Fusion in Sensor Networks with Communication Constraints*

Saeed A. Aldosari
saldosar@ece.cmu.edu

José M. F. Moura
moura@ece.cmu.edu

Department of Electrical and Computer Engineering
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213, USA

ABSTRACT

In this paper, we address the problem of optimizing the detection performance of sensor networks under communication constraints on the common access channel. Our work helps understanding tradeoffs between sensor network parameters like number of sensors, degree of quantization at each local sensor, and SNR. Traditionally, this problem is tackled using asymptotic assumptions on the number of sensors, an approach that leads to the abstraction of important details such as the structure of the fusion center. We adopt a non-asymptotic approach and optimize both, the sensing and the fusion sides with respect to the probability of detection error. We show that the optimal fusion rule has an interesting structure similar to the majority-voting rule. In addition, we study the convergence with respect to the number of sensors of the performance of the fusion rule. We show that convergence is SNR dependent and that, in low-SNR environments, asymptotics may require a large number of sensors.

Categories and Subject Descriptors

C.2.4 [**Computer Communication Networks**]: Distributed Systems—*Distributed applications*; E.4 [**Coding and Information Theory**]: Data compaction and compression,; H.1.1 [**Models and Prenciples**]: Systems and Information Theory—*Information theory*; C.2.1 [**Computer Communication Networks**]: Network Architecture and Design—*Distributed networks, Wireless communication*

General Terms

Design, Algorithm, Performance, Theory

Keywords

Decision Fusion, Decentralized Detection, Sensor Networks, Genetic Algorithm, Quantization

1. INTRODUCTION

Large-scale reconnaissance and surveillance through networks of geographically dispersed autonomous sensors is becoming a reality due to the levels of integration offered by technology. Distributed sensing is faced with many challenges pertaining to the scarcity of power, bandwidth, and computing resources. Effective design under such resource limitations is often a matter of compromise; a balance between competing goals and objectives. We focus here on the detection performance of sensor networks operating under a global rate constraint imposed by their common access communications channel.

Fundamental results on distributed detection go back to the early work of Tenney and Sandell [14]. Interested readers are referred to [20] as well as the book by Varshney [19] for an introduction and overview to the area of decentralized detection. Faced with many obstacles, most of the literature in this field resort to asymptotic assumptions and information-theoretic performance measures to simplify the analysis and design of sensor networks [15]-[3]. Analysis based on asymptotics leads to the abstraction of important details of the problem such as the structure of the fusion rule. Although there are few studies that avoid using asymptotic assumptions [13],[19], most of them are limited to simple networks and they fall short of providing insight into the structure of optimal fusion rules.

We consider here a specific architecture for the sensor network, a parallel architecture, see Fig. 1, where there is no communication among the local sensors, and the local detectors feed their quantized decisions to a single fusion center. We address the problem of optimizing the detection performance of such sensor networks under communication constraints. In particular, we concentrate on the design of the fusion center and the tradeoffs among parameters of interest like the number of sensors, how many bits per local decision, and SNR. Understanding these tradeoffs is important since it provides answers to important questions including the following: how many sensors of a particular type should be deployed in certain environments and whether or not it is advisable to use hard versus soft detectors. We adopt a non-asymptotic approach and develop algorithms for optimizing

both, the local sensors and the fusion center with respect to the exact probability of detection error. We show that optimal fusion rules have an interesting structure similar to the majority voting rule. We establish, under a channel rate constraint, when should we opt for fewer higher quality sensors (more bits per sensor) rather than more lower quality sensors (fewer bits per sensor). Also, we study the convergence of performance results towards those obtained under asymptotic assumptions, when the number of sensors $N \to \infty$, and show that conclusions derived from asymptotic analysis may be very different from the actual conclusions derived for networks with a given number of sensors. This is much more so at low-SNR environments, most likely the condition under which many such sensor networks will operate.

The remaining of the paper is organized as follows. In section 2, we present the model of parallel fusion networks and state the problems. Algorithms for optimizing the local detectors and the fusion rule are presented in section 3. Section 4 contains the optimization results along with discussions regarding the structure of the fusion center, the trade-offs between network parameters, and convergence toward the asymptotes. Finally, we present concluding comments in section 4.

2. MODEL AND PROBLEM STATEMENT

We consider here the following distributed detection problem for sensor networks: N sensors gather T measurements $y_{n,t}$ per sensor n, make a local decision $u_{n,t}$ per measurement and send the decisions $u_{n,t}$ to a single fusion center γ_0 through an error-free multiple-access channel (MAC) as shown in Fig. 1. This particular architecture is often referred to as a parallel fusion network [19]. We consider this problem when the MAC has a rate constraint R. The fusion center makes a global decision $\tilde{H}$ about the true state of nature H based on the collection of the local decisions gathered from all sensors. In the version of the problem we consider here explicitly, the fusion center does not sense measurements directly.

We cast the problem as a binary detection problem with hypotheses H_0 and H_1 with known prior probabilities π_0 and π_1, respectively. In addition, we assume that the observations $\{y_{n,t} : n = 1, 2, \ldots, N, t = 1, 2, \ldots, T\}$ are, conditioned on H, independent and identically distributed with conditional densities $f_0(y) = f(y|H_0)$ and $f_1(y) = f(y|H_1)$.

The spatio-temporal conditional independence assumption causes the space and time to be indistinguishable. Under such assumption, the probability of error $P_e(T, N, b, \boldsymbol{\lambda}, \gamma_0)$ is dependent on the total number of samples NT rather than the individual values of N and T. Effectively, this makes a sensor network with N sensors gathering T measurements per sensor equivalent to having NT sensors gathering single measurements each as long as the fusion rules in both cases are equivalent. Therefore, without loss of generality, we can assume that $T = 1$ and drop the dependence on T.

Since the fusion center makes the final decision, the output of the fusion rule γ_0 is binary, i.e., either H_0 or H_1. Local sensors, on the other hand, are not restricted to binary outputs: each sensor classifies each measurement $y_n \in \mathcal{Y}$ into $L = 2^b$ classes, where b is the number of transmitted bits per sensor per measurement. We can think of each classifier as a mapping from the observation space $\mathcal{Y}$ to the classification space $\mathcal{U}$, i.e., $\gamma_n : \mathcal{Y} \longrightarrow \mathcal{U}$. Similarly, the fusion rule maps
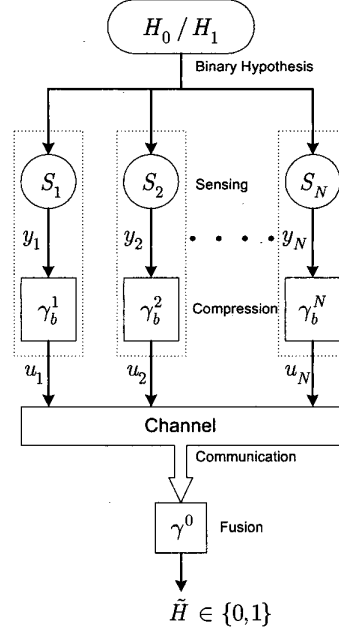

Figure 1: Parallel fusion network.

NT local decisions $\{u_n : n = 1, 2, \ldots, N\}$ into one of two classes, i.e., $\gamma_0 : \mathcal{Y}^N \longrightarrow \{0, 1\}$.

We address the following main issues:

1. Structure of the fusion rule that optimizes the probability of detection error $P_e = \mathrm{Pr}(\tilde{H} \neq H)$.

2. Tradeoffs between sensor network parameters such as the number of sensors N, number of bits per sensor b, and SNR.

3. Convergence of the error decay rate towards the asymptotes as $N \to \infty$.

3. OPTIMIZATION

In its general form, the problem of optimizing decentralized sensor networks is complex and computationally expensive due to the fact that the optimization is performed over all possible local classification rules $\gamma_n \in \Gamma$ and all possible fusion rules $\gamma_0 \in \Gamma_0$. When a discrete observation space is assumed, the resulting optimization problem is NP-complete. The problem can not be any easier if we consider a continuous observation space [17]. The assumptions introduced in the previous section yield considerable simplifications, which enable us to formulate a simpler version of the optimization problem. In particular, the conditional independence assumption simplifies the problem greatly since, in this case, optimal local classifiers are likelihood ratio tests characterized by a finite number of thresholds. The problem can be simplified further by assuming that the likelihood ratio $f_1(y)/f_0(y)$ is monotonic in y [10], in which case, we are allowed to quantize the measurements directly rather than their likelihood ratios. It was shown in [5] that, when the observations are Gaussian, at most $L(L-1)/2$ quantization

thresholds per local sensor are required in order to preserve global optimality of the sensor network. It is easy to show that the latter statement holds for any other distribution as long as the measurements are conditionally independent. Numerical results that we conducted for $b = 2$ on the asymptotic regime $(N \to \infty)$ show that optimizing a network with $L(L-1)/2$ thresholds per local sensor always converges to a simpler one having only $L - 1$ thresholds per local sensor. For this reason, and in order to reduce the computational complexity, we assume that the local quantizers are characterized by $L - 1$ thresholds as follows

$$
u_n = \begin{cases} 0 & \text{if } y_n \leq \lambda_{n,1} \\ 1 & \text{if } \lambda_{n,1} < y_n \leq \lambda_{n,2} \\ \vdots & \vdots \\ L-1 & \text{if } y_n > \lambda_{n,L-1} \end{cases}, \quad (1)
$$

where, y_n is the local measurement at the nth sensor, u_n is the corresponding local decision, and $\lambda_{n,1}, \lambda_{n,2}, \ldots, \lambda_{n,L-1}$, are the $L-1$ quantization thresholds of that sensor. In summary, for a fixed fusion rule γ_0, the optimization problem amounts to finding the set of optimal thresholds $\boldsymbol{\lambda} = \{\lambda_{n,i}, n = 1, 2, \ldots N, i = 1, 2, \ldots L - 1\}$ such that the average probability of detection error $P_e(N, b, \boldsymbol{\lambda}, \gamma_0)$ is minimized.

The structure of the fusion rule plays a crucial role regarding the overall performance of the sensor network since the fusion center makes the final decision about the state of the environment. While a few bad sensors might not greatly impact the overall performance, a badly designed fusion rule can lead to a poor performance even if the local detectors are well designed. Under the conditional independence assumption, the optimal fusion rule is known to be a likelihood ratio test on the probability mass functions (pmf) of the local decisions [19]. However, the coupling between the fusion rule and the local classification rules reduces the importance of this fact since the optimal thresholds of the local classifiers are not known *apriori*. This makes it necessary to consider all possible fusion rules when attempting to optimize the sensor network. One way to accomplish this is by searching the space of fusion rules and optimizing the local thresholds for each candidate rule. Other than for some simple cases (e.g., [19, pp.84-87]), the complexity of such an approach is prohibitive due to the exponential growth of the set of possible fusion rules with respect to the number of sensors N and the number of bits per sensor b.

Next, we consider the details of local thresholds and fusion rule optimization.

3.1 Representation of Fusion Rules

Before getting into details of the optimization algorithm, it is necessary to devise a way for representing fusion rules. We adopt a binary representation, which describes the output of the fusion rule under every possible combination of the local decisions. Since there are N sensors and each sensor classifies its measurement into L classes, a particular fusion rule should account for L^N local decision possibilities and, therefore, it can be represented as a string of L^N bits as follows

$$
h = \left(h_{L^N-1}\ h_{L^N-2}\ \cdots\ h_1\ h_0 \right),
$$

$$
h_q \in \{0, 1\}, \quad q = 0, 1, \ldots L^N - 1. \quad (2)
$$

Similarly, each possible combination of local decisions can

q	u_1	u_2	h_q
0 [0000]	0 [00]	0 [00]	h_0
1 [0001]	0 [00]	1 [01]	h_1
2 [0010]	0 [00]	2 [10]	h_2
3 [0011]	0 [00]	3 [11]	h_3
4 [0100]	1 [01]	0 [00]	h_4
16 [1111]	3 [11]	3 [11]	h_{15}

Figure 2: Fusion rule representation for a network of $N = 2$ sensors quantizing their measurements into $b = 2$ bits per measurement.

be represented by a vector of N integers

$$
\mathbf{u} = (u_1\ u_2\ \cdots\ u_N), \quad u_n \in \{0, 1, \ldots, L - 1\}. \quad (3)
$$

Assuming $L = 2^b$, $\mathbf{u}$ can be represented as a string of bN bits as follows

$$
\mathbf{u} = \left(u_1^1 u_1^2 \cdots u_1^b\ u_2^1 u_2^2 \cdots u_2^b\ \cdots\ u_N^1 u_N^2 \cdots u_N^b \right),
$$

$$
u_n^j \in \{0, 1\}. \quad (4)
$$

with representation (4), the space of all possible local decisions is spanned by a single bN-bit integer q ranging from 0 to $2^{bN} - 1$. For a particular combination of local decisions represented by an integer q, the individual values of the local decisions u_n, $n = 1, 2, \ldots, N$, can be extracted using a reverse mapping function $\Psi_n(q)$. This can be implemented by shifting the binary representation of q to the right by $b(N - n)$ bits then ANDing the result with $2^b - 1$. Alternatively, one could implement this using integer operations, $\Psi_n(q) = (q/2^{b(N-n)}) \bmod L$, where mod is the modulo operation.

Fig. 2 illustrates the relationship between the integer q, the local decisions u_1 and u_2, and the decision of the fusion center h_q for $N = 2$ sensors and $L = 4$ classes ($b = 2$ bits per sensor). Entry q in the table indicates the corresponding combination of local decisions and the output of the fusion rule h_q. For example, if the fusion rule is given by $h = (00 \cdots 01)$, then this means that the fusion center always decides for H_0 except when $u_1 = 00$ and $u_2 = 00$, in which case it decides in favor of H_1.

3.2 Fusion Rule Optimization

Searching for the fusion rule that leads to the minimum probability of error is the main bottleneck due to the discrete nature of this optimization process and the exponen-
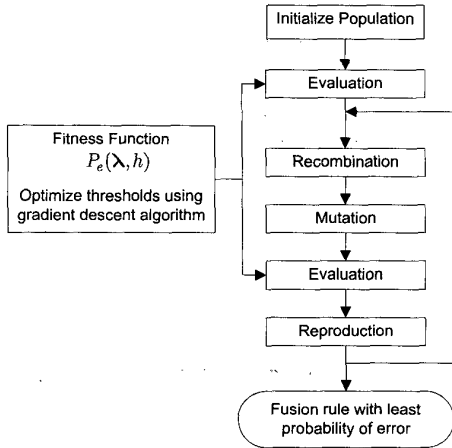
Figure 3: Sensor network optimization using the genetic algorithm.

tially large number of possible fusion rules. Except for few simple cases, exhaustive search strategies are usually infeasible since the search space grows exponentially with both, the number of sensors N and the number of bits per sensor b. Even if we assume identical sensors, the large number of possible fusion rules still precludes any practical algorithm based on exhaustive strategies. We develop an algorithm for optimizing the fusion rule by making use of the *Genetic Algorithm* (GA) to minimize the average probability of error. Unlike exhaustive search algorithms, GA uses evolution and survival-of-the-fittest mechanisms to guide the search toward the fittest candidates. The GA, introduced by John Holland in 1975 [4], has found a wide spread use in many fields including signal processing and communications (see for example [12] and the references therein).

Fig. 3 outlines the fusion rule search algorithm. The population comprises a group of chromosomes from which candidate solutions are selected. Each chromosome in the population represents a candidate fusion rule h represented by a string of L^N bits (genes). Initially, a population is generated randomly. The fitness of every chromosomes is evaluated by optimizing the local thresholds and calculating the objective function $P_e(\lambda, h)$ as described in the following section. A particular group of chromosomes (parents) is selected from the population to generate the offspring. The fitness of the offspring is evaluated similarly by evaluating their $P_e(\lambda, h)$. The chromosomes in the current population are then replaced by their offspring based on a replacement strategy. This cycle is repeated until a desired termination criterion is reached. If all goes well throughout this process of simulated evolution, the best chromosome in the final population can become a highly evolved solution to the problem.

GA-based algorithms usually suffer from slow convergence and high computational complexity, which may limit their use in real-time applications. However, recent developments in massively parallel computer architectures demonstrated the potential to eliminate the computational bottleneck in implementing evolution techniques due to the inherent parallel nature of the GA. In this regard, Twardowski [18] has shown the feasibility of GA real-time implementation using

parallel associative architectures. Furthermore, simulation results in this paper provide good heuristics that can be used without further optimization for real-time applications.

In the following section, we present a gradient-based algorithm for optimizing the thresholds of the local detectors. This optimization step is required for all candidate fusion rules within the GA population to evaluate the fitness of each chromosome.

3.3 Threshold Optimization

For each candidate fusion rule γ_0 represented by a binary string h, the set of $N(L - 1)$ local thresholds should be optimized with respect to the probability of error. Formally, this can be written as

$$\min_{\lambda} P_e(\lambda, h) \text{ subject to}$$

$$\lambda_{n,1} < \lambda_{n,2} < \cdots < \lambda_{n,L-1}, \quad n = 1, 2, \ldots N. \quad (5)$$

This is a $[N(L - 1)]$-dimensional nonlinear constrained optimization problem. Except for some simple cases, it is often hard, if not impossible, to find analytical solutions for the optimal thresholds, especially when the number of sensors or the number of bits per sensor is large.

We propose a gradient-based numerical approach to solve this optimization problem. Due to the number of constraints, instead of moving in the direction of the $N(L-1)$-dimensional gradient, we propose an algorithm similar to the *cyclic coordinate descent* explained in [8]. In this algorithm, each optimization step involves moving along the direction of the one-dimensional gradient with respect to one of the variables as long as the constraints are satisfied. The optimization is then carried out cyclically over all variables. Definitions and derivations of the probability of error $P_e(\lambda, h)$ and its gradients are presented in the appendix.

It might be tempting to think that optimal local classifiers should be identical, in which case the number of optimization variables can be greatly reduced. However, this is not true even if the observations are identically distributed. In fact, Tsitsiklis in [15] presented an example for which two sensors with identically distributed measurements have different decision rules. On the other hand, it has been shown in [15] that the performance loss due to this assumption is negligible when the number of sensors is large. In this paper, we study the general case except when indicated.

Table 1 outlines the threshold optimization algorithm. In this algorithm, each step involves the computation of the gradient with respect to a variable θ_i in the vector $\theta = (\theta_1, \theta_2, \ldots, \theta_{N(L-1)})$, then moving in the direction of that gradient as long as the constraints are not violated. In our work, the arrangement of the variables is such that $\theta = (\lambda_{1,1}, \lambda_{1,2}, \ldots, \lambda_{1,L-1}; \lambda_{2,1}, \ldots, \lambda_{N,L-1})$. The algorithm loops over all variables sequentially. The algorithm keeps searching until all gradients $\Delta_i, i = 1, 2, \ldots N(L - 1)$ approach zero or if a maximum number of iterations has been exceeded.

The search parameter $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_L)$ controls the convergence speed of the algorithm. The updating step of α_i is designed to speedup or stabilize the convergence of the algorithm: a too small value α_i leads to slow convergence; larger values may cause instabilities. At each step, the algorithm keeps track of the sign of the gradient Δ_i and uses it to update the value of α_i according to whether there was a sign change in Δ_i, indicating a switch in the search direc-

Table 1: Optimization of the local thresholds

1:	Initialize θ, α
2:	Compute the gradient $\Delta_i = \frac{\partial P_e(\theta, h)}{\partial \theta_i}$
3:	If constraints are violated, jump to 2
4:	$\theta_i \leftarrow \theta_i - \alpha_i \Delta_i$
5:	$i \leftarrow i + 1$ (reset if $i > N(L-1)$)
6:	If converged stop, else jump to 2

tion. Specifically, the α_i's are decreased by a small amount whenever the sign of Δ_i changes since this indicates that we stepped over a local optima and we should slow down the search. Similarly, if the sign of Δ_i does not change, this might indicate that we are still far away from the optima and we need to speed up the search in this direction by slightly increasing α_i.

Proper initialization plays another crucial role in the convergence of the algorithm. Since the algorithm is gradient-based, it is possible that the algorithm converges to an incorrect solution when the gradients become very small. Therefore, it is important to decide where to initialize the search in a way to avoid those regions where this might occur. Our results here and in the asymptotic study of [1] indicate that the thresholds should be initialized such that they are concentrated around the intersection between $f_0(y)$ and $f_1(y)$. This is intuitively reasonable since this is the region where it is hardest to discriminate between H_0 and H_1. In other words, it is not recommended to put the thresholds far away from the intersection point.

Convergence properties of the above algorithm are hard to analyze theoretically due to the high dimensionality and nonlinearity of the optimization problem at hand. Even if we succeeded in avoiding saddle points, the algorithm might get stuck in local minima. In the case of binary local detectors, a proof of quasi-convexity of the probability of error is provided in [11] for Gauss and some non-Gauss distributions. Aside from binary local detectors, convexity of the general case is much harder to prove. However, our results show good convergence properties and, for all case studies, the algorithm converges to a single point even if we use different reasonable initial conditions.

4. RESULTS

We consider a parallel fusion network with N sensors with a single measurement each and quantizing it to b bits per measurement. The sensors use possibly non-identical quantizers. Local observations are conditionally independent and identically distributed and, in addition, are assumed to follow the additive noise model $y = m_i + n$, where m_i is the signal mean under $H_i, i = 0, 1$ and n is a zero-mean noise with known distribution and variance σ^2. Our main objectives here are to investigate the structure of optimal fusion rules as well as to study the tradeoffs between sensor network parameters such as number of sensors N, number of bits per sensor b, and SNR. In addition, we study the convergence of the probability of error results towards those obtained under asymptotic assumption as $N \to \infty$.

4.1 Optimal fusion rule

For the purpose of exploring the structure of the fusion rule, we optimize the sensor network using the algorithm described in the previous section for different network set-
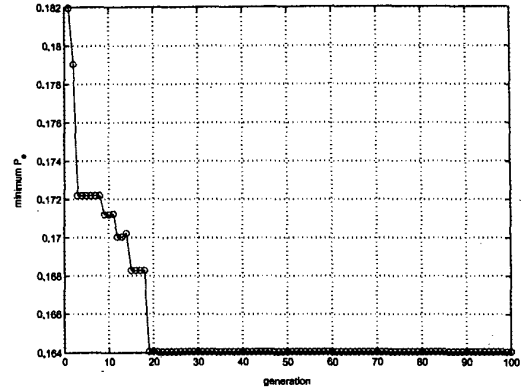
Figure 4: Evolution of the probability of error using the GA optimization algorithm for a network of $N = 4$ sensors and $b = 2$ bits per sensor. The measurements are Gaussian with $m_0 = 0$, $m_1 = 1$, and $\sigma^2 = 1$. The GA population size is set at 1000 chromosomes, while the crossover and mutation rates are 0.45 and 0.01, respectively.

tings. We use the notation (b, N) to refer to a network of N sensors with b bits per measurement. The aim here is to see if there is a unique structure to which the fusion rule converges.

Our study covered the cases of $(2, 2)$, $(2, 3)$, $(2, 4)$, $(3, 2)$, $(3, 3)$, and $(1, N)$ with N ranging from 2 to 8. Out of these cases, we were able to conduct an exhaustive search over all possible fusion rules for the $(2, 2)$ and the $(1, N), N = 2, 3, 4$ cases since the number of fusion rules is relatively small. For the remaining cases we used the GA algorithm to search for the fusion rule that minimizes the average probability of error. We note that this involves finding the optimal fusion rules and the optimal thresholds for the local detectors. These local detectors are not assumed to be identical, i.e., total number of the local thresholds is $N(L-1)$. In addition to considering different sensor network settings b and N, we also varied the prior probability π_0 as well as the parameters of the Gaussian observation model m_0, m_1, and σ^2. In all cases, we ran the algorithm several times to improve the odds of reaching the global minimum. Fig. 4 shows the evolution of the probability of error over 100 generations for a $(2, 4)$ case with prior probability $\pi_0 = 0.6$ and Gaussian observations with $m_0 = 0$, $m_1 = 1$, and $\sigma^2 = 1$, from which it is clear that the algorithm converged to a minimum after 20 generations. For this particular example, the population size is set at 1000 chromosomes while the crossover and mutation rates are 0.45 and 0.01, respectively.

The most important result in our study of fusion rules is that in all of the cases that we considered, the optimization algorithm converges to a fusion rule with a unique structure shown in Fig. 5. Interestingly, the best fusion rule that we found in our study came down to a simple threshold test on the integer sum of the local decisions. More interestingly, our results show that the best threshold λ^0 of this fusion rule is around

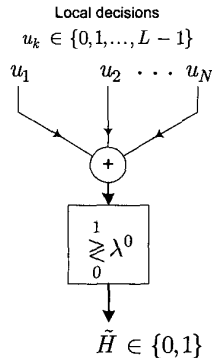$$\lambda^0 \simeq \frac{1}{2} N(L-1) \qquad (6)$$

112

Figure 5: Majority-like fusion rule.

Figure 6: Probability of error comparison of 8 binary ($b = 1$) versus 4 quaternary ($b = 2$) sensors when $\pi_0 = \pi_1 = 0.5$. The observations are Gaussian with $m_0 = -m_1$, and $\sigma^2 = 1$

regardless of the prior probabilities π_0 and π_1. It should be noted that there are several cases that we optimized using an exhaustive search algorithm for which we were able to show that this particular fusion rule is indeed optimal.

The fusion rule shown in Fig. 5 resembles the binary majority voting rule. It should be noted that the conclusion in [19] regarding the optimality of the majority voting rule is only specific for the case of binary local detectors ($b = 1$) and, in addition, has been derived under the assumption of identical local sensors. In contrast, our study provides an insight into the structure of the fusion rule in the general case ($b \geq 1$) and we do not restrict the local detectors to be identical.

The probability of error of the majority-like fusion rule in Fig. 5 can be analyzed using the saddle-point approach. Preliminary results show that a good accuracy is achievable using the Lugannani-Rice approximation [9]. This will be the subject of a subsequent paper.

4.2 Tradeoff between sensor network parameters

We discuss the tradeoffs between sensor network parameters such as the number of sensors N, number of bits per sensor b, and the SNR. For a fixed constraint on the total number of bits imposed for example by a MAC rate constraint R, we compare sensor networks with different number of bits b per sensor and different number of sensors N based on their detection performance. We use the fusion rule shown in Fig. 5 while the thresholds are optimized using the gradient-based algorithm described in section 3.3. The aim here is to see whether it is better to have a larger number of sensors N or a higher number of bits per sensor b when the total number of bits $R = bN$ is fixed due to constraints on the MAC channel. This problem has been addressed in [3] and [16], but with asymptotic assumptions on the number of sensors. Here we study this problem when the number of sensors is finite.

Fig. 6 shows the probability of error as a function of SNR of two sensor networks, one constructed using 8 binary ($b = 1$) sensors while the other uses only 4 quaternary ($b = 2$) sensors when $\pi_0 = \pi_1 = 0.5$, $m_0 = -m_1$, and $\sigma^2 = 1$. These both satisfy a MAC constraint of 8 bits per detection interval. Fig. 6 illustrate the superiority of the 8 binary sensors when both binary and quaternary sensors op-
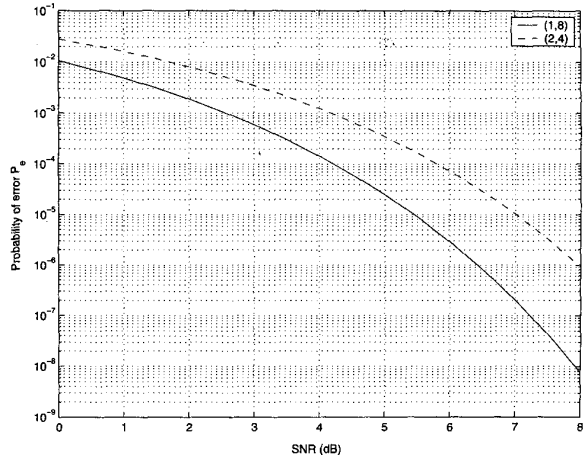
erate at the same SNR. This is consistent with the results of the asymptotic study in [3]. We note that the two curves in Fig. 6 are roughly 1.5 dB apart. So if the quaternary sensors operate at an SNR that exceeds by 1.5 dB the SNR of the binary sensors, then the 4 quaternary sensors outperform the 8 binary sensors. This contrasts with what is suggested in [3] and [16], in which it was concluded that we should always opt for the larger number of sensors. Again, we emphasize that their conclusion is reached under the assumption that all sensors offer the same SNR $\eta = \eta_1 = \eta_2 = \cdots = \eta_b$, where η_b is the SNR when b-bit per sensor quantizers are used. In practice, it is usually the case that low-cost hard sensors are of low-quality and, hence, provide lower SNR. The results here show that the preference of a particular sensor type over the others is dictated by the SNR that they can offer as well as their degree of quantization. Interestingly, we reached similar conclusion in our asymptotic ($N \to \infty$) study in [1] although the performance measure in that study is the error decay rate, which is somewhat different from the probability of error considered here.

4.3 Convergence towards the asymptotes

The assumption of asymptotically large number of sensors is often adopted when analyzing decentralized detection networks since it can lead to great simplifications resulting from the abstraction of the fusion center. Here, we study the convergence of our results based on finite number of sensors N toward those obtained when $N \to \infty$. Under the asymptotic assumption, an appropriate way of representing the performance of the system is through the error decay rate given by

$$C_b(\boldsymbol{\lambda}) = -\lim \frac{1}{N} \log P_e(N, b, \boldsymbol{\lambda}, \gamma_0), \qquad (7)$$

which is always less than C_b, the error decay rate of infinite-bandwidth systems (i.e., with unquantized local sensors). Fig. 7 shows the ratio of C_b, the error decay rates resulting from quantized measurements, to C_∞, the error decay rate when the sensors do not quantize their measurements, from
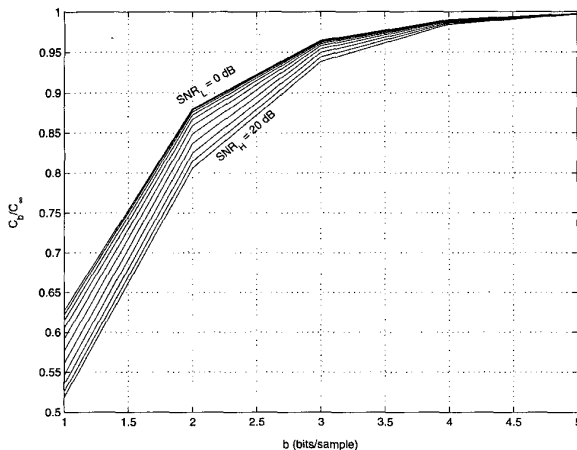
113

Figure 7: Ratio of the error decay rates of quantized to unquantized measurements. The measurements are Gaussian with $m_0 = -m_1$, $\sigma^2 = 1$. Each curve represents a different SNR ranging from $\mathrm{SNR_L} = 0$ dB to $\mathrm{SNR_H} = 20$ dB with 2 dB steps.

Figure 8: Convergence of the error decay rate towards the asymptotes when $b = 2$, $\pi_0 = \pi_1 = 0.5$ and the observations are Gaussian with $m_0 = -m_1$, and $\sigma^2 = 1$.

which it is clear that the performance loss due to quantization decays exponentially as the number of bits per sensor b is increased (more details in [1]). In this section, this ratio represents our main performance measure and we denote it by $\zeta_b(\eta) = C_b(\eta)/C_\infty(\eta)$, where η is the SNR.

Fig. 8 compares the ratio $\hat{\zeta}_b(N, \eta) = \hat{C}_b(N, \eta)/C_\infty(\eta)$ with the corresponding ratio $\zeta_b(\eta)$ obtained under the asymptotic assumption and when the local detectors are quaternary ($b = 2$). The approximate error decay rate $\hat{C}_b(N, \eta)$ is obtained by computing the probability of error P_e (this involves optimizing the local thresholds as explained in section 3.3 while the fusion rule is as shown in Fig. 5) at different numbers of sensors N and evaluating the decay rate numerically as N increases. The unquantized decay rate is given by $C_\infty(\eta) = \eta/2$ [3] while the quantized asymptotic error decay rate $C_b(\eta)$ is computed using the techniques presented in [1].

It is clear from Fig. 8 that the results obtained using a finite number N of sensors converge to the asymptotic results. More importantly, the convergence speed is highly dependent on the SNR. In particular, large number of sensors, possibly larger than 500 or 1000, are needed to approach the asymptotes when the SNR is low (as seen for SNR= -5 dB, a typical SNR environment with low quality sensors). This implies that, in low SNR environments, performance assessments based on asymptotic results lead to incorrect conclusions, especially when the number of sensors is not sufficiently large.

5. CONCLUSIONS

We addressed the problem of designing decentralized sensor networks when there are constraints on the communication resources. We considered a parallel fusion architecture and, unlike other studies, we adopted a non-asymptotic approach to the optimization of the local sensors and the fu-
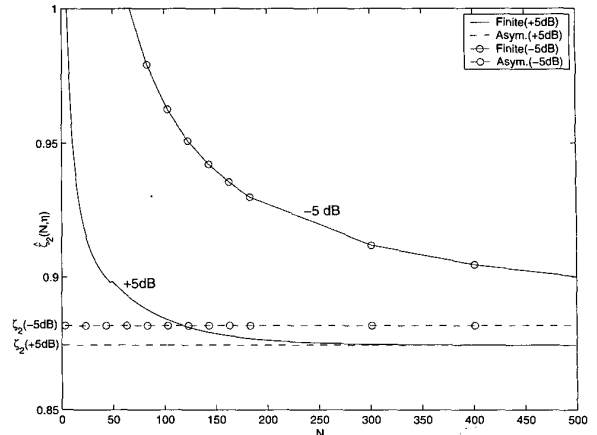
sion rule. Our results show that the optimal fusion rule has a structure that generalizes the majority rule for binary sensors. It should be noted that despite the complexity of the proposed optimization algorithm, simulation results in this paper provide good heuristics that can be used without further optimization for real-time applications. In addition, the obtained results enabled us to study the tradeoff between the number of bits per local quantizer and the SNR. We concluded that, for example, for binary ($b = 1$) versus quaternary ($b = 2$) sensors, if the SNR for the quaternary sensors is at least 1.5 dB larger than the SNR for the binary sensors, it is preferable to use a number of quaternary sensors that is only half the number of binary sensors. Finally, we addressed the question of how large should the number of sensors be so that performance results based on asymptotic analysis (with the number of sensors) are reliable. We find that the number is highly dependent on the SNR, and that it can be fairly large, on the order of hundreds if not thousands, for realistic (low) SNR values.

6. REFERENCES

[1] S. A. ALDOSARI AND J. M. F. MOURA. Detection in decentralized sensor networks. Accepted for presentation at *The IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'04)*, Montreal, Canada, (May 2004).

[2] R. S. BLUM AND M. C. DEANS. Distributed random signal detection with multibit sensor decisions. *IEEE Transactions on Information Theory* 44, 516–524 (Mar. 1998).

[3] J.-F. CHAMBERLAND AND V. V. VEERAVALLI. Decentralized detection in sensor networks. *IEEE Transactions on Signal Processing* 51, 407–416 (Feb. 2003).

[4] J. H. HOLLAND. "Adaptation in Natural and Artificial Systems". Ann arbor University of Michigan Press, Ann Arbor, MI (1975).

[5] W. W. IRVING AND J. N. TSITSIKLIS. Some properties of optimal thresholds in decentralized detection. *IEEE Transactions on Automatic Control* **39**, 835–838 (1994).

[6] C. C. LEE AND J. J. CHAO. Optimum local decision space partitioning for distributed detection. *IEEE Transactions on Aerospace and Electronic Systems* **AES-25**, 536–544 (July 1989).

[7] M. LONGO, T. D. LOOKABAUGH, AND R. M. GRAY. Quantization for decentralized hypothesis testing under communication constraints. *IEEE Transactions on Information Theory* **36**, 241–255 (Mar. 1990).

[8] D. G. LUENBERGER. "Introduction to linear and nonlinear programming". Addison-Wesley, Reading, MA (1973).

[9] R. LUGANNANI AND S. O. RICE. Saddlepoint approximation for the distribution of the sum of independent random variables. *Advances in Applied Probability* **12**, 475–490 (1980).

[10] L. L. SCHARF. "Statistical Signal Processing: Detection, Estimation, and Time Series Analysis". Addison-Wesley, Reading, MA (1991).

[11] W. SHI, T. W. SUN, AND R. D. WESEL. Quasi-convexity and optimal binary fusion for distributed detection with identical sensors in generalized Gaussian noise. *IEEE Transactions on Information Theory* **47**, 446–449 (Jan. 2001).

[12] K. S. TANG, K. F. MAN, AND Q. HE. Genetic algorithms and their applications. *IEEE Signal Processing Magazine* **13**, 22–37 (Nov. 1996).

[13] Z.-B. TANG, K. R. PATTIPATI, AND D. L. KLEINMAN. An algorithm for determining the decesion thresholds in a distributed detection problem. *IEEE Transactions on Systems, Man, and Cybernetics* **21**, 231–237 (Jan./Feb. 1991).

[14] R. R. TENNEY AND N. R. S. JR. Detection with distributed sensors. *IEEE Transactions on Aerospace and Electronic Systems* **AES-17**, 98–101 (July 1981).

[15] J. N. TSITSIKLIS. Decentralized detection by a large number of sensors. *Mathematics of Control, Signals, and Systems* **1**(2), 167–182 (1988).

[16] J. N. TSITSIKLIS. Decentralized detection. *In "Advances in Statistical Signal Processing: Vol 2 - Signal Detection,"* H. V. Poor, and J. B. Thomas, eds., JAI Press, Greenwich, CT, pp. 297–344 (Nov. 1993).

[17] J. N. TSITSIKLIS AND M. ATHANS. On the complexity of decentralized decision making and detection problems. *IEEE Transactions on Automatic Control* **30**, 440–446 (May 1985).

[18] K. TWARDOWSKIS. An associative architecture for genetic algorithm-based machine learning. *IEEE Computer* **27**, 27–38 (Nov. 1994).

[19] P. K. VARSHNEY. "Distributed Detection and Data Fusion". Springer-Verlag, New York (1996).

[20] R. VISWANATHAN AND P. VARSHNEY. Distributed detection with multiple sensors: Part I–Fundamentals. *Proceedings of the IEEE* **85**, 54–63 (Jan. 1997).

APPENDIX

A. GRADIENT OF THE PROBABILITY OF ERROR

Given a fusion rule h, the average probability of error at the fusion center is given by the weighted sum of type-I and type-II errors,

$$P_e(\lambda, h) = \sum_{k=0}^{1} \pi_k P_k^0(\bar{k}, \lambda, h), \qquad (8)$$

where π_k is the prior probability of hypothesis H_k, $P_k^0(\bar{k}, \lambda, h) = \Pr(u_0 = \bar{k}|H_k)$ is the false alarm probability when $k = 0$ or the miss probability when $k = 1$, and $\bar{k}$ is the binary negation of k (i.e., $\bar{k}$ =NOT(k)). From the L^N mutually exclusive possibilities of the local decisions, we sum over those that result in a $u_0 = k$ decision at the fusion center and we use the fact that the measurements at the local sensors are conditionally independent to write the probability of each error type $P_k^0(\lambda, h)$ as a sum of products,

$$P_k^0(\bar{k}, \lambda, h) = \sum_{\substack{q=0 \\ h_q=\bar{k}}}^{L^N-1} \prod_{n=1}^{N} P_k^n(\Psi_n(q), \lambda), \qquad (9)$$

where $P_k^n(m, \lambda) = \Pr(u_n = m|H_k)$ is the probability that the nth sensor decides m when H_k is present. Since u_n results from quantizing the measurement y_n, its probability is related to the pdf of y_n through the $L - 1$ quantization thresholds $\lambda_n = (\lambda_{n,1}, \lambda_{n,2}, \ldots, \lambda_{n,L-1})$, and since the measurements are identically distributed, $P_k^n(m, \lambda)$ is given by

$$P_k^n(m, \lambda) = \int_{\lambda_{n,m}}^{\lambda_{n,m+1}} f_k(y)dy = F_k(\lambda_{n,m+1}) - F_k(\lambda_{n,m}), \qquad (10)$$

where $\lambda_{n,0}$ and $\lambda_{n,L}$ are defined to be $-\infty$ and ∞, respectively, $f_k(y)$ is the probability density function of y conditioned on H_k, and $F_k(x)$ is the conditional cumulative density function (ccdf) of y defined as

$$F_k(x) = \int_{-\infty}^{x} f_k(y)dy. \qquad (11)$$

The gradient of the average probability of error $P_e(\lambda, h)$ with respect to a threshold $\lambda_{\nu,\tau}$ is given by

$$\frac{\partial}{\partial \lambda_{\nu,\tau}} P_e(\lambda, h) = \sum_{k=0}^{1} \pi_k \frac{\partial}{\partial \lambda_{\nu,\tau}} P_k^0(\bar{k}, \lambda, h), \qquad (12)$$

$$\frac{\partial}{\partial \lambda_{\nu,\tau}} P_k^0(\bar{k}, \lambda, h) = \sum_{\substack{q=0 \\ h_q=k}}^{L^N-1} \frac{\partial P_k^\nu(\Psi_\nu(q), \lambda)}{\partial \lambda_{\nu,\tau}} \prod_{\substack{n=1 \\ n \neq \nu}}^{N} P_k^n(\Psi_n(q), \lambda). \qquad (13)$$

The derivative of $P_k^n(m, \lambda)$, the probability of the nth sensor deciding in favor of m, with respect to a threshold $\lambda_{n,\tau}$ can be shown to be

$$\frac{\partial P_k^n(m, \lambda)}{\partial \lambda_{n,\tau}} = \begin{cases} -f_k(\lambda_{n,\tau}) & \text{if } m = \tau \\ f_k(\lambda_{n,\tau}) & \text{if } m = \tau - 1 \\ 0 & \text{otherwise} \end{cases} \qquad (14)$$