Testing Under Communication Constraints

SORA CHOI
BALAKUMAR BALASINGAM
PETER WILLETT

The problem of fault diagnosis with communication constraints is considered. Most approaches to fault diagnosis have been focused on detecting and isolating a fault under cost constraints such as economic factors and computational time. But in some systems, such as remote monitoring (e.g., satellite, sensor field) systems, there is a communication constraint between the system being monitored and the monitoring facility. In such circumstances it is desirable to isolate the faulty component with as few interactions as possible. The key consideration is that multiple tests are chosen at each stage in such a way that the tests within the chosen group complement each other. To this end we propose two algorithms for fault diagnosis under communications constraints. Their performances are analyzed in terms of the average number of testing stages as well as in terms of the required computational complexity.

Manuscript received August 2, 2012; revised June 8, 2013; released for publication October 13, 2013.

Refereeing of this contribution was handled by Mujdat Cetin.

This work acknowledges support from the U.S. Office of Naval Research under Grant N00014-10-10412, and from Qualtech Systems under a NASA STTR.

Authors' address: Department of Electrical and Computer Engineering, University of Connecticut, 371 Fairfield Way, U-2157, Storrs, Connecticut 06269 USA, E-mail: ({soc07002, willett, bala}@engr.uconn.edu).

1557-6418/13/$17.00 © 2013 JAIF

1. INTRODUCTION

Fault diagnosis is the process of detecting and isolating component failure in a system via reports from a suite of sensors, each of which monitors a subset of the components. Since the life-cycle maintenance cost of large integrated systems such as an aircraft or the space shuttle can, due to the large number of failure states and the need to rectify these failures quickly [4], [12], exceed the purchase cost, it has been recognized that testability must be built into the manufacturing process. Owing to the advent of intelligent sensors, onboard tests are available to the diagnostic/fusion center or operators, but the computational burden of processing test results in large-scale systems is still a factor.

Various techniques for computationally efficient test sequencing to identify component failure have been developed: test sequencing for single fault diagnosis [14]–[16], [21], dynamic single fault diagnosis [5], [24], multiple fault diagnosis [10], [18], [19], dynamic multiple fault diagnosis [17], [20] and test sequencing for complex systems [2], [3], to name a few. As another approach, some fault detection schemes for networked control systems use residual generation and evaluation without utilizing built-in smart sensors to detect component failures [11], [22], [23], [25], [26].

To date, the purpose of test sequencing (see [2], [3], [14]–[16], [19], [21]) has usually been to find an optimal or suboptimal solution minimizing a "testing cost" that include economic factors, testing time, etc. In general, a sequential testing algorithm repeats the procedure (called a stage) consisting of deciding to test, ordering to perform tests, and updating the state of the system using the received test results, until the failed component is identified. Moreover, at each state, communication between the system and the diagnostic (fusion) center is required for the sensors to transmit the test results and for the center to request the performance of tests.

There are, sometimes, systems placed in distributed or remote configurations, causing unusually long delays or restriction in communication of instructions from or results to a monitoring facility. One application might be remotely to determine and diagnose the health of a hard-to-reach system, for example a space vehicle or a craft in a deep sea (please see an illustration in Fig. 1). In this situation the monitoring facility should be able to diagnose the remote system using limited communication with the remote or distributed system. Under these circumstances the *number of instances of communication* becomes a primary constraint and the testing cost, while still an issue, becomes secondary. Our goal in this paper is hence to minimize the number of communication stages between the monitoring facility

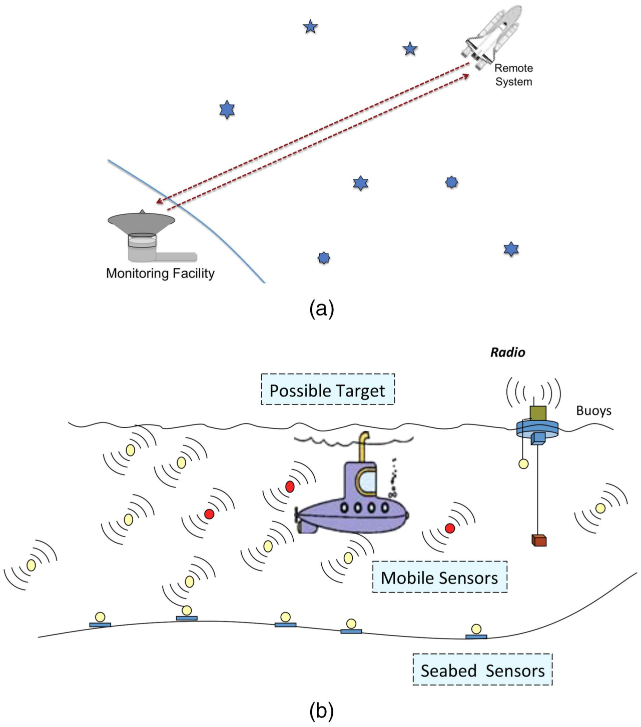


Fig. 1. Examples of a systems with limited communication between the system to be monitored and the monitoring facility. (a) A system with significant remote-ness, in which latency and communication costs are a concern. (b) A distributed surveillance system in which covertness is key.

and the remote system while keeping the testing cost within a specified level.

The optimal solution of fault diagnosis can be posed via generating a binary decision tree with DP (dynamic programing) and AND/OR graphs. Finding the optimal solution, however, is known to be an NP-complete problem for single fault diagnosis [7]–[9], [13] even when using just a single sensor at each stage, and (worse) an NP-hard problem for multiple fault diagnosis [1], meaning that it cannot be solved by an algorithm of polynomial complexity. Thus, a heuristic approach for test sequencing using multiple sensors at each stage is suggested. This involves:

1) selecting the sensors that will maximize the information gain (IG);
2) performing the test using those selected sensors;
3) updating conditional likelihood probability of all components' failure states, depending on the test outcomes;
4) pruning from consideration components that are unlikely to be faulty; and
5) repeating this procedure until the fault is isolated up to a certain specified probability of assurance.

We will discuss the IG heuristic later. Note that there are test inaccuracies due to unreliable sensors, electromagnetic interference, environmental conditions, and so on. Imperfect tests, for our purposes here, introduce two uncertainties to the diagnosis process: missed detections and false alarms. Under the former, of course, a component may fail yet the test that covers it can show "pass"; and vice versa for a false alarm. Consequently, even after collection of an arbitrarily amount of test signature evidence, one is never *certain*, just sure enough up to a given probability level.[1]

An important issue is that as we seek to reduce the number of iterations by selecting multiple sensors at a time, the computation—selecting the set of sensors that will maximize the information gain—increases rapidly both with number of tests and number of faults, and in many practical systems both are very large. We propose two algorithms for selecting multiple sensors at a time that maximize the information gain at an affordable computational complexity. The first algorithm introduces several thresholds in order to eliminate sensors that are less informative, so that fewer sensors form the candidate set for the maximization of information gain. The second approach populates the candidate set one-by-one, based on the correlation between the information state and the elements of the reachability matrix in addition to information gain. For simplicity, we assume that there is a single component failure during fault isolation, although the methods presented in this paper can be extended to multi-fault case by applying one of the multi-fault diagnosis techniques [10], [17]–[20] in order to mitigate (but not really avoid) the significant increase in resulting computational complexity.

The paper is organized as follows. In Section 2 we formulate the problem. In Section 3 two heuristic test sequencing algorithms are introduced. Section 4 presents the simulation results of the proposed algorithms and Section 5 concludes the paper.

2. PROBLEM FORMULATION

We consider the problem of single fault diagnosis: there are a set of components that might fail and a set of sensors each monitoring a subset of those components. The system is described in detail as given below.

1) A (finite) set of m possibly-faulty components $F = \{f_0, f_1, \ldots, f_m\}$ (loosely: "faults") is given, where f_0 denotes the no-fault condition and f_i denotes the ith faulty component. The state of faulty component f_i is expressed by x_i, where $x_i = 1$ if fault f_i occurs, otherwise $x_i = 0$.
2) A (finite) set of n binary sensors $S = \{s_1, s_2, \ldots, s_n\}$ is given, where sensor s_j monitors a known subset of faulty components and costs an amount $c_j (> 0)$ to apply.

[1] And indeed, depending on the test coverage (the R matrix to be defined shortly), it might *never* be possible even with perfect tests to isolate a fault more tightly than to a "ambiguity set."

3) The reachability matrix[2] $R = [r_{ij}]$ represents the relationship between the faulty components and the sensors: $r_{ij} = 1$ if sensor s_j monitors faulty component i, and, otherwise, $r_{ij} = 0$. In addition, $r_{0j} = 0$ for all j.

4) Test (sensor) s_j has associated with it a probability of false alarm

$$P_{f_j} = \text{Prob}(s_j = 1 \,|\text{no component monitored by } s_j \text{ has failed}), \tag{1}$$

and probability of detection

$$P_{d_j} = \text{Prob}(s_j = 1 \,|\text{at least one component monitored by } s_j \text{ has failed}). \tag{2}$$

Since only one fault is being considered, the probabilities of detection and false alarm can be combined with the reachability matrix R to as the likelihood matrix $D = [d_{i,j}]$, in which

$$d_{i,j} = \text{Prob}(s_j = 1 \mid x_i = 1) = r_{i,j}P_{d_j} + (1 - r_{i,j})P_{f_j}. \tag{3}$$

The element $d_{i,j}$ is the likelihood that the sensor j registers a "fail" despite the relation with a failure of component i, considering two cases: The first is that component i is failed and sensor s_j detects that there is a failure, when s_j is monitoring component i. The second is that sensor s_j is not monitoring component i, but s_j shows the result that there is a component failure as false alarm.

Let us address the single fault diagnosis problem under the following assumptions:

- The false-alarm and missed-detection probabilities of sensors are known and do not change with repeated testing.
- There is at most one component failure, which does not change over the course of (repeated) testing.
- It is possible that the system be fault-free.
- Each sensor's missed-detection/false-alarm process is independent of those of other sensors.
- Outcomes of sensors are binary, meaning that there are two outcomes: pass (0) and fail (1).

Let us denote the complete set of (possibly multiple) tests applied at the kth stage as $S_c(k) = \{s_{j_q} \in S \mid j_q \in J(k)\}$ where $J(k) = \{j_q\}$ is the set of indices of the applied sensors. The outcome of the tests at the kth stage is denoted as $O(k) = \{o_q(k)\}$ where $o_q(k)$ is the result of sensor s_{j_q}. Thus, the past information available for sensor selection at the $(k+1)$th stage is

$$I_k = \{(S_c(l), O(l))\}_{l=1}^{k}. \tag{4}$$

With the past information I_k, the conditional failure probabilities $\pi_i(k+1) = p(x_i = 1 \mid I_k)$ also known as the *information state* is updated from its previous state $\pi_i(k)$ based on Bayes' rule as

$$\begin{aligned}
\pi_i(k+1) &= p(x_i = 1 \mid I_k)\\
&= \frac{p((S_c(k),O(k)) \mid x_i = 1, I_{k-1})p(x_i = 1 \mid I_{k-1})}{p((S_c(k),O(k)) \mid I_{k-1})}\\
&= \frac{1}{c}\prod_{j_q\in\mathbf{J}(k)} p((s_{j_q}(k), o_q(k)) \mid x_i = 1)\pi_i(k)\\
&= \frac{1}{c}\prod_{j_q\in\mathbf{J}(k)} [o_q(k)d_{i,j_q} + (1 - o_q(k))(1 - d_{i,j_q})]\pi_i(k)
\end{aligned} \tag{5}$$

where the normalization factor is

$$c = \sum_{i=0}^{m}\prod_{j_q\in\mathbf{J}(k)} [o_j(k)d_{lj} + (1 - o_j(k))(1 - d_{lj})]\pi_i(k).$$

In addition, the prior failure probability $\pi_i(1)$ of component i is assumed to be known, and the probability of a healthy system $\pi_0(1)$ satisfies the following:

$$\pi_0(1) = \prod_{i=1}^{m}\text{Prob}(x_i = 0) = \prod_{i=1}^{m}(1 - \pi_i(1)). \tag{6}$$

The test sequencing algorithm with imperfect tests can never, except in trivial cases, identify the faulty component deterministically, but is assumed content with a (pre-specified) level of certainty α. We have:

Stopping rule: The algorithm stops when any information state reaches a level of certainty α, i.e.,

$$\pi_i(k) > \alpha. \tag{7}$$

Pruning criterion: If $\pi_i(k)$ satisfies

$$\pi_i(k) \le \beta\pi_i(0), \tag{8}$$

where threshold β is given, then it will be decided that the component i is not a faulty one and set $\pi_i(k) = 0$.

The *stopping rule* and *pruning criterion* in the algorithms will be described in later sections.

3. TEST SEQUENCING USING INFORMATION GAIN

Given the current information state $\{\pi_i(k)\}$ at the kth stage, the information gain achieved by testing with a set of sensors $S_c(k)$, i.e. the mutual information between the sensors and the information state, is written as

$$\text{IG}(\{\pi_i(k)\}, S_c(k)) = H(\{\pi_i(k)\}) - H(\{\pi_i(k)\} \mid S_c(k)) \tag{9}$$

where

$$H(\{\pi_i(k)\}) = -\sum_{i=0}^{m}\pi_i(k)\log\pi_i(k)$$

is the uncertainty (entropy) associated with the information state $\{\pi_i(k)\}_{i=1}^{m}$.

After performing some algebraic manipulations the information gain in (9) is written as follows (please see

[2] This is sometimes call the **D**-matrix, invoking variously test **d**ependency or **d**iagnosis. Here we use "D" for the test reliabilities.

Appendix for details):

$$\begin{aligned}\mathrm{IG}(\{\pi_i(k)\},S_c(k)) = \\ &-\sum_i \pi_i(k)\log\pi_i(k) \\ &+\sum_{\tilde{\mathbf{o}}_t\in\tilde{O}}\sum_{i=0}^{m}\pi_i(k)\prod_{j\in J}A(t,i,j)\cdot\log\left\{\pi_i(k)\prod_{q\in J}A(t,i,q)\right\} \\ &-\sum_{\tilde{\mathbf{o}}_t\in\tilde{O}}\log\left\{\sum_{p=0}^{m}\pi_p(k)\prod_{q\in J}A(t,p,q)\right\}\cdot\sum_{i=0}^{m}\pi_i(k)\prod_{j\in J}A(t,i,j)\end{aligned} \tag{10}$$

where

$$A(t,i,j) := (\tilde{\mathbf{o}}_t(j)d_{i,j} + (1-\tilde{\mathbf{o}}_t(j))(1-d_{i,j})). \tag{11}$$

Here, $\tilde{\mathbf{o}}_t$ denotes a vector whose element is a possible outcome (0 or 1) of a sensor in set $S_c(k)$, the set $\tilde{O} = \{\tilde{\mathbf{o}}_t\}$ consists of all possible vectors that can be generated by the sensors in set $S_c(k)$, and J is the set of indices of sensors in the set $S_c(k)$.

If the objective is only to minimize testing cost by using information heuristics, the sensor can be selected simply by maximizing the information gain per unit cost. That is, sensors $S_c(k)$, at the kth stage, can be selected based on the *Selection rule minimizing testing cost:*

$$S_c(k) = \arg\max_{\tilde{S}_c(k)\subset S}\frac{\mathrm{IG}(\{\pi_i(k)\},\tilde{S}_c(k))}{\sum_{j\in J(k)}c_j} \tag{12}$$

Our goal in this paper is to minimize the number of stages required to locate a fault, while limiting the cost spent at each stage to C_T. Hence, instead of performing one test at each stage, we propose to perform (possibly) several tests, where one should select the set of sensors $S_c(k)$ having the most information about the faulty component at each stage considering all possible combinations of sensors. This can be achieved via the following *Selection rule minimizing the number of stages:*

$$S_c(k) = \arg\max_{\tilde{S}_c(k)\subset S}\mathrm{IG}(\{\pi_i(k)\},\tilde{S}_c(k)) \tag{13}$$

subject to

$$\sum_{j\in J(k)} c_j \le C_T$$

where C_T is the cost threshold per stage.

If we countenance the use of an exhaustive search to select a set $S_c(k)$ according to the selection rule minimizing the number of stages, a test sequencing heuristic algorithm using the information gain will follow Algorithm 1.

ALGORITHM 1 *Exhaustive Search—ExS(N)*

1) After obtaining all possible combinations of sensors in the set S satisfying the cost constraint C_T, select $S_c(k)$ based on (13).
2) Obtain test outcomes of $S_c(k)$ and update information states using (5).
3) Apply the pruning criterion in (8): after pruning, normalize information states.
4) Repeat steps 1)–3) until the stopping rule in (7) is satisfied.

Algorithm 1 is computationally exhaustive due to its first step. For testing at most n_c sensors at a time, the number of all possible combinations of tests is

$$\sum_{n=1}^{n_c}\binom{n}{m}. \tag{14}$$

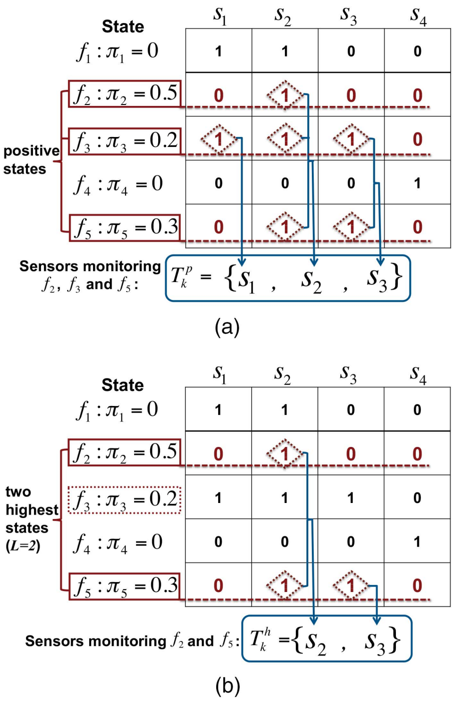
To give some perspective, if there are $m = 100$ sensors, considering pairs of tests jointly results in 4950 combinations and testing 3 tests at a time results in 161700 combinations. Thus, in what follows, we focus on finding suboptimal ways to find $S_c(k)$ that have good heuristics.

Before discussing suboptimal algorithms, we point out a useful property of the information gain in (10). It is observed that when the information states $\{\pi(k)\}$ are uniform, except for $\pi_0(k)$—because $\pi_0(k)$ corresponds to the fault-free state—the information gain in (10) reduces to

$$\begin{aligned}&\mathrm{IG}(\{\pi_i(k)\},\tilde{S}_c(k)) \\ &\quad = -n\pi(k)\log\pi(k) - \pi_0(k)\log\pi_0(k) \\ &\qquad + \pi(k)\prod_{j\in J}A(t,i,j)\sum_{\tilde{\mathbf{o}}_t\in\tilde{O}}\sum_{i=1}^{m}\left(\log\left\{\prod_{q\in J}A(t,i,q)\right\}\log\left\{\sum_{p=1}^{m}\prod_{q\in J}A(t,p,q)\right\} - \log\left\{\pi_0(k)\prod_{q\in J}A(t,0,q)\right\}\right)\end{aligned} \tag{15}$$

and (13) reduces to

$$\begin{aligned}S_c(k) &= \arg\max_{\tilde{S}_c(k)}\{\mathrm{IG}(\{\pi_i(k)\},\tilde{S}_c(k))\} \\ &= \arg\max_{\tilde{S}_c(k)}\left\{\prod_{j\in J}A(t,i,j)\sum_{\tilde{\mathbf{o}}_t\in\tilde{O}}\sum_{i=1}^{m}\left(\log\left\{\prod_{q\in J}A(t,i,q)\right\} - \log\left\{\sum_{p=1}^{m}\prod_{q\in J}A(t,p,q)\right\} - \log\left\{\pi_0(k)\prod_{q\in J}A(t,0,q)\right\}\right)\right\}.\end{aligned} \tag{16}$$


State
s_1 s_2 s_3 s_4
$f_1 : \pi_1 = 0$ 1 1 0 0
$f_2 : \pi_2 = 0.5$ 0 1 0 0
$f_3 : \pi_3 = 0.2$ 1 1 1 0
$f_4 : \pi_4 = 0$ 0 0 0 1
$f_5 : \pi_5 = 0.3$ 0 1 1 0
positive states
Sensors monitoring f_2, f_3 and f_5:
$T_k^p = \{s_1 , s_2 , s_3\}$
two highest states (L=2)
Sensors monitoring f_2 and f_5:
$T_k^h = \{s_2 , s_3\}$

(a)

(b)

Fig. 2. The relationship between information state and set T_0 or T_1. (a) Set T_0. (b) Set T_1: $L = 2$.

If the information state is uniform except for π_0, one can select the set of sensors maximizing information gain simply by appeal to the limit of cost—even before beginning to monitor the system according to (16)—which means that the set $S_c(k)$ can be chosen off-line. For example, in the case that the prior probability of fault elements be uniform, one can start to monitor the system with $S_c(1)$ selected by off-line.

Now, we propose two suboptimal algorithms to choose sensors. The first, described in Section 3.1, proposes to decrease the size of set from which the sensors are chosen based on the current information state. The second, described in Section 3.2, proposes to select sensors sequentially based on information gain and current information states, after the first sensor is chosen via correlation between information states and sensors.

3.1. Suboptimal algorithm based on Exhaustive Search (ES)

It should be noted that the application of the pruning rule in (8) at each stage reduces the number of possible faulty components up to an uncertainty, and similarly the size of set $T_0(k)$ where

$$T_0(k) = \{s_{i_q} \in S \mid R(j,i_q) = 1, \text{ for any } j \text{ s.t. } \pi_j(k) > 0\}. \tag{17}$$

As a result there is no need to test those sensors that are monitoring *only* those excluded components, as a result of pruning. Thus, at every stage k, as pruning is performed to eliminate some of components having little possibility of a fault the new set of sensors $T_0(k)$ satisfies $T_0(k) \subset T_0(k-1) \subset S$.

If the size of $T_0(k)$ is too large—possible in the early stages of testing—one can consider a new set $T_1(k) \subset T_0(k)$ whose elements are those sensors monitoring components having high probabilities of failure, as follows. Given a threshold L

$$T_1(k) = \{s_{i_q} \mid R(j_p,i_q) = 1, \text{ for any } p = 1,2,\ldots,L\} \tag{18}$$

where $\pi_{j_1}(k) \geq \pi_{j_2}(k) \geq \cdots \geq \pi_{j_{\tilde{l}}(k)} > 0$ and $L \geq \tilde{l}$. If $\tilde{l}$ is less than L, $T_0(k) = T_1(k)$. By considering all possible combinations of sensors in $T_1(k)$ within the cost limit, one selects $S_c(k)$.

It may be that the number of sensors used at the current stage becomes unacceptable due to a large allowable cost limit: the number of combinations of sensors become too high. A similar situation is that there are many sensors monitoring each component, which is tantamount to each row of the R matrix having many ones—that is, the tests, taken individually, are not specific. In such cases $|T_1(k)|$ will not be significantly decreased even by using a smaller value for L. Both of these cases require another threshold (denoted N) limiting the number of sensors chosen to perform testing at each stage, i.e. $|S_c|$ is limited to N. Hence, if the size of $T_1(k)$ is larger than the given threshold M, the number of sensors used at each stage will be limited by N, where M is decided based on cost. In summary, the upper bound in $|S_c(k)|$ is

$$|S_c(k)| \leq \begin{cases} |T_1(k)| & \text{if } |T_1(k)| \leq M \\ N & \text{if } |T_1(k)| > M \end{cases}. \tag{19}$$

Remark 1: If there are too many information states having the same probability, especially in the early stages, it becomes problematic to decide $T_1(k)$ just by using threshold L, since ordering becomes arbitrary. In this case we change the threshold: L is increased until all elements having the same probability are included in set $T_1(k)$ whereas threshold N is decreased depending on the size of $T_1(k)$.

A summary of the proposed fault diagnosis based on exhaustive search is summarized in Algorithm 2. An example of the sets T_0 and T_1 is illustrated in Fig. 2.

Algorithm 2 *Exhaustive Search*—ES(L,M,N)

1) If the prior probability is uniform, select the combination of sensors chosen off-line as $S_c(1)$. Otherwise, go to the next step.
2) Get $T_1(k)$ by using information states $\{\pi_i(k)\}$. If there are many information states having the same value, L is increased until all sensors monitoring those elements are included in $T_1(k)$.
3) If the number of elements in $T_1(k)$ is higher than M, obtain all possible combinations of at most N sensors satisfying the cost limit C_T. Otherwise, obtain all possible combination of sensors satisfying the limit.
4) Select $S_c(k)$ from the obtained combinations of sensors based on (18) and obtain test outcomes of $S_c(k)$.

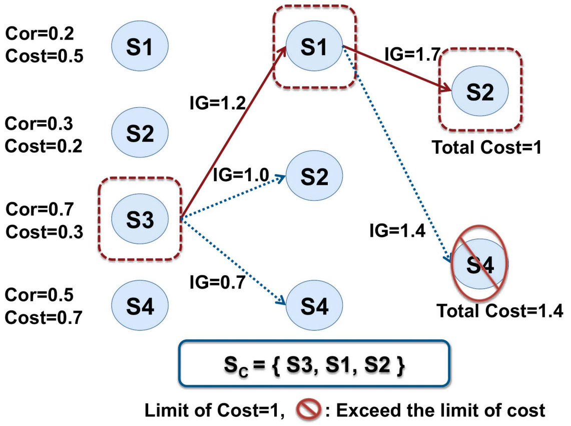


Fig. 3. An example of selecting $S_c(k)$ according to the CS approach.

5) Update information states using (5) and obtain information states $\{\pi(k+1)\}$ by applying the pruning criterion in (8).
6) Repeat steps 2)–5) until the stopping rule in (7) is satisfied.

3.2. Algorithm based on Correlative Search (CS)

While in the previous algorithm the sensors used at a single stage are selected jointly, this algorithm chooses these sensors one-by-one, which means that each sensor is added to the set of sensors chosen before as shown in Fig. 3. Before discussing how to choose sensors, let us define the correlation between matrix R and information state as follows. For each sensor j,

$$\mathrm{Cor}(j) = \sum_i r_{ij}\pi_i r_{ij}$$

The first sensor s_{j_1} is chosen based on correlation as follows:

$$s_{j_1} = \arg\max_{s_j \in \tilde{T}_1} \mathrm{Cor}(j) \tag{20}$$

where

$$\tilde{T}_1 = \{s_q \in T_0 \mid c_q \le C_T\}.$$

The second sensor is the one having the highest information gain calculated with the first sensor together:

$$s_{j_2} = \arg\max_{s_q \in \tilde{T}_2} \mathrm{IG}(\{\pi_i\} \mid \{s_{j_1}, s_q\}) \tag{21}$$

where

$$\tilde{T}_2 = \{s_q \in T_0 \mid c_q \le (C_T - c_{j_1})\} - \{s_{j_1}\}.$$

Assuming set $\tilde{T}_p$ is nonempty, the next sensor is selected in the same way as the second sensor:

$$s_{j_p} = \arg\max_{s_q \in \tilde{T}_p} \mathrm{IG}(\{\pi_i\} \mid \{s_{j_1}, s_{j_2}, \ldots, s_{j_{p-1}}, s_q\}), \tag{22}$$

where

$$\tilde{T}_p = \{s_q \in T_0 \mid c_q \le (C_T - \sum_{a=1}^{p-1} c_{j_a})\} - \{s_{j_1}, s_{j_2}, \ldots, s_{j_{p-1}}\}.$$

Similar to the previous algorithm a threshold N is used to limit the maximum number of sensors in $S_c(k)$, i.e., such that $|S_c(k)| < N$.

REMARK 2: If $\mathrm{IG}(\{\pi_i\} \mid \{s_{j_1}, s_{j_2} \ldots, s_{j_{p-1}}\}) = \mathrm{IG}(\{\pi_i\} \mid \{s_{j_1}, s_{j_2} \ldots, s_{j_p}\})$, which means that the information gain does not increase by adding more sensors, no more sensors will be added.

REMARK 3: It should be noted that the set T_0 can be replaced by set T_1 before correlative selection.

A summary of the proposed fault diagnosis based on correlative search is in Algorithm 3. An example of selecting the set $S_c(k)$ is shown in Fig. 3.

ALGORITHM 3 *Correlative Search—CS(N)*

1) If the prior probability is uniform, select the combination of sensors chosen off-line as $S_c(1)$. Otherwise, go to the next step.
2) After getting $T_0(k)$ by using information states $\{\pi_i(k)\}$, select the first sensor s_{j_1} using (20).
3) Repeat: select pth sensor s_{j_p} using (22) until one of the following is satisfied:
   - p reaches N
   - there are no more sensors to add.
   - the information gain between information states and sensors is not increased by adding another (any other) sensor.
4) Set $S = \tilde{T}_p$ and select $S_c(k)$ based on (13).
5) Obtain the sensor outcomes of $S_c(k)$ and update information states using (5) and obtain information states $\{\pi(k+1)\}$ by applying the pruning rule in (8).
6) Repeat steps 2)–5) until the stopping rule (7) is satisfied.

4. SIMULATION RESULTS

In this section the performance of the proposed methods are analyzed through simulations. The following three algorithms are considered in the simulations.

- ExS(N) for $N = 1$ and 2.
- ES(L,M,N) for $(L,M,N) = (20,10,4)$ and $(50,30,2)$.
- CS(N) for $N = 2$, 3 and 4.

4.1. Randomly generated R matrix

We have generated two different R matrices each with 100 rows and 70 columns, i.e., there are 100 possible faults and 70 sensors monitoring those faults in the simulated system. Each element of the first R matrix, denoted R_2, is generated as Bernoulli with success probability 0.2. In a similar fashion each element of the second R matrix, denoted as R_8, used success probability 0.8. The cost of the test by each sensor is assigned randomly following a uniform distribution between 0.5 and 1. The maximum cost allowed to be spent at each stage is 3. The stopping rule is defined using a level of uncertainty $\alpha = 0.99$ and the pruning criterion is defined with threshold $\beta = 0.005$.

Now, we define some performance metrics in order to compare various algorithms. The *average number of stages* $\bar{k}$ of a certain algorithm is obtained by repeating the algorithm over several Monte-Carlo runs and averaging the number of stages it took each time to locate the fault, i.e.,

$$\bar{k} = \frac{1}{N_m}\sum_{r=1}^{N_m} k_r \tag{23}$$

where k_r is the number of stages it took to locate the fault at the rth run and N_m is the number of Monte-Carlo runs. In all the simulations $N_m = 1000$ Monte-Carlo runs are used.

Figure 4 shows the average number of stages $\bar{k}$ of all the algorithms as the probability of detection P_d increases from 0.8 to 1. In particular, Fig. 4(a) shows the comparison of $\bar{k}$ vs. P_d of ExS and ES on different R matrices. The same is shown in Fig. 4(b) for ExS and CS. The summary of Fig. 4—comparison of all three algorithms—is shown in Fig. 4(c) which shows that the exhaustive search (ES) algorithm performs as well as the exhaustive search ExS(2), whereas the correlative search CS(N) outperforms the exhaustive search ExS(2) with increasing N.

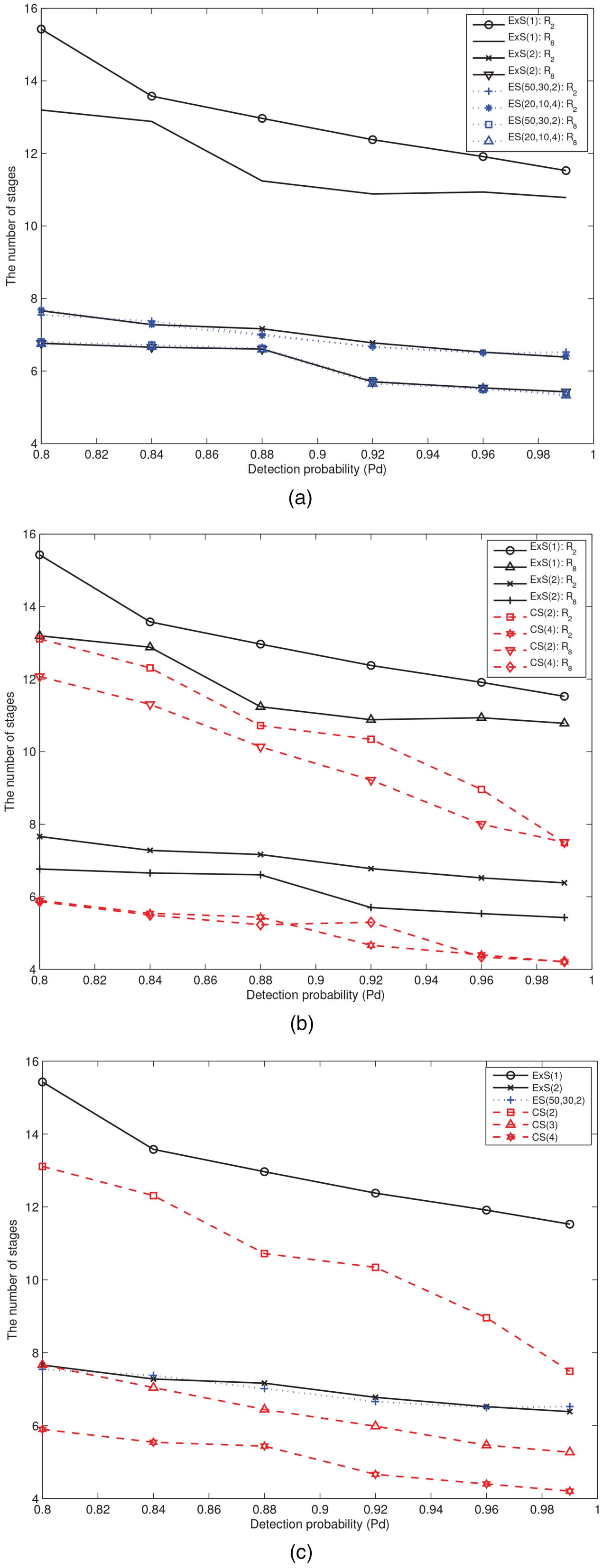
Figure 5 repeats the simulation analysis shown in Fig. 4 for various false alarm rates while fixing P_d at 0.99. Similar conclusion is arrived from Fig. 5 as well, where it can be noticed in the summary Fig. 5(c) that the ES algorithms outperform ExS(1), and perform essentially as well as ExS(2). On the other hand, CS(N) outperforms the others with increasing N.

4.2. Real System

Our algorithms are applied to a real system, the so-called “Documatch,” which is an R-matrix representation of the Pitney Bowes Integrated Mail System that takes an original document from a Microsoft Windows based personal computer and turns it into a finished and properly-addressed mail item in a sealed envelope (see, for example, [6]). The R matrix of this system, denoted hereafter as R_d, has 258 components and 179 sensors.

First, let us compare the simulated R matrices used in earlier simulations and the R_d of the Documatch system. Out of the simulated R matrices used in this section, we select R_2 for comparison with R_d. For each of these matrices we counted the *number of components monitored by each sensor* and the *number of sensors monitoring each component.* The result is summarized in Fig. 6.

The performance comparison of all three algorithms in terms of $\bar{k}$ vs. P_d is shown in Fig. 7. Due to the size of the R_d matrix, the exhaustive search is performed only for $N = 1$, i.e., we do not have ExS(2) in the simulations because of the time required to complete the simulation. The figure confirms the earlier conclusions arrived through the simulated R matrices.



(a)

(b)

(c)

Fig. 4. Comparison of algorithms, in terms of the average number of stages $\bar{k}$ vs. probability of detection P_d. The false alarm rate is fixed at $P_f = 0.01$ in all the figures. (a) Comparison of ExS and ES on matrices R_2 and R_8. (b) Comparison of ExS and CS on matrices R_2 and R_8. (c) Comparison of all algorithms, ExS, ES and CS, on matrix R_2.

4.3. Computational complexity analysis

The computational cost arises mainly based on how many combinations of sensors there are, i.e., the size of the candidate set for calculation of information gain. For example, if the system has 100 sensors, with the exhaustive search in ExS(2), we need to calculate information gain for $\sum_{p=1}^{2} 100!/((100-p)!p!) = 5050$ combinations of sensors at each stage, as 2 sensors are allowed to be tested at each stage and all possible combinations of two sensors can be used to test in the respect of the limit of cost. It means that $\tilde{S}_c(k) \in \{\text{all possible combinations of two sensors}\}$ and $|\{\tilde{S}_c(k)\}| = 5050$. If it takes 2 stages to isolate the faulty element, i.e. $k_r = 2$ for a particular run r, the total number of combinations (i.e. $n_t := \sum_{k=1}^{k_r} |\{\tilde{S}_c(k)\}|$) is 10100. Let us consider another example: Assume that only one sensor is allowed to test at each stage and it takes 110 stages to isolate the faulty component; in this case $n_t = 11000$. It should be noted that, in a real situation, due to pruning the number of these combinations $|\{\tilde{S}_c(k)\}|$ varies for each stage. In summary, n_t is the accumulated size of candidate set of $S_c(k)$ over all stages until the faulty component is isolated for a particular run. The average of n_t^r is defined as N_t, where

$$N_t = \frac{1}{N_m}\sum_{r=1}^{N_t} n_t^r, \qquad (24)$$

in which

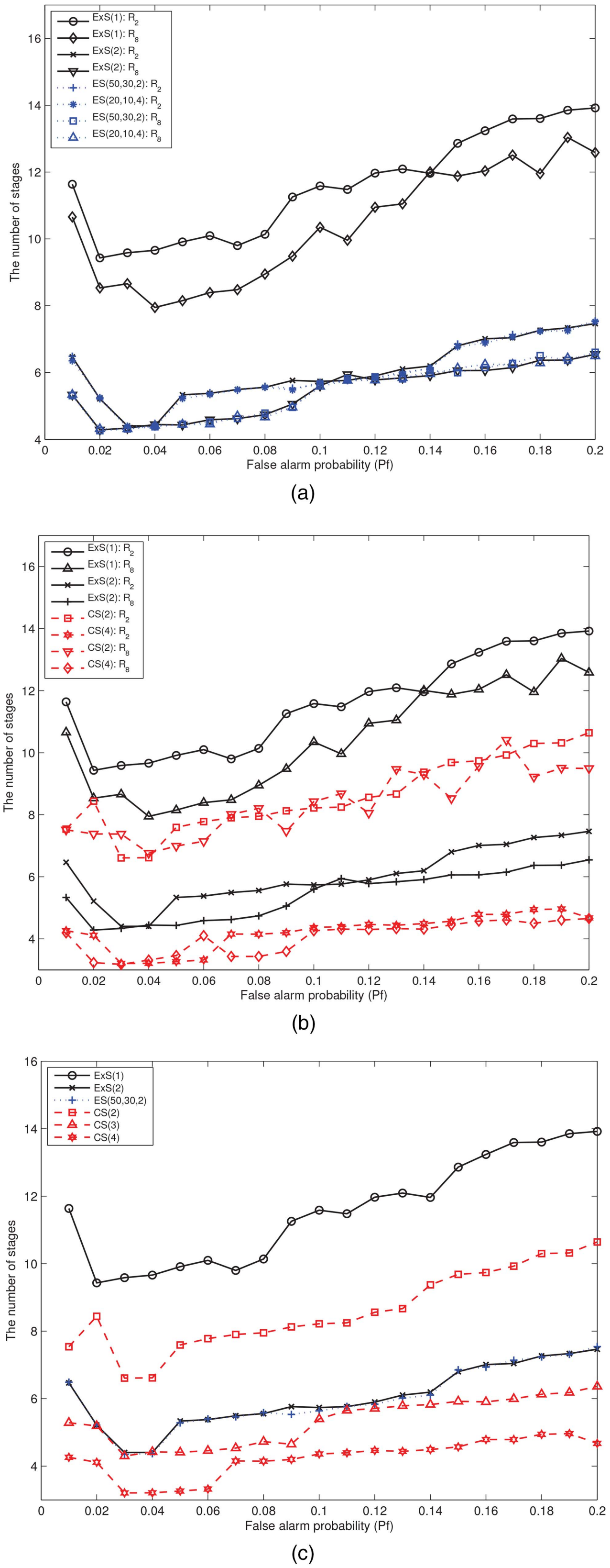
$$n_t^r = \sum_{k=1}^{k_r} |\{\tilde{S}_c^r(k)\}|. \qquad (25)$$

The comparison of all the algorithms in terms of N_t vs. P_d is shown in Fig. 8. It shows that CS(N) requires much reduced computation compared to ExS(2) for $N = 2, 3$, and 4. Further it must be emphasized that N_t reduces with increasing N for CS whereas N_t increases with increasing N for ExS(N). At this point it must be re-emphasized that the CS(N) algorithm also reduces the average number of iterations required to isolate the faulty element with increasing number of N (see Figs. 4 and 5) so that N_t (or computations) is reduced even with increasing N. Fig. 9 shows the comparison of all the algorithms in terms of N_t vs. P_f for fixed P_d. It shows that CS(N) requires a similar amount of—and at times lower—computation compared to ExS(1).

The comparison of N_t for the Documatch system is shown in Fig. 10. Due to the size of the system the computational complexity of ExS(2) was not computed. The figure confirms the results of the experiments performed on the R_2 and R_8 matrices. It confirms that CS(N) requires slightly less computation with increasing N than ExS(1), especially with higher probability of detection P_d and lower false alarm rate P_f.

5. CONCLUSIONS

Two algorithms for single fault diagnosis under communication constraints are presented. Both of the al-



(a)

(b)

(c)

Fig. 5. Comparison of algorithms, in terms of the average number of stages $\bar{k}$ vs. probability of false-alarm P_f. The probability of detection is fixed at $P_d = 0.99$ in all the figures. (a) Comparison of ExS and ES on matrices R_2 and R_8. (b) Comparison of ExS and CS on matrices R_2 and R_8. (c) Comparison of all algorithms, ExS, ES and CS, on matrix R_2.

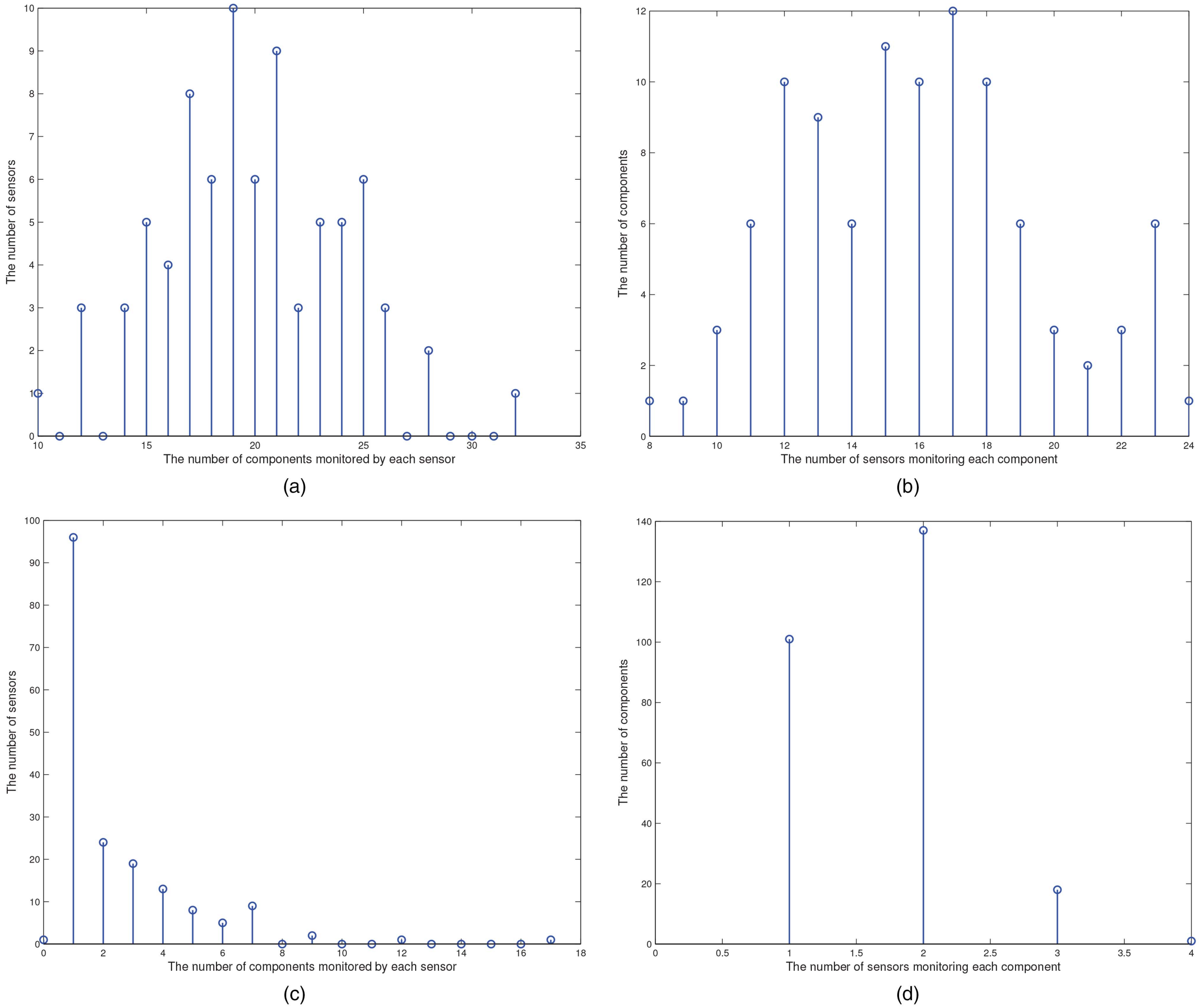


Fig. 6. Summary of comparisons: Simulated R_2 matrix vs. practical Documatch R_d matrix. (a) Simulated R_2 matrix. (b) Simulated R_2 matrix. (c) Documatch R_d matrix. (d) Documatch R_d matrix.

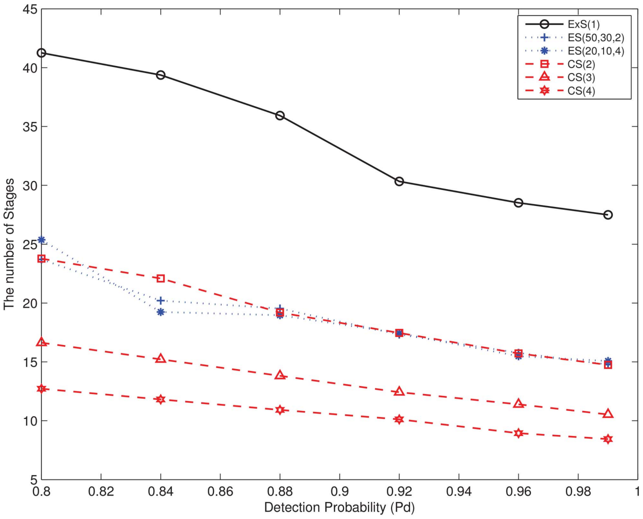
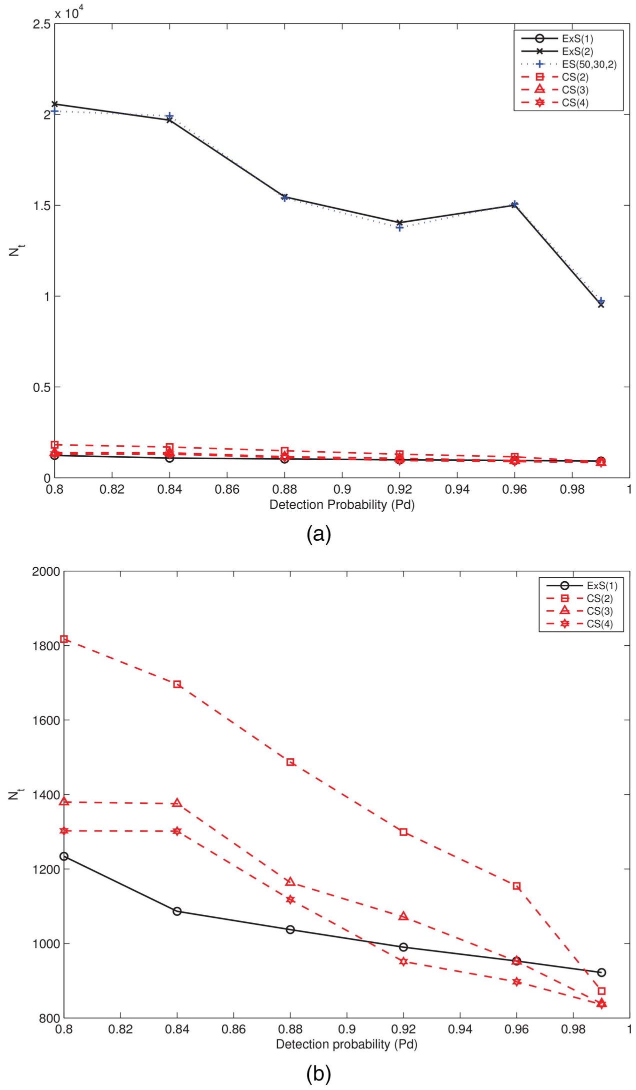


Fig. 7. Comparison of all algorithms in terms of $\bar{k}$ vs. P_d on Documatch system for $P_f = 0.01$.

gorithms are concerned with the selection of (possibly) several sensors at a time in order to reduce the number of iterations/communication. The algorithms differ in their approach in selecting multiple sensors that maximize the information gain. The first algorithm, termed the exhaustive search (ES) method, introduces several thresholds in order to eliminate sensors that are less informative, so that fewer sensors form the candidate set for the maximization of information gain. The second approach, referred to as correlative search (CS), selects the candidate set one-by-one based on the correlation between the information state and the elements of the reachability matrix. Both of the proposed approaches demonstrate their ability to reduce the number of iterations in fault diagnosis.

APPENDIX

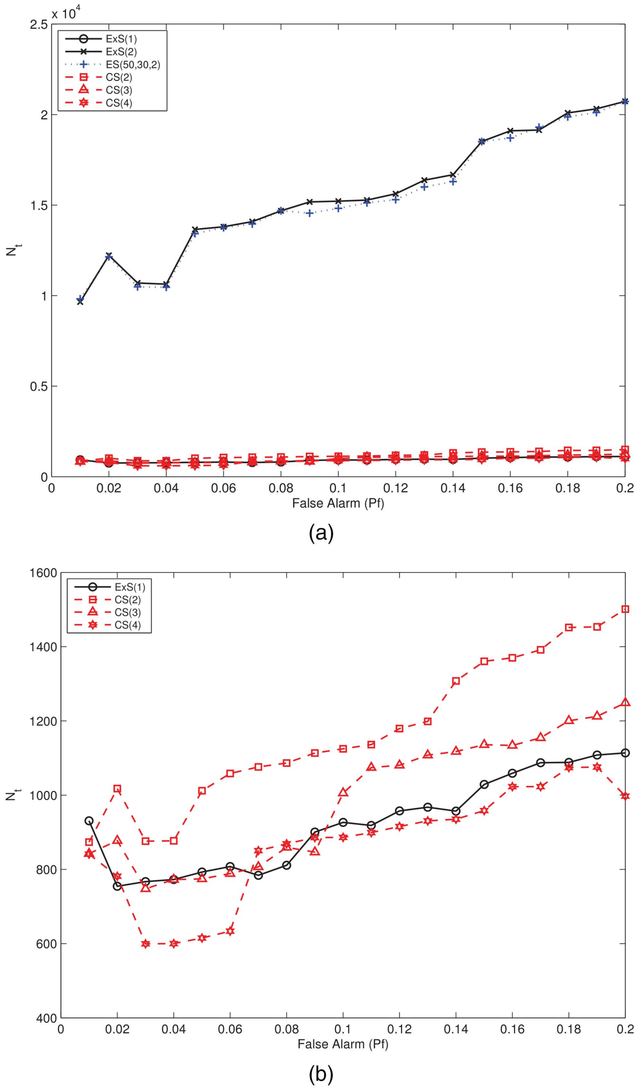
Given a subset S_c of S, $\tilde{\mathbf{o}}_t$ denotes a vector of outcomes of the sensors in a set S_c, and the set $\tilde{O} = \{\tilde{\mathbf{o}}_t\}$



(a)

(b)

Fig. 8. Comparison of all algorithms on R_2 for varying P_d with $P_f = 0.01$. (a) Comparison of N_t for various algorithms. (b) Closer view of the bottom portion of Fig. 8(a) above.



(a)

(b)

Fig. 9. Comparison of all algorithms on R_2 for varying P_f with $P_d = 0.99$. (a) Comparison of N_t for various algorithms. (b) Closer view of the bottom portion of Fig. 9(a) above.

consists of all possible vector which can be generated by the sensors. And J is the set of the indices of sensors in the set S_c. The information gain is defined with information state as

$$\mathrm{IG}(\{\pi_i(k)\},\mathbf{S}(k)) = H(\{\pi_i(k)\}) - H(\{\pi_i(k)\} \mid S_c(k)) \tag{26}$$

where

$$H(\{\pi_i(k)\}) = -\sum_i \pi_i(k)\log\pi_i(k). \tag{27}$$

The following is obtained:

$$\begin{aligned}
&\mathrm{IG}(\{\pi_i(k)\},S_c(k))\\
&\quad = -\sum_i \pi_i(k)\log\pi_i(k)\\
&\qquad + \sum_{\tilde{\mathbf{o}}_t\in\tilde{O}}\sum_{i=0}^{m}\pi_i(k)\prod_{j\in J}(\tilde{\mathbf{o}}_t(j)d_{i,j} + (1-\tilde{\mathbf{o}}_t(j))(1-d_{i,j}))\\
&\qquad \times \log\left\{\pi_i(k)\prod_{q\in J}(\tilde{\mathbf{o}}_t(q)d_{i,q} + (1-\tilde{\mathbf{o}}_t(q))(1-d_{i,q}))\right\}\\
&\qquad - \sum_{\tilde{\mathbf{o}}_t\in\tilde{O}}\sum_{i=0}^{m}\pi_i(k)\prod_{j\in J}(\tilde{\mathbf{o}}_t(j)d_{i,j} + (1-\tilde{\mathbf{o}}_t(j))(1-d_{i,j}))\\
&\qquad \times \log\left\{\sum_{p=0}^{m}\prod_{q\in J}\pi_i(k)(\tilde{\mathbf{o}}_t(q)d_{p,q} + (1-\tilde{\mathbf{o}}_t(q))(1-d_{p,q}))\right\}.
\end{aligned} \tag{28}$$

PROOF The conditional entropy of information state is described by

$$H(\{\pi_i(k)\} \mid S_c) = H(\{\pi_i(k)\} \mid S_c, I_{k-1}) \tag{29}$$

$$= \sum_{\tilde{\mathbf{o}}_t\in\tilde{O}} \mathrm{Prob}(\tilde{\mathbf{o}}_t \mid I_{k-1}) H(\{\pi_i(k)\} \mid \tilde{\mathbf{o}}_t, I_{k-1}). \tag{30}$$

The entropy $H(\{\pi_i(k)\} \mid \tilde{\mathbf{o}}_t, I_{k-1})$ is given as

$$\begin{aligned} &H(\{\pi_i(k)\} \mid \tilde{\mathbf{o}}_t, I_{k-1}) \\ &\quad = -\sum_{i=0}^{m} \mathrm{Prob}(\pi_i(k) \mid \tilde{\mathbf{o}}_t, I_{k-1}) \log(\mathrm{Prob}(\pi_i(k) \mid \tilde{\mathbf{o}}_t, I_{k-1})) \\ &\quad = -\sum_{i=0}^{m} \frac{\pi_i(k)}{\mathrm{Prob}(\tilde{\mathbf{o}}_t \mid I_k)} \prod_{j\in J} (\tilde{\mathbf{o}}_t(j) d_{i,j} + (1-\tilde{\mathbf{o}}_t(j))(1-d_{i,j})) \\ &\qquad \times \log\left\{ \frac{\pi_i(k)}{\mathrm{Prob}(\tilde{\mathbf{o}}_t \mid I_k)} \prod_{j\in J} (\tilde{\mathbf{o}}_t(j) d_{ij} + (1-\tilde{\mathbf{o}}_t(j))(1-d_{ij})) \right\} \end{aligned} \tag{31}$$

where

$$\begin{aligned} &\mathrm{Prob}(\pi_i(k) \mid \tilde{\mathbf{o}}_t, I_{k-1}) \\ &\quad = \mathrm{Prob}(\tilde{\mathbf{o}}_t \mid \pi_i(k), I_{k-1}) \frac{\mathrm{Prob}(\pi_i(k) \mid I_{k-1})}{\mathrm{Prob}(\tilde{\mathbf{o}}_t \mid I_{k-1})} \\ &\quad = \frac{\pi_i(k)}{\mathrm{Prob}(\tilde{\mathbf{o}}_t \mid I_{k-1})} \prod_{j\in J} (\tilde{\mathbf{o}}_t(j) d_{ij} + (1-\tilde{\mathbf{o}}_t(j))(1-d_{ij})) \end{aligned} \tag{32}$$

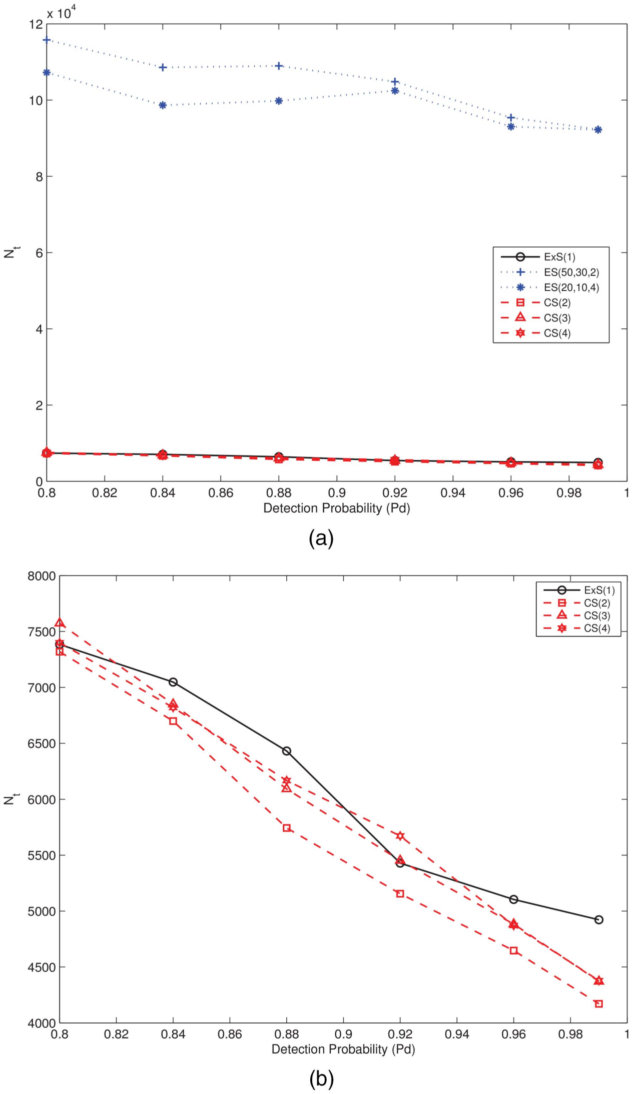
and

$$\begin{aligned} &\mathrm{Prob}(\tilde{\mathbf{o}}_t \mid I_{k-1}) \\ &\quad = \sum_{i=0}^{m} \prod_{j\in J} \mathrm{Prob}(\tilde{\mathbf{o}}_t(j) \mid x_i = 1, I_{k-1}) \mathrm{Prob}(x_i = 1 \mid I_{k-1}) \\ &\quad = \sum_{i=0}^{m} \prod_{j\in J} \pi_i(k)(\tilde{\mathbf{o}}_t(j) d_{ij} + (1-\tilde{\mathbf{o}}_t(j))(1-d_{ij})). \end{aligned} \tag{33}$$

Thus,

$$\begin{aligned} &H(\{\pi_i(k)\} \mid S_c) \\ &\quad = \sum_{\tilde{\mathbf{o}}_t \in \tilde{O}} \mathrm{Prob}(\tilde{\mathbf{o}}_t \mid I_{k-1}) H(\{\pi_i(k)\} \mid \tilde{\mathbf{o}}_t, I_{k-1}) \\ &\quad = -\sum_{\tilde{\mathbf{o}}_t \in \tilde{O}} \sum_{i=0}^{m} \pi_i(k) \prod_{j\in J} (\tilde{\mathbf{o}}_t(j) d_{i,j} + (1-\tilde{\mathbf{o}}_t(j))(1-d_{i,j})) \\ &\qquad \times \log\left\{ \pi_i(k) \prod_{j\in J} (\tilde{\mathbf{o}}_t(j) d_{i,j} + (1-\tilde{\mathbf{o}}_t(j))(1-d_{i,j})) \right\} \\ &\qquad + \sum_{\tilde{\mathbf{o}}_t \in \tilde{O}} \sum_{i=0}^{m} \pi_i(k) \prod_{j\in J} (\tilde{\mathbf{o}}_t(j) d_{i,j} + (1-\tilde{\mathbf{o}}_t(j))(1-d_{i,j})) \\ &\qquad \times \log\left\{ \sum_{p=0}^{m} \prod_{q\in J} \pi_i(k)(\tilde{\mathbf{o}}_t(q) d_{p,q} + (1-\tilde{\mathbf{o}}_t(q))(1-d_{p,q})) \right\} \end{aligned} \tag{34}$$

and hence the information gain is as in (28).



(a)

(b)

Fig. 10. Comparison of all algorithms on Documatch for varying P_d with $P_f = 0.01$. (a) Comparison of N_t for various algorithms. (b) Closer view of the bottom portion of Fig. 10(a) above.

REFERENCES

[1] D. Blough and A. Pelc. Complexity of fault diagnosis in comparison models. *IEEE Transactions on Computers*, 41(3):318–324, Mar. 1992.

[2] R. Boumen, I. S. M. de Jong, J. W. H. Vermunt, J. M. van de Mortel-Fronczak, and J. E. Rooda. Test sequencing in complex manufacturing systems. *IEEE Transactions on Systems, Man, and Cybernetics: Part A.*, 38(1):25–37, Jan. 2008.

[3] R. Boumen, S. Ruan, I. S. M. de Jong, J. M. van de Mortel-Fronczak, J. E. Rooda, and K. R. Pattipati. Hierarchical test sequencing for complex systems. *IEEE Transactions on Systems, Man, and Cybernetics: Part A.*, 39(3):640–649, May 2009.

[4] J. M. Christensen and J. M. Howard, editors. *Human Detection and Diagnosis of System Failure*, chapter Field experience in maintenance. New York: Plenum, 1981.

[5] O. Erdinc, C. Brideau, P. Willett, and T. Kirubarajan. Fast diagnosis with sensors of uncertain quality. *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, 38(4):1157–1165, 2008.

[6] O. Erdinc, C. Brideau, P. Willett, and T Kirubarajan. The problem of test latency in machine diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics—Part A.*, 38(1):88–92, 2008.

[7] M. R. Garey. Optimal binary identification procedures. *SIAM Journal on Applied Mathematics*, 23(2):173–186, 1972.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. New York: W. H. Freeman and Company, 1979.

[9] L. Hyafil and R. L. Rivest. Constructing optimal binary decision trees in NP-complete. *Information Processing Letters*, 5(1):15–17, 1976.

[10] T. Le and C. Hadjicostis. Max-product algorithms for the generalized multiple-fault diagnosis problem. *IEEE Transactions on Systems, Man, and Cybernetics: Part B*, 37(6):1607–1621, Dec. 2007.

[11] Z. Mao, B. Jiang, and P. Shi. H_∞ fault detection filter design for networked control systems modelled by discrete Markovian jump systems. In *2010 IEEE International conference on Information and Automation*, Jun. 2010.

[12] M. Mastrianni. Virtual integrated test and diagnosis. *Sikosrky Pres. ARPA*, Nov. 9 1994.

[13] B. M. E. Moret. Decision trees and diagrams. *Computing Surveys*, 14(4):593–623, 1982.

[14] K. R. Pattipati and M. G. Alexandridis. Application of heuristic search and information theory to sequential fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics*, 20(4):872–887, 1990.

[15] V. Raghavan, M. Shakeri, and K. R. Pattipati. Optimal and near-optimal test sequencing algorithms with realistic test models. *IEEE Transactions on Systems, Man, and Cybernetics: Part A*, 29(1):11–26, 1999.

[16] V. Raghavan, M. Shakeri, and K. R. Pattipati. Test sequencing algorithms with unreliable tests. *IEEE Transactions on Systems, Man, and Cybernetics: Part A*, 29(4):347–357, 1999.

[17] S. Ruan, Y. Zhou, F. Yu, K. R. Pattipati, P. Willett, and A Patterson-Hine. Dynamic multiple-fault diagnosis with imperfect tests. *IEEE Transactions on Systems, Man, and Cybernetics: Part A*, 39(6):1224–1236, Nov. 2009.

[18] M. Shakeri, K. R. Pattipati, V. Raghavan, and A. Patterson-Hine. Optimal and near-optimal algorithms for multiple fault diagnosis with unreliable tests. *IEEE Transactions on Systems, Man, and Cybernetics: Part C*, 28(3):431–440, 1998.

[19] M. Shakeri, V. Raghavan, K. R. Pattipati, and A. Patterson-Hine. Sequential testing algorithms for multiple fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Part A*, 30(1):1–14, 2000.

[20] S. Singh, A. Kodali, K. Choi, K. Pattipati, S. M. Namburu, S. C. Sean, D. V. Prokhorov, and L. Qiao. Dynamic multiple fault diagnosis: Mathematical formulations and solution techniques. *IEEE Transactions on Systems, Man, and Cybernetics: Part A*, 39(1):160–176, 2009.

[21] F. Tu and K. R. Pattipati. Rollout strategy for sequential fault diagnosis. *IEEE Transactions on Systems, Man, and Cybernetics: Part A*, (1):86–99, 2003.

[22] Y. Wang, S. Ding, H. Ye, L. Wei, P. Zhang, and G. Wang. Fault detection of networked control systems with packet based periodic communication. *International Journal of Adaptive Control and Signal Processing*, 23(8):682–698, 2009.

[23] Y. Wang, H. Ye, S. Ding, Y. Cheng, P. Zhang, and G. Wang. Fault detection of networked control systems with limited communication. *International Journal of Control*, 82(7):1344–1356, 2009.

[24] J. Ying, T. Kirubarajan, K. R. Pattipati, and A. Patterson-Hine. A hidden Markov model-based algorithm for fault diagnosis with partial and imperfect tests. *IEEE Transactions on Systems, Man, and Cybernetics: Part C*, 30(4), 2000.

[25] P. Zhang, S. X. Ding, P. M. Frank, and M. Sader. Fault detection of networked control systems with missing measurements. In *5th Asian Control Conference*, volume 2, pages 1258–1263, Jul. 2004.

[26] P. Zhang, S. X. Ding, G. Z. Wang, and D. H. Zhou. Fault detection of linear discrete-time periodic systems. *IEEE Transactions on Automatic Control*, 50(2):239–244, 2005.

Sora Choi received her Ph.D. degree, in Mathematics, from Seoul National University, Seoul, Korea in 2003. She was a researcher at the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea from 2003 to 2004, and was an instructor at Seoul National University between 2000 and 2002, Kyunghee and Konkuk Universities, in Korea, in 2007. She received a Graduate Predoctoral Fellowship in 2011 and is currently a graduate student/research assistant at the University of Connecticut, Storrs, where she is pursuing a Ph.D. degree in Electrical Engineering. Her research interests are in target tracking, detection theory, signal processing, and fault diagnosis.

Balakumar Balasingam earned his Ph.D. from McMaster University, Canada in 2008. Currently, he is an assistant research professor at the University of Connecticut in the Department of Electrical and Computer Engineering. Before joining UConn in 2010 as a postdoctoral fellow, he has worked as a postdoctoral fellow at the University of Ottawa, Canada from 2008. His research interests include signal processing, distributed information fusion and machine learning.

Peter Willett has been a faculty member in the Electrical and Computer Engineering Department at the University of Connecticut since 1986, following his Ph.D. from Princeton University. His areas include statistical signal processing, detection, sonar/radar, communications, data fusion, and tracking. He is an IEEE Fellow. He was EIC for *IEEE Transactions on AES*, and remains AE for the *IEEE AES Magazine* plus ISIF's *Journal of Advances in Information Fusion* and senior editor for *IEEE Signal Processing Letters*, and is a member of the editorial board of IEEE's Special Topics in Signal Processing. He is a member of the IEEE AESS Board of Governors (and is now VP Pubs) and of the IEEE Signal Processing Society's Sensor-Array and Multichannel (SAM) technical committee (and is now Vice Chair). He has served as Technical Chair for the 2003 SMCC and the 2012 SAM workshop; and as Executive/General Chair for the 2006, 2008 and 2011 FUSION conferences.