Blind Adaptive Decision Fusion for Distributed Detection

GHASEM MIRJALILY

ZHI-QUAN LUO, Member, IEEE

TIMOTHY N. DAVIDSON, Member, IEEE
McMaster University
Canada

ÉLOI BOSSÉ
Defense Research Establishment Valcartier
Canada

We consider the problem of decision fusion in a distributed detection system. In this system, each detector makes a binary decision based on its own observation, and then communicates its binary decision to a fusion center. The objective of the fusion center is to optimally fuse the local decisions in order to minimize the final error probability. To implement such an optimal fusion center, the performance parameters of each detector (i.e., its probabilities of false alarm and missed detection) as well as the a priori probabilities of the hypotheses must be known. However, in practical applications these statistics may be unknown or may vary with time. We develop a recursive algorithm that approximates these unknown values on-line. We then use these approximations to adapt the fusion center. Our algorithm is based on an explicit analytic relation between the unknown probabilities and the joint probabilities of the local decisions. Under the assumption that the local observations are conditionally independent, the estimates given by our algorithm are shown to be asymptotically unbiased and converge to their true values at the rate of $O(1/k^{1/2})$ in the rms error sense, where k is the number of iterations. Simulation results indicate that our algorithm is substantially more reliable than two existing (asymptotically biased) algorithms, and performs at least as well as those algorithms when they work.

Manuscript received April 1, 2000; revised May 16, 2002; released for publication August 29, 2002.

IEEE Log No. T-AES/39/1/808660.

Refereeing was handled by X. R. Li.

This research is supported by a research grant from the Natural Sciences and Engineering Research Council of Canada and by a grant from the Defense Research Establishment Valcartier (DREV), Québec, Canada. The second author is also supported by the Canada Research Chair programme.

Authors' current addresses: G. Mirjalily, Dept. of Electrical Engineering, Yazd University, Yazd, Iran; Z-Q. Luo, and T. N. Davidson, Dept. of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario L8S 4K1, Canada; É. Bosse, Defense Research Establishment Valcartier, Decision Support Technologies Section, 2459 PIE XI Nord, B.P. 8800, Courcelette, Québec, G0A 1R0, Canada.

I. INTRODUCTION

The problem of distributed detection has been the subject of several recent studies [4, 14]. It is well known that the deployment of multiple sensors for signal detection in a military surveillance application may substantially enhance system survivability and result in improved detection performance, shorter decision time, and other benefits [14]. Although many organizational structures for distributed detection systems can be constructed, the most popular choice has been the parallel topology as shown in Fig. 1. In this topology each local detector LD_i makes a decision u_i based on its own observation Y_i and then transmits this decision to the final decision-maker (fusion center). Based on the received local decisions, the fusion center makes the final decision u_0 by employing the optimal fusion rule (discussed below). Throughout this work, we consider binary hypothesis testing, therefore u_i will be binary. We adopt the convention that $u_i = 1$ (respectively, $u_i = 0$) if the ith local detector favors the hypothesis H_1 (respectively, the hypothesis H_0). The a priori probabilities of the two hypotheses are denoted by $P(H_0) = P_0$ and $P(H_1) = P_1$ such that $P_0 + P_1 = 1$.

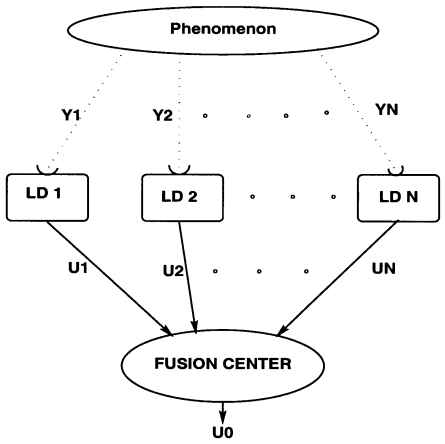


Fig. 1. Parallel distributed detection system.

The general problem of determining the optimal local decision rules and optimal fusion rule for the system in Fig. 1 is known to be NP-hard [13]. Thus, to efficiently design an optimal distributed detection system, it is necessary to assume that the local observations are structured in some way. The standard assumption that we employ here is that the local observations are conditionally independent in the sense that $P(Y_i, Y_j \mid H_k) = P(Y_i \mid H_k)P(Y_j \mid H_k)$, for all $i \neq j$ and all k. Chair and Varshney [4] showed that for N conditionally independent local detectors, the optimal fusion rule based on the minimum probability of error criterion can be

written as

$$u_0 = \begin{cases} 1, & \text{if} \quad w_0 + \sum_{i=1}^{N} w_i > 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where

$$w_0 = \log\left(\frac{P_1}{P_0}\right) \tag{2}$$

and for $i = 1,\ldots,N$,

$$w_i = \begin{cases} \log((1-P_i^m)/P_i^f), & \text{if} \quad u_i = 1 \\ \log(P_i^m/(1-P_i^f)), & \text{if} \quad u_i = 0. \end{cases} \tag{3}$$

Here P_i^f and P_i^m represent the probabilities of false alarm and missed detection at the ith local detector:

$$P_i^f = \Pr(u_i = 1 \mid H_0), \qquad P_i^m = \Pr(u_i = 0 \mid H_1). \tag{4}$$

As can be seen, each weight w_i is a function of the probabilities of false alarm and missed detection errors at the corresponding detector. So to implement the optimal fusion rule (1), we need to know the a priori probability of the hypothesis P_1 (recall that $P_0 = 1 - P_1$), and the probabilities of false alarm and missed detection $\{P_i^f, P_i^m\}$. Unfortunately these parameters are not always available in practice or may be time varying [1, 9]. The primary objective of the work presented here is to develop an on-line blind adaptive algorithm for estimating these parameters at the fusion center. By "blind" we mean that the adaptive algorithm does not have access to any training signals.

A few early works have been reported on the on-line blind estimation of error probabilities. Naim and Kam [9] proposed an on-line adaptive algorithm to estimate the values of $\{P_1, P_i^f, P_i^m\}$ for the Bayesian hypothesis testing problem. The local detectors are assumed to be conditionally independent and to perform local hypothesis tests based on their own observations. However, the estimator given in Naim and Kam's algorithm [9] is statistically biased and requires bias-correction to reduce the estimation error, thus making the algorithm computationally complex. Ansari and his coauthors presented, in a series of papers [1, 2, 5], a learning algorithm which estimates the weights of the fusion center directly without estimating error probabilities. Similar to [9], the estimator of Ansari et al. lacks certain desirable statistical properties such as unbiasedness. To reduce the error in the estimation of the weights for each sensor, they perform a threshold test to determine the extent to which the central decision is "reliable" in the absence of that sensor. Only central decisions which are reliable without that sensor's local decision are used to update that sensor's weight. However, dropping such decisions may compromise the convergence behavior of the algorithm. Furthermore, a proper procedure for the selection of the (optimal) reliability threshold remains unresolved in the aforementioned studies.

In this work, we propose an on-line algorithm which, like that in [9], estimates the a priori probability P_1, and the probabilities P_i^f and P_i^m and then uses them to implement the fusion rule (1)–(3). Our algorithm is based on an explicit analytic relation between the unknown a priori probabilities and the joint probabilities of the local decisions. The latter can be estimated via a simple averaging algorithm and then be used to calculate the unknown a priori probabilities P_1 and P_i^f, P_i^m. Our method allows us to estimate the performance of each detector efficiently. Unlike Naim and Kam's estimators [9], our estimators of the unknown probabilities $\{P_1, P_i^f, P_i^m\}$ are shown to be asymptotically unbiased, provided that the local observations are conditionally independent. Moreover, we prove that the estimates converge to their true values at a rate of $O(1/k^{1/2})$ in the rms sense, where k is the number of iterations. Our simulation studies indicate that our algorithm is substantially more reliable than the (asymptotically biased) algorithms of Naim and Kam [9] and Ansari et al. [2], and performs at least as well as those algorithms when they work. We also suggest modifications to the algorithms of Naim and Kam [9] and Ansari et al. [2] which make them more reliable (see Appendix C), however the resulting algorithms are still asymptotically biased and the modified version of Naim and Kam's algorithm is still much more computationally expensive than our method.

The remainder of this paper is organized as follows. In Section II we explain our approach and present the algorithm for the case of three detectors. (Note that our algorithm is not applicable in systems with two detectors.) The main results on the three-detector case are established, and some simulation results are presented. Also, several simple methods which enable the algorithm to track changes faster are proposed. In Section III we generalize our algorithm to the case of more than three local detectors. Performance comparisons with two existing methods are given in Section IV and some concluding remarks and suggestions are given in Section V.

II. ADAPTIVE DECISION FUSION FOR THE THREE-DETECTOR CASE

In this section we introduce an adaptive fusion algorithm for a distributed detection system with exactly three detectors. In Section III, this algorithm is generalized for systems with more than three detectors.

A. An Invertibility Result

Consider a distributed detection system with exactly three detectors and a fusion center arranged in a parallel structure (c.f., Fig. 1). Each detector employs a predetermined local decision rule, and we assume that conditioned on each hypothesis, the

local binary decisions are statistically independent. Our purpose is to estimate the unknown probabilities P_1, P_i^f, and P_i^m, $i = 1,2,3$, by using the local binary decisions u_1, u_2, and u_3. We accomplish this goal indirectly by estimating the unconditional probability of each decision and their joint probabilities. Note that in this case we have three binary decisions, giving a total of eight different combined outcomes. Let us use the short notation P_{ijk} to denote the probability of $u_1 = i$, $u_2 = j$ and $u_3 = k$, where $i,j,k = 0$ or 1. Using Bayes rule and conditional independence, we have

$$\begin{aligned} P_{ijk} &= \Pr(u_1 = i,\ u_2 = j,\ u_3 = k) \\ &= P(u_1 = i \mid H_1)P(u_2 = j \mid H_1)P(u_3 = k \mid H_1)P_1 \\ &\quad + P(u_1 = i \mid H_0)P(u_2 = j \mid H_0)P(u_3 = k \mid H_0)(1 - P_1). \end{aligned} \tag{5}$$

Notice that

$$P(u_j = i \mid H_1) = \begin{cases} 1 - P_j^m, & \text{if} \quad i = 1 \\ P_j^m, & \text{if} \quad i = 0 \end{cases} \tag{6}$$

and

$$P(u_j = i \mid H_0) = \begin{cases} P_j^f, & \text{if} \quad i = 1 \\ 1 - P_j^f, & \text{if} \quad i = 0. \end{cases} \tag{7}$$

Rewriting (5) in terms of P_i^f, P_i^m $(i = 1,2,3)$ yields a nonlinear system of equations

$$\begin{cases} P_{000} = P_1^m P_2^m P_3^m P_1 + (1 - P_1^f)(1 - P_2^f)(1 - P_3^f)(1 - P_1) \\ P_{001} = P_1^m P_2^m (1 - P_3^m) P_1 + (1 - P_1^f)(1 - P_2^f) P_3^f (1 - P_1) \\ P_{010} = P_1^m (1 - P_2^m) P_3^m P_1 + (1 - P_1^f) P_2^f (1 - P_3^f)(1 - P_1) \\ P_{100} = (1 - P_1^m) P_2^m P_3^m P_1 + P_1^f (1 - P_2^f)(1 - P_3^f)(1 - P_1) \\ P_{111} = (1 - P_1^m)(1 - P_2^m)(1 - P_3^m) P_1 + P_1^f P_2^f P_3^f (1 - P_1) \\ P_{011} = P_1^m (1 - P_2^m)(1 - P_3^m) P_1 + (1 - P_1^f) P_2^f P_3^f (1 - P_1) \\ P_{110} = (1 - P_1^m)(1 - P_2^m) P_3^m P_1 + P_1^f P_2^f (1 - P_3^f)(1 - P_1) \\ P_{101} = (1 - P_1^m) P_2^m (1 - P_3^m) P_1 + P_1^f (1 - P_2^f) P_3^f (1 - P_1). \end{cases} \tag{8}$$

Notice that the system of nonlinear equations (8) provides a link between the values of P_{ijk} and the unknown parameters P_1, P_i^f, P_i^m $(i = 1,2,3)$. Clearly, the values of P_{ijk} can be estimated easily by using the local observations $\{u_i, i = 1,2,3\}$. Once the values of P_{ijk} are estimated, we can use the nonlinear system (5) to infer the unknown parameters P_1, P_i^f, P_i^m $(i = 1,2,3)$. As a result, it becomes necessary to solve the nonlinear system (8) by representing P_1, P_i^f, P_i^m $(i = 1,2,3)$ in terms of the values of P_{ijk}. It is well known that in general a numerical procedure is required to solve a nonlinear system. However, the rather surprising main result of this section is that the nonlinear system (8) has some special algebraic structure which affords it an analytic solution. Needless to say, it is much more desirable to invert the nonlinear system (8) analytically, rather than numerically.

As can be seen from (8), the probability values $\{P_{ijk}\}$ depend on a total of seven unknown (independent) parameters P_1, P_i^f and P_i^m, $i = 1,2,3$. Notice that only seven out of the eight equalities in (8) are algebraically independent because of the normalizing condition

$$\sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} P_{ijk} = 1.$$

In other words, we have a total of seven nonlinear algebraic equations relating the joint probabilities of local decisions to the seven unknown parameters that we wish to estimate. In principle, we should be able to represent the unknown parameters in terms of the probabilities P_{ijk} (thus, inverting the nonlinear system (8)).

In what follows, we assume

$$0 < P_1 < 1, \qquad P_i^f + P_i^m < 1, \qquad i = 1,2,3. \tag{9}$$

Under this assumption, we can derive an explicit representation of P_1, P_i^f and P_i^m $(i = 1,2,3)$ in terms of P_{ijk}.

THEOREM 1 *Consider a distributed system with exactly three detectors. Let P_1 denote the a priori probability that the hypothesis H_1 holds. Suppose P_i^f, P_i^m are the probabilities of false alarm and missed detection for the ith detector respectively, and assume that they satisfy* (9). *Then the nonlinear system* (8) *can be inverted analytically. In particular, let*

$$\begin{aligned} X &= \frac{\gamma_* - \gamma_1\gamma_2\gamma_3 - (\gamma_1 a_2 a_3 + \gamma_2 a_1 a_3 + \gamma_3 a_1 a_2)}{\sqrt{(\delta_{12} - \gamma_1\gamma_2)(\delta_{13} - \gamma_1\gamma_3)(\delta_{23} - \gamma_2\gamma_3)}} \\ &= \frac{\gamma_* - \gamma_1\gamma_2\gamma_3 - (\gamma_1|\delta_{23} - \gamma_2\gamma_3| + \gamma_2|\delta_{13} - \gamma_1\gamma_3| + \gamma_3|\delta_{12} - \gamma_1\gamma_2|)}{\sqrt{(\delta_{12} - \gamma_1\gamma_2)(\delta_{13} - \gamma_1\gamma_3)(\delta_{23} - \gamma_2\gamma_3)}} \end{aligned} \tag{10}$$

where

$$\begin{aligned} &\gamma_1 = \sum_{j,k} P_{1jk}, \qquad \gamma_2 = \sum_{i,k} P_{i1k}, \\ &\gamma_3 = \sum_{i,j} P_{ij1}, \qquad \gamma_* = P_{111} \\ &\delta_{12} = P_{110} + P_{111}, \qquad \delta_{13} = P_{111} + P_{101}, \\ &\delta_{23} = P_{111} + P_{011} \end{aligned} \tag{11}$$

and

$$\begin{cases} a_1 = \sqrt{\dfrac{(\delta_{12} - \gamma_1\gamma_2)(\delta_{13} - \gamma_1\gamma_3)}{\delta_{23} - \gamma_2\gamma_3}} \\ a_2 = \sqrt{\dfrac{(\delta_{12} - \gamma_1\gamma_2)(\delta_{23} - \gamma_2\gamma_3)}{\delta_{13} - \gamma_1\gamma_3}} \\ a_3 = \sqrt{\dfrac{(\delta_{13} - \gamma_1\gamma_3)(\delta_{23} - \gamma_2\gamma_3)}{\delta_{12} - \gamma_1\gamma_2}}. \end{cases} \tag{12}$$

Then, P_1, P_0 can be computed as

$$P_1 = 0.5 - \frac{X}{2\sqrt{X^2+4}}, \qquad P_0 = 1 - P_1. \tag{13}$$

Furthermore, we have

$$P_i^f = \gamma_i - a_i\sqrt{\frac{P_1}{1-P_1}}, \qquad P_i^m = 1 - \gamma_i - a_i\sqrt{\frac{1-P_1}{P_1}}, \quad i = 1,2,3. \tag{14}$$

The proof of Theorem 1 involves straightforward, but tedious, algebraic manipulation and has been relegated to Appendix A. Theorem 1 suggests that the unknown probabilities P_0, P_1 and P_i^m, P_i^f can be calculated uniquely from the joint (unconditional) probabilities γ_i, δ_{ij} and γ_* (or equivalently, from the joint probabilities P_{ijk}, $i,j,k = 0,1$), provided that the (reasonable) assumption (9) is satisfied.

It should be pointed out that Theorem 1 assumes the presence of (at least) three detectors (or sensors) in the distributed system. Although this assumption seems restrictive, it is actually necessary, as we explain below. Suppose there are N detectors in the system, then there will be $2N + 1$ unknown parameters to be estimated (P_1, plus P_i^f, P_i^m, $i = 1,2,\ldots,N$). At the same time, we have $2^N - 1$ independent joint probabilities (of local decisions) in the form of (8). This is because there are exactly 2^N different combinations of N local decisions and there is one normalizing condition. In order to be able to estimate the $2N + 1$ unknowns from the $2^N - 1$ joint probabilities of local decisions, one must have $2^N - 1 \geq 2N + 1$, implying $N \geq 3$. For $N = 2$, there are $2N + 1 = 5$ unknown parameters (P_1, P_1^f, P_2^f, P_1^m, P_2^m) to be estimated, and there are only $2^N - 1 = 3$ algebraic relations in the form of (8). Therefore, we cannot expect to estimate the unknown parameters from the local decisions. Of course, if two of the 5 unknowns are given or if additional side information is available, then we may still be able to recover the unknown parameters when there are only two detectors.

B. An Adaptive Decision Fusion Algorithm

Notice that the values of the (unconditional) joint probabilities γ_i ($i = 1,2,3$), δ_{ij} ($i,j = 1,2,3$, $j \neq i$) and γ_* can be conveniently approximated via time-averaging [10]; e.g.,

$$\hat{\gamma}_i^k = \frac{1}{k}\sum_{j=1}^{k} u_i^j$$

where u_i^j is the decision of ith detector at the jth time step. The above equation can be written recursively as

$$\hat{\gamma}_i^k = \frac{1}{k}u_i^k + \frac{k-1}{k}\hat{\gamma}_i^{k-1} = \hat{\gamma}_i^{k-1} + \frac{1}{k}(u_i^k - \hat{\gamma}_i^{k-1}) \tag{15}$$

with an arbitrary initial guess. Similar to (15), we have

$$\hat{\delta}_{ij}^k = \frac{1}{k}u_i^k u_j^k + \frac{k-1}{k}\hat{\delta}_{ij}^{k-1} \tag{16}$$

$$\hat{\gamma}_*^k = \frac{1}{k}u_1^k u_2^k u_3^k + \frac{k-1}{k}\hat{\gamma}_*^{k-1}. \tag{17}$$

Once the estimated values $\hat{\gamma}_i^k$, $\hat{\delta}_{ij}^k$ and $\hat{\gamma}_*^k$ are obtained, all of the unknown probabilities P_0, P_1 and P_i^m, P_i^f can be estimated by substituting these estimated values for γ_i, δ_{ij} and γ_* in equations (10)–(14).

Equations (15)–(17) are stochastic approximations [as emphasized in (15)] which are convergent with probability 1 if the sequence of local decisions $\{(u_1^k, u_2^k, u_3^k)\}$ is ergodic [3]. These estimations are also unbiased with variance decaying to zero at the rate $1/k$, as k increases [10]. From the above discussions, we propose the following adaptive fusion algorithm.

Adaptive Fusion Algorithm

Initial Step: Select initial values: $\hat{\gamma}_i^0$ ($i = 1,2,3$), $\hat{\delta}_{ij}^0$ ($i,j = 1,2,3$, $j \neq i$) and $\hat{\gamma}_*^0$.

Iterations:

a) Update the values of unconditional probabilities $\hat{\gamma}_i^k$, $\hat{\delta}_{ij}^k$ and $\hat{\gamma}_*^k$ by observing local decisions u_1^k, u_2^k and u_3^k and using (15)–(17).

b) Use the updated values $\hat{\gamma}_i^k$, $\hat{\delta}_{ij}^k$ and $\hat{\gamma}_*^k$ obtained from (a) to calculate $\hat{a}_i$ ($i = 1,2,3$) and $\hat{X}$ according to (12) and (10). Then compute $\hat{P}_1$, $\hat{P}_i^f$ and $\hat{P}_i^m$ ($i = 1,2,3$) according to

$$\begin{cases} \hat{P}_1 = [0.5 - 0.5\hat{X}(\hat{X}^2+4)^{-1/2}]_+ \\ \hat{P}_0 = [1 - \hat{P}_1]_+ = 1 - \hat{P}_1 \\ \hat{P}_i^f = [\hat{\gamma}_i - \hat{P}_1^{1/2}\hat{P}_0^{-1/2}\hat{a}_i]_+, \qquad i = 1,2,3 \\ \hat{P}_i^m = [1 - \hat{\gamma}_i - \hat{P}_0^{1/2}\hat{P}_1^{-1/2}\hat{a}_i]_+, \qquad i = 1,2,3 \end{cases} \tag{18}$$

where $[\cdot]_+$ denotes the projection operator onto the unit interval, that is, $[z]_+ = \max\{\min\{1, \mathrm{Re}\{z\}\}, 0\}$, for all $z \in \mathbb{C}$. Geometrically, $[z]_+$ denotes the point in $[0,1]$ that is closest to z. (For notational simplicity, we have dropped the index k from the notations $\hat{X}$, $\hat{a}_i$, $\hat{P}_i^f$, $\hat{P}_i^m$, etc. It should be clear from the context that these estimates are updated at each iteration and thus vary with k.)

c) Use the estimated values of $\hat{P}_0$, $\hat{P}_1$, $\hat{P}_i^f$ and $\hat{P}_i^m$ ($i = 1,2,3$), and (2) and (3) to form estimates, $\hat{w}_0$ and $\hat{w}_i$, of the optimal fusion weights.

Due to the recursive nature of the updating rules (15)–(17), it can be easily checked that in the above adaptive fusion algorithm only a fixed amount data storage is required at all times. Furthermore, the algorithm's computational requirement per iteration is constant; that is, it does not increase with the number

of iterations. Clearly, these are important features for an adaptive fusion algorithm.

As another remark, we note that the estimators (18) are exactly as prescribed by Theorem 1 except for the inclusion of the projection operator $[\cdot]_+$. The use of this projection operator ensures that the estimators will always remain real and bounded (even in the case when the denominators of (10)–(14) are small). This boundedness property allows us to prove the asymptotic unbiasedness of the estimators and establish their rate of convergence (see Theorem 2 below).

THEOREM 2 *Let* $\hat{P}_1$, $\hat{P}_i^f$, $\hat{P}_i^m$ $(i = 1,2,3)$ *be the estimators given by* (18). *Assume that the local observations* $\{(u_1^k, u_2^k, u_3^k)\}$ *are ergodic and that*

$$\epsilon \le P_1 \le 1-\epsilon, \qquad 1 - P_i^f - P_i^m \ge \epsilon,$$
$$i = 1,2,3, \qquad \text{for some} \quad \epsilon > 0. \tag{19}$$

Then, with probability one,

$$\lim_{k\to\infty} \hat{P}_1 = P_1, \qquad \lim_{k\to\infty} \hat{P}_i^f = P_i^f,$$
$$\lim_{k\to\infty} \hat{P}_i^m = P_i^m, \qquad i = 1,2,3,$$

which, by the Lebesgue Dominance Theorem, implies the asymptotic unbiasedness of the estimators:

$$\lim_{k\to\infty} E(\hat{P}_1) = P_1, \qquad \lim_{k\to\infty} E(\hat{P}_i^f) = P_i^f,$$
$$\lim_{k\to\infty} E(\hat{P}_i^m) = P_i^m, \qquad i = 1,2,3.$$

Moreover, there holds

$$E|\hat{P}_1 - P_1|^2 = O(1/k), \qquad E|\hat{P}_i^f - P_i^f|^2 = O(1/k),$$
$$E|\hat{P}_i^m - P_i^m|^2 = O(1/k), \qquad i = 1,2,3$$

where $O(1/k)$ *denotes a nonnegative function such that* $\limsup_{k\to\infty} O(1/k)k < \infty$.

The proof of Theorem 2 is quite complicated and has been relegated to Appendix B. Theorem 2 shows that the probability estimates $\hat{P}_1$, $\hat{P}_i^f$, $\hat{P}_i^m$ $(i = 1,2,3)$ will eventually converge to their true values as k (the number of iterations) increases. An important corollary of this result is that the estimated fusion weights $\hat{w}_0$, $\hat{w}_i$ $(i = 1,2,3)$ will also converge eventually converge to their optimal values. This further implies the resulting final error rate will converge to its minimum value as $k \to \infty$.

Finally, we remark that in practice the assumption (19) is almost always satisfied. However, from a theoretical standpoint, it will be desirable to develop an analysis of the estimators (18) in the absence of this assumption.

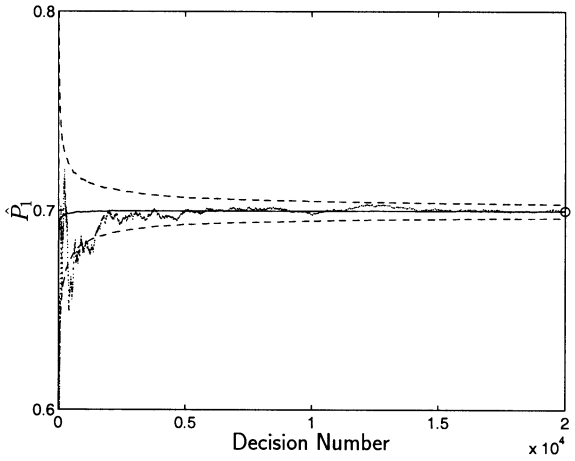


Fig. 2. Adaptation of $\hat{P}_1$ for the example in Section IIC1. Legend–solid: average over 1000 realizations; dashed: average plus and minus one standard deviation; dotted: typical realization; circle: true value of P_1.

C. Simulation Results

1) *Convergence*: To examine the effectiveness of the proposed algorithm, we have performed some computer simulations for a distributed detection system consisting of three detectors. In these simulations, the target and local detectors are simulated as binary sources with probabilities:

$$P_1 = 0.7, \qquad P_1^f = 0.05,$$
$$P_2^f = 0.09, \qquad P_3^f = 0.15, \tag{20}$$
$$P_1^m = 0.03, \qquad P_2^m = 0.07, \qquad P_3^m = 0.12.$$

The initial values for $\hat{\gamma}_i$, $\hat{\delta}_{ij}$ and $\hat{\gamma}_*$ are selected to be 0.5. Figs. 2 and 3 show the convergence behavior of $\hat{P}_1$, and $\hat{P}_i^f$ and $\hat{P}_i^m$ $(i = 1,2,3)$, respectively. Each figure contains the average estimate (over 1000 realizations), the average plus and minus one standard deviation, and the behavior of a typical realization. Note that the estimates converge to their true values in (20). (The true values are denoted by the circles in the figures.) The detail of the typical realizations of $\hat{P}_i^f$ and $\hat{P}_i^m$ in Fig. 4 shows that the projection operator in (18) is highly active in the initial stages of the algorithm. However, Figs. 2 and 3 indicate that the projection operator is inactive once the algorithm has obtained reasonable estimates of the probabilities. Fig. 5 shows how the error probability of the fusion center,

$$P_e = \Pr(u_0 = 1 \mid H_0)P_0 + \Pr(u_0 = 0 \mid H_1)P_1 \tag{21}$$

converges to its minimum value as system reaches its steady state. [The minimum value is obtained using (21), the optimal fusion rule (1) and the optimal weights, (2) and (3).]

2) *Tracking*: To illustrate the fusion algorithm's ability to adaptively fuse the local decisions, we

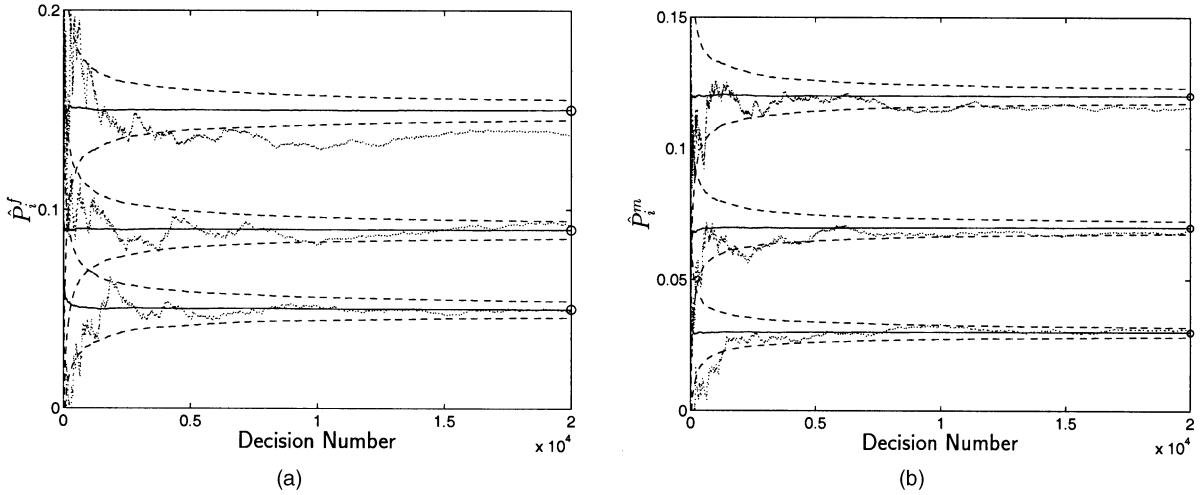


Fig. 3. Adaptation of $\hat{P}_i^f$ and $\hat{P}_i^m$ $(i = 1,2,3)$ for the example in Section IIC1. Legend is same as Fig. 2, except circles denote true values of probabilities being estimated. (a) Adaptation of $\hat{P}_i^f$. (b) Adaptation of $\hat{P}_i^m$.

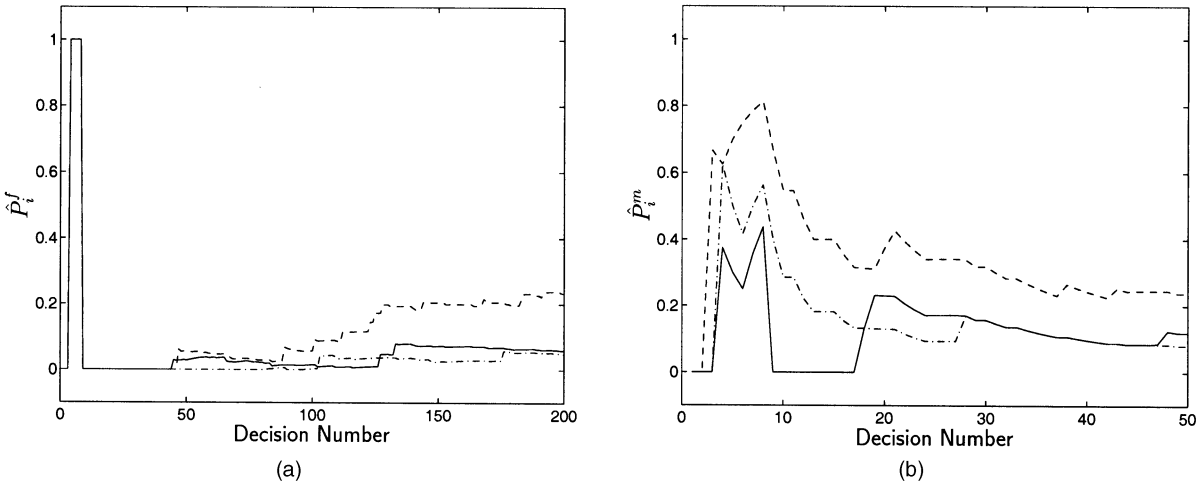


Fig. 4. Detail of adaption of $\hat{P}_i^f$ and $\hat{P}_i^m$ for typical realization of example in Section IIC1. Legend–solid: $i = 1$; dash-dot: $i = 2$; dashed: $i = 3$. Note that time scales of parts (a) and (b) are different. (a) Adaptation of $\hat{P}_i^f$. (b) Adaptation of $\hat{P}_i^m$.

have considered a situation where the underlying a priori probability P_1 is changed in the middle of the simulation. Fig. 6(a) shows how $\hat{P}_1$ adapts when P_1 is changed from 0.7 to 0.85 at the 20,000th decision. Once again, the figures contain the average of the estimates (over 1000 realizations), the average plus and minus one standard deviation, and the behavior of a typical realization. Notice that our estimate $\hat{P}_1$ can still track the change in P_1, but that the convergence is rather slow. In addition, Fig. 7(a) indicates that the estimates of P_i^f are essentially unaffected by the change in P_1. (The corresponding plot for the estimates of P_i^m is similar, but has been omitted for space reasons.) The slow tracking of P_1 is due to the "long memory" of the time averaging process (15)–(17). As a result, the role of the old data fades too slowly. To circumvent this shortcoming, we may use an exponentially weighted time-averaging process in which the contribution from the old data is

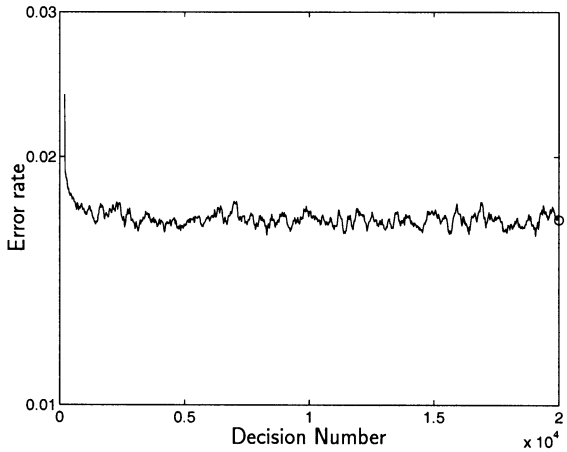


Fig. 5. Error rate of fusion center, on logarithmic scale, for example in Section IIC1. Legend–solid: error rate over previous 200 decisions, averaged over 1000 realizations; circle: minimum P_e.

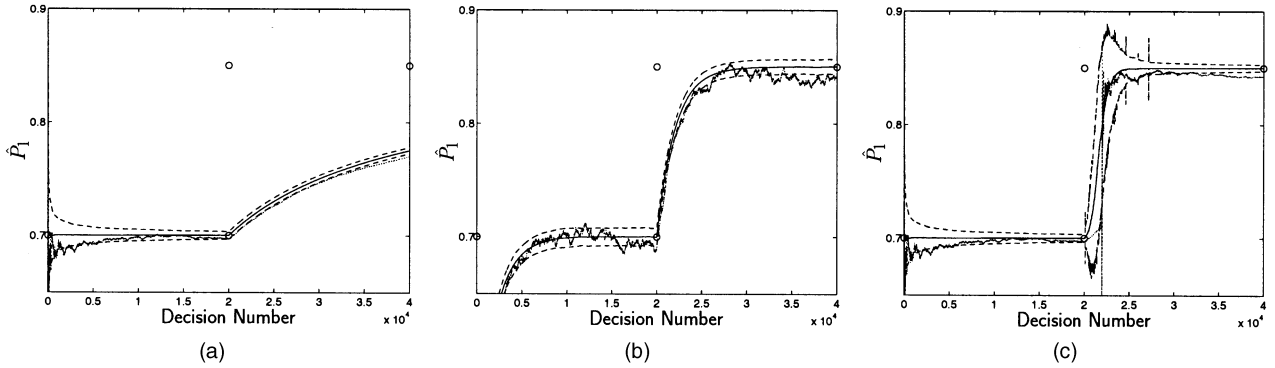


Fig. 6. Adaptation of $\hat{P}_1$ for example in Section IIC2. Legend is same as in Fig. 2, except circles denote true values of P_1 before and after step change at 20,000th decision. (a) Direct form. (b) Exp. weighted, $\lambda = 0.9995$. (c) Slope detection.

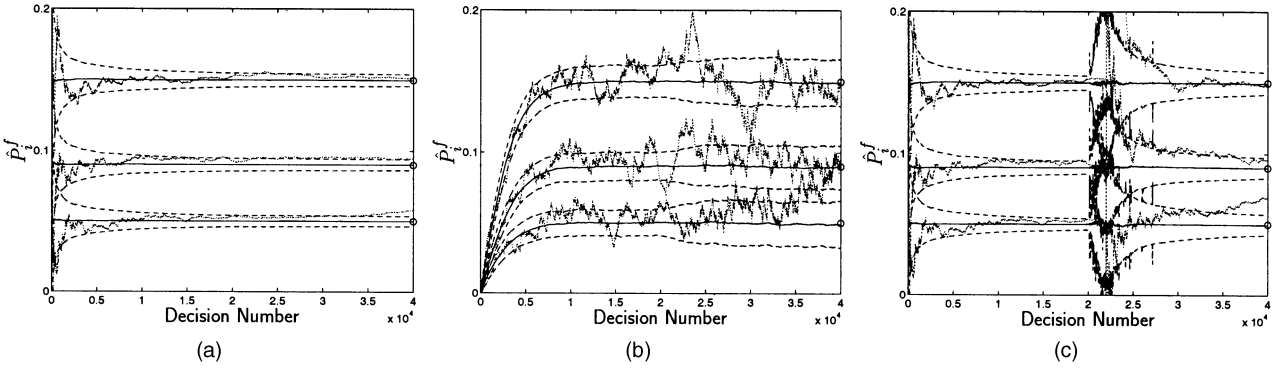


Fig. 7. Adaptation of $\hat{P}_i^f$ $(i = 1,2,3)$ for example in Section IIC2. Legend is same as Fig. 3. (a) Direct form. (b) Exp. weighted, $\lambda = 0.9995$. (c) Slope detection.

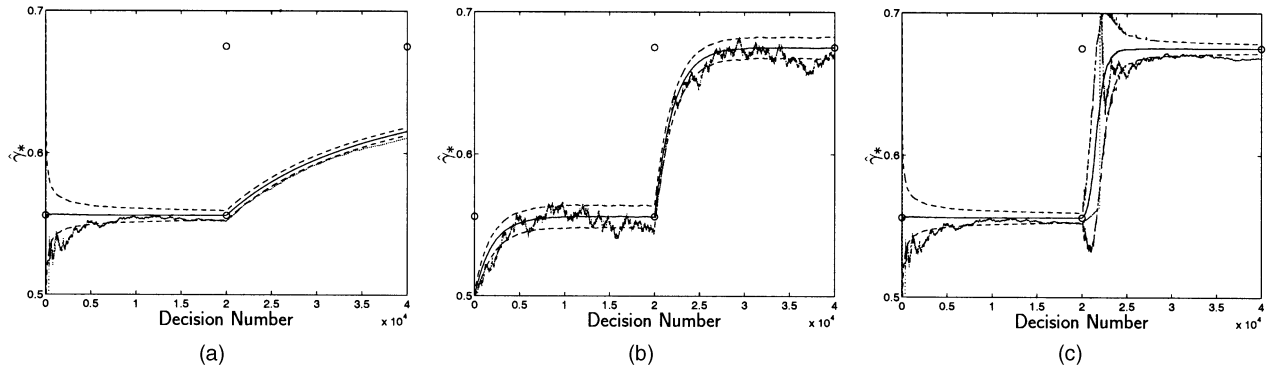


Fig. 8. Adaptation of $\hat{\gamma}_*$ $(i = 1,2,3)$ for example in Section IIC2. Legend is analogous to Fig. 6. (a) Direct form. (b) Exp. weighted, $\lambda = 0.9995$. (c) Slope detection.

reduced exponentially (rather than just $O(1/k^{1/2})$). For example, we might replace (15) by $\hat{\gamma}_i^k = (1-\lambda)u_i^k + \lambda\hat{\gamma}_i^{k-1}$, where λ is a positive constant just less than one. Although this exponentially weighted scheme is able to track changes more quickly than our original algorithm, the variance of the estimates does not decay to zero (see part (b) of Figs. 6–8). This results in sub-optimal asymptotic error rate performance (see Fig. 9).

An alternative to the exponentially weighted scheme, and other window-based schemes, is to use the "slope detection" procedure described below to detect changes in the environment and then restart the algorithm to take advantage of the fast convergence of the direct form of the algorithm.

3) *Fast Adaptation by Slope Detection*: According to (28) in Appendix A, γ_* is a function of all unknown probabilities: P_1, the P_i^ms and the P_i^fs. Therefore, any change in one of these a priori probabilities will affect the value of γ_* and its estimated value $\hat{\gamma}_*$. For example, Fig. 8(a) shows how $\hat{\gamma}_*$ changes when P_1 is increased. Observe the slow adaptation of the algorithm as P_1 is changed. This slow convergence is in contrast to the relatively

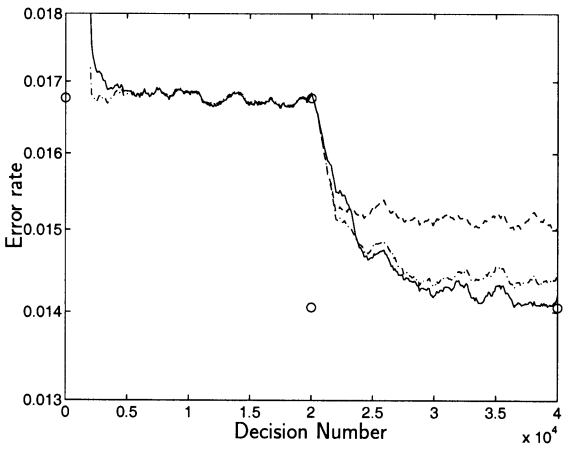


Fig. 9. Error rate of the fusion center over previous 2000 decisions for example in Section IIC2, averaged over 997 out of 1000 realizations for which slope detector produced no false alarms nor missed detections. Legend–dashed: direct form; dash-dot: exponentially weighted, $\lambda = 0.9995$; solid: direct form with slope detection; circle: minimum P_e on each side of step change in P_1.

fast convergence in the initial transient phase of the algorithm; see Fig. 8(a). This suggests that we can increase the speed of adaptation by "restarting" the algorithm when a change in $\hat{\gamma}_*$ is detected and using the fast initial transient convergence to track the changes in γ_*. The "restart" entails reinitialization of the stochastic time-averaging processes (15)–(17). The slow adaptation produces a non-zero slope as shown in Fig. 8(a). Therefore, the change of parameters can be detected by observing the behavior of $\hat{\gamma}_*$ and detecting a non-zero slope. To do so, we observe $\hat{\gamma}_*$ over a (sliding) window of K decisions. If the absolute value of $\Delta_{\hat{\gamma}_*} = \hat{\gamma}_*{}^k - \hat{\gamma}_*{}^{k-K+1}$ is greater than a predetermined fraction ρ, of the average value of $\hat{\gamma}_*{}^k$ over the window, then the slope over the window is deemed to be significant, and a counter is incremented or decremented according to the sign of $\Delta_{\hat{\gamma}_*}$. When the absolute value of the counter exceeds a given threshold T, then we recognize a non-zero slope. Since we expect $\hat{\gamma}_*{}^k$ to change during the transient phases of the algorithm, so the counter is disabled for a predetermined "settling period" after a restart. The details of the procedure are described below.

Slope Detection Method

Initial Step: Set counter to zero, and select appropriate values for K, ρ, T and the settling time. Execute the following steps after each iteration of the adaptive fusion algorithm once the number of decisions since the last restart is greater than the settling time.

Main Steps:

a) Compute (recursively) the average value of $\hat{\gamma}_*{}^k$ over the window; i.e., $\mu_{\hat{\gamma}_*} = 1/K \sum_{m=k-K+1}^{k} \hat{\gamma}_*{}^m$.

b) Compute $\Delta_{\hat{\gamma}_*} = \hat{\gamma}_*{}^k - \hat{\gamma}_*{}^{k-K+1}$.

c) If $|\Delta_{\hat{\gamma}_*}| > \rho\mu_{\hat{\gamma}_*}$, set counter := counter + sgn($\Delta_{\hat{\gamma}_*}$).

d) If the absolute value of counter is greater than T, report a non-zero slope, restart the adaptive fusion algorithm (i.e., reinitialize the time-averaging processes (15)–(17)) and reset counter.

Part (c) of Figs. 6–8 and Fig. 9 illustrate the performance of our algorithm equipped with this slope detection method. The parameters of the slope detector are $K = 100$, $\rho = 10^{-5}$, $T = 750$ and the settling time was 10,000 decisions. The improved tracking of the scheme with the slope detector is clear from these figures. Of course, the performance of the slope-detection-based scheme is sensitive to the false alarm and missed detection probabilities of the slope detector. In our 1000 trials, there were 3 trials with a single false alarm and no trials with missed detection. These trials were removed before Fig. 9 and part (c) of Figs. 6–8 were generated.

III. ADAPTIVE DECISION FUSION FOR N DETECTOR CASE

Let us now consider the problem of adaptive decision fusion for a distributed detection system with N detectors ($N > 3$). In this case, we have $2N + 1$ unknown parameters to be estimated: P_1, and P_i^f and P_i^m ($i = 1, \ldots, N$). In what follows, we propose a method which is a generalized version of the adaptive fusion algorithm developed for the three detector case (Section II).

A. N-Detector Adaptive Fusion Method

Using (26) and (27) in Appendix A, we have

$$P_j^f = \frac{\dfrac{P_1\delta_{ij} - \gamma_i\gamma_j}{1 - P_1} + \gamma_j P_i^f}{-\gamma_i + P_i^f} \tag{22}$$

and

$$P_j^m = 1 + \frac{\dfrac{\gamma_i\gamma_j - (1 - P_1)\delta_{ij}}{P_1} - \gamma_j(1 - P_i^m)}{1 - \gamma_i - P_i^m}. \tag{23}$$

Equations (22) and (23) show a relationship between the probabilities P_i^f, P_i^m of detector i and the probabilities P_j^f, P_j^m of detector j, for each pair of i, j. Thus, if we have estimated the probabilities P_i^f, P_i^m for detector i, we can use (22) and (23) to obtain estimates of probabilities P_j^f, P_j^m for any other detector j. This observation suggests that we can arbitrarily select any set of three detectors, say LD_1, LD_2, LD_3, and use the adaptive fusion algorithm in Section IIB on these three detectors. As a result, we obtain estimates for P_1, and P_i^f, P_i^m for these three detectors. Next, we select one of the three detectors as the "reference detector," say LD_1. Then by using (22) and the estimated probability of false alarm of the

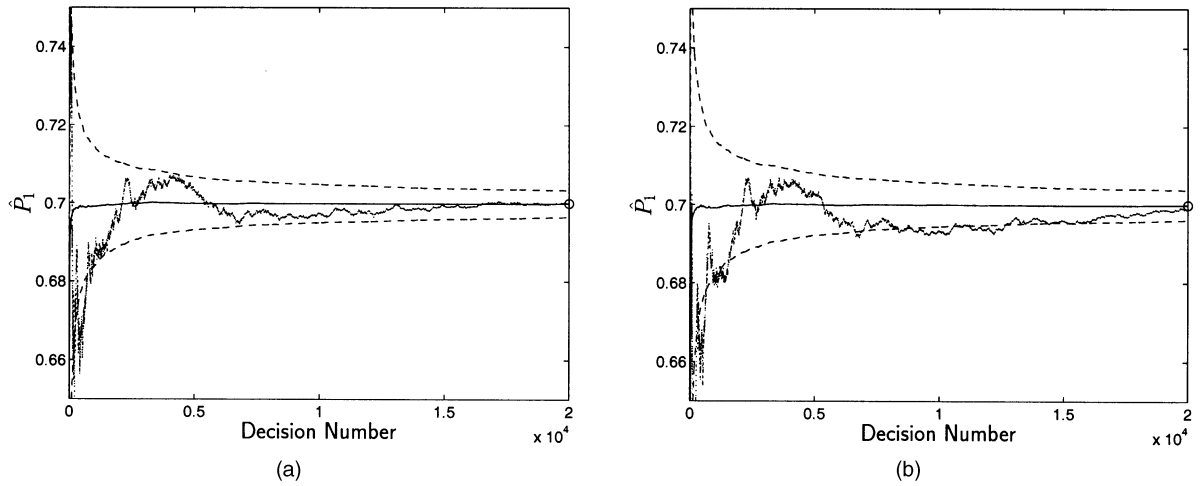


Fig. 10. Adaptation of $\hat{P}_1$ for example of Section IIIB. Legend is same as Fig. 2. (a) Reference detector is detector 1. (b) Reference detector is detector 5.

reference detector $\hat{P}_1^f$, we can calculate the estimates for P_j^f $(j = 4,\ldots,N)$. Similarly, we can use (23) and $\hat{P}_1^m$ to calculate $\hat{P}_j^m$ $(j = 4,\ldots,N)$. In this estimation process, we need to estimate $2N + 1$ unconditional joint probabilities: γ_i $(i = 1,\ldots,N)$, δ_{1j} $(j = 2,\ldots,N)$, δ_{23} and γ_*. The details of the N-detector adaptive fusion method are summarized below.

N-Detector Adaptive Fusion Method

Initial Step: Select a set of three arbitary detectors, say, LD_1, LD_2, and LD_3. Let LD_1 be the reference detector. Select initial values: $\hat{\gamma}_i^0$ $(i = 1,\ldots,N)$, $\hat{\delta}_{1j}^0$ $(j = 2,\ldots,N)$, $\hat{\delta}_{23}^0$ and $\hat{\gamma}_*^0$.

Iterations:

a) Use the adaptive fusion algorithm described in Section IIB to update $\hat{P}_i^f$, $\hat{P}_i^m$ $(i = 1,2,3)$ and $\hat{P}_1$.

b) Use time-averaging of local decisions u_i to update the values of $\hat{\gamma}_i$ $(i = 4,\ldots,N)$ and $\hat{\delta}_{1i}$ $(i = 4,\ldots,N)$; c.f., (15) and (16).

c) Use (22) and (23) and the updated values of $\hat{P}_i^f$, $\hat{P}_i^m$ $(i = 1,2,3)$ and $\hat{P}_1$ (see Step (b)) to calculate $\hat{P}_i^f$ and $\hat{P}_i^m$ $(i = 4,\ldots,N)$. Note that the projection operator $[\cdot]_+$ in (18) will also be required here, but that it can be simplified because its argument is always real.

d) Use (2) and (3) to calculate estimates of the optimal fusion weights, w_0, w_i, $i = 1,2,3,\ldots,N$.

We point out that an analysis similar to Theorem 2 can be performed on the above N-Detector adaptive fusion method. However, we do not provide the details here as they are largely the same as those in Theorem 2. Essentially, we can prove that the estimators provided by the above N-detector adaptive fusion method are asymptotically unbiased, and that the estimators converge to their final true values at the rate $O(1/k^{1/2})$ (measured in the rms error sense), where k is the iteration number.

B. Simulation Results

We now present some computer simulations to examine the effectiveness of the N-detector adaptive fusion method. In these simulations, we considered a distributed detection system with five local detectors and set:

$$\begin{cases} P_1 = 0.7, & P_1^f = 0.05, & P_2^f = 0.1, \\ P_3^f = 0.08, & P_4^f = 0.03, & P_5^f = 0.12, \\ P_1^m = 0.02, & P_2^m = 0.05, & P_3^m = 0.1, \\ P_4^m = 0.12, & P_5^m = 0.2. & \end{cases} \tag{24}$$

The initial values for $\hat{\gamma}_i$, $\hat{\delta}_{ij}$ and $\hat{\gamma}_*$ are selected to be 0.5. We ran the N-detector adaptive fusion method for the case where detector 1 was the reference detector, and detectors 2 and 3 were the other detectors used in part (a) of the algorithm. The results are shown in Figs. 10(a) and 11(a). It can be seen that $\hat{P}_1$, and the $\hat{P}_i^f$s converge to their true values. (The convergence plot of the $\hat{P}_i^m$s is similar to that of the $\hat{P}_i^f$s, but has been omitted for space reasons.) In that scenario, the reference detector was the most reliable detector. A useful property of our algorithm is that the quality of the estimates of P_1 and the $\hat{P}_i^f$s and $\hat{P}_i^m$s depends (only) on the accuracy of the estimates of the false alarm and missed detection probabilities of the reference detector. As a result, our algorithm is rather insensitive to the error rate of the reference detector itself. This property is illustrated in Figs. 10(b) and 11(b), where detector 5, the most error prone detector, is chosen as the reference detector. (Detectors 2 and 3 are retained as the other detectors in part (a) of the algorithm. The simulations which generated parts (a) and (b) of Figs. 10 and 11 contained exactly the same sequences of local decisions.) The solid and dotted curves in Fig. 12 show how the error probability of the fusion center P_e converges to its minimum value when the system reaches to its steady state. Although

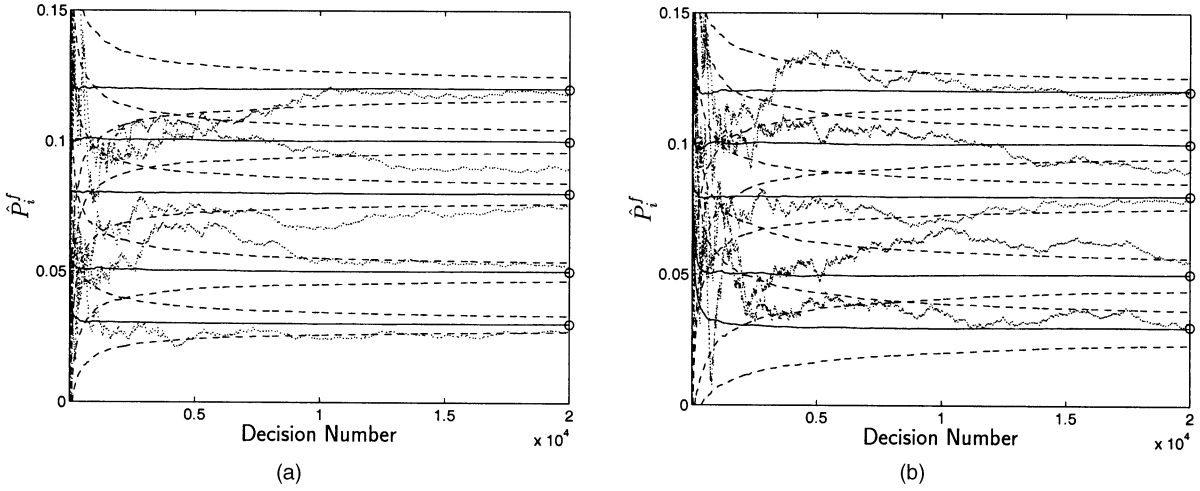


Fig. 11. Adaptation of $\hat{P}_i^f$ $(i = 1,\ldots,5)$ for example of Section IIIB. Legend is same as Fig. 3. (a) Reference detector is detector 1. (b) Reference detector is detector 5.

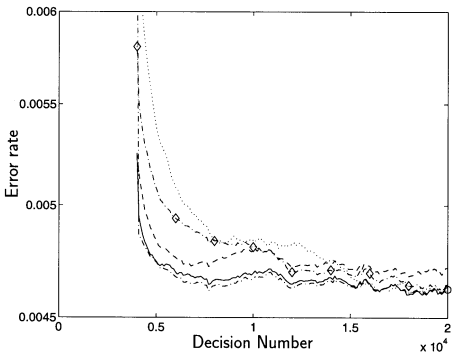


Fig. 12. Error rate of fusion center over previous 4000 decisions, for example in Section IV, averaged over 795 out of 1000 realizations for which direct form of Naim and Kam's algorithm did not fail. Legend–solid: our algorithm (ref. detect.= 1); dotted: our algorithm (ref. detect.= 5); dash-dot, ⋄: Naim and Kam [9]; dash-dot: our modification of Naim and Kam; dashed: Ansari et al. [2], with counters initialized to 10 and $\alpha = 0.3$; circle: minimum P_e.

both choices of the reference detector provide good performance, the slightly faster convergence of $\hat{P}_1$, and the $\hat{P}_i^f$s and $\hat{P}_i^m$s when the reference detector is detector 1 (see, e.g., the dashed curves in Fig. 11) results in a tangible improvement in the convergence speed of the error rate of the fusion center.

IV. COMPARISON WITH EXISTING METHODS

In this section, we compare the performance of our algorithm with those of Naim and Kam [9] and Ansari et al. [2]. Like our algorithm, Naim and Kam's algorithm [9] estimates P_1, P_i^f, and P_i^m and then uses these values to calculate estimates of the weights w_i in (1). In contrast, Ansari's algorithm [2] estimates the weights directly. As mentioned in the Introduction, our algorithm has the desirable property that the estimates are asymptotically unbiased (see Theorem 2), whereas the estimates generated by the existing algorithms in [2, 9] are biased. This bias arises because both existing algorithms are based on the assumption that the decision of the fusion center is correct. Naim and Kam [9] employ a bias estimation and compensation technique which reduces the bias of their method, but greatly increases its computational cost. Ansari et al. [2] attempt to reduce the bias of their algorithm by using a threshold test to determine whether the fusion center's decision is reliable in the absence of a given detector's local decision. Those central decisions which are not reliable in the absence of that detector's local decision are not used to update the estimates of that detector's weight. Unfortunately, removing these decisions may compromise the convergence behavior of the algorithm, as shown below. Furthermore, optimal selection of the threshold for determining reliability remains an unresolved problem.

To compare the performance of these algorithms in practice, we simulated them in the scenario of Section IIIB. Each algorithm was exposed to 1000 identical realizations of 20,000 local decisions from each detector. The implementation of our algorithm followed the method outlined in Section IIIA, with the initial values of $\hat{\gamma}_i$, $\hat{\delta}_{ij}$ and $\hat{\gamma}_*$ each being 0.5. This leads to reliable performance after 20,000 decisions, as shown in Section IIIB and Table I. However, our direct implementation of Naim and Kam's algorithm (described in Appendix C) provided rather unreliable performance after 20,000 decisions in this scenario, especially when the bias estimation and compensation step was used, as shown in Table I. Our direct implementation of Ansari's algorithm (described in Appendix C) required some fine tuning of the initialization in order to obtain reliable performance (see Table I). The results in Table I are discussed in more detail below, but first we address the unreliable performance of the existing algorithms.

The convergence failure of the existing algorithms appears to be due to the bias induced by the premise

TABLE I
Reliability and Offsets for Various Algorithms in Example in Section IV

	Our Method		Naim and Kam [9]				Ansari *et al.* [2]							
	ref. detect.		direct		modified		count. init. = 5,‡ thresh. coeff. α,§				count. init. = 10 thresh. coeff. α			
	1	5	nbc*	bc†	nbc	bc	0	0.1	0.3	0.4	0	0.1	0.3	0.4
Failed runs (from 1000)¶														
Partial failure	0	0	46	127	0	0	37	40	37	52	0	0	0	3
Mild failure	0	0	0	60	0	0	9	9	3	5	0	0	0	0
Moderate failure	0	0	3	14	0	0	20	7	1	0	0	0	0	0
Complete failure	0	0	3	4	0	0	32	29	53	116	0	0	0	0
Total	0	0	52	205	0	0	98	85	94	173	0	0	0	3
Offsets ($\times 10^{-4}$),‖														
in $\hat{w}_0$,	3.36	1.70	28.5	11.2	8.48	6.69	5.51	15.1	18.7	47.3	47.3	54.9	54.5	77.4
in $\hat{w}_i$ for $u_i = 1$,**	17.7	103	293	220	227	166	194	205	198	349	403	373	229	393
in $\hat{w}_i$ for $u_i = 0$,**	24.0	31.5	128	151	116	147	129	233	491	805	181	284	584	861

* The bias estimation and compensation step was not used.
† The bias estimation and compensation step was used.
‡ Initial value of the counters; see Appendix C for more details.
§ The absolute value of the reliability threshold was set to be $\alpha y_{\max}$, as in Section 4.2 of [2]; see Appendix C for more details.
¶ Mis- or slow convergence of at least one weight leading to a fusion centre error rate over decisions 15,000–20,000 substantially greater than the minimum error rate—Partial: 2–5 times the minimum error rate; Mild: 5–10 times; Moderate: 10–20 times; Complete: more than 20 times. (These ranges were selected following a study of error histograms.)
‖ If $\bar{\tilde{w}}$ represents the average value of the weight at the 20,000th decision (averaged over the 650 out of 1000 realizations for which none of the methods in the table failed), and if w_{opt} represents the optimal value, the offset is defined to be $|\bar{\tilde{w}} - w_{\text{opt}}|$.
** Averaged over the five sensors

that the fusion center's decision is correct. Any detector whose (local) decision agrees with that of the fusion center has its weight increased. In Ansari's algorithm the weight is increased directly, whereas in Naim and Kam's algorithm it is increased indirectly through a reduction of $\hat{P}_i^f$ or $\hat{P}_i^m$. As a result, the fusion center's next decision is more heavily influenced by that detector (if that detector makes the same local decision at the next instant). Hence, it is more likely that this detector's next local decision will agree with the fusion center's next decision. Consequently, the weight of a detector (or detectors) may grow without bound, leading to central decisions which are influenced by only that detector (or detectors). Most of the "failures" in Table I were caused in this way. In a small number of realizations, the algorithms are able to "rein in" this "explosive" weight growth, but the resulting convergence times can be very long (see, e.g., Fig. 14(b), below). A few of the failures in Table I occurred when the algorithm was recovering from initially explosive weight growth, but was unable to completely recover in under 20,000 decisions. In our experiments, we found that for both existing algorithms, the occasional explosive weight growth was exacerbated by the rather large updating steps that the algorithms take for the first few updates after the initialization.

In order to obtain fair comparisons between our algorithm and Naim and Kam's algorithm [9], we modified their algorithm by bounding the false alarm and missed detection probability estimates away from zero and one, and by adjusting the initialization (see Appendix C for the details). As indicated in Table I, this leads to a reliable algorithm. However, it does require prior knowledge of the performance ranges of the detectors. This knowledge was not made available to our algorithm. To obtain a fair comparison with Ansari's algorithm [2], we have provided results for several settings of the initial counter values in that algorithm (see Appendix C for the details). Increasing these initial values reduces the step size of the algorithm and hence increases convergence times, but as can bee seen from Table I, doing so leads to more reliable performance. We have included several settings of the reliability threshold for Ansari's algorithm in the table in order to give an indication of the achievable performance. When the threshold is small (i.e., α is small) almost all of the central decisions are considered reliable and as the threshold is increased (i.e., α is increased) more of the central decisions are considered unreliable and hence are excluded from the weight updating process. As one might expect, this can improve performance, but if the threshold is too large, then convergence problems can arise. We point out that the setting of the reliability threshold involves some prior knowledge of the performance of the detectors. This knowledge was not made available to our algorithm.

In addition to the reliability information, we have also included the weight estimation offsets at the 20,000th decision in Table I. Since our algorithm is asymptotically unbiased and the existing algorithms are biased, one might expect our algorithm to produce

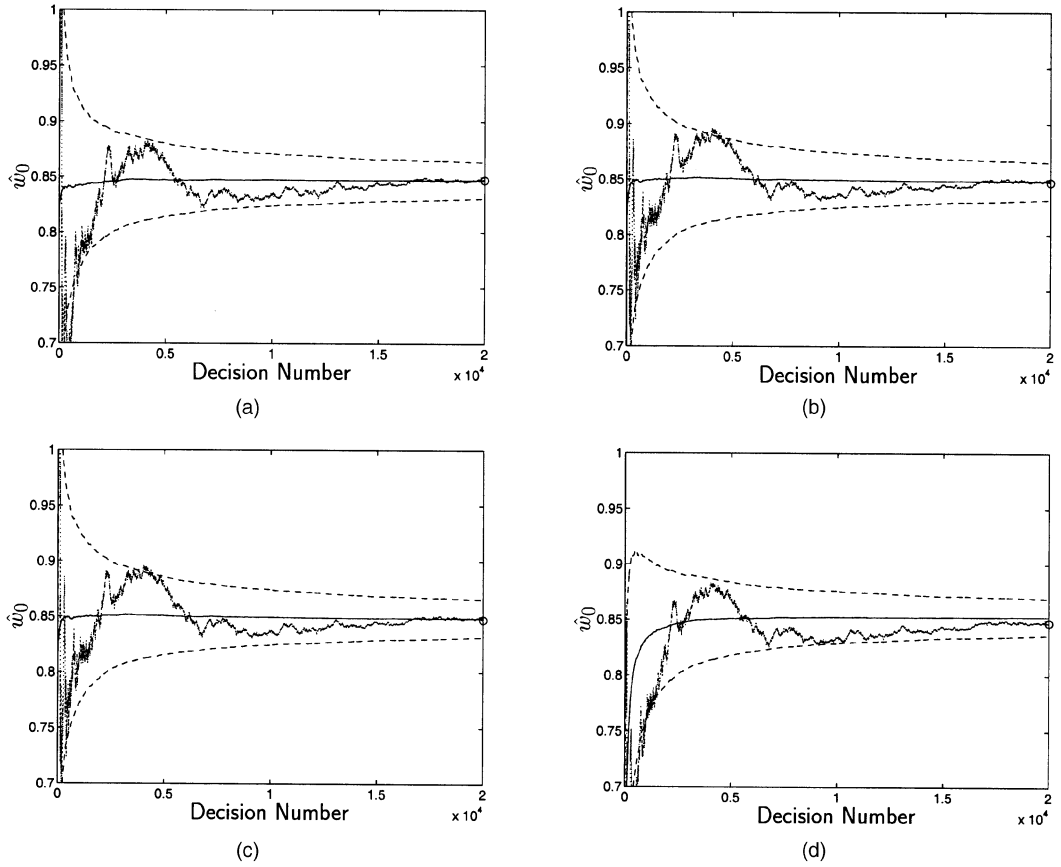


Fig. 13. Adaptation of $\hat{w}_0$ for 795 out of 1000 realizations for which direct form of Naim and Kam's algorithm did not fail in example in Section IV. Legend is analogous to Fig. 2. (a) Our algorithm. (b) Naim and Kam [9]. (c) Modified Naim and Kam. (d) Ansari et al. [2].

substantially lower offsets, and this is indeed the case. It is interesting to note that Naim and Kam's bias estimation and compensation step reduces the weight offset when $u_i = 1$ at the expense of increasing the weight offsets for $u_i = 0$. This is probably because in our scenario the probability that $u_i = 1$ is greater than $1/2$; i.e., $P_1(1 - P_i^m) + (1 - P_1)P_i^f > 1/2$. Hence the weights for $u_i = 1$ are used more often than those for $u_i = 0$. Similarly, Ansari's reliability threshold trades an increase in the weight offsets for $u_i = 0$ for a reduction in the offsets for $u_i = 1$.

The error rate performance of the fusion center for five of the methods from Table I is provided in Fig. 12. For our method we have plotted curves for both choices of the reference detector in Table I. For Naim and Kam's algorithm we have plotted curves for the direct implementation and the modified version with bias estimation and compensation, and for Ansari's algorithm we have initialized the counters to 10 and set $\alpha = 0.3$. These settings provided the best performance at convergence of the instances of Ansari's algorithm in Table I. Note, however, that the bias in Ansari's algorithm is evident from its slightly higher error rate at convergence. From the figure, it can also be seen that our algorithm (with detector 1 as the reference detector) provides better performance than the direct implementation of the existing algorithms and is competitive with our modified version of Naim and Kam's algorithm. (Recall that our modified version of Naim and Kam's algorithm has information regarding the range of sensor performance, whereas our algorithm does not; see Appendix C for further details.) Our algorithm achieves this performance reliably and at a computational cost which is far lower than that of Naim and Kam's algorithm (with bias estimation and compensation) and is competitive with Ansari's algorithm.

To further compare the performance of the algorithms, we have plotted the convergence behavior of the estimates of the weights in Figs. 13–15. The methods considered are our method (with the first detector as the reference detector), the direct implementation and our modified version of Naim and Kam's algorithm, and Ansari's method with counters initialized to 10 and $\alpha = 0.3$. The plots have been generated from the 795 out of 1000 realizations for which the direct form of Naim and Kam's algorithm did not fail. It is clear from Fig. 13 and the first row of Table I that our algorithm and both versions of Naim and Kam's algorithm obtain good estimates of w_0 after 20,000 decisions. The typical realization

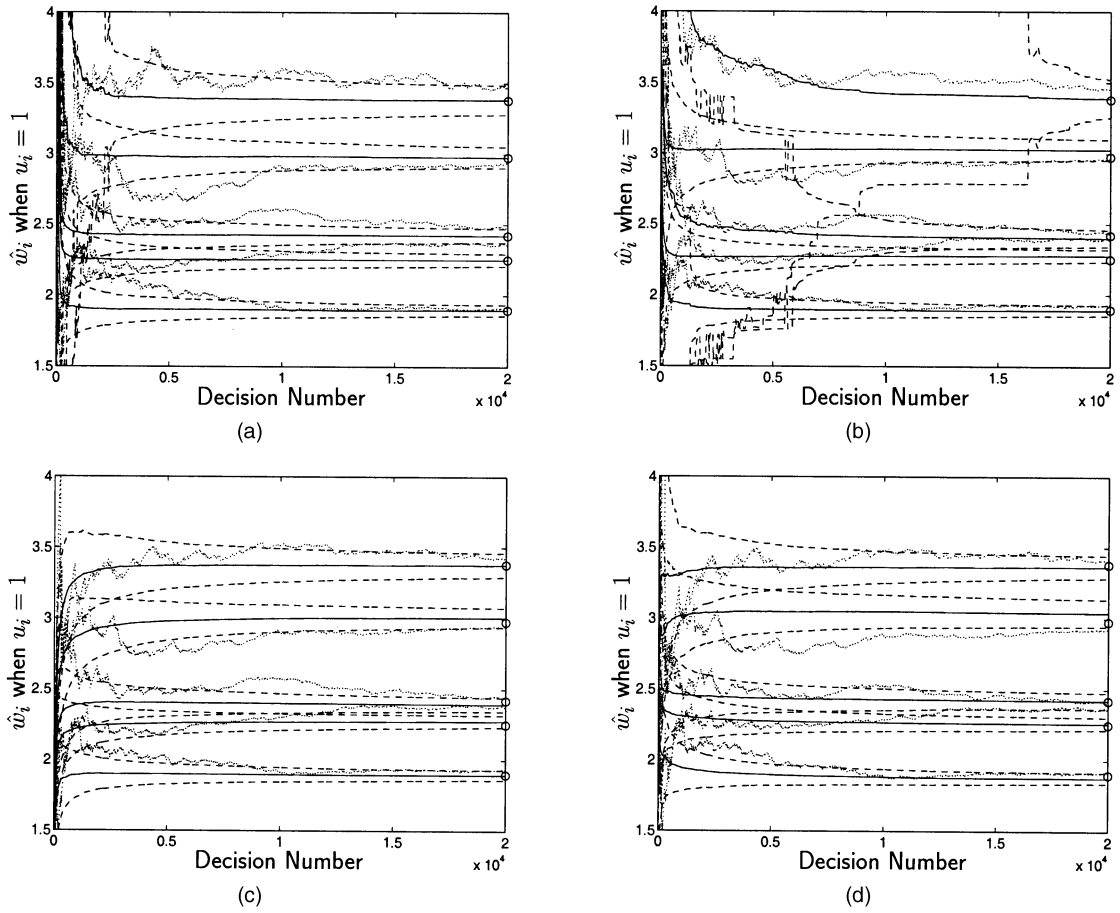


Fig. 14. Adaptation of $\hat{w}_i$ when $u_i = 1$ for 795 out of 1000 realizations for which direct form of Naim and Kam's algorithm did not fail in the example in Section IV. Legend is analogous to Fig. 3. (a) Our algorithm. (b) Naim and Kam [9]. (c) Modified Naim and Kam. (d) Ansari et al. [2].

of Ansari's algorithm in Fig. 13(d) also obtains a good estimate of w_0 after 20,000th decisions, but on average, the bias in Ansari's method is visible at the scale of Fig. 13. The bias of the direct and modified versions of Naim and Kam's algorithm and Ansari's algorithm is apparent from the estimates of w_i for $u_i = 1$ and $u_i = 0$ plotted in Figs. 14 and 15, respectively. In contrast, the offsets for our method at the 20,000th decision are negligible at the scale of the figures. Parts (b) and (c) of Figs. 14 and 15 also highlight the improved performance of our modified version of Naim and Kam's algorithm. The apparent slow decay of the standard deviation of three weight estimates (two in Fig. 14(b) and one in Fig. 15(b)) for the direct implementation of Naim and Kam's algorithm is actually due to a small number of realizations (less than 10) in which a weight initially "exploded" but was eventually "reined in." (In another 205 realizations the direct implementation of Naim and Kam's algorithm was unable to rein in an exploding weight by the 20,000th decision, leading to the failure designation.) In contrast, weight explosion is explicitly contained in our modified version of Naim and Kam's algorithm. This helps to eliminate the failures (see Table I) and provides faster convergence of the weights (see part (c) of Figs. 14 and 15).

V. CONCLUDING REMARKS

In real world fusion applications, the probability of the hypothesis and the performance of local detectors may be unknown or variable. Under such circumstances, we need an adaptive fusion center in order to obtain optimal performance. In this paper, we have proposed a recursive algorithm based on the time-averaging of local decisions and the use of these time averages to estimate the error probabilities of the local detectors and the a priori probabilities. The heart of this algorithm is an explicit analytic solution of the problem in the three-detector case. The algorithm is suitable for a time-varying environment. The estimators provided by our adaptive fusion algorithms have been shown to be asymptotically unbiased, and the rate at which the estimators converge to their true values has been shown to be $O(1/k^{1/2})$ (in the rms error sense), where k is the iteration number. Furthermore, our simulation studies suggest that our algorithm is substantially more reliable than the two existing (asymptotically biased) algorithms [2, 9] and

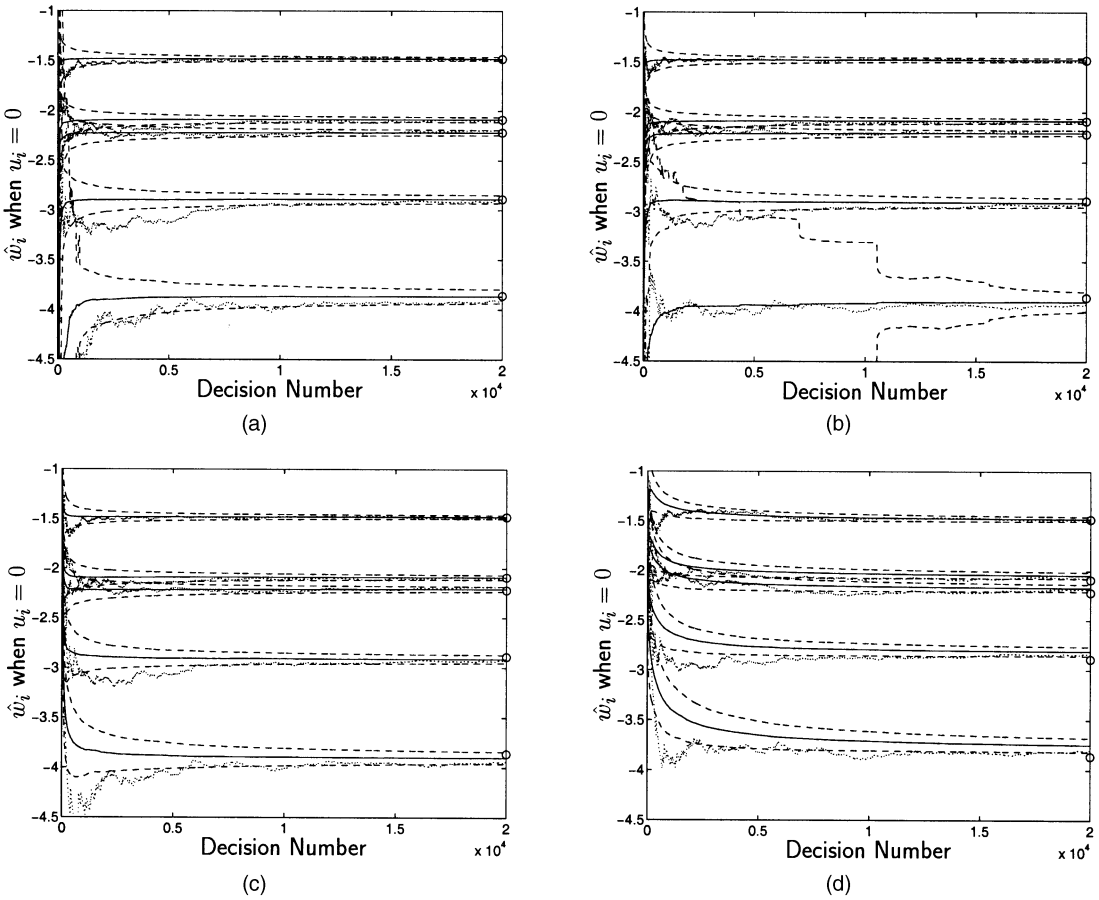


Fig. 15. Adaptation of $\hat{w}_i$ when $u_i = 0$ for 795 out of 1000 realizations for which direct form of Naim and Kam's algorithm did not fail in the example in Section IV. Legend is analogous to Fig. 3. (a) Our algorithm. (b) Naim and Kam [9]. (c) Modified Naim and Kam. (d) Ansari et al. [2].

performs at least as well as those methods when they work.

In closing, we mention a few possible extensions of our work.

1) *Adaptive Decision Fusion for Neyman–Pearson Distributed Detection*: In this paper we used an optimal fusion rule based on the minimum probability of error. In some applications (e.g., radar) the goal may be to minimize the probability of missed detection induced by the final decision-maker, subject to the constraint that the final false alarm rate does not exceed a given constant value. This approach is called the Neyman–Pearson approach; see [11, 12] for a discussion on the optimal decision fusion rule for the Neyman–Pearson criterion. It will be interesting to see if our adaptive fusion rule can be modified to suit the Neyman–Pearson criterion.

2) *Adaptive Multilevel Decision Fusion*: In this paper, we considered binary decisions. It will be interesting to see whether one can extend the results to the case of multilevel decisions. The concept of multilevel decision fusion is considered in [8].

3) *Adaptive Detection at Local Detectors*: Here we assumed the local decision rules are fixed. Is it possible to use the error probability estimates obtained in this work to adjust detection thresholds at each local detector? A related question was considered in [9].

APPENDIX A: PROOF OF THEOREM 1

First of all, we notice that by definition

$$\begin{cases} \gamma_i = \Pr(u_i = 1), \\ \delta_{ij} = \Pr(u_i = 1, u_j = 1), \\ \gamma_* = \Pr(u_1 = 1, u_2 = 1, u_3 = 1), \end{cases} \quad i,j = 1,2,3, \quad i \neq j. \tag{25}$$

Using these definitions we can simplify (8) algebraically to obtain

$$\gamma_i = (1 - P_i^m)P_1 + P_i^f(1 - P_1), \qquad i = 1,2,3 \tag{26}$$

$$\delta_{ij} = (1 - P_i^m)(1 - P_j^m)P_1 + P_i^f P_j^f (1 - P_1), \quad i,j = 1,2,3, \quad j \neq i \tag{27}$$

$$\gamma_* = (1 - P_1^m)(1 - P_2^m)(1 - P_3^m)P_1 + P_1^f P_2^f P_3^f (1 - P_1), \tag{28}$$

To solve the above nonlinear equations, we first use (26) and (27) to compute

$$\begin{aligned}
&\delta_{12}-\gamma_1\gamma_2\\
&\quad=(1-P_1^m)(1-P_2^m)P_1+P_1^fP_2^f(1-P_1)\\
&\qquad-((1-P_1^m)P_1+P_1^f(1-P_1))((1-P_2^m)P_1+P_2^f(1-P_1))\\
&\quad=(1-P_1^m)(1-P_2^m)P_1+P_1^fP_2^f(1-P_1)\\
&\qquad-(1-P_1^m)(1-P_2^m)P_1^2\\
&\qquad-P_1^fP_2^f(1-P_1)^2-P_1^f(1-P_2^m)P_1(1-P_1)\\
&\qquad-(1-P_1^m)P_2^fP_1(1-P_1)\\
&\quad=P_1(1-P_1)((1-P_1^m)(1-P_2^m)+P_1^fP_2^f-P_1^f(1-P_2^m)\\
&\qquad-(1-P_1^m)P_2^f)\\
&\quad=P_1(1-P_1)(1-P_1^m-P_1^f)(1-P_2^m-P_2^f). \qquad (29)
\end{aligned}$$

Similarly, we have

$$\begin{cases}\delta_{23}-\gamma_2\gamma_3=P_1(1-P_1)(1-P_2^m-P_2^f)(1-P_3^m-P_3^f)\\ \delta_{13}-\gamma_1\gamma_3=P_1(1-P_1)(1-P_1^m-P_1^f)(1-P_3^m-P_3^f).\end{cases} \qquad (30)$$

By assumption (9), we have

$$\delta_{12}-\gamma_1\gamma_2>0,\qquad \delta_{23}-\gamma_2\gamma_3>0,\qquad \delta_{13}-\gamma_1\gamma_3>0.$$

Let us define

$$\begin{aligned}
a_1&=\sqrt{\frac{(\delta_{12}-\gamma_1\gamma_2)(\delta_{13}-\gamma_1\gamma_3)}{\delta_{23}-\gamma_2\gamma_3}}\\
a_2&=\sqrt{\frac{(\delta_{12}-\gamma_1\gamma_2)(\delta_{23}-\gamma_2\gamma_3)}{\delta_{13}-\gamma_1\gamma_3}} \qquad (31)\\
a_3&=\sqrt{\frac{(\delta_{13}-\gamma_1\gamma_3)(\delta_{23}-\gamma_2\gamma_3)}{\delta_{12}-\gamma_1\gamma_2}}.
\end{aligned}$$

Then, the preceding relations (29)–(30) easily imply

$$a_i=\sqrt{P_1(1-P_1)}(1-P_i^m-P_i^f),\qquad i=1,2,3.$$

Substituting this into (26) yields

$$P_i^f=\gamma_i-a_i\sqrt{\frac{P_1}{1-P_1}},\qquad P_i^m=1-\gamma_i-a_i\sqrt{\frac{1-P_1}{P_1}},\qquad i=1,2,3. \qquad (32)$$

Thus, if P_1 is known, then all the remaining parameters P_i^f, P_i^m $(i=1,2,3)$ can be readily determined using (32).

It remains to determine P_1. To this end, we substitute (32) back into (28) to obtain

$$\begin{aligned}
\gamma_*&=\prod_i\left(\gamma_i+a_i\sqrt{\frac{1-P_i}{P_1}}a_i\right)P_1\\
&\quad+\prod_i\left(\gamma_i-a_i\sqrt{\frac{P_i}{1-P_1}}\right)(1-P_1)\\
&=P_1\left(\gamma_1\gamma_2\gamma_3+\left(\frac{1-P_1}{P_1}\right)^{3/2}a_1a_2a_3\right.\\
&\qquad\left.+b\sqrt{\frac{1-P_1}{P_1}}+c\frac{1-P_1}{P_1}\right)\\
&\quad+(1-P_1)\left(\gamma_1\gamma_2\gamma_3-\left(\frac{P_1}{1-P_1}\right)^{3/2}a_1a_2a_3\right.\\
&\qquad\left.-b\sqrt{\frac{P_1}{1-P_1}}+c\frac{P_1}{1-P_1}\right)\\
&=\gamma_1\gamma_2\gamma_3+c\\
&\quad+a_1a_2a_3\left((1-P_1)\sqrt{\frac{1-P_1}{P_1}}-P_1\sqrt{\frac{P_1}{1-P_1}}\right)
\end{aligned}$$

where

$$\begin{aligned}
b&:=\gamma_1\gamma_2a_3+\gamma_1\gamma_3a_2+\gamma_2\gamma_3a_1,\\
c&:=\gamma_1a_2a_3+\gamma_2a_1a_3+\gamma_3a_1a_2.
\end{aligned}$$

Therefore, if we define

$$\begin{aligned}
X&=\frac{\gamma_*-\gamma_1\gamma_2\gamma_3-c}{a_1a_2a_3}\\
&=\frac{(\gamma_*-\gamma_1\gamma_2\gamma_3)-(\gamma_1a_2a_3+\gamma_2a_1a_3+\gamma_3a_1a_2)}{\sqrt{(\delta_{12}-\gamma_1\gamma_2)(\delta_{13}-\gamma_1\gamma_3)(\delta_{23}-\gamma_2\gamma_3)}}
\end{aligned} \qquad (33)$$

then it follows from the above relation

$$(1-P_1)\sqrt{\frac{1-P_1}{P_1}}-P_1\sqrt{\frac{P_1}{1-P_1}}=X.$$

Multiplying both sides by $\sqrt{P_1(1-P_1)}$ and simplifying yields

$$1-2P_1=X\sqrt{P_1(1-P_1)}.$$

Consequently, X has an opposite sign from $P_1-\frac{1}{2}$. Squaring both sides of the above equality and after some algebraic manipulation, we obtain

$$\left(P_1-\frac{1}{2}\right)^2=\frac{P_1(1-P_1)}{4}X^2=\frac{X^2}{4}\left(\frac{1}{4}-\left(P_1-\frac{1}{2}\right)^2\right).$$

It follows that

$$\left(P_1-\frac{1}{2}\right)^2=\frac{X^2}{4(X^2+4)}.$$

Since $P_1 - \frac{1}{2}$ has an opposite sign from X, the above equality implies a unique solution for P_1 and P_0:

$$P_1 = 0.5 - \frac{X}{2\sqrt{X^2+4}}, \qquad P_0 = 1 - P_1 \tag{34}$$

where X is defined by (33). This completes the derivation for P_1. The remaining six unknown probabilities P_i^f, P_i^m $(i = 1,2,3)$ can also be calculated explicitly using (32). This completes the proof of Theorem 1.

APPENDIX B: PROOF OF THEOREM 2

Let $\hat{\gamma}_i^k$, $\hat{\delta}_{ij}^k$ and $\hat{\gamma}_*$ be given by (15)–(17). Also, let $\hat{a}_i$, $\hat{X}$, $\hat{P}_1$, $\hat{P}_i^f$ and $\hat{P}_i^m$ $(i = 1,2,3)$ be defined by the adaptive fusion algorithm. (Again, for simplicity, we have removed the index k from these notations.)

By the assumption (19) and (29)–(30), there exists some $\mu > 0$ such that

$$\epsilon \le P_1 \le 1 - \epsilon, \qquad \delta_{ij} - \gamma_i\gamma_j \ge \mu, \quad \text{for} \quad 1 \le i,j \le 3, \quad i \ne j. \tag{35}$$

By (10)–(13), P_1 is a continuous function of δ_{ij}, γ_i, and γ_*. Hence, there exists $\nu > 0$ such that

$$\frac{\epsilon}{2} \le \bar{P}_1 \le 1 - \frac{\epsilon}{2}, \quad \bar{\delta}_{ij} - \bar{\gamma}_i\bar{\gamma}_j \ge \mu/2, \quad \text{for} \quad 1 \le i,j \le 3, \quad i \ne j \tag{36}$$

whenever

$$|\bar{\gamma}_i - \gamma_i| \le \nu, \quad |\bar{\gamma}_* - \gamma_*| \le \nu, \quad |\bar{\delta}_{ij} - \delta_{ij}| \le \nu \tag{37}$$

where $\bar{P}_1$ denotes value of P_1 calculated with (10)–(13) using $\bar{\delta}_{ij}$, $\bar{\gamma}_i$, and $\bar{\gamma}_*$.

Since the local observations $\{(u_1^k, u_2^k, u_3^k)\}$ are ergodic, we know that

$$\lim_{k\to\infty} \hat{\gamma}_i^k = \gamma_i, \qquad \lim_{k\to\infty} \hat{\gamma}_*^k = \gamma_*, \qquad \lim_{k\to\infty} \hat{\delta}_{ij}^k = \delta_{ij}, \quad i \ne j, \quad 1 \le i,j \le 3 \tag{38}$$

with probability 1. In fact, if we let

$$F_k = \text{the event that } \{|\hat{\gamma}_i^k - \gamma_i| \le \nu,\ |\hat{\gamma}_*^k - \gamma_*| \le \nu,\ |\hat{\delta}_{ij}^k - \delta_{ij}| \le \nu\}$$

and let F_k^c denote its complement, then there exists some $\alpha \in (0,1)$ and some $\rho > 0$ such that

$$\Pr(F_k^c) \le \rho\alpha^k, \qquad k \ge 1.$$

Let $G_{\bar{k}} = \bigcap_{\bar{k}}^{\infty} F_k$. Clearly, there holds

$$\Pr(G_{\bar{k}}^c) = \Pr\left(\bigcup_{k=\bar{k}}^{\infty} F_k^c\right) \le \sum_{k=\bar{k}}^{\infty} \Pr(F_k^c) \le \frac{\rho\alpha^{\bar{k}}}{1-\alpha}. \tag{39}$$

Moreover, on the set $G_{\bar{k}}$, we have from (36)

$$\frac{\epsilon}{2} \le \hat{P}_1 \le 1 - \frac{\epsilon}{2}, \qquad \hat{\delta}_{ij}^k - \hat{\gamma}_i^k\hat{\gamma}_j^k \ge \mu/2, \quad \text{for} \quad 1 \le i,j \le 3, \quad i \ne j \quad \text{and for all} \quad k \ge \bar{k}. \tag{40}$$

This shows that the denominators in the definitions of $\hat{a}_i$, $\hat{X}$, $\hat{P}_1$, $\hat{P}_i^f$ and $\hat{P}_i^m$ $(i = 1,2,3)$ (cf. (10), (12), and (18)) are all bounded away from zero over $G_{\bar{k}}$. Also, notice that the projection operator $[\cdot]_+$ is a Lipschitz continuous mapping. This, together with the property that the denominators are bounded away from zero, imply that $\hat{a}_i$, $\hat{X}$, $\hat{P}_1$, $\hat{P}_i^f$ and $\hat{P}_i^m$ $(i = 1,2,3)$ are all Lipschitz continuous functions of $\hat{\gamma}_i^k$, $\hat{\gamma}_*^k$ and $\hat{\delta}_{ij}^k$ over the set $G_{\bar{k}}$. Thus, there exists some constant $L > 0$ (independent of k, but possibly dependent on $\bar{k}$) such that

$$\begin{cases} |\hat{P}_1 - P_1| \le L(|\hat{\gamma}_i^k - \gamma_i| + |\hat{\gamma}_*^{\,k} - \hat{\gamma}_*| + \sum_{i\ne j} |\hat{\delta}_{ij}^k - \delta_{ij}|), \\ |\hat{P}_i^f - P_i^f| \le L(|\hat{\gamma}_i^k - \gamma_i| + |\hat{\gamma}_*^{\,k} - \hat{\gamma}_*| + \sum_{i\ne j} |\hat{\delta}_{ij}^k - \delta_{ij}|), \\ |\hat{P}_i^f - P_i^f| \le L(|\hat{\gamma}_i^k - \gamma_i| + |\hat{\gamma}_*^{\,k} - \hat{\gamma}_*| + \sum_{i\ne j} |\hat{\delta}_{ij}^k - \delta_{ij}|), \end{cases} \quad i = 1,2,3, \qquad \forall \quad k \ge \bar{k} \tag{41}$$

holds over the set $G_{\bar{k}}$. Letting $k \to \infty$ in the above inequalities and using (38) shows that

$$\lim_{k\to\infty} \hat{P}_1 = P_1, \qquad \lim_{k\to\infty} \hat{P}_i^f = P_i^f, \qquad \lim_{k\to\infty} \hat{P}_i^m = P_i^m, \quad i = 1,2,3 \tag{42}$$

over the set $G_{\bar{k}}$. Since (42) holds for all large $\bar{k}$, it follows that (42) holds on $G_\infty := \lim_{\bar{k}\to\infty} G_{\bar{k}}$. Since $\lim_{\bar{k}\to\infty} \Pr(G_{\bar{k}}^c) = 0$ (cf. (39)), it follows $\Pr(G_\infty) = 1$. In other words, (42) holds with probability 1, as desired.

Since $\hat{P}_1$, $\hat{P}_i^f$ and $\hat{P}_i^m$ $(i = 1,2,3)$ are bounded (thanks to the projection operator $[\cdot]_+$ in (18)), the almost sure convergence as given in (42) implies L^1 convergence. In other words, we have the asymptotic unbiasedness of the estimators:

$$\lim_{k\to\infty} E(\hat{P}_1) = P_1, \qquad \lim_{k\to\infty} E(\hat{P}_i^f) = P_i^f,$$

$$\lim_{k\to\infty} E(\hat{P}_i^m) = P_i^m, \qquad i = 1,2,3.$$

It remains to establish the rate of convergence. It is well known that

$$E|\hat{\gamma}_i^k - \gamma_i|^2 = O(1/k), \qquad E|\hat{\gamma}_*^k - \gamma_*|^2 = O(1/k),$$

$$E|\hat{\delta}_{ij}^k - \delta_{ij}|^2 = O(1/k), \qquad i \ne j, \quad 1 \le i,j \le 3.$$

Fixing some large $\bar{k}$ and combining the above estimates with (41) yields

$$\begin{aligned} &E[|\hat{P}_1 - P_1|^2 \mid G_{\bar{k}}]\, P(G_{\bar{k}}) \\ &\quad \le L' \left(E|\hat{\gamma}_i^k - \gamma_i|^2 + E|\hat{\gamma}_*{}^k - \hat{\gamma}_*|^2 + \sum_{i \ne j} E|\hat{\delta}_{ij}^k - \delta_{ij}|^2 \right) \\ &\quad = O(1/k) \end{aligned}$$

for some constant $L' > 0$ (independent of k, but may depend on $\bar{k}$). Now we only need to note that

$$\begin{aligned} E|\hat{P}_1 - P_1|^2 &= E[|\hat{P}_1 - P_1|^2 \mid G_{\bar{k}}]P(G_{\bar{k}}) \\ &\quad + E[|\hat{P}_1 - P_1|^2 \mid G_{\bar{k}}^c]\, P(G_{\bar{k}}^c) \\ &\le O(1/k) + \frac{\rho\alpha^k}{1-\alpha} \end{aligned}$$

where the last step follows (39) and the fact that $|\hat{P}_1 - P_1|$ is bounded by 1 (again thanks to the projection operator $[\cdot]_+$ in (18)). The above inequality shows $E|\hat{P}_1 - P_1|^2 = O(1/k)$ as desired. The other two inequalities

$$E|\hat{P}_i^f - P_i^f|^2 = O(1/k), \qquad E|\hat{P}_i^m - P_i^m|^2 = O(1/k), \quad i = 1,2,3$$

can be established in a similar fashion. This completes the proof of the theorem.

APPENDIX C: IMPLEMENTATION AND MODIFICATIONS OF THE EXISTING ALGORITHMS

Naim and Kam's Algorithm [9]

Our direct implementation of Naim and Kam's algorithm followed [9, Table 2], with the correction of a couple of minor typographical errors and the addition of the projection operator $[\cdot]_+$ in (18) to ensure that the estimated probabilities remain in $[0,1]$. (The estimated probabilities in Naim and Kam's published algorithm are always real, but can stray outside $[0,1]$.) The algorithm was initialized at instant $k = 1$ and the initial estimate of P_1 was chosen to be 0.5. Given our prior knowledge of the true detector performance, we chose the initial estimates of P_i^f and P_i^m to be 0.1. (In the different scenario in [9, Example 1], the initial estimates of P_i^f and P_i^m were chosen to be 0.05, but that choice lead to worse performance in our scenario.) As can be seen from Table I, the performance of this direct implementation was rather unreliable, and hence we sought modifications of the algorithm to improve its performance. The main modification of Naim and Kam's algorithm was to exploit some prior knowledge of the performance of the detectors to choose an ϵ and constrain the estimates $\hat{P}_i^f$ and $\hat{P}_i^m$ so that they lie in $[\epsilon, 1-\epsilon]$. This bounds the magnitudes of the corresponding weights by $\log((1-\epsilon)/\epsilon)$. If ϵ is appropriately chosen, this bound not only contains the weight explosion problem described in Section IV, but may also reduce the convergence time. (For our scenario we chose $\epsilon = 10^{-3}$.) The projection operator $[\cdot]_+$ used in our algorithm [see (18)] could also be modified in a similar way to improve the convergence speed. However, since our algorithm is not based on the assumption that the fusion center makes the correct decision, we obtain reliable performance by simply bounding the estimated probabilities to lie in $[0,1]$. The initialization of Naim and Kam's algorithm was also modified to improve the algorithm's reliability. In the modified implementation we set the initial estimates of P_i^f and P_i^m to be quite high in order to ensure that the detectors' initial weights were quite small. We found that initial estimates of 0.35 performed the most reliably. In addition, we initialized Naim and Kam's algorithm at instant $k = 9$ rather than $k = 1$ in order to reduce the step size of the probability estimates (particularly in the early stages of the algorithm). Although this results in slower convergence of the probability estimates, it does lead to a more reliable algorithm (see Table I). Slow convergence of this type is what we were trying to avoid in the modified versions of our algorithm discussed in Section IIC2, but for Naim and Kam's algorithm it provides a substantial improvement in reliability.

Ansari et al.'s Algorithm [2]

Our direct implementation of Ansari's algorithm followed [2, Table 1], where in the second row of that table it is understood that there are two weights for each sensor; one for when $u_i = 1$ and one for when $u_i = 0$. Our initial estimates for the weights w_i were chosen as $\log((1-0.1)/0.1)$, which corresponds to the initial estimates of P_i^f and P_i^m being 0.1, and hence appears to be a natural choice. Our initial estimate of w_0 was chosen to be 0, which corresponds to the initial estimate of P_1 being 0.5. The reliability threshold of Ansari's algorithm was set using the procedure described in [2, Sect. 4.2]. The absolute value of the threshold was set to be $\alpha y_{\max}$, where $y_{\max}$ is a function of the detector performance, and α is a coefficient of our choosing. When $\alpha = 0$ all decisions are deemed reliable, and as α is increased the reliability threshold is raised. The calculation of $y_{\max}$ requires prior knowledge of the performance of the detectors. We have used this knowledge in our simulations, but in practice accurate knowledge of the performance of the detectors would not be available. (A threshold can still be chosen in that case, but determining an appropriate choice may be more difficult.) The final initialization that is required in Ansari's algorithm is the initial values of the counters m, $m_{1,i}$ and $m_{0,i}$ in [2, Sect. 3]. These counters count the number of times $u_0 = 1$, the number of times $u_i = 1$ and $u_0 = 1$, and the number

of times $u_i = 0$ and $u_0 = 0$, respectively [see (1)–(3) for definitions]. The step sizes of the algorithm are inversely proportional to these counters, and hence their initial values must be chosen as a compromise between large values which result in small step sizes and hence long convergence times, and small values which may reduce the convergence time, but expose the algorithm to the weight explosion problem discussed in Section IV. Appropriate initial values for these counters were not suggested by Ansari et al. [2], but our experiments have suggested that an initial value of 10 for each of these counters provides good performance in the scenario we are considering (see Table I). As a final observation, we point out that we could modify Ansari's algorithm by choosing an ϵ and bounding the magnitude of the weights by $\log((1-\epsilon)/\epsilon)$. However, we found that when the initial values of the counters were set to 10 and $\epsilon = 10^{-3}$, the bound was only active for large α. (That being said, it did remove the three failures from the right hand column of Table I.) When the counters were initialized to 5, enforcing the bound reduced the number of failures by 30–50%. For simplicity, we have not implemented this bound in the simulations reported in Section IV.

REFERENCES

[1] Ansari, N., Hou, E., Zhu, B., and Chen, J. (1996)
An adaptive fusion model for distributed detection systems.
IEEE Transactions on Aerospace and Electronic Systems, **32**, 2 (1996), 524–531.

[2] Ansari, N., Chen, J-G., and Zhang, Y-Z. (1997)
Adaptive decision fusion for unequiprobable sources.
IEE Proceedings on Radar, Sonar, Navigation, **144**, 3 (June 1997), 105–111.

[3] Blum, J. R. (1954)
Approximation methods which converge with probability one.
Annals of Mathematical Statistics, **25**, 2 (June 1954), 382–388.

[4] Chair, Z., and Varshney, P. (1986)
Optimum data fusion in multiple sensor detection systems.
IEEE Transactions on Aerospace and Electronic Systems, **22**, 1 (1986), 98–101.

[5] Chen, J., and Ansari, N. (1998)
Adaptive fusion of correlated local decisions.
IEEE Transactions on Systems , Man and Cybernetics—Part C: Applications and Reviews, **28**, 2 (1998), 276–281.

[6] Drakopoulos, E., and Lee, C. (1991)
Optimum multi-sensor fusion of correlated local decision.
IEEE Transactions on Aerospace and Electronic Systems, **27** (July 1991), 593–605.

[7] Kam, M., Zhu, Q., and Gray, W. S. (1992)
Optimum data fusion of correlated local decisions in multiple sensor detection systems.
IEEE Transactions on Aerospace and Electronic Systems, **28** (July 1992), 916–920.

[8] Lee, C., and Chao, J. (1989)
Optimal local decision space partitioning for distributed detection.
IEEE Transactions on Aerospace and Electronic Systems, **25**, 4 (1989), 536–544.

[9] Naim, A., and Kam, M. (1994)
On-line estimation of probabilities for distributed bayesian detection.
Automatica, **30**, 4 (1994), 633–642.

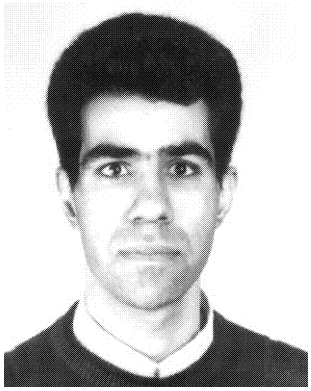
[10] Robbins, H., and Monro, S. (1951)
A stochastic approximation method.
Annals of Mathematical Statistics, **22** (1951), 400–407.

[11] Thomopoulos, S., Viswanathan, R., and Bougoulias, D. (1987)
Optimal decision fusion in multiple sensor systems.
IEEE Transactions on Aerospace and Electronic Systems, **AES-23**, 5 (1987), 644–665.

[12] Thomopoulos, S., Viswanathan, R., and Bougoulias, D. (1989)
Optimal distributed decision fusion.
IEEE Transactions on Aerospace and Electronic Systems, **25** (Sept. 1989), 761–765.

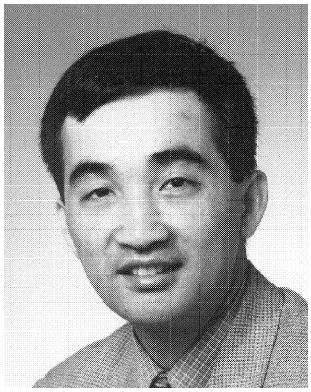
[13] Tsitsiklis, J. N., and Athans, M. (1985)
On the complexity of distributed decision problems.
IEEE Transactions on Automatic Control, **AC-30** (May 1985), 440–446.

[14] Viswanathan, R., and Varshney, P. (1997)
Distributed detection with multiple sensors: Part I—Fundamentals.
Proceedings of the IEEE, **85**, 1 (1997), 54–63.



Ghasem Mirjalily was born in Iran in 1969. He received the B.Eng. degree from Mashhad University, Iran, in 1991 and the M.S. degree from K. N. Toosi University of Technology, Iran, in 1994, both in electrical engineering and with honors. In 2000 he completed the Ph.D. program at Tarbiat Modarres University, Iran. He was a visiting researcher at the Communications Research Laboratory at McMaster University, from June 1998 to December 1998, where he worked on the problem of distributed detection.

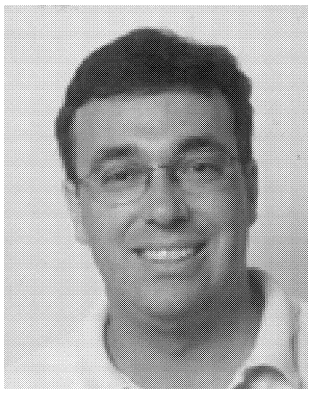
Dr. Mirjalily joined Yazd University as an assistant professor in 2001 where he is head of the computer and informatic center. His research interests are in signal processing, distributed detection and data fusion.



Zhi-Quan Luo (M'90) was born in Nanchang, Jiangxi province, the People's Republic of China. He received the B.Sc. degree in applied mathematics in 1984 from Peking University, Beijing, China. During the academic year of 1984 to 1985, he studied at the Nankai Institute of Mathematics, Tianjin, China. From 1985 to 1989, he studied at the Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, where he received the Ph.D. degree in operations research.

In 1989, he joined the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Ontario, Canada, where he is now the department chair and holds the Canada Research Chair in Information Processing. His research interests lie in the union of large-scale optimization, information theory and coding, data communications and signal processing.

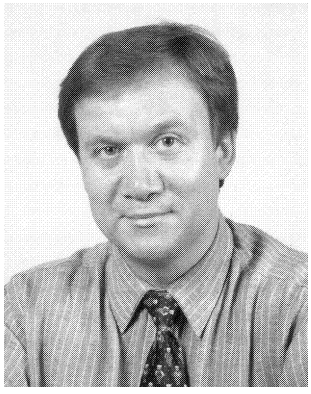
Dr. Luo is a member of SIAM and MPS. He is presently serving as an Associate Editor for the *Journal of Optimization Theory and Applications*, *SIAM Journal on Optimization*, *Mathematics of Computation*, *Mathematics of Operations Research*, *IEEE Transactions on Signal Processing*, and *Optimization and Engineering*.



Timothy N. Davidson (M'96) received the B.Eng. (Hons. I) degree in electronic engineering from The University of Western Australia (UWA), Perth, in 1991 and the D.Phil. degree in engineering science from the The University of Oxford, England, in 1995.

He is currently an Assistant Professor in the Department of Electrical and Computer Engineering at McMaster University, Hamilton, Ontario, Canada. His research interests are in signal processing, communications and control, with current activity focused on signal processing for digital communication systems. He has held research positions at the Communications Research Laboratory at McMaster University, the Adaptive Signal Processing Laboratory at UWA and the Australian Telecommunications Research Institute at Curtin University of Technology, Perth, Western Australia.

Dr. Davidson was awarded the 1991 J. A. Wood Memorial Prize (for "the most outstanding [UWA] graduand" in the pure and applied sciences), and the 1991 Rhodes Scholarship for Western Australia.



Éloi Bossé received the B.A.Sc., M.A.Sc. and Ph.D. degrees in electrical engineering from Laval University, Québec.

In 1981 he joined the Communications Research Centre, Ottawa, Canada, where he worked on radar signal processing, high resolution spectral analysis and radar tracking in multipath. In 1988 he was transferred to the Defence Research Establishment Ottawa (DREO) and in 1992 to the Defence Research Establishment Valcartier (DREV) where he worked on data and information fusion. He now heads the Decision Support Technology Section. His current interests include data and information fusion, reasoning under uncertainty, decision support. He is also the Leader of the Maritime C3I Thrust.

Dr. Bossé has published over 80 papers in journals and conference proceedings.