $t_{13} = 0$, $t_{23} = 1$, and $t_{33} = 0$.

For modeling channel transmission errors, we incorporate a probabilistic fault model similar to that proposed in [14] and [15] into the DCFECC approach. Let $p_{1j} = P(u_j = 1 | u_j^* = 0)$ be the probability that the fusion center receives $u_j = 1$ when the local decision output in sensor j , u_j^* , is 0. The probability $p_{0j} = P(u_j = 0 | u_j^* = 1)$ is similarly defined.

According to the above cost assignment, the probability of decision error can be written as

$$P_e = \sum_{i_1, \dots, i_N, i_1^*, \dots, i_N^*, \ell} \int_{y_1, \dots, y_N} P_\ell P(u_1 = i_1, \dots, u_N = i_N, u_1^* = i_1^*, \dots, u_N^* = i_N^* | y_1, \dots, y_N | H_\ell) C_{i_1, \dots, i_N}^\ell. \quad (1)$$

By Assumption 5 and Assumption 6, we can rewrite the previous equation as

$$P_e = \sum_{i_1, \dots, i_N, i_1^*, \dots, i_N^*, \ell} \int_{y_1, \dots, y_N} P_\ell P(u_1 = i_1 | u_1^* = i_1^*) \times \dots \times P(u_N = i_N | u_N^* = i_N^*) P(u_1^* = i_1^* | y_1) \times \dots \times P(u_N^* = i_N^* | y_N) P(y_1, \dots, y_N | H_\ell) C_{i_1, \dots, i_N}^\ell.$$

Let us employ the person-by-person optimization (PBPO) procedure [7]. P_e is minimized if we set the local decision rule at sensor k as

$$\sum_{\ell} P(y_k | H_\ell) \mathbf{K}_{k\ell} \begin{matrix} u_k^*=1 \\ \geq 0 \\ u_k^*=0 \end{matrix} \quad (2)$$

where

$$\begin{aligned} \mathbf{K}_{k\ell} = & \sum_{i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_N, i_1^*, \dots, i_{k-1}^*, i_{k+1}^*, \dots, i_N^*} \\ & \times \int_{y_1, \dots, y_{k-1}, y_{k+1}, \dots, y_N} P_\ell P(u_1 = i_1 | u_1^* = i_1^*) \\ & \times \dots \times P(u_{k-1} = i_{k-1} | u_{k-1}^* = i_{k-1}^*) \\ & \times P(u_{k+1} = i_{k+1} | u_{k+1}^* = i_{k+1}^*) \\ & \times \dots \times P(u_N = i_N | u_N^* = i_N^*) \\ & \times P(u_1^* = i_1^* | y_1) \times \dots \times P(u_{k-1}^* = i_{k-1}^* | y_{k-1}) \\ & \times P(u_{k+1}^* = i_{k+1}^* | y_{k+1}) \times \dots \times P(u_N^* = i_N^* | y_N) \\ & \times P(y_1, \dots, y_{k-1}, y_{k+1}, \dots, y_N | y_k, H_\ell) \\ & \times \left[C_{i_1, \dots, i_{k-1}, 0, i_{k+1}, \dots, i_N}^\ell - C_{i_1, \dots, i_{k-1}, 1, i_{k+1}, \dots, i_N}^\ell \right] \\ & \times (1 - p_{1k} - p_{0k}). \end{aligned}$$

$\mathbf{K}_{k\ell}$ can be further simplified as

$$\begin{aligned} \mathbf{K}_{k\ell} = & \sum_{i_1, \dots, i_{k-1}, i_{k+1}, \dots, i_N} P_\ell P(u_1 = i_1 | H_\ell) \\ & \times \dots \times P(u_{k-1} = i_{k-1} | H_\ell) P(u_{k+1} = i_{k+1} | H_\ell) \\ & \times \dots \times P(u_N = i_N | H_\ell) \\ & \times \left[C_{i_1, \dots, i_{k-1}, 0, i_{k+1}, \dots, i_N}^\ell - C_{i_1, \dots, i_{k-1}, 1, i_{k+1}, \dots, i_N}^\ell \right] \\ & \times (1 - p_{1k} - p_{0k}) \quad (3) \end{aligned}$$

due to Assumption 3.

Thus, $K_{k\ell}$ represents a weight that depends on the decision rule at other sensors due to the conditional probabilities $P(u_j | H_\ell)$, $j = 1, \dots, N$, $j \neq k$, and the coupled costs C_{i_1, \dots, i_N}^ℓ . It can be interpreted as the expected cost difference of sensor k making the decision 1 or 0 with respect to the local decisions of all other sensors. The term $(1 - p_{1k} - p_{0k})$ appears due to the possible transmission error.

The above optimal local decisions are determined when the fault-tolerant fusion rule is given. Whenever the code matrix is changed, the corresponding optimal local decision rules are also modified. The system performance of the DCFECC approach is then related to the given code matrix. Therefore, it is important to find a good code matrix such that the probability of decision error is low.

It is easy to see that the statistic at sensor k on the left hand side of (2) depends on the binary decision rules at the other sensors. Even when the code matrix is given, it is very difficult to find the globally optimal threshold. Instead of finding a globally optimal threshold, a locally optimal threshold may be sufficient in many applications. An algorithm that could be used to search for the optimal threshold is the iterative Gauss–Seidel cyclic coordinate descent algorithm [7] which may converge to a locally optimal threshold. Usually, convergence depends on the chosen initial values of $P(u_1 | H_\ell)$, $P(u_2 | H_\ell)$, \dots , $P(u_N | H_\ell)$.

IV. CODE DESIGN METHODOLOGY

The objective while designing a good code matrix is to have the fusion system exhibit good performance in both fault-free and faulty situations. In general, the minimum Hamming distance in a code matrix should be as large as possible since larger Hamming distance between codewords provides the system the ability to tolerate more faults. However, for the code matrix used in the DCFECC, larger Hamming distance does not always ensure good performance in both fault-free and faulty situations. System performance also depends on the patterns of columns in the code matrix, which determine the performance of local binary classifiers (detectors). If a code matrix has larger Hamming distance but results in poor binary classifiers, then the overall system performance degrades. Moreover, when some local detectors fail to perform normally, the operation of the system will rely only on the other functional local detectors. Obviously, the fault-tolerance ability will also be degraded when poor binary local classifiers dominate system performance. Therefore, a good code matrix should have a large minimum Hamming distance and simultaneously result in good local binary classifiers.

The coupling with local decision rules in distributed classification makes the code design quite complicated. To achieve

system-wide optimization, local sensors often use different decision strategies from the case when they are not in collaboration. Thus, the code matrix design can not be viewed as the independent design of individual column vectors (binary classifiers). Instead of analytically designing the code matrix with these interwoven rules, we propose two heuristic algorithms to efficiently solve the code design problem.

A. Code Design by Cyclic Column Replacement

For the cyclic column replacement approach, we first set a minimum Hamming distance constraint based on the required fault-tolerance capability. We then initialize the algorithm by picking a code matrix $\mathbf{T}^{(0)} = \{\mathbf{t}^{(0),1}, \mathbf{t}^{(0),2}, \dots, \mathbf{t}^{(0),N}\}$, where $\mathbf{T}^{(0)}$ must satisfy the minimum Hamming distance constraint. $\mathbf{t}^{(0),j}$ represents the j th column vector of code matrix $\mathbf{T}^{(0)}$. In the following, the first superscript s of each column in this matrix indicates that this column is the result after s th iteration, while the second superscript indicates the column index of the code matrix. During the first iteration, the column vector $\mathbf{t}^{(0),1}$ is first replaced by $\mathbf{t}^{(1),1}$ so as to minimize $P_e(\{\mathbf{t}^1, \mathbf{t}^{(0),2}, \dots, \mathbf{t}^{(0),N}\})$ and ensure that the code matrix $\{\mathbf{t}^1, \mathbf{t}^{(0),2}, \dots, \mathbf{t}^{(0),N}\}$ fulfills the minimum Hamming distance requirement, while other column vectors $\mathbf{t}^{(0),2}, \dots, \mathbf{t}^{(0),N}$ remain fixed, where $P_e(\{\mathbf{t}^1, \mathbf{t}^{(0),2}, \dots, \mathbf{t}^{(0),N}\})$ denotes the probability of misclassification when $\{\mathbf{t}^1, \mathbf{t}^{(0),2}, \dots, \mathbf{t}^{(0),N}\}$ is the code matrix in (1). Next, the second column $\mathbf{t}^{(0),2}$ is replaced by $\mathbf{t}^{(1),2}$ so as to minimize $P_e(\{\mathbf{t}^{(1),1}, \mathbf{t}^2, \dots, \mathbf{t}^{(0),N}\})$ and at the same time meet the minimum Hamming distance requirement. This procedure is continued for all the columns. The first iteration is complete after all the column vectors have been updated. Once the first iteration is complete, the next iterations are run in a similar way. The iterative algorithm terminates when the code matrix remains the same after an iteration.

Although the search process for each column in this algorithm needs to search 2^M possible column vectors, only the column vectors satisfying the minimum Hamming distance requirement are needed to determine the probability of error. It is surprising that this algorithm converged very fast in all the simulations we have conducted. The algorithm took only one or two iterations to converge. This indicates that there are many local optimal solutions that this code search procedure converges to. In order to avoid convergence to a local optimal solution, we propose to use simulated annealing to search for a global optimal solution.

B. Code Design by Simulated Annealing

Simulated annealing is a stochastic algorithm for obtaining approximate solutions to combinatorial optimization problems. It has been successful in many diverse applications. This algorithm is designed to search for the global optimum and is usually robust. Gamal *et al.* [19] have constructed good source codes, error-correcting codes, and spherical codes by using simulated annealing. For constructing the optimal code matrix here, the energy function is set to the probability of misclassification as shown in (1). The minimum Hamming distance is also set to meet the fault-tolerance requirement.

The annealing schedule is decided as follows. We first generate some random code matrices and determine their corresponding optimal local decision rules. We then use them to determine the encountered range of values of $\Delta E = P_e(\mathbf{T}') - P_e(\mathbf{T})$, where \mathbf{T}' and \mathbf{T} are potential candidates for code matrices. Then, an initial temperature is set sufficiently high so that the algorithm has a better chance of finding the global optimum. We set $T = 1$ in our case. The cooling control parameter $\alpha = 0.9$ is chosen for lowering the temperature using $T \leftarrow \alpha T$. This chosen value of α ensures slow cooling in the annealing process for our problem, and is essential for achieving a low-energy state.

Random changes in the code configuration are achieved by perturbation of the code matrix. The amount of perturbation could be done on a case by case basis. In our code design process, a codeword of the current code matrix is randomly selected and then one or two randomly selected bits of the codeword are flipped. Each time a new code matrix is generated, we have to check whether or not the new code matrix meets the minimum Hamming distance requirement. If not, a new code matrix is generated according to the above code matrix perturbation until the minimum Hamming distance constraint is satisfied for the new code matrix. Specifically, the algorithm is given as follows.

- Step 1) Set the minimum Hamming distance requirement, and initialize the algorithm by selecting a random code matrix \mathbf{T} satisfying the minimum Hamming distance constraint. Determine the optimal local decisions according to (2) and (3) based on \mathbf{T} . Set the temperature $T = 1$. Compute the energy $P_e(\mathbf{T})$ according to (1).
- Step 2) (Main iteration.)
 - a) Obtain \mathbf{T}' by perturbing \mathbf{T} until the minimum Hamming distance constraint is satisfied.
 - b) Based on \mathbf{T}' , the local decision rules are optimized according to (2) and (3).
 - c) Compute $\Delta E = P_e(\mathbf{T}') - P_e(\mathbf{T})$, and replace \mathbf{T} by \mathbf{T}' if $\Delta E < 0$; otherwise, replace \mathbf{T} by \mathbf{T}' with probability $e^{-\Delta E/T}$.
 - d) Repeat Step 2a) to Step 2c) until the number of times the energy drops exceeds a prescribed number or too many iterations, whichever comes first.
 - e) Lower temperature by $T \leftarrow \alpha T$, and return to Step 2a).
- Step 3) Terminate the algorithm when a stable code matrix configuration is observed or a prescribed running time is expired.

Example 1: A fusion center and ten independent local sensors are considered to identify four equally likely hypotheses H_0, H_1, H_2, H_3 in the example presented here. Furthermore, we assume that all the sensor measurements are identically distributed. The observations are statistically independent. The probability density function for each hypothesis is assumed to be a Gaussian distribution with the same variance ($\sigma^2 = 1$) but with different means $0, s, 2s$, and $3s$, respectively. The signal-to-noise power ratio (SNR) of observations at each local

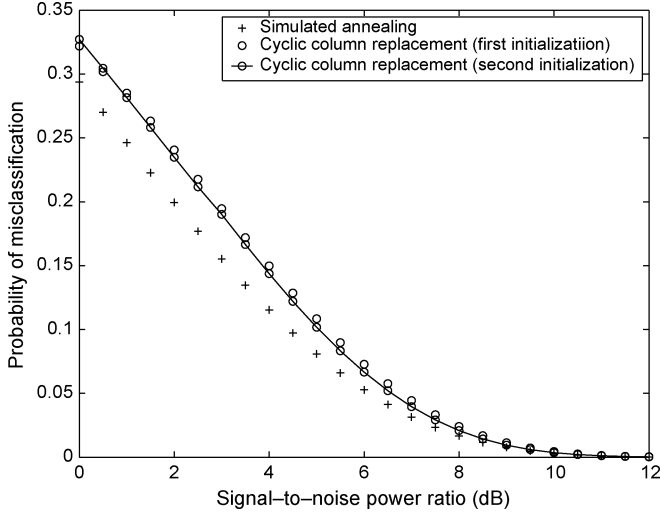


Fig. 2. Performance comparison of 4×10 code matrices designed at 0 dB by the cyclic column replacement approach (both initial strategies) and simulated annealing.

sensor is $20 \log_{10} s$ since the variance of the noise is equal to one. We assume that data transmission is over ideal channels in this example. The algorithms based on the cyclic column replacement approach and simulated annealing were started with two different initial code matrices given in Code Matrix I and

Code Matrix II : (7, 14, 9, 12, 7, 10, 14, 11, 13, 13)

respectively. It is easy to see that the minimum Hamming distance between any pair of the codewords in both matrices is 5. This value is set as the minimum Hamming distance requirement in both algorithms. We then searched the optimal code matrix when $\text{SNR} = 0$ dB. The optimal local decision rules were computed via the Gauss–Seidel algorithm. Probability of misclassification as shown in (1) was computed by numerical integration. The code matrices obtained from the cyclic column replacement approach with these two initial code matrices are

Code Matrix III : (3, 6, 5, 14, 3, 8, 8, 1, 2, 11)

and

Code Matrix IV : (9, 1, 9, 12, 7, 5, 1, 4, 2, 2)

respectively. Since the code matrix obtained from simulated annealing with both initial code matrices is the same, we only present one result, namely

Code Matrix V : (6, 12, 4, 9, 12, 9, 12, 6, 1, 3).

The probability of misclassification for each matrix is computed and plotted in Fig. 2. One can observe that the performance of the cyclic column replacement approach is dependent on its initial code matrix. The performance of the code matrix obtained by simulated annealing is better than that obtained by the cyclic column replacement approach at all SNR values considered even though the code matrices are obtained at $\text{SNR} = 0$ dB.

V. PERFORMANCE ANALYSIS

In this section, we give the asymptotic performance analysis for the proposed DCFECC approach. We provide the results for both fault-free and faulty situations.

A. Decision Error Under Fault-Free Situation

In the following derivations, we assume that an identical local decision rule is employed at each sensor. Under this assumption, the probability of making the decision H_j given that H_i is the true hypothesis is denoted by $P(H_j|H_i) = h_{j,i}$ for each sensor. Assume independence across sensor observations. Also, let $d_{\min} = \min_{i \neq j} d_H(\mathbf{t}_i, \mathbf{t}_j)$. Let $\delta_N = (1/N)[(d_{\min} - 1)/2]$, and $\beta_i = 1 - h_{i,i}$. The probability of correct detection can be expressed as

$$\begin{aligned}
 P_{\text{correct}} &= \frac{1}{M} \sum_{i=0}^{M-1} P(\text{decision} = H_i | H_i) \\
 &\stackrel{(a)}{\geq} \frac{1}{M} \sum_{i=0}^{M-1} P\left(d_H(\mathbf{u}, \mathbf{t}_i) \leq \left\lfloor \frac{(d_{\min} - 1)}{2} \right\rfloor \middle| H_i\right) \\
 &\stackrel{(b)}{\geq} \frac{1}{M} \sum_{i=0}^{M-1} P\left(\text{At most } \left\lfloor \frac{(d_{\min} - 1)}{2} \right\rfloor \text{ sensors} \right. \\
 &\quad \left. \text{assume wrong hypothesis} | H_i\right) \\
 &= \frac{1}{M} \sum_{i=0}^{M-1} \sum_{k=0}^{N\delta_N} \binom{N}{k} (1 - h_{i,i})^k h_{i,i}^{N-k} \\
 &\stackrel{(c)}{\geq} \frac{1}{M} \sum_{i=0}^{M-1} \left(1 - e^{-ND(\delta_N \| \beta_i)}\right)
 \end{aligned}$$

where (a) follows from the fact that d_{\min} is the minimum Hamming distance between any pair of codewords in the code matrix, (b) follows from the fact that the sensor could decide on wrong hypothesis but the output (1 or 0) from the sensor may still be correct, (c) follows since we assume² $\max_{0 \leq i \leq M-1} \beta_i < \delta_N$, and $D(\delta_N \| \beta_i)$ is the binary Kullback–Leibler divergence of δ_N and β_i . Thus, the probability of decision error can be expressed as

$$P_e \leq \frac{1}{M} \sum_{i=0}^{M-1} e^{-ND(\delta_N \| \beta_i)} \leq e^{-N \min_{0 \leq i \leq M-1} D(\delta_N \| \beta_i)}$$

which implies that given that $\max_{0 \leq i \leq M-1} \beta_i < \delta_N$

$$-\frac{1}{N} \log P_e \geq \min_{0 \leq i \leq M-1} D(\delta_N \| \beta_i).$$

²For binary independent identically distributed (i.i.d.) variables $\{X_j\}_{j=1}^{\infty}$ with $\Pr(X_j = 1) = 1 - \Pr(X_j = 0) = \beta < \delta$,

$$\begin{aligned}
 &\Pr\{X_1 + X_2 + \dots + X_N \geq N\delta\} \\
 &= \Pr\{e^{s(X_1 + X_2 + \dots + X_N)} \geq e^{Ns\delta}\}, \quad \text{where } s > 0 \\
 &\leq \frac{E^N[e^{sX}]}{e^{Ns\delta}} \\
 &\leq (\exp\{-[s\delta - \log(1 - \beta + \beta e^s)]\})^N.
 \end{aligned}$$

Taking $s = \log(\delta(1 - \beta)/\beta(1 - \delta)) > 0$ yields

$$\Pr\{X_1 + \dots + X_N \geq N\delta\} \leq \exp\{-ND(\delta \| \beta)\}.$$

Now, suppose $\delta_N \rightarrow \delta$. We can conclude that under the condition of $\delta > \max_{0 \leq i \leq M-1} \beta_i$

$$\liminf_{N \rightarrow \infty} -\frac{1}{N} \log P_e \geq \min_{0 \leq i \leq M-1} D(\delta \parallel \beta_i) > 0.$$

Hence, as long as d_{\min} satisfies $\delta > \max_{0 \leq i \leq M-1} \beta_i$, the probability of decision error for DCFECC approaches zero as N goes to infinity.

B. Decision Error With Possible Faulty Sensors

Assume that there are μ mis-behaved sensors that might be due to damaged sensors. Then the resultant new \bar{d}_{\min} will satisfy

$$d_{\min} \geq \bar{d}_{\min} \geq d_{\min} - \mu.$$

Thus, using the same notations

$$\delta_N \geq \bar{\delta}_N \geq \delta_N - \nu_N$$

where $\nu_N = (1/N)\lceil \mu/2 \rceil$. By assuming that $\nu_N \rightarrow \nu$, we have

$$\delta \geq \limsup_N \bar{\delta}_N \geq \liminf_N \bar{\delta}_N \geq \delta - \nu.$$

Consequently, if

$$\delta - \nu > \max_{0 \leq i \leq M-1} \beta_i$$

the probability of decision error for DCFECC goes to zero as N goes to infinity. An operational implication of the above inequality is as follows. As long as the largest minimum pairwise Hamming distance (i.e., d_{\min}) asymptotically beats the number of misbehaved sensors (i.e., μ) plus twice of the expected number of sensors that give wrong local classifications (i.e., $2N \cdot \max_{0 \leq i \leq M-1} \Pr[\text{decision} \neq H_i | H_i]$), the DCFECC approach can provide vanishing global decision error.

VI. PERFORMANCE EVALUATION IN THE PRESENCE OF FAULTS

In this section, we evaluate the performance of the DCFECC approach in the presence of deterministic faults such as stuck-at faults and random faults such as channel transmission errors. Unlike the stuck-at faults in logic circuits, we assume that the sensors with stuck-at faults always send 1 or 0 decisions to the fusion center. We also give an example to show the effect of the minimum Hamming distance on the fault-tolerance capability. Note that all the performance evaluations in this paper are based on analytical results. Performance in terms of probability of misclassification is computed using (1). Numerical methods are used to obtain integration values. The problem considered in the following examples is defined as follows.

- Four equally likely hypotheses H_0, H_1, H_2, H_3 are to be identified.
- All the sensor measurements are identically distributed.
- The probability density function for each hypothesis is the same as that given in Example 1.
- For each SNR value, the Gauss–Seidel algorithm is used to compute the optimal local decision rules of the DCFECC approach, as well as the optimal local decision rules and the Chair–Varshney fusion rule of both CA and FCA approaches. CA and FCA will be defined later.

Example 2: In this example, stuck-at faults at local sensors are considered and data transmission is assumed to be over ideal

channels. When a stuck-at fault occurs at sensor k , the local decision u_k^* of sensor k is either always 1 or always 0. Consequently, if sensor k always sends 1 to the local fusion center, the conditional probabilities $P(u_k^* = 1 | H_\ell)$, $\ell = 0, 1, 2, 3$, become 1 regardless of the hypothesis present. Similarly, the conditional probabilities $P(u_k^* = 0 | H_\ell)$ become 1 when the faulty sensor always sends 0 to the local fusion center. Since we assume an ideal channel, $P(u_k^* = 0 | H_\ell)$ is equivalent to $P(u_k = 0 | H_\ell)$ in this example.

A system with a fusion center and seven local sensors is considered in this example. In the following, the performance of the DCFECC approach is evaluated and it is compared with that of the 1-bit CA in both the fault-free and faulty situations. By the 1-bit CA, we mean distributed M -ary detection based on binary local decisions received at the fusion center that employs the Chair–Varshney fusion rule [17]. That is, the local sensors send their one bit information to the fusion center, and the fusion center fuses the binary decision vector into a multiclass decision by

$$\text{assigning } H_\omega \text{ to } u_0 \text{ if } \omega = \arg \max_{0 \leq \ell \leq M-1} P(\mathbf{u} | H_\ell) P_\ell. \quad (4)$$

From (4), we can see that the Chair–Varshney fusion rule is the MAP fusion rule and is optimal when the local decision rules are given. It can be obtained by extending the binary hypotheses version given in [7]. Note that the CA is different from the approaches used in [20] and [21], where an identical local decision rule is used for each sensor. For the parallel fusion network, the use of identical local decision rules is optimal based on error exponent only in an asymptotical sense when the number of sensors becomes arbitrarily large, i.e., the error exponents of the identical local decision rules and the nonidentical local decision rules obtained by system-wide optimization are equal. In general, however, system-wide optimization will result in nonidentical local decision rules for which readers may refer to [22] for more details. An example that demonstrates that identical local classification rule is only suboptimal in the sense of global decision error can be found in [22, Appendix B]. In [20], the authors show that identical local decision rules are also asymptotically optimal in the sense of error exponent when the system is subject to a power constraint.

In this example, the code matrix designed by simulated annealing is chosen for the DCFECC approach and is given by

$$\text{Code Matrix VI : } (3, 8, 14, 12, 9, 12, 9).$$

It is easy to see that the minimum Hamming distance between any two codewords in this code matrix is 3.

In this example, we also let SNR range from 0 to 12 dB with step size equal to 0.5 dB. Since the Gauss–Seidel algorithm may converge to a local minimum, depending on the initial values of $P(u_k | H_\ell)$, where $k = 1, \dots, 7$ and $\ell = 0, 1, 2, 3$, two initialization strategies are considered. In the first strategy, we initialize the algorithm with identical decision rules at all the local sensors. The identical decision rule was found by considering only an isolated sensor where the isolated optimum decision rule is for the classification of $H_k^1 = \{H_0, H_1\}$ and $H_k^0 = \{H_2, H_3\}$. In the second strategy, we initialize the algo-

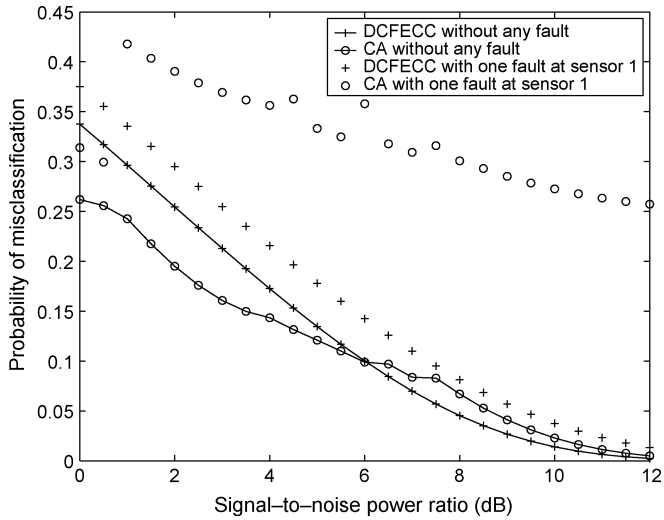


Fig. 3. Performance comparison of DCFECC and the CA when one fault is present and the first initialization strategy is employed.

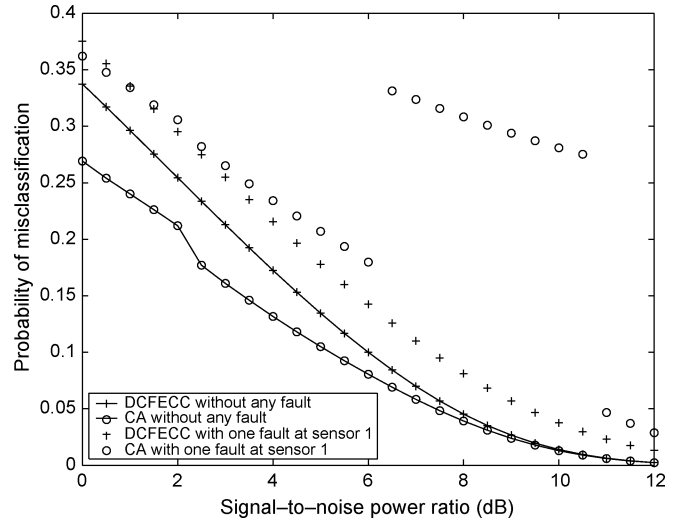


Fig. 5. Performance comparison of DCFECC and the CA when one fault is present and the second initialization strategy is employed.

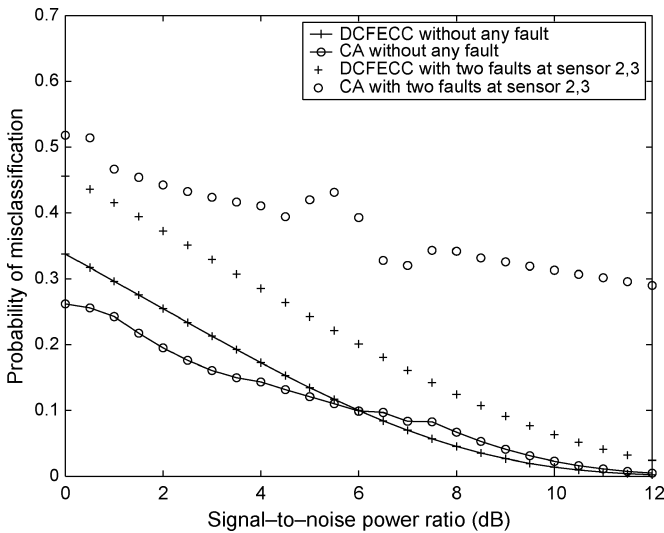


Fig. 4. Performance comparison of DCFECC and the CA when two faults are present and the first initialization strategy is employed.

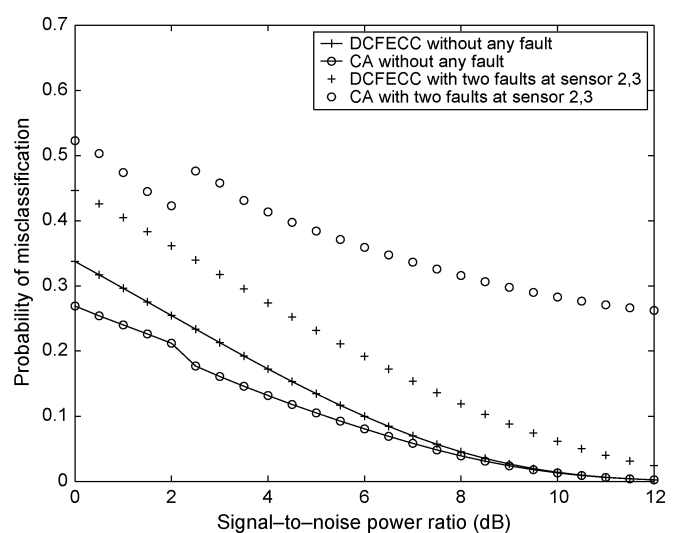


Fig. 6. Performance comparison of DCFECC and the CA when two faults are present and the second initialization strategy is employed.

gorithm with the probabilities $P(u_k|H_\ell)$ when all the local sensors operate independently with their optimum decision rule for classification corresponding to the columns of the code matrix designed for DCFECC.

We assume that the faulty sensors always send the decision 1 to the fusion center. The performance with one faulty sensor ($k = 1$) and two faulty sensors ($k = 2, 3$) are shown in Figs. 3 and 4, respectively, when the first initialization strategy is employed. The performance of fault-free situation is also included for comparison. Performance results with one faulty sensor ($k = 1$) and two faulty sensors ($k = 2, 3$) are presented in Figs. 5 and 6 when the second initialization strategy is employed. Again, the performance with no faults is also included.

There are several observations that can be made from these figures. First, from Fig. 3, the performance of DCFECC in the

presence of one fault is better than that of the CA at all SNR ranges except when SNR is less than 0.5 dB. However, as shown in Fig. 4, the performance of DCFECC in the presence of two faults is better than that of the CA at all SNR ranges considered. A possible reason for this unusual situation is that the error resulting from one faulty sensor is smaller than the error resulting from the overall decision error of the network at very low SNRs. That is, the error resulting from the overall decision error of the network dominates the system performance at very low SNRs when there is only one faulty sensor. Second, from Figs. 3 and 4, the performance of DCFECC without any fault, when the first initialization strategy is employed, is worse than the CA, when SNR is less than 6.0 dB. This is obvious because the CA employs the Chair–Varshney fusion rule. However, the performance of DCFECC is slightly better than that of the CA when SNR is larger than 6.0 dB.

This situation is caused by two facts. One is that the thresholds of the local decision rules and fusion rules determined by the Gauss–Seidel algorithm are not the global optimum and depend on the initial conditions given to the algorithm. The other reason is that when the cardinality of the local decisions is less than the number of hypotheses and all local decision rules are identical, the system performance starts to become worse when the SNR grows over a critical point. This surprising phenomenon was also observed in [10]. We can see as SNR becomes very large, certain local decision probabilities, e.g., $P(u_j = 1|H_0)$ and $P(u_j = 1|H_1)$, $j = 1, \dots, N$, become closer in value. Therefore, it becomes harder to distinguish between hypotheses H_0 and H_1 at all local sensors. One can see from Figs. 5 and 6 that this phenomenon does not happen with the second initialization strategy, since, under nonidentical local decision rules, some sensors can have good performance in distinguishing particular hypotheses sets, even when other sensors have poor performance. As for the third observation, the performance of the CA in the presence of one fault is improved when the second initialization strategy is employed. This is evident in Figs. 3 and 5. However, it is still worse than that of the proposed DCFECC approach at most of the SNR values. Finally, the misclassification probability of the proposed DCFECC approach decreases when SNR increases for both fault-free and faulty situations. On the other hand, the performance of the CA becomes unpredictable when any fault occurs in the system. Thus, the performance of the DCFECC approach is more predictable and graceful in the presence of faults.

For the implementation of the Chair–Varshney fusion rule, as expressed in (4), $N \times M$ real parameters, namely, $P(u_k|H_\ell)$, $k = 1, \dots, N$, $\ell = 0, \dots, M - 1$ must be stored at the fusion center to perform the decision. This needs more space as compared with the proposed fault-tolerant fusion rule, which only requires $N \times M$ integers (binary code matrix) to be stored. Moreover, performing the Chair–Varshney fusion rule requires real number operations. This results in more bit computations as compared with the fault-tolerant fusion rule, whose computations involve only integer (fewer bits) computations.

Example 3: In this example, we evaluate the performance of the DCFECC approach in the presence of channel errors. Ten local sensors and one fusion center are used. Code Matrix V is again used in this example. For performance comparison purposes, we design the fault-tolerant system with binary local decisions and employ the Chair–Varshney fusion rule obtained by extending the design of binary detection systems with channel errors [15]. We call this the one bit fault-tolerant conventional approach (FCA).

In this example, we assume that $p_{1j} = p_{0j} = p_j$, and design both the DCFECC and FCA at 5 dB SNR, while assuming that the probabilities of channel transmission errors, p_j , are 0.01, 0.05, 0.1, and 0.25 for $j = 1, \dots, N$. In order to investigate the robustness of the two approaches, we evaluate their performances when the channel transmission error probability, p_{ce} , is varied from 0 to 0.3. For the FCA, we only provide the results for the second initialization strategy mentioned in Example 2. The FCA with the first initialization strategy performs much worse than the DCFECC, and the result is omitted

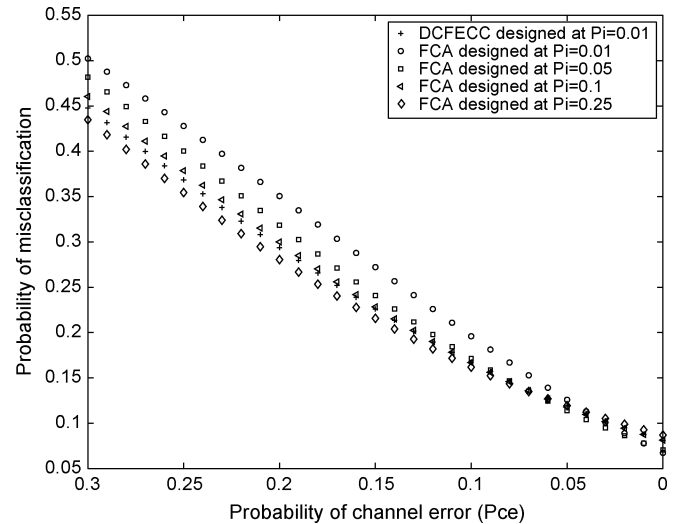


Fig. 7. Performance comparison of DCFECC and FCA when channel transmission errors are present and the second initialization strategy is employed. Both approaches are designed at 5 dB SNR and $p_j = 0.01, 0.05, 0.1$, and 0.25.

here due to the space limitation. The performance comparison of DCFECC and FCA is provided in Fig. 7. Since we found that the performances for the DCFECC approach under these four channel transmission error probabilities are very close, we only plot one result in Fig. 7. This phenomenon indicates that employing the fault-tolerant fusion rule has already captured most of the effect of channel errors. As illustrated in Fig. 7, the performance of DCFECC is better than those of the FCA designed at $p_j = 0.01$ and 0.05, when $p_{ce} \geq 0.05$. However, when the FCA is designed for a higher probability of channel error, the robustness is close to the DCFECC approach. We see the performance of the FCA designed at $p_j = 0.1$ is almost the same as the performance of DCFECC. When FCA is designed at $p_j = 0.25$, the performance is better than that of DCFECC when $p_{ce} > 0.05$ and is slightly worse than that of DCFECC when $p_{ce} < 0.05$.

From the results of the previous examples, we see the DCFECC can tolerate many types of faults.

Example 4: In this example, we investigate the effect of the minimum Hamming distance on the fault-tolerance capability of the DCFECC approach. Ten local sensors and one fusion center are used. Three code matrices, with different minimum Hamming distance are designed in this example for comparison purposes, and Code Matrix V and the two other matrices are given by

$$\text{Code Matrix VII : } (6, 14, 4, 9, 12, 9, 12, 6, 1, 3)$$

and

$$\text{Code Matrix VIII : } (7, 14, 4, 9, 12, 9, 12, 7, 1, 1).$$

One can see that the minimum Hamming distance in these code matrices are 5, 4, and 3, respectively.

The performance comparison of systems employing these three code matrices is provided in Fig. 8. As the results in this figure show, the performances of the system employing the

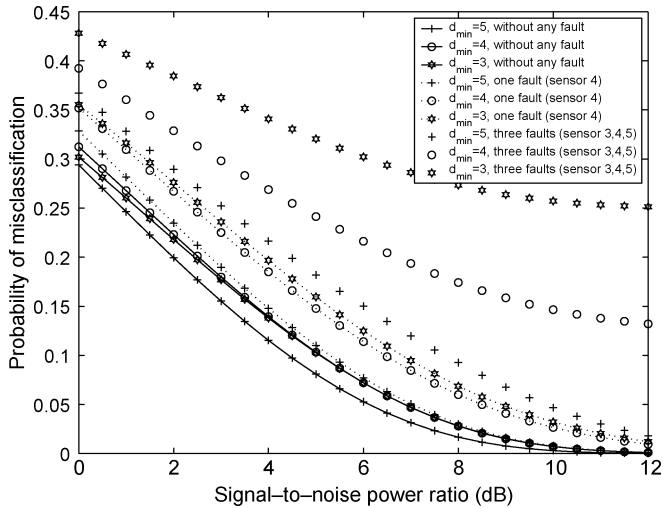


Fig. 8. Performance comparison of DCFECC employing three code matrices with different minimum Hamming distances.

three different code matrices in the fault-free case are only slightly different. However, when one fault occurs at sensor 4, the performance of the system employing the code matrix with minimum Hamming distance 5 becomes better than those employing code matrices with minimum Hamming distances 3 and 4. When three faults occur at sensors 3, 4, and 5, these performance differences become much more apparent. The system employing the code matrix with the largest minimum Hamming distance performs much better than the others, while the system employing the code matrix with the smallest minimum Hamming distance performs the worst. Therefore, the system employing a code matrix with larger minimum Hamming distance is able to improve the fault-tolerance capability against more faults.

VII. SUMMARY AND CONCLUSION

In this paper, the problem of fault-tolerant distributed classification in wireless sensor networks was considered. The proposed DCFECC approach is based on the theory of error-correcting codes and is applicable to harsh environments. Based on the fault-tolerant fusion rule, the local decision rules for each sensor were derived. Taking advantage of the characteristics of the error correcting fusion rule, the system provides fault-tolerance capability. Computational complexity and memory requirements are also reduced due to the simplification of the decoding rules required for fusion processing.

In order to achieve good performance, we have developed two efficient algorithms to search for good code matrices for implementing the DCFECC approach. The cyclic column replacement approach is usually fast but may converge to a local optimum depending on the chosen initial code matrix. On the other hand, the simulated annealing approach is robust to the selection of the initial code matrix, and has better performance even though it takes more time to converge.

The conditions for asymptotic detection of the correct hypothesis by the DCFECC were also developed to show the relationship between the minimum Hamming distance of the employed code matrix and the fault-tolerance capability.

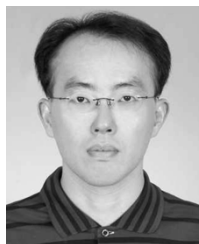
Finally, performance comparisons of the DCFECC and conventional approaches were provided by numerical examples. The results showed that the former outperforms the latter in the presence of faults.

ACKNOWLEDGMENT

The authors would like to thank Prof. A. M. Sayeed for his helpful suggestions during the review process. They would also like to thank the anonymous reviewers for their valuable comments.

REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Commun. Mag.*, vol. 40, no. 8, pp. 102–114, Aug. 2002.
- [2] L. Dan, K. D. Wong, H. H. Yu, and A. M. Sayeed, "Detection, classification, and tracking of targets," *IEEE Signal Process. Mag.*, vol. 19, pp. 17–29, Mar. 2002.
- [3] H. Wang, J. Elson, L. Girod, D. Estrin, and K. Yao, "Target classification and localization in habitat monitoring," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2003)*, vol. 4, Hong Kong, China, Apr. 2003, pp. IV-844–7–29.
- [4] A. D'Costa and A. M. Sayeed, "Data versus decision fusion in sensor networks," presented at the IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP 2003), Hong Kong, China, Apr. 2003.
- [5] —, "Data versus decision fusion for classification in sensor networks," in *Proc. 6th Int. Conf. Inf. Fusion*, Jul. 2003, pp. 889–895.
- [6] S. A. Aldosari and J. M. F. Moura, "Detection in decentralized sensor networks," in *IEEE Int. Conf. Acoust., Speech, Signal Process.*, Montreal, Canada, May 2004, pp. 277–280.
- [7] P. K. Varshney, *Distributed Detection and Data Fusion*. New York: Springer, 1997.
- [8] F. A. Sadjadi, "Hypotheses testing in a distributed environment," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 22, no. 2, pp. 134–137, Mar. 1986.
- [9] B. Dasarthy, "Operationally efficient architectures for fusion of binary-decision sensors in multidetection environments," *Opt. Eng.*, vol. 36, no. 3, pp. 632–641, Mar. 1997.
- [10] X. Zhu, M. Kam, and C. Corres, "M-ary hypothesis testing with binary local decisions," in *Proc. Conf. Inf. Sci. Syst.*, Mar. 1998, pp. 107–112.
- [11] Q. Zhang and P. K. Varshney, "Decentralized M-ary detection via hierarchical binary decision fusion," *Inf. Fusion*, vol. 2, pp. 3–16, 2001.
- [12] G. G. L. Meyer and H. L. Weinert, "On the design of fault-tolerant signal detectors," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, no. 4, pp. 973–978, Aug. 1986.
- [13] A. R. Reibman and L. W. Nolte, "Optimal fault-tolerant signal detection," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 1, pp. 179–180, Jan. 1990.
- [14] —, "Optimal design and performance of distributed signal detection systems with faults," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 10, pp. 1771–1782, Oct. 1990.
- [15] S. C. Thomopoulos and L. Zhang, "Distributed decision fusion in the presence of networking delays and channel errors," *Inf. Sci.*, vol. 66, no. 1–2, pp. 91–118, Dec. 1992.
- [16] R. E. V. Dyck and L. E. Miller, "Distributed sensor processing over an ad hoc wireless network: Simulation framework and performance criteria," in *Proc. IEEE Military Commun. Conf.*, vol. 2, McLean, VA, Oct. 2001, pp. 28–31.
- [17] Z. Chair and P. K. Varshney, "Optimal data fusion in multiple sensor detection systems," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-22, pp. 98–101, Jan. 1986.
- [18] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. New York: Elsevier, 1977.
- [19] A. E. Gamal, L. A. Hemachandra, I. Shperling, and V. K. Wei, "Using simulated annealing to design good codes," *IEEE Trans. Inf. Theory*, vol. IT-33, no. 1, pp. 116–123, Jan. 1987.
- [20] J.-F. Chamberland and V. V. Veeravalli, "Asymptotic results for decentralized detection in power constrained wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 1007–1015, Aug. 2004.
- [21] A. D'Costa, V. Ramachandran, and A. M. Sayeed, "Distributed classification of Gaussian space-time sources in wireless sensor networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 6, pp. 1026–1036, Aug. 2004.
- [22] P.-N. Chen and A. Papamarcou, "New asymptotic results in parallel distributed detection," *IEEE Trans. Inf. Theory*, vol. 39, no. 6, pp. 1847–1863, Nov. 1993.



Tsang-Yi Wang (S'01–M'04) received the B.A. and B.S. degrees from the National Sun Yat-Sen University, Kashiung, Taiwan, in 1994 and 1996, respectively, and the Ph.D. degree in electrical engineering from Syracuse University, Syracuse, NY, in 2003.

He joined the National Chi Nan University, Nantou, Taiwan, in 2004, where he is currently an Assistant Professor in the Graduate Institute of Communication Engineering. His research interests are in wireless sensor networks, distributed detection,

and wireless communication.



Yunghsiang S. Han (S'90–M'93) was born in Taipei, Taiwan, on April 24, 1962. He received the B.S. and M.S. degrees in electrical engineering from the National Tsing-Hua University, Hsinchu, Taiwan, in 1984 and 1986, respectively, and the Ph.D. degree from the School of Computer and Information Science, Syracuse University, Syracuse, NY, in 1993.

From 1993 to 1997, he was an Associate Professor in the Department of Electronic Engineering, Hua Fan College of Humanities and Technology, Taipei,

Taiwan. From 1997 to 2004, he was with the Department of Computer Science and Information Engineering, National Chi Nan University, Nantou, Taiwan. He was promoted to Full Professor in 1998. From June 2001 to October 2001, he was a Visiting Scholar in the Department of Electrical Engineering, University of Hawaii, Manoa, and from September 2002 to January 2004, he was the SUPRIA Visiting Research Scholar in the Department of Electrical Engineering and Computer Science and CASE Center, Syracuse University. He is now with the Graduate Institute of Communication Engineering, National Taipei University, Taipei, Taiwan. His research interests are in wireless networks, security, and error-control coding.

Dr. Han received the 1994 Syracuse University Doctoral Prize.



Pramod K. Varshney (F'97) was born in Allahabad, India, on July 1, 1952. He received the B.S. degree in electrical engineering and computer science (with highest honors), and the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois at Urbana-Champaign in 1972, 1974, and 1976, respectively.

Since 1976, he has been with Syracuse University, Syracuse, NY, where he is currently a Professor of Electrical Engineering and Computer Science and the Research Director of the New York State Center for

Advanced Technology in Computer Applications and Software Engineering. He has supervised 34 Ph.D. dissertations and has published extensively. His current research interests are in distributed sensor networks and data fusion, detection and estimation theory, wireless communications, image processing, and radar signal processing.

Dr. Varshney was a James Scholar, a Bronze Tablet Senior, and a Fellow, while at the University of Illinois. He is a member of Tau Beta Pi. He is the recipient of the 1981 ASEE Dow Outstanding Young Faculty Award. In 2000, he received the Third Millennium Medal from the IEEE and Chancellor's Citation for Exceptional Academic Achievement at Syracuse University. He was elected to the grade of Fellow of the IEEE in 1997 for his contributions in the area of distributed detection and data fusion. He serves as a Distinguished Lecturer for the Audio Engineering Society (AES) of the IEEE. He was the President of International Society of Information Fusion during 2001.



Po-Ning Chen (S'93–M'95–SM'01) was born in Taipei, Taiwan, in 1963. He received the B.S. and M.S. degrees in electrical engineering from the National Tsing-Hua University, Hsinchu, Taiwan, in 1985 and 1987, respectively, and the Ph.D. degree in electrical engineering from University of Maryland, College Park, in 1994.

From 1985 to 1987, he was with the Image Processing Laboratory, National Tsing-Hua University, where he worked on the recognition of Chinese characters. During 1989, he was with Star Tech.

Inc., where he focused on the development of fingerprint recognition systems. In 1994, he joined Wan Ta Technology, Inc., as a Vice General Manager, conducting several projects on point-of-sale systems. In 1995, he became a Research Staff Member in the Advanced Technology Center, Computer and Communication Laboratory, Industrial Technology Research Institute, Taiwan, where he led a project on Java-based Network Managements. Since 1996, he has been an Associate Professor in the Department of Communications Engineering, National Chiao-Tung University, Hsinchu, Taiwan, and became a Full Professor in 2001. His research interests generally lie in information and coding theory, large deviation theory, and distributed detection.

Dr. Chen received the 2000 Young Scholar Paper Award from Academia Sinica, Taiwan.