Dynamic Multiple Fault Diagnosis: Mathematical Formulations and Solution Techniques

Satnam Singh, *Member, IEEE*, Anuradha Kodali, Kihoon Choi, Krishna R. Pattipati, *Fellow, IEEE*, Setu Madhavi Namburu, Shunsuke Chigusa Sean, Danil V. Prokhorov, and Liu Qiao

***Abstract*—Imperfect test outcomes, due to factors such as unreliable sensors, electromagnetic interference, and environmental conditions, manifest themselves as missed detections and false alarms. This paper develops near-optimal algorithms for dynamic multiple fault diagnosis (DMFD) problems in the presence of imperfect test outcomes. The DMFD problem is to determine the most likely evolution of component states, the one that best explains the observed test outcomes. Here, we discuss four formulations of the DMFD problem. These include the deterministic situation corresponding to perfectly observed coupled Markov decision processes to several partially observed factorial hidden Markov models ranging from the case where the imperfect test outcomes are functions of tests only to the case where the test outcomes are functions of faults and tests, as well as the case where the false alarms are associated with the nominal (fault free) case only. All these formulations are intractable NP-hard combinatorial optimization problems. Our solution scheme can be viewed as a two-level coordinated solution framework for the DMFD problem. At the top (coordination) level, we update the Lagrange multipliers (coordination variables, dual variables) using the subgradient method. At the bottom level, we use a dynamic programming technique (specifically, the Viterbi decoding or Max-sum algorithm) to solve each of the subproblems, one for each component state sequence. The key advantage of our approach is that it provides an approximate duality gap, which is a measure of the suboptimality of the DMFD solution. Computational results on real-world problems are presented. A detailed performance analysis of the proposed algorithm is also discussed.**

***Index Terms*—Dynamic faults, hidden Markov models, imperfect tests, intermittent faults, multiple fault diagnosis.**

I. Introduction

ONLINE vehicle health monitoring and fault diagnosis is essential to improve the vehicle availability via condition-based and opportunistic maintenance, and to reduce maintenance and operational costs by seamlessly integrating the onboard and offline diagnosis, thereby reducing troubleshooting time. During online (dynamic) fault diagnosis, the test outcomes are obtained over time as compared to static fault diagnosis where the observed test outcomes are available as a block. Online vehicle health monitoring heavily relies on the extensive processing of data in real time, which is made possible by smart onboard sensors. Using these intelligent sensors, the system parameters that are essential to vehicle fault diagnosis can be transmitted to an onboard diagnostic inference engine.

A significant technical challenge in onboard vehicle health monitoring is the quality of tests. Generally, the tests are imperfect due to unreliable sensors, electromagnetic interference, environmental conditions, or aliasing inherent in the signature analysis of onboard tests. The onboard tests can be symptoms, manifestations, alarms, or residuals generated from sensor data using trending, range checking, dynamic thresholding, etc. The imperfect tests introduce additional elements of uncertainty into the diagnostic process: The *pass* outcome of a test does not guarantee the integrity of components under test because the test may have missed a fault; on the other hand, a *fail* outcome of a test does not mean that one or more of the implicated components are faulty because the test outcome may have been a false alarm. Hence, it is desired that an onboard diagnostic algorithm should be able to accommodate missed detections and false alarms in test outcomes. The performance of onboard diagnosis can be improved by incorporating the knowledge of reliabilities of tests and by incorporating temporal correlations of test outcomes.

The hidden Markov model (HMM) is a natural choice here to represent the individual component states of the system. The HMM is a doubly embedded stochastic process with an underlying unobservable (hidden) stochastic process (individual component state evolution), which can be observed through another set of stochastic processes (i.e., uncertain test outcome sequences). The individual component state HMMs are coupled through the observation process. Consequently, the fault diagnosis problem corresponds to a factorial HMM (FHMM), where each HMM characterizes the individual component states of the system. The sequence of uncertain test outcomes are probabilistic functions of the underlying Markov chains characterizing the evolution of system states. In this paper, we investigate the

Manuscript received August 29, 2007; revised February 24, 2008. Current version published December 17, 2008. This work was supported in part by the Toyota Technical Center and in part by the Office of Naval Research under Contract 00014-00-1-0101. Earlier versions of this paper were published at the IEEE Aerospace Conference, Big Sky, MT, March 2007 and the DX-07 Workshop, Nashville, TN, May 2007. This paper was recommended by Associate Editor G. Biswas.

S. Singh is with General Motors India Science Laboratory, Bangalore 560066, India (e-mail: satnam.singh@gm.com).

A. Kodali, K. Choi, and K. R. Pattipati are with the Electrical and Computer Engineering Department, University of Connecticut, Storrs, CT 06269 USA (e-mail: krishna@engr.uconn.edu).

S. M. Namburu, S. C. Sean, D. V. Prokhorov, and L. Qiao are with the Technical Research Department, Toyota Motor Engineering and Manufacturing North America, Inc., Erlanger, KY 41018 USA.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2008.2007986

problem of determining the most likely states of components, given a set of partial and unreliable test outcomes over time.

A. *Organization of This Paper*

This paper is organized as follows. In Section II, we discuss the previous research on multiple fault diagnosis (MFD). We formulate the NP-hard dynamic MFD (DMFD) problem with imperfect test outcomes in Section III. Four formulations of the DMFD problem are also discussed in Section III. In Section IV, we decompose the DMFD problem formulation 1 using Lagrangian relaxation algorithm. The DMFD problem is decoupled into a set of parallel subproblems (involving dynamic single HMM state estimation problems) using Lagrange multipliers. A dynamic programming technique (the Viterbi algorithm) is used to solve each of the subproblems, and their solutions are used to update the Lagrange multipliers via the subgradient method. Feasible (primal) solutions are constructed using the dual solutions. In Sections V–VII, we discuss the details of DMFD problems 2, 3, and 4, respectively. Online DMFD problem is solved using a sliding window method, which is presented in Section VIII. The simulation results of DMFD problem 1 is performed on several real-world data sets to validate our approach. Section IX discusses the simulation results of both block and online DMFD problems. Finally, this paper concludes with a summary and future research directions in Section X.

II. Previous Work

The MFD problem originates in several fields, such as medical diagnosis [1], [2], error correcting codes (ECCs), speech recognition, distributed computer systems, and networks [3]. The MFD problem in large-scale systems with unreliable tests was first considered by Shakeri *et al.* in [4]. They proposed near-optimal algorithms using Lagrangian relaxation and subgradient optimization methods for the static MFD problem. In the area of distributed system management, the MFD problem is studied by Odintsova *et al.* in [3]. They utilized an adaptive diagnostic technique, termed active probing, for fault diagnosis and isolation. A probe can be viewed as a test in our terminology; the purpose of a probe is to check the set of system components on the probed path. The probe outcomes determine if one or more of the components on the probed path are faulty or normal. Given the probe outcomes, a diagnostic matrix (dependence matrix (D-matrix), diagnostic dictionary, reachability matrix) defining the relationship among the probes and component faults, as well as the initial system state, they developed a sequential multifault algorithm to diagnose the system state. They considered the probe outcomes as being deterministic, which is analogous to the assumptions made in our Problem 4 and in the work described in [5]–[7]. In [8], Le and Hadjicostis applied graphical model-based decoding algorithms to the MFD problem in the presence of unreliable tests. They proposed a suboptimal belief propagation algorithm used to decode low-density parity check codes. They considered a fault model, where tests are asymmetric, i.e., the D-matrix is not binary and the test outcomes are also unreliable, and they termed it the Y model. Their implementation is parallel to our Problem formulation 1; however, they considered only the static case.

The dynamic single fault diagnosis problem using HMM formalism was first proposed by Ying *et al.* [9], where it is assumed that, at any time, the system has at most one component state present. This modeling is somewhat unrealistic for most real-world systems. Another version of the dynamic fault diagnosis problem was studied in [10]: Unknown probabilities of sensor error, incompletely populated sensor observations, and multiple faults were allowed, but the faults could only occur or clear once per sampling interval.

In the dynamic single fault framework [9], a hidden Markov modeling framework was adopted, and a moving window Viterbi algorithm was used to infer the evolution of component states. In the multiple fault case, the state space of HMM increases exponentially from $(m+1)$ to 2^m, where m is the number of possible component states. Consequently, the HMM-based method would be viable only for small-sized systems. The solution method proposed in [10] is a multiple hypothesis tracking approach, where at each observation epoch, k best component state configurations are stored. In that paper, the missed-detection/false-alarm process was a property of the sensor rather than the fault, with the effect that the underlying inference process could not be decoupled into an FHMM. In [10], at each epoch, all candidate fault sets derived from the previously identified faults are listed, based on at most one change per epoch assumption. Then, of all $k(m+1)$ possible candidate sets, each has its score calculated; the candidate set which obtains the highest score is selected as the inference result at the epoch, and the candidates with the k-best scores are updated. The method is equivalent to enumeration in a limited search space; consequently, it is either computationally expensive or far from optimal. A major contribution of this paper is that the missed-detection/false-alarm process is modeled as being a property of the component state: The model is perhaps less realistic, but the computational benefit of an FHMM is large.

Another approach, developed by Ruan *et al.* [11], decomposes the original DMFD problem into a series of decoupled subproblems, one for each epoch. For a single-epoch MFD, they developed a deterministic simulated annealing (DSA) method, which is inspired by its sibling stochastic simulated annealing and the approximate belief revision heuristic algorithm [1], [2]. The single-epoch MFD was extended to incorporate component states of multiple consecutive epochs. In addition, they applied a local search and update scheme to further smooth the "noisy" diagnoses stemming from imperfect test results and, consequently, increase the accuracy of fault diagnosis.

The DMFD problem can be viewed as an FHMM, which is discussed in the machine learning literature [12]. Here, the HMM state is factored into multiple state variables and represented in a distributed manner. The authors discussed an exact algorithm for inference computations in the FHMM framework. In this framework, inference and learning involves computing the posterior probabilities of multiple hidden layers (or states), given the test outcomes. However, due to the combinatorial nature of the hidden state representation, the exact algorithm is intractable. They presented approximate inference algorithms based on Gibbs sampling and variational methods. The latter

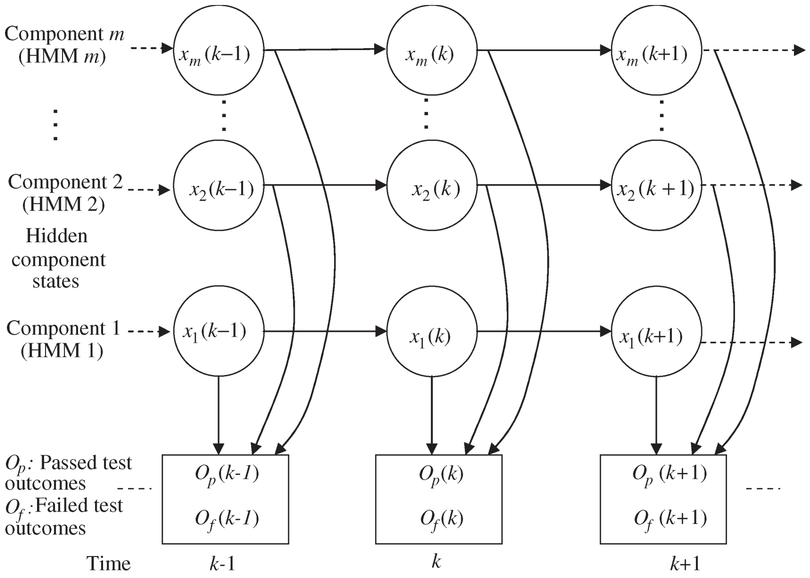

Fig. 1. DMFD problem viewed as an FHMM.

methods are similar to Lagrangian relaxation, although motivated from a Fenchel duality perspective [1], [2], [4], [5], [13].

In this paper, we extend the work of Ruan *et al.* [11], Shakeri *et al.* [4], and Tu *et al.* [5] on MFD to solve the DMFD problem by combining the Viterbi algorithm and Lagrangian relaxation in an iterative way. Depending on the probabilistic assumptions on fault–test relationships and test outcomes, one obtains various DMFD formulations. In summary, the contributions of this paper are the following: 1) a primal–dual optimization framework to solve the DMFD problem; 2) four formulations of the DMFD problem along with their solutions; 3) simulation results on several real-world systems for the first and most general formulation of the DMFD problem; 4) a comparison of the results between the subgradient and the DSA methods [11]; and 5) simulation results, along with performance analysis, of the online DMFD problem using a sliding window method.

III. DMFD Problem Formulations

The DMFD problem consists of a set of possible component states in a system and a set of binary test outcomes that are observed at each sample (observation, decision) epoch. Formally, $S = \{s_1, \ldots, s_m\}$ is a finite set of m components (failure sources) associated with the system. The state of component s_i is denoted by $x_i(k)$ at epoch k, where $x_i(k) = 1$ if failure source s_i is present; otherwise, $x_i(k) = 0$. Here, $\kappa = \{0, 1, \ldots, k, \ldots, K\}$ is the set of discretized observation epochs. The status of all component states at epoch k is denoted by $\underline{x}(k) = \{x_1(k), x_2(k), \ldots, x_m(k)\}$. We assume that the initial state $\underline{x}(0)$ is known (or its probability distribution is known). Component states are assumed to be independent. Each test outcome provides information on a subset of the component states. At each sample epoch, a subset of test outcomes is available. Tests are imperfect in the sense that the outcomes of some of the tests could be missing, and tests have missed-detection/false-alarm processes associated with them. The observations consist of imperfect binary test outcomes and are characterized by sets of passed test outcomes O_p and failed test outcomes O_f.

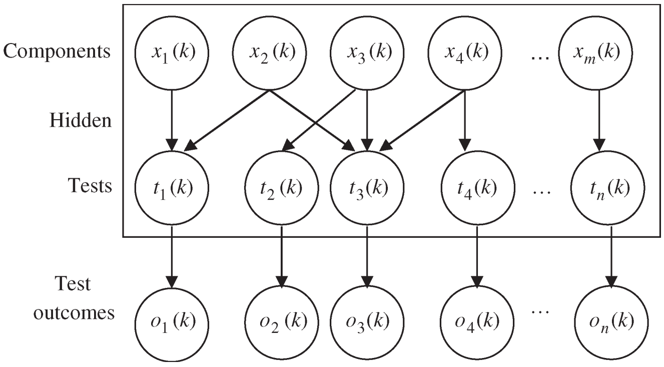

Fig. 2. Tripartite digraph for DMFD problem.

Our problem is to determine the time evolution of component states based on imperfect test outcomes observed over time. Fig. 1 shows the DMFD problem viewed as an FHMM. The hidden component state of the ith HMM at time epoch k is denoted by $x_i(k)$. Each component state $x_i(k)$ can be $x_i(k) = 1$ if failure source s_i is present; otherwise, $x_i(k) = 0$. The observations at each epoch are subsets of binary outcomes[1] of tests $O = \{o_1, o_2, \ldots, o_n\}$, i.e., $o_j \in \{\text{pass}, \text{fail}\} = \{0, 1\}$. Fig. 2 shows the DMFD problem as a tripartite digraph at epoch k. Component states, tests, and test outcomes represent the nodes of the digraph. Here, the true states of the component states and tests are hidden. $P = \{Pd, Pf\}$ represents a set of probabilities of detection and false alarm, which is defined differently for each of the DMFD problem formulations. Here, we assumed that the probabilities $P = \{Pd, Pf\}$ are available; otherwise, they can be learned during the training and validation. We also

[1] Extension to multivalued component states and test outcomes is straightforward. In the general case, each component i will be in one of l_i states, and each test j has multiple outcomes o_j. The detection capabilities of each test for a component is defined by an $l_i \times (o_j - 1)$ matrix. Equation (12) needs to be modified to account for multistate components and multioutcome tests.

define the matrix $D = [d_{ij}]$ as the D-matrix, which represents the full-order dependence among failure sources and tests.

Each component state is modeled as a two-state nonhomogenous Markov chain. For each component state, e.g., for component s_i at epoch k, $\Pi = (Pa_i(k), Pv_i(k))$ denotes the set of fault appearance probability $Pa_i(k)$ and fault disappearance probability $Pv_i(k)$ defined as $Pa_i(k) = \Pr(x_i(k) = 1 | x_i(k - 1) = 0)$ and $Pv_i(k) = \Pr(x_i(k) = 0 | x_i(k - 1) = 1)$, respectively. These probabilities are required to model the intermittent faults. Here, $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of n available binary tests, where the integrity of the system can be ascertained. We denote the set of passed tests T_p and failed tests T_f. At each observation epoch, $k \in \kappa$, test outcomes up to and including epoch k are available, i.e., we let $O^k = \{O(b) = (O_p(b), O_f(b))\}_{b=1}^{k}$, where O^k is the set of observed test outcomes at epoch k, with $O_p(b) (\subseteq O)$ and $O_f(b) (\subseteq O)$ as the corresponding outcomes of sets of passed and failed tests at epoch b, respectively. The tests are partially observed in the sense that outcomes of some tests may not be available, i.e., $(O_p(b) \cup O_f(b)) \subseteq O$. In addition, tests exhibit missed detections and false alarms. Here, we also make the noisy-OR ("causal independence") assumption [14].

Formally, we represent the DMFD problem as $DM = \{S, \kappa, T, O, D, P, \Pi\}$, where $S = \{s_1, \ldots, s_m\}$ is a finite set of m components (failure sources) associated with the system, $\kappa = \{0, 1, \ldots, k, \ldots, K\}$ is the set of discretized observation epochs, $T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of n available binary tests, O is a finite set of test outcomes up to and including epoch K, $D = [d_{ij}]$ is the D-matrix, $P = \{Pd, Pf\}$ is a set of probabilities of detection and false alarm, and $\Pi = (Pa_i(k), Pv_i(k))$ denotes the set of fault appearance probability $Pa_i(k)$ and fault disappearance probability $Pv_i(k)$. The DMFD problem can be formulated in the following ways, arranged from the general to simplified.

Problem 1: The probability of detection (Pd_{ij}) and false alarm probability (Pf_{ij}) are associated with each failed test and each failure source, i.e., $Pd_{ij} = \Pr(o_j(k) = 1 | x_i(k) = 1)$ and $Pf_{ij} = \Pr(o_j(k) = 1 | x_i(k) = 0)$ of a failure source s_i and test t_j. For notational convenience, when s_i does not affect the outcome of test t_j, we let the corresponding $Pd_{ij} = Pf_{ij} = 0$. This problem scenario frequently arises in medical fault diagnosis. For example, the Quick Medical Reference, Decision-Theoretic database used in the domain of internal medicine contains approximately 600 disease nodes (faults or failure sources) and 4000 symptoms (tests) [1], [2]. Each of the symptoms could have a probability pair (Pd_{ij}, Pf_{ij}) associated with the symptom and the disease node. Fig. 3 shows the bipartite graph, where the edges represent the probability pair (Pd_{ij}, Pf_{ij}). These probabilities can be obtained from the tripartite digraph (Fig. 2) using the total probability theorem as follows:

$$\begin{aligned}\Pr\left(o_j(k) | x_i(k)\right) &= \sum_{t_j \in \{0,1\}} \Pr\left(o_j(k), t_j(k) | x_i(k)\right) \\ &= \sum_{t_j \in \{0,1\}} \Pr\left(o_j(k) | t_j(k)\right) \Pr\left(t_j(k) | x_i(k)\right). \qquad (1)\end{aligned}$$

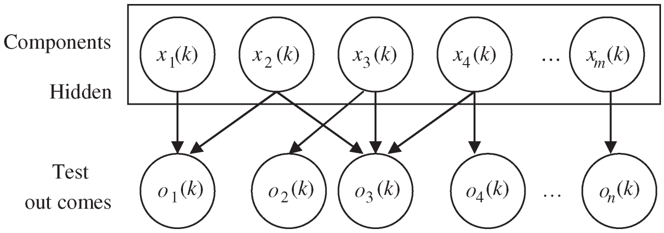

Fig. 3. Bipartite graph for the DMFD problem.

Problem 2: In situations where the probability of detection (Pd_{ij}) is associated with each failure source–test pair, but the false alarm probability is specified only for the normal system state, i.e., $Pf_j = P(o_j(k) = 1 | x_1(k) = 0, \ldots, x_m(k) = 0)$, we obtain a slightly complicated variation of Problem formulation 1 (in terms of computational complexity, but not in terms of parameterization). This type of scenario arises when we design class-specific classifiers that distinguish between normal system operation and failure source, s_i only, or when the false alarms are defined on an overall system basis. Here, the probability pair (Pd_{ij}, Pf_j) is associated with test outcomes to model imperfect test outcomes [4]. This model is also called the Z model in [8]. Similar to problem 1, the probability pair (Pd_{ij}, Pf_j) is shown as edges between the hidden component states and test outcomes in Fig. 3, and they can be obtained from the tripartite digraph (Fig. 2) using the total probability theorem on the nodes of the test layer.

Problem 3: The detection probability (Pd_j) and false alarm probability (Pf_j) are associated with each test t_j only. The probability pair (Pd_j, Pf_j) is shown as the edges between the tests and test outcomes in the tripartite digraph (Fig. 3). This formulation is quite useful in classifier fusion using ECCs. In the ECC matrix, each column corresponds to a binary classifier with the associated (Pd_j, Pf_j) pair. In this case, the fault–test relationships are deterministic, but the test outcomes are unreliable and depend on the concomitant test only. This type of formulation is also considered in [10].

This formulation provides a nice vehicle for the dynamic fusion of classifiers, where each column of the ECC matrix is a classifier, and their associated probability pairs $(Pd_j, Pf_j)$are uncertainties associated with classifier outcomes. When the learned parameters and the ECC matrix are fed as an input to the DMFD algorithm, it performs a dynamic fusion of classifier outputs over time. Note that the sampling interval of the dynamic fusion algorithm can be different from the sampling interval of the raw sensor data.

Problem 4: This is the deterministic case when tests are perfect, i.e., $Pd_{ij} = 1$ and $Pf_{ij} = 0$ [5]. This formulation reduces the tripartite digraph in Fig. 2 to a bipartite graph between the components and tests. This scenario is useful in situations where the tests are highly reliable (e.g., automated testing of electronic cards), and this leads to a novel dynamic set covering problem.

Next, we discuss the DMFD formulations in detail.

IV. DMFD Problem 1

In this problem, we assume that the detection and false alarm probabilities (Pd_{ij}, Pf_{ij}) are associated with

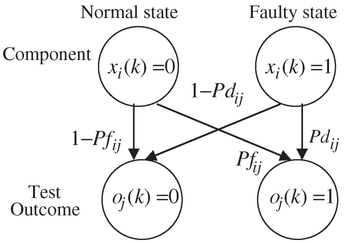

Fig. 4. Detection and false alarm probabilities for problem 1.

each failure source and each test. Fig. 4 shows these probabilities.

The DMFD problem is one of finding, at each decision epoch k, the most likely fault state candidates $\underline{x}(k) \in \{0,1\}^m$, i.e., the fault state evolution over time, $X^K = \{\underline{x}(1), \ldots, \underline{x}(K)\}$, that best explains the observed test outcome sequence O^K. We formulate this as one of finding the maximum *a posteriori* configuration

$$\widehat{X}^K = \arg\max_{X^K} \Pr\left(X^K | O^K, \underline{x}(0)\right). \tag{2}$$

Applying the Bayes rule in (2), the objective function is equivalent to

$$\widehat{X}^K = \arg\max_{X^K} \Pr\left(O^K | X^K, \underline{x}(0)\right) \Pr\left(X^K | \underline{x}(0)\right).$$

With passed and failed test outcomes being conditionally independent given the status of component states ("the noisy-OR assumption") and the Markov property of component state evolution, the problem is equivalent to

$$\widehat{X}^K = \arg\max_{X^K} \prod_{k=1}^{K} \left\{\Pr\left(O_p(k)|\underline{x}(k)\right) \Pr\left(O_f(k)|\underline{x}(k)\right) \times \Pr\left(\underline{x}(k)|\underline{x}(k-1)\right)\right\} \tag{3}$$

where $O_p(k) \subseteq O$ and $O_f(k) \subseteq O$ denote the sets of passed and failed test outcomes at epoch k, respectively. To simplify the problem, we maximize the ln (a monotonic function) of (3). We define a new function $f_k(\underline{x}(k), \underline{x}(k-1))$ as

$$f_k\left(\underline{x}(k), \underline{x}(k-1)\right) = \ln\left\{\Pr\left(O_p(k)|\underline{x}(k)\right) \Pr\left(O_f(k)|\underline{x}(k)\right) \times \Pr\left(\underline{x}(k)|\underline{x}(k-1)\right)\right\}. \tag{4}$$

Given the component state status $\underline{x}(k)$, the test outcomes are independent. Consequently

$$\Pr\left(O_p(k)|\underline{x}(k)\right) = \prod_{o_j(k)\in O_p(k)} \Pr\left(o_j(k)=0|\underline{x}(k)\right) \tag{5}$$

$$\Pr\left(O_f(k)|\underline{x}(k)\right) = \prod_{o_j(k)\in O_f(k)} \Pr\left(o_j(k)=1|\underline{x}(k)\right). \tag{6}$$

For test t_j to pass at epoch k, it shall pass on all its associated component states so that

$$\Pr\left(o_j(k)=0|\underline{x}(k)\right) = \prod_{i=1}^{m} \Pr\left(o_j(k)=0|x_i(k)\right) \tag{7}$$

where

$$\begin{aligned}\Pr\left(o_j(k)=0|x_i(k)\right) &= \begin{cases} 1-Pf_{ij}, & x_i(k)=0 \\ 1-Pd_{ij}, & x_i(k)=1 \end{cases} \\ &= (1-Pd_{ij})^{x_i(k)}(1-Pf_{ij})^{1-x_i(k)} \\ &\qquad x_i(k)\in\{0,1\}.\end{aligned} \tag{8}$$

Evidently

$$\Pr\left(o_j(k)=1|\underline{x}(k)\right) = 1 - \Pr\left(o_j(k)=0|\underline{x}(k)\right). \tag{9}$$

In the same vein, the assumption of the independent evolution of component states leads to

$$\Pr\left(\underline{x}(k)|\underline{x}(k-1)\right) = \prod_{i=1}^{m} \Pr\left(x_i(k)|x_i(k-1)\right) \tag{10}$$

where

$$\Pr\left(x_i(k)|x_i(k-1)\right) = \begin{cases} 1-Pa_i(k) & x_i(k-1)=0;\ x_i(k)=0 \\ Pa_i(k) & x_i(k-1)=0;\ x_i(k)=1 \\ Pv_i(k) & x_i(k-1)=1;\ x_i(k)=0 \\ 1-Pv_i(k) & x_i(k-1)=1;\ x_i(k)=1. \end{cases}$$

Equivalently

$$\begin{aligned}&\Pr\left(x_i(k)|x_i(k-1)\right) \\ &\quad = \left(1-Pa_i(k)\right)^{(1-x_i(k-1))(1-x_i(k))} \\ &\qquad \times Pa_i(k)^{(1-x_i(k-1))x_i(k)} \cdot Pv_i(k)^{x_i(k-1)(1-x_i(k))} \\ &\qquad \times \left(1-Pv_i(k)\right)^{x_i(k-1)x_i(k)}, \qquad x_i(k-1), x_i(k)\in\{0,1\}.\end{aligned} \tag{11}$$

Therefore, the problem that is equivalent to (3) is as follows:

$$\widehat{X}^K = \arg\max_{X^K} \sum_{k=1}^{K} f_k\left(\underline{x}(k), \underline{x}(k-1)\right) \tag{12}$$

where

$$\begin{aligned}&f_k\left(\underline{x}(k), \underline{x}(k-1)\right) \\ &\quad = \sum_{o_j\in O_p(k)} \sum_{i=1}^{m} \left[x_i(k)\ln(1-Pd_{ij}) + (1-x_i(k))\ln(1-Pf_{ij})\right] \\ &\qquad + \sum_{o_j\in O_f(k)} \ln\left[1 - \prod_{i=1}^{m} (1-Pd_{ij})^{x_i(k)}(1-Pf_{ij})^{(1-x_i(k))}\right] \\ &\qquad + \sum_{i=1}^{m} \left\{(1-x_i(k-1))\,(1-x_i(k))\ln\left(1-Pa_i(k)\right)\right. \\ &\qquad\qquad + (1-x_i(k-1))\,x_i(k)\ln\left(Pa_i(k)\right) \\ &\qquad\qquad + x_i(k-1)\,(1-x_i(k))\ln\left(Pv_i(k)\right) \\ &\qquad\qquad \left. + x_i(k-1)x_i(k)\ln\left(1-Pv_i(k)\right)\right\}, \\ &\qquad\quad \underline{x}(k), \underline{x}(k-1)\in\{0,1\}^m.\end{aligned} \tag{13}$$

The primal DMFD problem posed in (12) and (13) is NP-hard. Indeed, even the single-epoch problem, i.e., $\widehat{\underline{x}}(k) = \arg\max_{\underline{x}(k)} f_k(\underline{x}(k), \widehat{\underline{x}}(k-1))$, is NP-hard [4], which for all practical purposes, means that it cannot be solved to optimality within a polynomially bounded computation time.

A. Primal–Dual Optimization Framework

The NP-hard nature of the primal DMFD problem motivates us to decompose it into a primal–dual problem using a Lagrangian relaxation approach [13]. By defining new variables and constraints, the DMFD problem reduces to a combinatorial optimization problem with a set of equality constraints. The constraints are relaxed via Lagrange multipliers. The relaxation procedure generates an upper bound for the objective function. The procedure of minimizing the upper bound via a subgradient optimization produces a sequence of dual and concomitant primal feasible solutions to the DMFD problem. If the objective function value for the best feasible solution and the upper bound are the same, the feasible solution is the optimal solution. Otherwise, the difference between the upper bound and the feasible solution, termed the approximate duality gap, provides a measure of suboptimality of the DMFD solution; this is a key advantage of our approach. Another advantage of the primal–dual method is that, although the primal DMFD problem is not concave, the dual DMFD problem is a piecewise convex function, which can be optimized via the subgradient method. In order to write the primal DMFD problem, we define new variables $Y^K = \{y(1), y(2), \ldots, y(K)\}$ and $y(k) = \{y_j(k), \forall o_j \in O_f(k)\}$ such that

$$\ln y_j(k) = \sum_{i=1}^{m} c_{ij} x_i(k) + \eta_j \qquad \forall\, o_j \in O_f(k)$$

where

$$c_{ij} = \ln\left(\frac{1 - Pd_{ij}}{1 - Pf_{ij}}\right) \qquad \eta_j = \sum_{i=1}^{m} \ln(1 - Pf_{ij}). \tag{14}$$

After simple algebraic manipulations of (13) and using (12) and (14), the primal problem can be written as

$$\max_{X^K, Y^K} J(X, Y) = \max_{X^K, Y^K} \sum_{k=1}^{K} f_k\left(\underline{x}(k), \underline{x}(k-1), \underline{y}(k)\right) \tag{15}$$

where the component state sequence is $X^K = \{x(1), x(2), \ldots, x(K)\}$. Here, the primal objective function for an individual component state, i.e., $f_k(\underline{x}(k), \underline{x}(k-1), \underline{y}(k))$, is defined as

$$\begin{aligned} f_k &\left(\underline{x}(k), \underline{x}(k-1), \underline{y}(k)\right) \\ &= \sum_{o_j \in O_p(k)} \sum_{i=1}^{m} c_{ij} x_i(k) + \sum_{i=1}^{m} \mu_i(k) x_i(k) \\ &\quad + \sum_{o_j \in O_f(k)} \ln(1 - y_j(k)) + \sum_{i=1}^{m} \sigma_i(k) x_i(k-1) \\ &\quad + \gamma(k) + g(k) + \sum_{i=1}^{m} h_i(k) x_i(k) x_i(k-1) \end{aligned} \tag{16}$$

where

$$\begin{aligned} \gamma(k) &= \sum_{o_j \in O_p(k)} \eta_j \\ \mu_i(k) &= \ln\left(\frac{Pa_i(k)}{1 - Pa_i(k)}\right) \\ \sigma_i(k) &= \ln\left(\frac{Pv_i(k)}{1 - Pa_i(k)}\right) \\ h_i(k) &= \ln\left(\frac{(1 - Pa_i(k))\,(1 - Pv_i(k))}{Pa_i(k) Pv_i(k)}\right) \\ g(k) &= \sum_{i=1}^{m} \ln(1 - Pa_i(k)). \end{aligned} \tag{17}$$

Note that the multiple HMMs are coupled here because their states are observed only via a set of test outcomes. In (16), the terms involving $y_j(k)$ and $h_i(k)$ show the coupling effects.

By appending constraints (14)–(16) via Lagrange multipliers $\{\lambda_j(k)\}_{o_j \in O_f(k)}$, the Lagrangian function $L(X, Y, \Lambda)$ can be written as

$$\begin{aligned} L(X, Y, \Lambda) &= \sum_{k=1}^{K} f_k\left(\underline{x}(k), \underline{x}(k-1), \underline{y}(k)\right) \\ &\quad + \sum_{\forall o_j \in O_f(k)} \lambda_j(k) \left(\ln y_j(k) - \sum_{i=1}^{m} c_{ij} x_i(k) - \eta_j\right) \end{aligned} \tag{18}$$

where $\Lambda = \{\lambda_j(k) \geq 0, k \in (1, K), o_j \in O_f(k)\}$ is the set of Lagrange multipliers. In (18), Lagrange multipliers $\{\lambda_j(k)\}$ are nonnegative despite equality constraints (14), because the $y_j(k)$ needs to be nonnegative. Using the Lagrange multiplier theorem, we optimize the Lagrangian function in (17) w.r.t. $y_j(k)$ to obtain optimal $y_j^*(k)$ as

$$y_j^*(k) = \frac{\lambda_j(k)}{1 + \lambda_j(k)}. \tag{19}$$

The dual of the primal DMFD problem as posed in (15)–(17) can be written as

$$\begin{aligned} &\min_{\Lambda} \quad Q(\Lambda) \\ &\text{subject to } \Lambda = \{\lambda_j(k) \geq 0, k \in (1, K), o_j \in O_f(k)\} \end{aligned} \tag{20}$$

where the dual function $Q(\Lambda)$ is defined by

$$Q(\Lambda) = \max_{X^K, Y^K} L(X, Y, \Lambda). \tag{21}$$

The optimization problem described in (20) and (21) is the dual of the primal problem in (15)–(17) because it dualizes the maximization problem into a minimization problem.

Substituting (19) into (20) and simplifying further by rearranging and combining the terms, we obtain the dual function as

$$Q(X, \Lambda) = \max_{X^K} \sum_{i=1}^{m} Q_i(\Lambda). \tag{22}$$

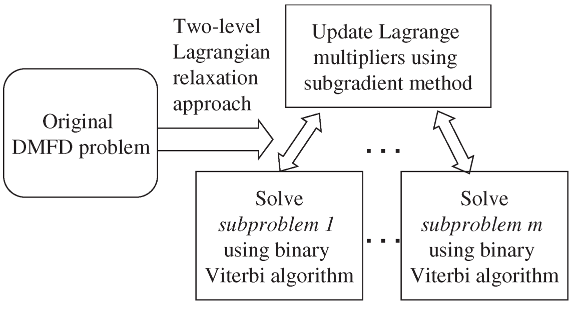

Fig. 5. Decomposition of the original DMFD problem.

Here

$$Q_i(\Lambda) = \sum_{k=1}^{K} \xi_i \left(x_i(k), x_i(k-1), \lambda_j(k)\right) + \frac{1}{m} w_k(\Lambda) \tag{23}$$

$$\begin{aligned}\xi_i &\left(x_i(k), x_i(k-1), \lambda_j(k)\right)\\ &= \left(\sum_{o_j \in O_p(k)} c_{ij} + \mu_i(k) - \sum_{o_j \in O_f(k)} c_{ij}\lambda_j(k) \right) x_i(k)\\ &\quad + \sigma_i(k) x_i(k-1) + h_i(k) x_i(k) x_i(k-1) \end{aligned} \tag{24}$$

$$\begin{aligned} w_k(\Lambda) = \gamma(k) + g(k) + &\sum_{\forall o_j \in O_f(k)} \lambda_j(k) \ln \lambda_j(k) - \lambda_j(k)\eta_j\\ &- \sum_{\forall o_j \in O_f(k)} (1+\lambda_j(k)) \ln (1+\lambda_j(k)) \end{aligned} \tag{25}$$

represents the dual function for the ith component. The main benefit of (22) is that, now, the original problem is separable. As shown in Fig. 5, we employed the Lagrangian relaxation method to decompose the original DMFD problem into m separable subproblems, one for each component state sequence $\underline{x}_i$, where $\underline{x}_i = (x_i(1), x_i(2)), \ldots, x_i(K)), x_i(k) \in \{0, 1\}$ and $i \in \{1, m\}$. This scheme can be viewed as a two-level coordinated solution framework for the DMFD problem. At the top (coordination) level, we update Lagrange multipliers $\Lambda = \{\lambda_j(k), k \in (1, K), o_j \in O_f(k)\}$ using the subgradient method based on the decoupled solutions of the individual subproblems. This level facilitates coordination among each of the subproblems and can thus reside in a diagnostic control unit. At the bottom level, we use the dynamic programming technique (the Viterbi algorithm) to solve each of the subproblems with a computational complexity $\mathbf{O}(K)$, i.e., we optimize the ξ_i function in (24) to obtain the optimal state sequence $\underline{x}_i^*$ for each component state, given a fixed set of Lagrange multipliers $\Lambda = \{\lambda_j(k), k \in (1, K), o_j \in O_f(k)\}$. The Viterbi algorithm is a dynamic programming technique to find the most likely fault sequence [15]. It finds a recursive optimal solution to the problem of estimating the state sequence of a finite state Markov chain observed in memoryless noise. The key feature of the Viterbi algorithm is that the objective function can be written as a sum of *merit* functions depending on one state and its preceding one. We obtain the optimal state sequence for each component state, i.e., $X^* = \{\underline{x}_1^*, \underline{x}_2^*, \ldots, \underline{x}_m^*\}$, using a binary Viterbi algorithm. The key steps of the Viterbi algorithm are described in Appendix I.

B. Approximate and Exact Duality Gap

After evaluating the optimum state sequence X^* for fixed Λ, the problem reduces to one of minimizing the dual function value $Q^l(\Lambda) = Q(X, Y, \Lambda)$ at iteration l, which is computed using (22)–(25). Q^* denotes the optimal dual function value, i.e., $Q^* = Q(\Lambda^*) = \min_\Lambda Q(\Lambda)$, where the dual problem is given by (22)–(25). The optimal primal solution is denoted by $J^* = J(X^*, Y^*) = \max_{X^K, Y^K} J(X, Y)$, where the primal problem is given by (15)–(17).

The difference between the optimal dual and the primal function values, i.e., $(Q^* - J^*)$, is termed the *exact duality gap*. Since the DMFD problem is NP-hard, it is difficult to obtain the global optimal solution J^*. However, we can obtain several feasible solutions from the dual solution and select the best feasible solution from the set. If $J_f = J(\Lambda^*, X^f, Y^f)$ is the best feasible value, then we have

$$J_f \le J^* \le Q^* \le Q^l. \tag{26}$$

Using this method, we can obtain an *approximate duality gap* $Q^* - J_f = (Q^* - J^*) + (J^* - J_f) \ge 0$, which provides an overestimate of the error between the global optimal solution and the best feasible solution. To summarize, we update feasible solutions, i.e., X^f, Y^f, and the lower bound Q_{lb} as follows: If $J(X^*(\Lambda^l), Y(X^*(\Lambda^l)) \ge Q_{\mathrm{lb}}$, then $X^f = X^*(\Lambda^l)$, $Y^f = Y(X^*(\Lambda^l))$ and

$$Q_{\mathrm{lb}} = J_f = J\left(X^*(\Lambda^l), Y(X^*(\Lambda^l)\right). \tag{27}$$

The upper bound Q_{ub} is obtained using the current dual value Q^l as follows:

$$Q_{\mathrm{ub}} = Q_{\min} = \min(Q_{\min}, Q^l). \tag{28}$$

Since the dual function $Q^l(\Lambda)$ is a piecewise differentiable function of Lagrange multipliers Λ, this problem cannot be solved using differentiable optimization algorithms. We use a subgradient algorithm to compute a sequence of upper bounds for $Q^l(\Lambda)$ [13]. The details of a subgradient method are described in Appendix II.

Fig. 6 shows the flowchart of our algorithm. There are five major steps. In step 1, we initialize the Lagrange multipliers and the input fault universe, i.e., fault and test information along with the associated probabilities (Pd_{ij}, Pf_{ij}, Pa_i, and Pv_i). We also input test outcomes for the K epochs. In step 2, we run m binary Viterbi algorithms to obtain the optimal state sequences corresponding to the m faults. In step 3, we update the feasible solutions, i.e., X^f, Y^f, and the lower and upper bounds, i.e., Q_{lb} and Q_{ub} using (24)–(26). Next, the Lagrange multipliers are updated using the subgradient method, which is described in Appendix II. If stopping criteria, defined in Appendix II, are met, then the algorithm outputs the most likely component state sequence for the m components.

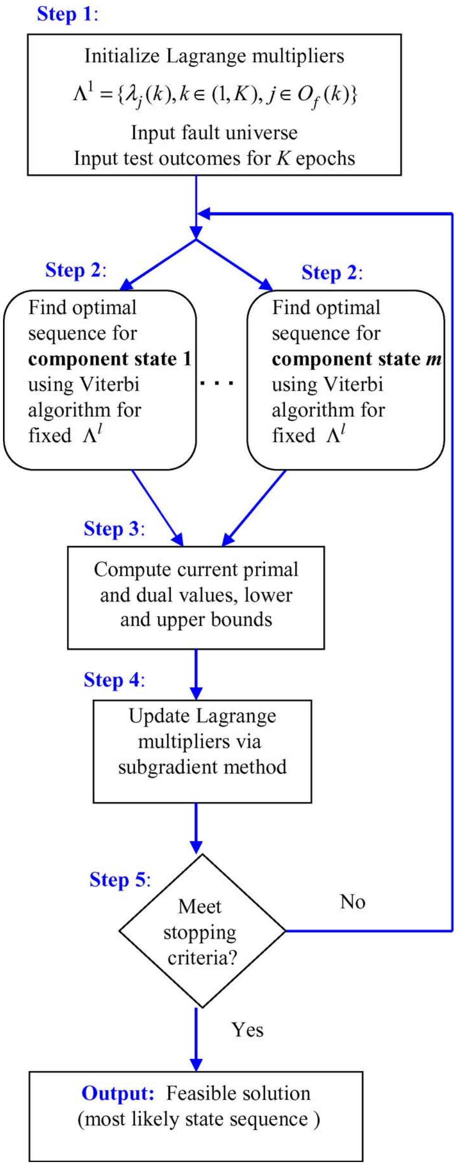
Step 1:
Initialize Lagrange multipliers
$\Lambda^1 = \{\lambda_j(k), k \in (1,K), j \in O_f(k)\}$
Input fault universe
Input test outcomes for K epochs
Step 2:
Find optimal sequence for **component state 1** using Viterbi algorithm for fixed Λ^l
. . .
Step 2:
Find optimal sequence for **component state *m*** using Viterbi algorithm for fixed Λ^l
Step 3:
Compute current primal and dual values, lower and upper bounds
Step 4:
Update Lagrange multipliers via subgradient method
Step 5:
Meet stopping criteria?
No
Yes
Output: Feasible solution (most likely state sequence)

Fig. 6. Flowchart of the algorithm.

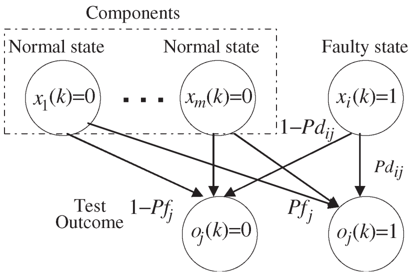

Fig. 7. Detection and false alarm probabilities for problem 2.

V. DMFD Problem 2

In this formulation, we define Pd_{ij} as $Pd_{ij} = \Pr(o_j(k) = 1|x_i(k) = 1)$ and $Pf_j = \Pr(o_j(k) = 1|x_1(k) = 0,\ x_2(k) = 0, \ldots, x_m(k) = 0)$. This scenario is shown in Fig. 7.

$$\Pr(o_j(k) = 0|\underline{x}(k)) = (1 - Pf_j)^{(1-x_1(k)),\ldots,(1-x_m(k))} \prod_{i=1}^{m} (1 - Pd_{ij})^{x_i(k)} \tag{29}$$

Using a new variable $z(k) = \prod_{i=1}^{m}(1 - x_i(k))$ and

$$\Pr(o_j(k) = 0|\underline{x}(k)) = (1 - Pf_j)^{z(k)} \prod_{i=1}^{m} (1 - Pd_{ij})^{x_i(k)}$$

taking log

$$\ln z(k) = \sum_{i=1}^{m} \ln(1 - x_i(k)) \tag{30}$$

$$\ln(\Pr(o_j(k) = 0|\underline{x}(k))) = z(k)\ln(1 - Pf_j) + \sum_{i=1}^{m} x_i(k)\ln(1 - Pd_{ij}). \tag{31}$$

Following steps similar to those in problem 1, we have

$$\begin{aligned}
&\ln(\Pr(O_p(k)|\underline{x}(k))) \\
&\quad= \sum_{o_j(k)\in O_p(k)} \ln(\Pr(o_j(k) = 0|\underline{x}(k))) \\
&\quad= \sum_{o_j(k)\in O_p(k)} z(k)\ln(1 - Pf_j) \\
&\qquad+ \sum_{o_j(k)\in O_p(k)} \sum_{i=1}^{m} x_i(k)\ln(1 - Pd_{ij}) \\
&\quad= z(k)\eta_j(k) + \sum_{i=1}^{m} \sum_{o_j(k)\in O_p(k)} x_i(k)\ln(1 - Pd_{ij})
\end{aligned} \tag{32}$$

where $\eta_j(k) = \sum_{o_j(k)\in O_p(k)} \ln(1 - Pf_j)$ and $z(k)$ is defined in (30). For failed tests

$$\begin{aligned}
\ln(\Pr(O_f(k)|\underline{x}(k))) &= \sum_{o_j(k)\in O_f(k)} \ln(\Pr(o_j(k) = 1|\underline{x}(k))) \\
&= \sum_{o_j(k)\in O_f(k)} \ln(1 - y_j(k))
\end{aligned}$$

where $y_j(k) = \Pr(o_j(k) = 0|\underline{x}(k))$, and using (31)

$$\ln(y_j(k)) = z(k)\ln(1 - Pf_j) + \sum_{i=1}^{m} x_i(k)\ln(1 - Pd_{ij}). \tag{33}$$

Here, the DMFD problem is equivalent to

$$\widehat{X}^K = \arg\max_{X^K} \sum_{k=1}^{K} f_k\left(\underline{x}(k), \underline{x}(k-1), \underline{y}(k), z(k)\right) \tag{34}$$

where the primal objective function for an individual component state, i.e., $f_k(\underline{x}(k), \underline{x}(k-1), \underline{y}(k), z(k))$ is defined as

$$\begin{aligned}
&f_k\left(\underline{x}(k), \underline{x}(k-1), \underline{y}(k), z(k)\right) \\
&\quad= z(k)\eta_j(k) + \sum_{i=1}^{m} \sum_{o_j(k)\in O_p(k)} x_i(k)\ln(1 - Pd_{ij}) \\
&\qquad+ \sum_{o_j(k)\in O_f(k)} \ln(1 - y_j(k)) + \sum_{i=1}^{m} \tau_i(k)x_i(k) \\
&\qquad+ \sum_{i=1}^{m} \sigma_i(k)x_i(k-1) + \sum_{i=1}^{m} h_i(k)x_i(k)x_i(k-1) + g(k)
\end{aligned} \tag{35}$$

where

$$\begin{aligned}
\ln z(k) &= \sum_{i=1}^{m} \ln\left(1 - x_i(k)\right) \\
\eta_j(k) &= \sum_{o_j(k)\in O_p(k)} \ln(1 - Pf_j) \\
\ln\left(y_j(k)\right) &= z(k)\ln(1 - Pf_j) + \sum_{i=1}^{m} x_i(k)\ln(1 - Pd_{ij}) \\
\tau_i(k) &= \ln\left(\frac{Pa_i(k)}{1 - Pa_i(k)}\right) \\
\sigma_i(k) &= \ln\left(\frac{Pv_i(k)}{1 - Pa_i(k)}\right) \\
h_i(k) &= \ln\left(\frac{\left(1 - Pa_i(k)\right)\left(1 - Pv_i(k)\right)}{Pa_i(k)Pv_i(k)}\right) \\
g(k) &= \sum_{i=1}^{m} \ln\left(1 - Pa_i(k)\right). \qquad (36)
\end{aligned}$$

Appending constraints (30) and (33) via Lagrange multipliers $\mu(k)$, $\{\lambda_j(k)\}_{j\in O_f(k)}$, the Lagrangian function $L(X, Y, z, \Lambda)$ can be written as

$$\begin{aligned}
&L(X, Y, z, \Lambda) \\
&= \sum_{k=1}^{K} f_k\left(\underline{x}(k), \underline{x}(k-1), \underline{y}(k), z(k)\right) \\
&\quad + \mu(k)\left(\ln z(k) - \sum_{i=1}^{m} \ln\left(1 - x_i(k)\right)\right) \\
&\quad + \sum_{\forall o_j\in O_f(k)} \lambda_j(k)\left(\ln y_j(k) - z(k)\ln(1 - Pf_j)\right) \\
&\quad - \sum_{\forall o_j\in O_f(k)} \sum_{i=1}^{m} x_i(k)\lambda_j(k)\ln(1 - Pd_{ij}) \qquad (37)
\end{aligned}$$

where $\Lambda = \{\mu(k), \lambda_j(k) \geq 0, k \in (1, K), o_j \in O_f(k)\}$ is the set of Lagrange multipliers. Using the Lagrange multiplier theorem, we optimize the Lagrangian function in (34) w.r.t. $y_j(k)$ to obtain optimal $y_j^*(k)$ as

$$y_j^*(k) = \frac{\lambda_j(k)}{1 + \lambda_j(k)} \qquad (38)$$

and optimizing w.r.t. $z(k)$, we obtain optimal $z^*(k)$ as

$$z^*(k) = \frac{\mu(k)}{-\eta_j(k) + \sum\limits_{\forall o_j\in O_f(k)} \lambda_j(k)\ln(1 - Pf_j)}. \qquad (39)$$

The dual function $Q(\Lambda)$ of problem 4 is defined by

$$Q(\Lambda) = \max_{X^K, Y^K, z} L(X, Y, z, \Lambda). \qquad (40)$$

Substituting (38) and (39) into (37) and simplifying further by rearranging and combining the terms, we obtain the dual function as

$$Q(\Lambda) = \max_{X^K} \sum_{i=1}^{m} Q_i(\Lambda) \qquad (41)$$

where

$$\begin{aligned}
Q_i(\Lambda) &= \sum_{k=1}^{K} \xi_i\left(x_i(k), x_i(k-1), \lambda_j(k), \mu(k)\right) \\
&\quad + \frac{1}{m} w_k\left(\lambda_j(k), \mu(k)\right) \qquad (42) \\
\xi_i&\left(x_i(k), x_i(k-1), \lambda_j(k), \mu(k)\right) \\
&= \sum_{o_j(k)\in O_p(k)} x_i(k)\ln(1 - Pd_{ij}) + x_i(k)\tau_i(k) \\
&\quad - \sum_{o_j(k)\in O_f(k)} \lambda_j(k)x_i(k) \\
&\quad \times \ln(1 - Pd_{ij}) + \sigma_i(k)x_i(k-1) \\
&\quad + h_i(k)x_i(k)x_i\left(k-1\right) - \mu(k)\ln(1 - x_i(k)) \qquad (43) \\
w_k&\left(\lambda_j(k), \mu(k)\right) \\
&= \mu(k)\left(\frac{\eta_j(k) + \ln\left(\mu(k)\right)}{-\eta_j(k) + \sum\limits_{\forall o_j\in O_f(k)} \lambda_j(k)\ln(1 - Pf_j)}\right) + g(k) \\
&\quad + \sum_{\forall o_j\in O_f(k)} \left[\lambda_j(k)\ln\lambda_j(k) - \left(1 + \lambda_j(k)\right)\right. \\
&\qquad \left. \times \ln\left(1 + \lambda_j(k)\right)\right] - \mu(k) \\
&\quad \times \left(\sum_{\forall o_j\in O_f(k)} \frac{\lambda_j(k)\ln(1 - Pf_j)}{-\eta_j(k) + \sum\limits_{\forall o_j\in O_f(k)} \lambda_j(k)\ln(1 - Pf_j)}\right). \qquad (44)
\end{aligned}$$

The dual problem posed in (40)–(44) is separable, and it can be solved by following a procedure similar to that used for solving problem 1. The only difference is that we also need to update the Lagrange multiplier $\mu(k)$ using a subgradient method.

VI. DMFD Problem 3

In this formulation, we consider the case where the probabilities of detection and false alarm (Pd_j, Pf_j) are associated only with each test t_j (see Fig. 8). Formally, $Pd_j = \Pr(o_j(k) = 1|t_j(k) = 1)$ and $Pf_j = \Pr(o_j(k) = 1|t_j(k) = 0)$. We can convert these probabilities into a special case of problem 1 by computing (Pd_{ij}, Pf_{ij}) using the total probability theorem

$$\begin{aligned}
\Pr(o_j(k)|x_i(k)) &= \sum_{t_j(k)\in\{0,1\}} \Pr(o_j(k), t_j(k)|x_i(k)) \\
&= \sum_{t_j(k)\in\{0,1\}} \Pr(o_j(k)|t_j(k))\Pr(t_j(k)|x_i(k)), \\
Pd_{ij} &= (d_{ij})Pd_j + (1 - d_{ij})Pf_j \qquad (45) \\
Pf_{ij} &= (d_{ij})Pf_j + (1 - d_{ij})Pd_j. \qquad (46)
\end{aligned}$$

Here, $D = [d_{ij}]$ is the D-matrix. The solution of Problem 3 can be obtained by substituting Pd_{ij} and Pf_{ij} in (45) and (46) in the solution of problem 1.

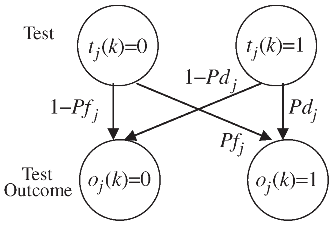

Fig. 8. Detection and false alarm probabilities for problem 3.

VII. DMFD Problem 4

Next, we consider the case when the system consists of reliable tests and the fault–test relationships are deterministic, i.e., $Pd_{ij} = 1$ and $Pf_{ij} = 0$ for $i = 1, \dots, m$ and $j = 1, \dots, n$, or equivalently, the D-matrix completely characterizes the fault–test relationships [5]. This formulation can be represented as a bipartite graph between the components and tests. In this case, if some tests have passed, then we can infer that all the failure sources covered by these tests are good components.

Thus, we need to infer failed components from those covered by the failed tests only, i.e., by excluding those components covered by the passed tests. Consequently, the size of the DMFD problem can be reduced by removing all failure sources $\{s_i | Pf_{ij} = 0,\ Pd_{ij} = 1, \text{ and } t_j(k) \in T_p(k)\}$. For each failed test $t_j(k) \in T_f(k)$, the optimal solution contains at least one component state $x_i(k) = 1$ that satisfies $d_{ij} = 1$. Thus, there must be one or more failure sources that cover the failed tests. Let us consider a matrix A, which has each row representing the list of failure sources covered by a failed test. After excluding the failure sources covered by the passed tests, the resulting matrix A is a binary matrix such that $a_{ij} = d_{ji}$. After substituting $Pd_{ij} = 1$ and $Pf_{ij} = 0$ in (13), the reliable test scenario with a binary D-matrix simplifies to a dynamic set covering problem with the following objective function term at epoch k:

$$f_k\left(\underline{x}(k), \underline{x}(k-1)\right) = \sum_{i=1}^{m} \{\mu_i(k)x_i(k) + \sigma_i(k)x_ik-1) + h_i(k)x_i(k)x_i(k-1)\} + g(k) \quad (47)$$

subject to the following constraints.

$A(k)\underline{x}(k) \geq \underline{e}$ for $t_j(k) \in T_f(k)$ where $\underline{e}$ is a vector of one's. By appending constraints to (47) via Lagrange multipliers $\Lambda = \{\lambda_j(k) \leq 0, k \in (1, K), t_j \in T_f(k)\}$, the Lagrangian function $L(X, \Lambda)$ can be written as

$$L(X, \Lambda) = \sum_{k=1}^{K} f_k\left(\underline{x}(k), \underline{x}(k-1)\right) + \sum_{\forall t_j \in T_f(k)} \lambda_j(k)\left(1 - \sum_{i=1}^{m} a_{ji}(k)x_i(k)\right). \quad (48)$$

After rearranging the terms, the Lagrangian function of the original problem is shown as a sum of the Lagrangian functions of each subproblem as follows:

$$L(X, \Lambda) = \sum_{i=1}^{m} L_i(x_i(k), \Lambda) \quad (49)$$

where

$$L_i\left(x_i(k), \Lambda\right) = \sum_{k=1}^{K}\left[\mu_i(k)x_i(k) - \sum_{t_j \in T_f(k)} a_{ji}(k)\lambda_j(k)x_i(k) + \sigma_i(k)x_i(k-1) + h_i(k)x_i(k)x_i(k-1) + \frac{1}{m}\left(g(k) + \sum_{\forall t_j \in T_f(k)} \lambda_j(k)\right)\right]. \quad (50)$$

The dual function $Q(\Lambda)$ is defined by

$$Q(\Lambda) = \max_{X^K} L(X, \Lambda). \quad (51)$$

Simplifying further by rearranging and combining the terms, we obtain the dual function as

$$Q(\Lambda) = \max_{X^K} \sum_{i=1}^{m} Q_i(\Lambda) \quad (52)$$

where

$$Q_i(\Lambda) = \sum_{k=1}^{K} \xi_i\left(x_i(k), x_i(k-1), \lambda_j(k)\right) + \frac{1}{m} w_k(\Lambda) \quad (53)$$

$$\xi_i\left(x_i(k), x_i(k-1), \lambda_j(k)\right) = \mu_i(k)x_i(k) + \sigma_i(k)x_i(k-1) + h_i(k)x_i(k)x_i(k-1) - \sum_{t_j \in T_f(k)} \lambda_j(k)a_{ji}(k)x_i(k) \quad (54)$$

$$w_k(\Lambda) = g(k) + \sum_{\forall t_j \in T_f(k)} \lambda_j(k). \quad (55)$$

The dual problem defined in (52)–(55) is separable. The Viterbi algorithm is used to solve each subproblem corresponding to each component state sequence $\{x_i(k)\}_{k=1}^{K}$. This algorithm can be viewed as a dynamic set covering problem, which is NP-hard. Thus, the dynamic set covering problem is solved by combining the Viterbi algorithm and Lagrangian relaxation. This generalizes Beasley's Lagrangian relaxation algorithm for the static set covering problem [5], [16] to dynamic settings. We will explore the applications of this algorithm in [18].

VIII. Sliding Window DMFD Method

During the online monitoring of a system, the observations and potential fault sequences are usually very long. Hence, in order to reduce the amount of computation and storage, the DMFD problem is solved using a sliding window method. The window size W is selected based on the performance criteria, such as low classification error and low false isolation rate (FI). One of the key advantages of the sliding window method is that Lagrange multipliers are available $W - 1$ samples ahead, which improves the speed of dual optimization. The sliding window method involves the following steps.

Step 1) Solve the DMFD problem for the window size $W (W \leq K)$. Make a decision at epoch $k = 1$.

TABLE I
SMALL-SCALE SCENARIO FOR SIMULATIONS

<table>
<tr><td colspan="6">Fault model</td></tr>
<tr><td></td><td colspan="3">$(s_i, t_j, Pd_{ij}, Pf_{ij})$</td><td>$P_a$</td><td>$P_v$</td></tr>
<tr><td>s_1</td><td>(1,3,0.80,0.01)</td><td>(1,13,0.75,0)</td><td>(1,19,0.74,0)</td><td>0.0050</td><td>0.00026</td></tr>
<tr><td>s_2</td><td>(2,10,0.86,0.01)</td><td>(2,12,0.88,0)</td><td></td><td>0.0049</td><td>0.00027</td></tr>
<tr><td>s_3</td><td>(3,1,0.80,0.015)</td><td>(3,16,0.72,0.01)</td><td></td><td>0.0050</td><td>0.00026</td></tr>
<tr><td>.
.
.</td><td>.
.
.</td><td>.
.
.</td><td>.
.
.</td><td>.
.
.</td><td>.
.
.</td></tr>
<tr><td>s_{19}</td><td>(19,12,0.85,0.015)</td><td>(19,18,0.88,0.016)</td><td></td><td>0.0051</td><td>0.00033</td></tr>
<tr><td>s_{20}</td><td>(20,8,0.735,0.011)</td><td>(20,11,0.82,0)</td><td></td><td>0.0051</td><td>0.00028</td></tr>
<tr><td colspan="6">Test outcomes</td></tr>
<tr><td>k</td><td>No. of failed tests</td><td>Failed test outcomes $O_f(k)$</td><td>No. of passed tests</td><td colspan="2">Passed test outcomes $O_p(k)$</td></tr>
<tr><td>1</td><td>1</td><td>3</td><td>19</td><td colspan="2">1 2 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20</td></tr>
<tr><td>2</td><td>0</td><td>ϕ</td><td>20</td><td colspan="2">1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20</td></tr>
<tr><td>3</td><td>2</td><td>11 18</td><td>18</td><td colspan="2">1 2 3 4 5 6 7 8 9 10 12 13 14 15 16 17 19 20</td></tr>
<tr><td>.
.
.</td><td>.
.
.</td><td>.
.
.</td><td>.
.
.</td><td colspan="2">.
.
.</td></tr>
<tr><td>19</td><td>5</td><td>3 10 12 16 17</td><td>15</td><td colspan="2">1 2 4 5 6 7 8 9 11 13 14 15 18 19 20</td></tr>
<tr><td>20</td><td>6</td><td>3 4 10 12 16 17</td><td>14</td><td colspan="2">1 2 5 6 7 8 9 11 13 14 15 18 19 20</td></tr>
</table>

Step 2) Move the window by one time epoch, i.e., $k = 2$ to $k = W + 1$.
1) Initialize $(W - 1)$ Lagrange multipliers using previous window.
2) Initialize component states at $k = 2$ using the results of previous window.
3) Solve the online DMFD problem using the data from $k = 2$ to $k = W + 1$.
4) Make a decision at epoch $k = 2$.

Step 3 Continue sliding the window until $k = K - W + 1$. The selection of window size is a key issue, and it depends on the system and fault behavior, i.e., permanent or intermittent faults.

IX. SIMULATIONS AND RESULTS

We implemented and applied the solution of problem 1, the most general version of the DMFD problem formulation, to a small-scale system and few real-world models.

A. Small-Scale System

We randomly generated a small-scale system to illustrate the inputs and outputs of our algorithm. The model was constructed for a system with 20 components, 20 tests, and 20 observation epochs. Each component can have binary states, i.e., normal and faulty. The detection probabilities were set between 0.7 and 0.9, and the false alarm probabilities were set between 0 and 0.02, and the tests uniformly cover the component states. The fault appearance and disappearance probabilities were varied between 0.0049 and 0.0051, and 0.00025 and 0.00033, respectively. These probabilities were chosen such that the average number of faults was two over a span of 20 epochs. The true fault state set and, accordingly, the test outcomes of each epoch are generated using the aforementioned model parameters. The stopping criteria as defined in Appendix II were used for the subgradient method. We used the following metrics to evaluate the performance of our algorithm.

CI: Correct isolation rate (CI) is the percentage of true fault states which are detected by the algorithm at epoch k. Let $\hat{x}(k)$ be the fault state set at epoch k detected by the algorithm, and $r(k)$ is the true fault state set at epoch k. Then, CI and the average $\overline{CI}$ over all epochs are obtained as follows:

$$CI(k) = \frac{|\hat{x}(k) \cap r(k)|}{|r(k)|} \tag{56}$$

$$\overline{CI} = \frac{\sum_{k=1}^{K} CI(k)}{K}. \tag{57}$$

FI: FI is the percentage of fault states which are falsely detected by the algorithm as fault state at epoch k. FI and average $\overline{FI}$ are computed as

$$FI(k) = \frac{|\hat{x}(k) \cap \neg r(k)|}{|S| - |r(k)|} \tag{58}$$

$$\overline{FI} = \frac{\sum\limits_{k=1}^{K} FI(k)}{K}. \tag{59}$$

Table I shows only partial data (five rows) of a small-scale example. Any component state can be detected by several tests. For example, the state of component 1 can be detected by

TABLE II
RESULTS FOR SMALL-SCALE SCENARIO

Approximate duality gap (D) (%)	13.2
Correct isolation rate (CI)	100.0
False isolation rate (FI)	0.0
Primal function value (J)	-68
Dual function value (Q)	-59
Computation time (sec) (t)	0.12

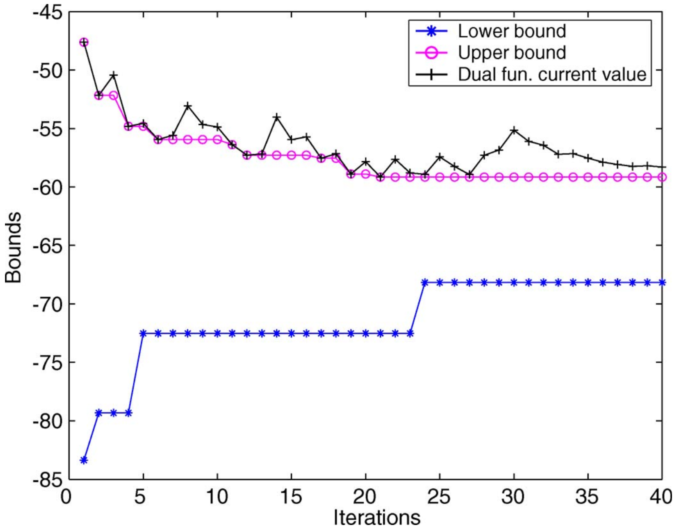

Fig. 9. Approximate duality gap.

t_3, t_{13}, and t_{19} with $(Pd_{1,3}, Pf_{1,3}) = (0.80, 0.01)$, $(Pd_{1,13}, Pf_{1,13}) = (0.75, 0)$, and $(Pd_{1,19}, Pf_{1,19}) = (0.74, 0)$, respectively. This implies that, when component state s_1 occurs, test t_3 detects it with probability 0.80, and if component state s_1 does not occur, test t_3 has a 0.01 probability of falsely implicating it. Similarly, test t_{13} detects s_1 with probability of 0.75 and has no false alarms. If s_1 is not present at epoch k, then at epoch $k+1$, the probability of having s_1 present is 0.005, while if s_1 is present at epoch k, s_1 has a 0.00026 probability of disappearing at epoch $k+1$. The test outcomes at epochs $k = 1, 2, 3, 19$, and 20 are shown in the table; test outcomes at other epochs are not shown here for simplicity and for saving space. For example, at $k = 1$, the outcome of tests t_3 is observed as having *failed*, while the outcome of all other tests except t_3 is observed as having *passed*. The DMFD problem here is to identify the evolution of component states over the 20 epochs. The results for this model are shown in Table II. Here, J, Q, D, CI, FI, and t denote the primal function value, dual function value, approximate duality gap, CI, FI, and computation time per epoch. The primal and dual function values are computed using (15)–(17) and (22)–(25), respectively. The approximate duality gap (D) is computed as a ratio of the difference between Q and J divided by the absolute value of the primal feasible value J. The algorithms were implemented in MATLAB. We used a standard PC having a Pentium 4 processor with 3.0-GHz clock speed and 512-MB RAM. The approximate duality gap is also shown in Fig. 9 is 13.2%. The duality gap reduces as the number of iterations increases, and the subgradient method converges to the minimum dual function value.

TABLE III
REAL-WORLD MODELS

	m	n	$Pd_{ij} Pf_{ij}$	c, Pa_i
Automotive	22	60	(0.85-0.95), 0-0.02	3, 9.13e-04
Docmatch	257	180	(0.6-1),0	9, 3.12e-04
Powerdist	96	98	(0.6-1),0	3,3.13e-04
Helitrans	34	51	(0.6-1),0	2, 2.95e-04
EngineSim	53	30	(0.6-1),0	2,5.68e-04

B. Real-World Data Sets

Table III illustrates the model parameters of an *automotive* system, a document matching system (*Docmatch*), a power distribution system (*Powerdist*), a UH-60 helicopter transmission system (*Helitrans*), and an engine simulator (*EngineSim*). Details of these models are provided in [5]. Here, m, n, and c denote the number of components (failure sources), number of tests, and average number of intermittent faults that can occur over a span of 100 epochs. The fault appearance probabilities (Pa_i) were computed based on the average number of intermittent faults (c). These real-world systems are not ideal because they have fewer tests as compared to failure sources; hence, some failure sources are not covered by any tests. The fault disappearance probabilities (Pv_i) were varied between 0.0025 and 0.0049 to allow c intermittent faults, on average. The probabilities of detection and false alarm were varied as shown in Table III. Here, $\overline{J}$, $\overline{Q}$, $\overline{D}$, $\overline{CI}$, $\overline{FI}$, and $\overline{t}$ denote the average primal function value, average dual function value, average approximate duality gap, average CI, average FI, and average computation time per epoch. The maximum number of subgradient iterations was set at 80 and 100 Monte Carlo runs were used to generate the test outcomes.

Table IV shows the results obtained using the subgradient (S) and DSA [11] methods. The subgradient method (S) achieves higher CIs as compared to DSA for all systems except *Helitrans*. However, the DSA method achieves better primal function value and is also effective in reducing the computation time (t). Also, note that we can obtain a hybrid duality gap by taking the maximum primal solution from the subgradient (S) and DSA methods, and the dual function value from the subgradient (S) method. The hybrid DSA–subgradient (HS) duality gaps are also shown in Table V. The average computation time $(\overline{t})$ is measured in seconds. These numbers are attractive practically, and they can be further reduced significantly by a careful implementation in the C language.

We also showed an application of the DMFD Problem 3 formulation in our recent paper [17] where we performed the dynamic fusion of classifiers over time for automotive engine fault diagnosis. The temporal correlations considered by dynamic fusion improve classification accuracy over a variety of static fusion techniques (based on batch data).

TABLE IV
RESULTS ON REAL-WORLD MODELS

		$\overline{J}$	$\overline{Q}$	$\overline{D}$ (%)	$\overline{CI}$	$\overline{FI}$	$\overline{t}$
Automotive	S	-658	-481	27	99.5	0.05	0.43
	DSA	-775	–	–	75	1.30	0.01
	HS	-658	-481	27			
Docmatch	S	-541	-311	42.5	88.2	0.36	4.53
	DSA	-405	–	–	69	0.70	0.24
	HS	-405	-311	23.2			
Powerdist	S	-232	-125	46.1	91.6	0.75	1.56
	DSA	-157	–	–	84	0.30	0.05
	HS	-157	-125	20.3			
Helitrans	S	-15	-14	6.7	94.8	0.31	0.47
	DSA	-15	–	–	100	0.0	0.02
	HS	-15	-14	6.7			
EngineSim	S	-85	-33	61.1	95.1	2.28	0.64
	DSA	-51	–	–	86	0.3	0.02
	HS	-51	-33	35.3			

TABLE V
TYPE OF FAULTS

	Fault disappearance probabilities	Fault behavior
Case 1	0.0247 to 0.0484	Intermittent
Case 2	6.8089e-004 to 9.2327e-004	Permanent
Case 3	0.2235 to 0.3926	Highly intermittent

C. Sliding Window DMFD Results

Figs. 10–12 show the boxplots of CI and FI for various real-world models. These plots were obtained using the sliding-window DMFD method.

The boxplot shows the dispersion of the data with lines at lower quartile (25%), median, and upper quartile (75%). The whiskers are shown by extending the lines from each end of the box, and the maximum length of the line is kept as a function of the interquartile range. The outliers are shown as "o" in the figures. The window size is selected such that it gives a high CI with a small boxsize and a low FI with a low boxsize. The most suitable window size using the aforementioned criterion for *Automotive*, *Docmatch*, *Powerdist*, *Helitrans*, and *EngineSim* systems are 15, 10, 20, 15, and 15, respectively. For Helitrans and EngineSim systems, window sizes of 15 and 25 achieves nearly the same performance; however, a smaller window size is preferred because it will reduce the time delay in making the diagnostic decisions.

Next, we perform simulations to study the effect of intermittent faults on the performance of the sliding-window DMFD method. The *automotive* system is used for simulations. The fault appearance probability was kept so that, on average, three faults occur over a span of 100 epochs.

Fig. 13 shows the CI for various fault behaviors. The results show the mean value, and the vertical lines on the data points indicate the standard deviation. These results were obtained using 1000 Monte Carlo runs. The results demonstrate that the algorithm achieves low variance when the fault behavior is highly intermittent.

The FI plot (Fig. 14) also illustrates the same behavior as the CI plot, i.e., the algorithm achieves the least variance for the highly intermittent fault types. This illustrates that the DMFD algorithm is highly suitable for intermittent faults.

D. Complexity

The algorithm presented here reduces the overall complexity from $\mathbf{O}(K(2^m))$ to $\mathbf{O}(K(m + O_f))$ where m is the number of component states, K is the number of epochs, and O_f is the set of failed tests. More specifically, the complexities of a binary Viterbi algorithm over all component states and the subgradient method are $\mathbf{O}(Km)$ and $\mathbf{O}(KO_f)$, respectively, per iteration; this is a substantial improvement over extensive approaches based on exact inference.

X. CONCLUSION

In this paper, we discussed the problem of DMFD with imperfect tests. The original DMFD problem is an intractable NP-hard combinatorial optimization problem. Using a Lagrangian relaxation-based coordination framework, we decomposed the original DMFD problem into parallel decoupled subproblems coordinated via Lagrange multipliers. Each subproblem corresponds to finding the optimal state sequence of a fault with fixed Lagrange multipliers. The subproblems were solved using a binary Viterbi decoding algorithm. The coordination among the subproblems was facilitated by Lagrange multipliers, which were updated using a subgradient method.

We discussed four formulations of the DMFD problem. Analogous forms of these formulations have been studied widely in fault diagnosis community in a static context and applied in various fields. Here, we provided a unified formulation of all the MFD formulations in a dynamic context. The

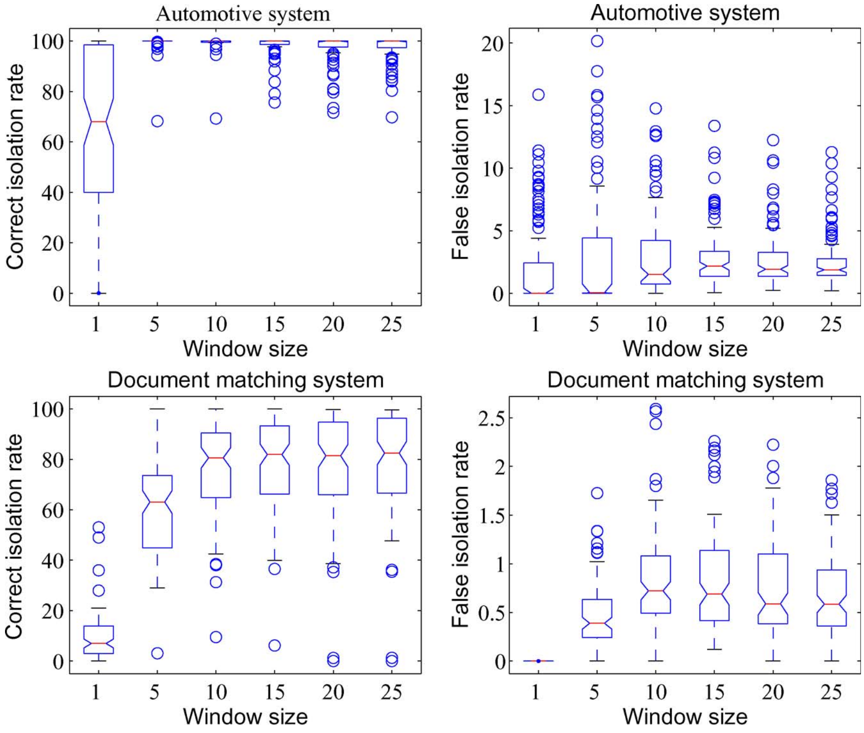

Fig. 10. Boxplots of CI and FI for automotive and document matching system.

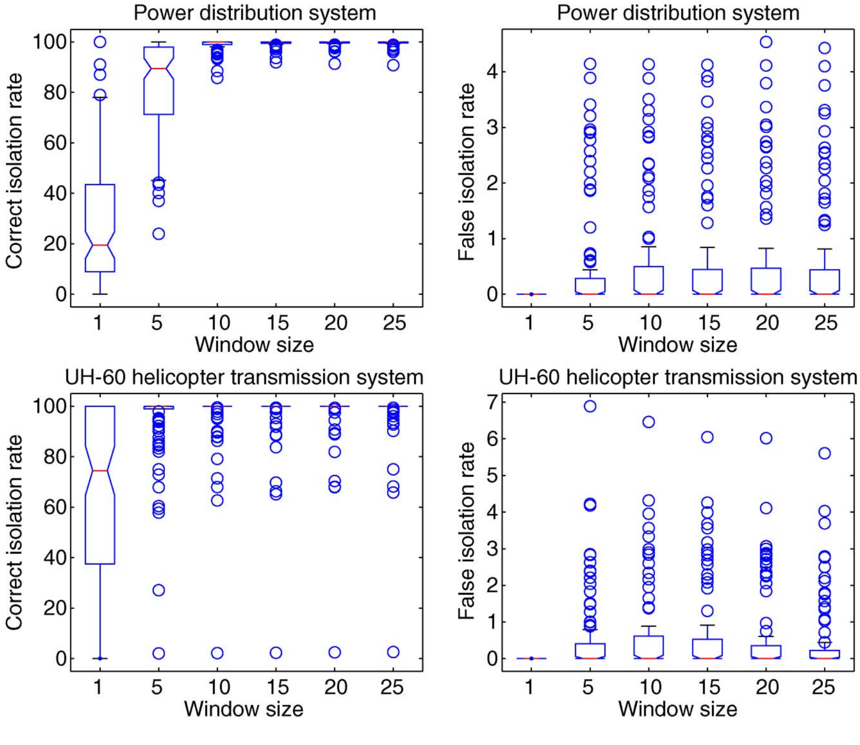

Fig. 11. Boxplots of CI and FI for power distribution and UH-60 helicopter transmission system.

first formulation refers to a generalized version of the DMFD problem when the detection and false alarm probabilities are associated with each test and fault. In the second formulation, the false alarm probability is associated with a fault-free case only. The solution to the second formulation was shown to be quite similar to that of problem formulation 1, except for the need to update an additional Lagrange multiplier. The third formulation considers the case where the uncertainties are associated with only test outcomes. This models the dynamic fusion of classifier outputs. In the fourth formulation, we considered the deterministic case, which led to a novel dynamic set covering problem. We implemented the algorithm on several real-world data sets, and the results validated the theory. The key advantage of our approach is that the method provides an approximate duality gap, which is a *worst case* indicator of the difference between the feasible and optimal solutions. Our

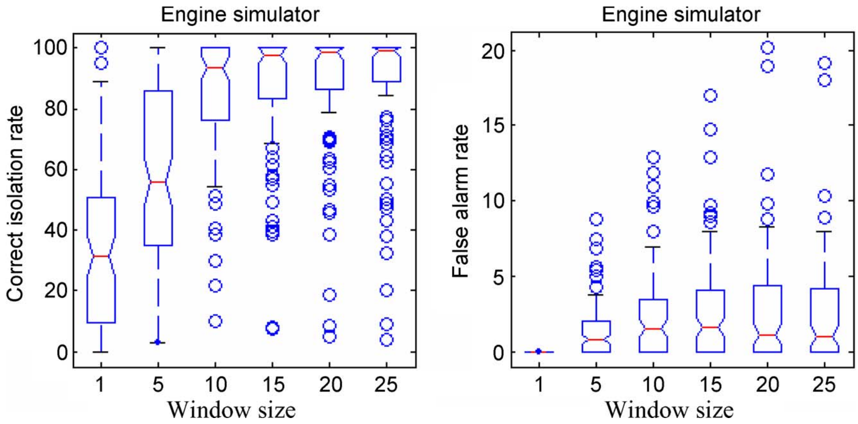

Fig. 12. Boxplots of CI and FI for engine simulator system.

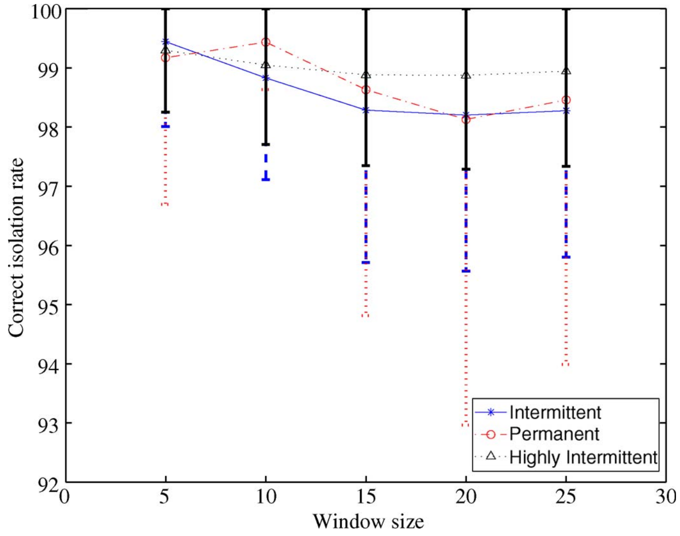

Fig. 13. CI for various fault behaviors.

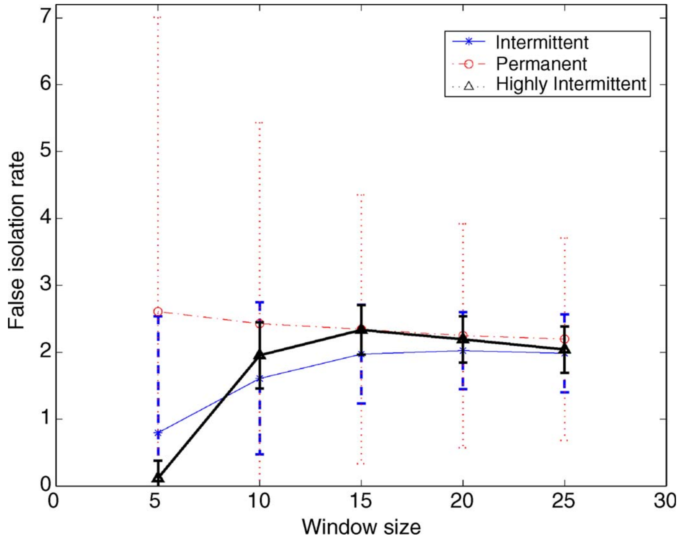

Fig. 14. FI for various fault behaviors.

results demonstrate that our algorithm achieves a high isolation rate as compared to the DSA method. The latter provides better primal function value as compared to the subgradient method. In this paper, we assumed that the DMFD model parameters are known and that faults evolve independently and are coupled through the test outcomes via the diagnostic matrix (D-matrix). In our future work, we will implement techniques to learn the DMFD model parameters from the observed test outcome sequences and relax the independence assumption to solve the DMFD problem when faults are dependent. Coupled HMMs offer a promising platform for the solution of the dependent fault problem [19]. We will also focus on improving the primal solution using a soft Viterbi algorithm.

Appendix I
Solving Subproblems Using Viterbi Algorithm

In this Appendix, we discuss the key steps of the Viterbi algorithm which is used to solve each subproblem corresponding to each component state sequence $\underline{x}_i$.

Initialization: In this step, the objective function is computed at $k = 1$ for each node (component state). It is assumed that the initial state $x(0)$ is known for all the component states. The maximum function value of ξ_i in (24) at time k is denoted by $\delta_k(x_i(k))$, and the value of x_i where the function value is maximum is denoted by $\psi_k(x_i(k))$. For the binary case, we have used the notation $\delta_k(0) = \delta_k(x_i(k) = 0)$ and $\delta_k(1) = \delta_k(x_i(k) = 1)$. At time $k = 1$

$$\begin{aligned}\delta_1\left(x_i(1)\right) =& \xi_i\left(x_i(1), x_i(0), \{\lambda_j(1)\}\right)\\ =& \left\{\mu_i(1) - \sum_{o_j \in O_f(1)} c_{ij}\lambda_j(1) + \sum_{o_j \in O_p(1)} c_{ij}\right\} x_i(1)\\ &+ \sigma_i(1)x_i(0) + h_i(1)x_i(1)x_i(0)\\ \psi_1\left(x_i(1)\right) =& \phi, \qquad \text{where} \quad x_i(0) \in \{0,1\}. \end{aligned} \tag{60}$$

Recursion: The recursion step involves maximizing the objective function at each epoch k.

$$\begin{aligned}\delta_k\left(x_i(k)\right) =& \left\{\sum_{o_j \in O_p(k)} c_{ij} + \mu_i(k)\right\} x_i(k)\\ &- \sum_{o_j \in O_f(k)} c_{ij}\lambda_j(k)x_i(k)\\ &+ \max_{x_i(k-1) \in \{0,1\}} \left[\delta_{k-1}\left(x_i(k-1)\right) + \sigma_i(k)x_i(k-1)\right.\\ &\left. + h_i(k)x_i(k)x_i(k-1)\right] \end{aligned} \tag{61}$$

for $2 \le k \le K$; $x_i(k) \in \{0,1\}$

$$\psi_k\left(x_i(k)\right) = \underset{x_i(k-1)\in\{0,1\}}{\arg\max} \left[\delta_{k-1}(x_i(k-1)) + \sigma_i(k)x_i(k-1) + h_i(k)x_i(k)x_i(k-1)\right]. \quad (62)$$

Termination: This step computes the objective function at time epoch $k = K$.

$$F^* = \max_{x_i(K)\in\{0,1\}} \left\{\delta_K\left(x_i(K)\right)\right]$$

$$x_i^*(K) = \underset{x_i(K)\in\{0,1\}}{\arg\max} \left[\delta_K\left(x_i(K)\right)\right]. \quad (63)$$

Optimal State Sequence Backtracking: The backtracking step computes the optimal state sequence by tracing the path backwards. The optimal state $x_i^*(k)$ of ith fault at time epoch k is given by

$$x_i^*(k) = \psi_{k+1}(x_i(k+1)^*), \qquad k = K-1, \ldots, 1. \quad (64)$$

Similar to the recursion step, we can further simplify termination and backtracking for the binary case.

APPENDIX II
UPDATING LAGRANGE MULTIPLIERS VIA SUBGRADIENT METHOD

Lagrange multipliers are updated via

$$\lambda_j^{l+1}(k) = \max\left(0, \lambda_j^l(k) + \beta^l(k)d_j^l(k)\right) \quad (65)$$

for $j \in O_f(k)$ and $k \in (1, K)$ where the subgradients $d_j^l(k)$ at iteration l and epoch k are

$$d_j^l(k) = \ln\left(y_j^*(k)\right) - \sum_{i=1}^{m} c_{ij}x_i^*(k) - \eta_j \quad (66)$$

and step size $\beta^l(k)$ is

$$\beta^l(k) = -\upsilon \frac{(Q^l - Q^*)}{\sum\limits_{j=1}^{T_f} \left(d_j^l(k)\right)^2}. \quad (67)$$

Lagrange multipliers were initialized at iteration $l = 0$ with values equal to 0.5. Since the optimal dual function value is not available, it is estimated using the primal feasible solution J_f and best current dual value $Q_{\min}$ using (27) and (28), respectively. We estimate the optimal dual function as

$$\hat{Q}^* = \frac{\omega(J_f + Q_{\min})}{2} \quad (68)$$

and initial value $\upsilon = 0.01$ is used. If the best current dual value $Q_{\min}$ does not decrease in the previous 20 iterations of the subgradient procedure with the current value of υ, then υ is reduced by a factor. To improve the subgradient convergence, we also vary ω, which is increased or decreased based on whether the dual function value is decreasing or not [13]. We used the following stopping criteria for the subgradient method.

1) Stop if $\sum_{j=1}^{O_f}(d_j^l(k))^2 = 0$ since we cannot define a suitable step size in this case.
2) Stop if $\upsilon \le 10^{-4}$ because step sizes become too small.
3) Stop if the number of iterations crossed the maximum number of iterations, i.e., $l \ge 100$.

ACKNOWLEDGMENT

Any opinions expressed in this paper are solely those of the authors and do not represent those of the sponsor.

REFERENCES

[1] F. Yu, F. Tu, H. Tu, and K. R. Pattipati, "Multiple disease (fault) diagnosis with applications to the QMR-DT problem," in *Proc. Comput. Commun. Control Technol. Int. Conf.*, Austin, TX, 2004, pp. 227–233.

[2] F. Yu, F. Tu, H. Tu, and K. R. Pattipati, "Multiple disease (fault) diagnosis with applications to the QMR-DT problem," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, pp. 746–757, Sep. 2007.

[3] N. Odintsova, I. Rish, and S. Ma, "Multifault diagnosis in dynamic systems," in *Proc. IM*, 2005.

[4] M. Shakeri, K. R. Pattipati, V. Raghavan, and A. Patterson-Hine, "Optimal and near-optimal algorithms for multiple fault diagnosis with unreliable tests," *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, vol. 28, no. 3, pp. 431–440, Aug. 1998.

[5] F. Tu, K. R. Pattipati, S. Deb, and V. N. Malepati, "Computationally efficient algorithms for multiple fault diagnosis in large graph-based systems," *IEEE Trans. Syst., Man Cybern.*, vol. 33, no. 1, pp. 73–85, Jan. 2003.

[6] K. R. Pattipati and M. G. Alexandridis, "Application of heuristic search and information theory to sequential fault diagnosis," *IEEE Trans. Syst., Man Cybern.*, vol. 20, no. 4, pp. 872–887, Jul./Aug. 1990.

[7] V. Raghavan, M. Shakeri, and K. R. Pattipati, "Optimal and near-optimal test sequencing algorithms with realistic test models," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 29, no. 1, pp. 11–27, Jan. 1999.

[8] T. Le and C. N. Hadjicostis, "Graphical inference methods for fault diagnosis based on information from unreliable sensors," in *Proc. Int. Conf. Control, Autom., Robot. Vis.*, Singapore, Dec. 2006, pp. 1012–1017.

[9] J. Ying, T. Kirubarajan, and K. R. Pattipati, "A hidden Markov model based algorithm for fault diagnosis with partial and imperfect tests," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 30, no. 4, pp. 463–473, Nov. 2000.

[10] O. Erdinc, C. Raghavendra, and P. Willett, "Real-time diagnosis with sensors of uncertain quality," in *Proc. SPIE Conf.*, Orlando, FL, Apr. 2003, pp. 1490–1499.

[11] S. Ruan, Y. Zhou, F. Yu, K. R. Pattipati, P. Willett, and A. Patterson-Hine, "Dynamic multiple fault diagnosis and imperfect tests," in *Proc. AUTOTESTCON*, 2004, pp. 395–401.

[12] Z. Ghahramani and M. I. Jordan, "Factorial hidden Markov models," in *Machine Learning*. Boston, MA: Kluwer, 1997.

[13] D. Bertsekas, *Nonlinear Programming*, 2nd ed. Belmont, MA: Athena Scientific, 2003.

[14] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1988.

[15] D. Forney, Jr., "The Viterbi algorithm," *Proc. IEEE*, vol. 61, no. 3, pp. 268–278, Mar. 1973.

[16] J. E. Beasley, "An algorithm for set covering problems," *Eur. J. Oper. Res.*, vol. 31, pp. 85–93, 1987.

[17] S. Singh, K. Choi, A. Kodali, K. Pattipati, S. M. Namburu, S. Chigusa, D. V. Prokhorov, and L. Qiao, "Dynamic fusion of classifiers for fault diagnosis," in *Proc. IEEE SMC Conf.*, Montreal, QC, Canada, Oct. 2007, pp. 2467–2472.

[18] A. Kodali, S. Singh, K. Choi, K. Pattipati, S. M. Namburu, S. Chigusa, D. V. Prokhorov, and L. Qiao, "Diagnostic inference with nearly perfect tests," in *Proc. IEEE Aerosp. Conf.*, Big Sky, MT, Mar. 2008.

[19] M. Brand, "Coupled hidden Markov models for modeling interacting processes," in *Neural Comput.*, Nov. 1996.

Satnam Singh (S'00–M'08) received the B.E. degree from the Indian Institute of Technology, Rookree, India, in 1996, the M.S. degree in electrical engineering from the University of Wyoming, Laramie, in 2003, and the Ph.D. degree in electrical engineering from the University of Connecticut, Storrs, in 2007.

He is currently a Senior Researcher with General Motors India Science Laboratory, Bangalore, India. Prior to his graduate studies, he was a Design Engineer with Engineers India Ltd., New Delhi, India, for four years.

Dr. Singh is a reviewer for IEEE Potentials, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, and IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT. His research interests include data mining, graphical models, pattern recognition, fault detection and diagnosis, and optimization theory.

Anuradha Kodali received the B.E. degree in electronics and communications engineering from Andhra University, Visakhapatnam, India, in 2006. She is currently working toward the M.S. degree in electrical and computer engineering in the Electrical and Computer Engineering Department, University of Connecticut, Storrs.

Her research interests include data mining, pattern recognition, fault detection and diagnosis, and optimization theory.

Kihoon Choi is currently working toward the Ph.D. degree in the electrical and computer engineering in the Electrical and Computer Engineering Department, University of Connecticut, Storrs.

His research interests are in the areas of fault diagnosis of complex systems, system real-time fault diagnosis, and analysis and optimization of large-scale systems.

Mr. Choi is the recipient of the Best Technical Paper Award at the 2004 IEEE AUTOTEST Conference.

Krishna R. Pattipati (S'77–M'80–SM'91–F'95) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kharagpur, India, in 1975 and the M.S. and Ph.D. degrees in control and communication systems from the University of Connecticut (UConn), Storrs, in 1977 and 1980, respectively.

He is currently a Professor of electrical and computer engineering with the Electrical and Computer Engineering Department, UConn. He has published over 370 articles, primarily in the application of systems theory and optimization techniques to large-scale systems. His research has been primarily in the application of systems theory and optimization techniques to complex systems.

Dr. Pattipati was the Editor-in-Chief of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS (SMC)—PART B: CYBERNETICS during 1998–2001. He is the recipient of the Centennial Key to the Future Award, in 1984, from the IEEE SMC Society, the Andrew P. Sage Award for the Best SMC Transactions Paper for 1999, the Barry Carlton Award for the Best Aerospace and Electronic Systems Transactions Paper for 2000, the 2002 and 2008 NASA Space Act Awards, the 2003 American Association of University Professors Research Excellence Award, and the 2005 School of Engineering Teaching Excellence Award at UConn. He also the recipient of the Best Technical Paper Awards at the 1985, 1990, 1994, 2002, 2004, and 2005 IEEE AUTOTEST Conferences, and at the 1997 and 2004 Command and Control Conferences.

Setu Madhavi Namburu received B.Tech. degree from Jawaharlal Nehru Technological University, Hyderabad, India, in 2002 and the M.S. degree from the University of Connecticut, Storrs, in 2006, both in electrical engineering.

Currently, she is as a Research Engineer with Toyota Motor Engineering and Manufacturing North America, Inc., Erlanger, KY. Her research interests include data mining, knowledge discovery, systems analysis, simulation, and fault diagnostics and prognostics.

Shunsuke Chigusa Sean received the M.S. degree in electrical engineering from Osaka University, Osaka, Japan.

He is a Senior Principal Research Scientist with the Technical Research Department, Toyota Motor Engineering and Manufacturing North America, Inc., Erlanger, KY. He started his automotive career as an Autonomous Vehicle Control System Scientist and later as an Intelligent Diagnosis Scientist with Toyota Motor Corporation, Japan.

Danil V. Prokhorov received the M.S. degree (with honors) from the State Academy of Aerospace Engineering (former LIAP), St. Petersburg, Russia, in 1992 and the Ph.D. degree in electrical engineering from Texas Tech University, Lubbock, in 1997.

He was with Ford Research Laboratory, Dearborn, MI, from 1997 until 2005. He has been engaged in application-driven studies of neural networks and training algorithms. He is currently a Research Manager with the Technical Research Department, Toyota Motor Engineering and Manufacturing North America, Inc., Erlanger, KY. His research interest is in machine learning algorithms and their applications to decision making under uncertainty. He has authored 80 technical publications including several patents.

Dr. Prokhorov was the recipient of the International Neural Network Society (INNS) Young Investigator Award in 1999. He was the INNS-IEEE International Joint Conference on Neural Networks (IJCNN) 2005 General Chair and the IJCNN 2001 Program Chair. He has been a reviewer for numerous journals and conferences, a program committee member for many conferences, and a panel expert for the National Science Foundation every year since 1995.

Liu Qiao received the B.S. degree in engineering from Beijing University of Technology, Beijing, China, and the M.S. and Ph.D. degrees in electrical engineering from Tohoku University, Sendai, Japan.

He is a General Manager and Chief Technologist with the Technical Research Department, Toyota Motor Engineering and Manufacturing North America, Inc., Erlanger, KY. Switching from a university faculty position, he started his automotive career as an Advanced Automotive Control System Expert, Advanced Technology Manager, eBusiness Manager, and Research Manager. He successfully led a Canadian hybrid vehicle project and its market introduction. He is an active member/supporter of many academic associations.