Model-Based Diagnosis of Hybrid Systems

Sriram Narasimhan and Gautam Biswas, *Senior Member, IEEE*

***Abstract*—Techniques for diagnosing faults in hybrid systems that combine digital (discrete) supervisory controllers with analog (continuous) plants need to be different from those used for discrete or continuous systems. This paper presents a methodology for online tracking and diagnosis of hybrid systems. We demonstrate the effectiveness of the approach with experiments conducted on the fuel-transfer system of fighter aircraft.**

***Index Terms*—Fault detection and isolation, hybrid systems, model-based diagnosis (MBD).**

I. Introduction

THIS PAPER discusses methods for online monitoring and diagnosis of complex hybrid systems. The behavior of these systems, common in the avionics, automotive, and robotics domains, is composed of continuous plant dynamics described by continuous-state variables and supervisory control that generates actuator signals at discrete time points to change the set points of regulators or the plant configuration. Therefore, hybrid-system analysis has to seamlessly integrate discrete and continuous behaviors and multiple-system models. As a result, tasks like monitoring, fault diagnosis, and control require model selection and switching to be performed online as system behavior evolves.

Faults in hybrid systems can occur in the plant, controller [either physical (hardware) or logical (software)], and in the interface between the discrete and continuous components of the system. In this paper, we focus on faults that occur in plant components. We develop parameterized plant models, where faults are represented by abrupt changes in system parameters (i.e., the change in parameter value is faster than the system dynamics). Fault diagnosis involves the detection of anomalous system behavior, its isolation to specific causes, and estimation of the amount of the change. Insufficient diagnostic capability and inability to perform timely fault diagnosis reduces system availability and operational readiness while increasing life-cycle costs. Hence, diagnostic capabilities are a key determinant of system-performance and cost-effectiveness goals.

We apply model-based diagnosis (MBD) methodologies to hybrid systems. MBD exploits the analytical redundancy between the model and the system measurements [1], [2]. MBD does not require the pre-enumeration of the explicit association between faults and their externally visible manifestations, since the model captures the constraints between faults and discrepancies in a compact form. Current MBD techniques are designed for dynamic systems whose behavior is modeled by discrete event [4], [5] or continuous models [1], [6], [7]. Discrete-event diagnosis approaches abstract nominal and faulty-system behavior in the form of event trajectories. This process may result in loss of information critical for fault isolation and control. Our work in continuous diagnosis has demonstrated that behavior transients are the key to quick diagnosis of abrupt faults [8]. Traditional algorithms for continuous diagnosis use models that do not accommodate discrete changes. Therefore, discrete effects of mode changes have to be modeled by complex nonlinear functions that are hard to analyze online in real time.

Recent work on hybrid-system diagnosis [9]–[11] has focused on discrete failures (e.g., valve stuck open). These algorithms require the pre-enumeration of the model in all modes to perform diagnosis. We present an online MBD methodology for parametric faults in hybrid systems that is based on tracking hybrid behaviors (continuous behaviors interspersed with discrete changes), but unlike hybrid-automata models [12], pre-enumeration of all system modes is avoided by generating models at runtime as mode switches occur. After a fault occurs, the system model is no longer correct; therefore, predicting autonomous mode changes (i.e., mode changes that are dependent on system variables) is not possible. To address this, the monitoring task for fault isolation has to hypothesize mode changes while tracking faulty behavior, and multiple behavior trajectories may have to be tracked until the true fault is isolated and identified. We have developed a tracking, fault-detection, and fault-isolation scheme, which address all of the issues outlined above. Our method is also designed to diagnose partial failures (e.g., a 10% change in a pipe-resistance parameter).

This paper discusses our MBD engine for hybrid systems, which includes: a unified-modeling methodology geared to the diagnosis task, observers and fault detectors for tracking system behavior and identifying discrepancies between predicted and observed behavior, and fault-isolation and identification schemes for determining the faulty component and computing the magnitude of the change. The rest of this paper discusses each of these methods.

II. Unified-Modeling Framework

Parsimonious models that establish relevant links between faults and observed system behavior form the key to developing efficient-diagnosis algorithms. However, tradeoffs between

Manuscript received September 23, 2003; revised November 16, 2005. This work was supported in part by the DARPA/IXO SEC program (F30602-96-2-0227), by the NASA IS NCC 2-1238, by the NASA ALS NCC 9-159, and by the Boeing Company (Kirby Keller and Tim Bowman). This paper was recommended by Associate Editor P. K. Willett.

S. Narasimhan is with the University of California, Santa Cruz, and also with the NASA Ames Research Center, Moffett Field, CA 94035 USA (e-mail: sriram@e-mail.arc.nasa.gov).

G. Biswas is with the Department of Electrical Engineering and Computer Sciences and Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN 37235 USA (e-mail: gautam.biswas@vanderbilt.edu).

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.2007.893487

accuracy and simplicity of the models have to be considered. Detailed and accurate models may generate more precise diagnoses, but the increased computational complexity makes online analysis computationally infeasible. Simpler models reduce computational complexity, but they produce large ambiguity sets (multiple-fault hypotheses that cannot be distinguished using the model and available measurements).

Hybrid-system models capture the continuous and discrete characteristics of the system behavior and their interactions. We employ a unified physics-based modeling framework called hybrid bond graphs (HBGs) [14] that extends the continuous bond-graph modeling paradigm [13] to provide a compact representation for hybrid models. The modeling paradigm is topological, component-based and hierarchical, and facilitates causal analysis of system dynamics. It also provides standard techniques for deriving state space and input–output equation formulations.

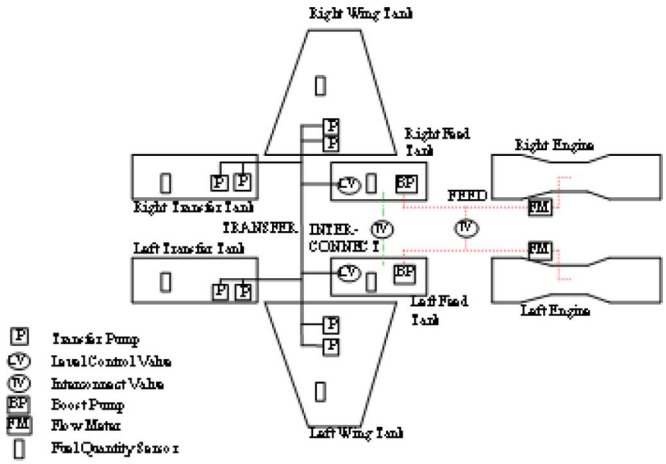

Fig. 1. Fuel-transfer-system schematic.

A. Hybrid Bond Graphs

Bond graphs (BGs) are a domain-independent topological-modeling language that capture energy-based interactions among the processes that make up a physical system [13]. Energy exchange among generic-component processes, such as capacities, inertias, and resistances, are modeled by bonds. 1- and 0-junctions model the equivalent of series and parallel topologies, respectively. HBGs extend BGs by incorporating switched junctions [14] to enable discrete changes in the system configuration. The BG junctions may be dynamically switched on (active) and off (inactive) as system behavior evolves. An active junction behaves like a regular BG junction, whereas in the OFF state, all bonds incident on the junction are deactivated. Therefore, an inactive junction and the connected bonds do not play any part in determining the system dynamics. Activating and deactivating junctions may affect the behavior at adjoining junctions.

A two-state (OFF and ON) finite-state automaton (FSA) implements the junction-switching function. Transitions are defined by external-controller signals (controlled switching) and by internal variables crossing-boundary values (autonomous switching). The overall system mode is determined by a parallel composition of modes of the individual switched junctions. More formally, the FSA associated with an HBG-switched junction is defined by a five tuple: $M = (Q, \sum, \delta, q_0)$, where $Q = \{\text{on}, \text{off}\}$ is the set of states, Σ is the set of controlled and autonomous events, $q_0 \in Q$ is the initial state, and $\delta : Q \times \sum \rightarrow Q$ is the transition function that defines state changes after the occurrence of events in Σ. The reset function γ for hybrid-automata models can be directly derived from the switched HBG model. In other work [15], we have derived systematic procedures for converting HBG models to hybrid-automata models [12].

As an example, we look at the HBG model for part of a generic fuel-transfer system for fighter aircraft, as illustrated in Fig. 1. The system is designed to provide an uninterrupted supply of fuel at a constant rate to the aircraft engines while maintaining the center of gravity of the aircraft. The system is symmetrically divided into left and right parts (top and bottom

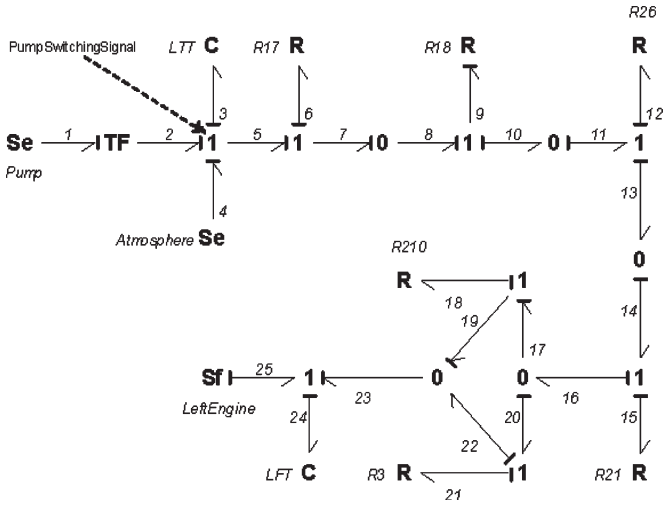

Fig. 2. HBG for part of fuel system.

in the schematic). The four supply tanks [Left Wing (LWT), Right Wing (RWT), Left Transfer (LTT), and Right Transfer (RTT)] are full initially and so are the two receiving tanks [Left Feed (LFT) and Right Feed (RFT)] that directly feed the engine. During engine operation, fuel is transferred from the supply tanks through a common manifold to the two feed tanks in a sequence determined by the fuel-system controller. The controller generates ON/OFF signals for the pumps in the supply tanks and the valves in the pipes to achieve different flow configurations.

The HBG in Fig. 2 models the part of the system which transfers fuel from the LTT through the LFT to the aircraft engine. The rest of the system includes other fuel flow paths from the transfer and wing tanks through the feed tanks to the engines. Fuel is pumped from the LTT, modeled as a capacitor, using a pump, to create a pressure head. In this paper, we have modeled pumps simply as a source of effort (pressure) Se. The pipes are modeled as nonlinear resistances, and fuel transfers through a shared manifold to the feed tank through a main pipe (resistance R3) and a parallel-bleed path, R210. The engine is modeled to drain fuel from the feed tank (capacitor) at a constant rate. In addition to the regular BG elements, the

TABLE I
TCG LABELS

Label	Propagation Type	Effect
1	Instantaneous Propagation	Direct
-1		Inverse
=		Equal
p		Direct
$1/p$		Inverse
$\int(p)\partial t$	Time Delayed Propagation	Direct
$\int(1/p)\partial t$		Inverse

model includes switching signals (the ones associated with the LTT path is shown in Fig. 2) generated by the controller to turn the pumps and connecting valves on and off. The HBG models associate one or more parameters with system components. We exploit this in defining a component-based diagnosis methodology, where faults in components are represented as deviations in their parameter values. For example, there are five potential fault candidates in the fuel-transfer subsystem modeled in Fig. 2: pump efficiency, tank capacities, and tank and switched-pipe resistances (blocks and leaks).

B. Additional Model Representations

The topological HBG representation can be used to systematically derive other model representations, such as state-space equations for the observer model, temporal causal graphs (TCGs) for qualitative fault isolation, and input–output equations for parameter estimation to estimate fault magnitudes. The method for deriving the state space and input–output models are presented elsewhere [15]. We focus on the TCGs, the underlying model for the innovative qualitative diagnosis scheme described in this paper and earlier work [6], [8].

A TCG captures causal and temporal relations among the system variables that characterize dynamic system behavior. They are an extended form of signal flow graphs [16] with vertices representing the system variables (e.g., pressures, temperatures, and flow rates) and labeled directed edges capturing the relations between the variables. The labels on the edges further qualify the relations between the vertices.

Table I lists the possible labels on the TCG edges and their meaning. Consider an edge $\nu_1 \rightarrow \nu_2$ between two system variables, ν_1 and ν_2. A label of ± 1 on this edge implies that ν_2 is directly (inversely) proportional to ν_1. In other words, if ν_1 increases, there is a proportional increase (decrease) in ν_2. The $=$ label implies $\nu_1 = \nu_2$. The symbol p on an edge corresponds to a nonjunction element (capacity, inertia, resistance, transformer, and gyrator) parameter in the BG model. The implication is that ν_2 is directly proportional to $(p \times \nu_1)$. An increase in ν_1 or an increase in p would cause an increase in ν_2. A $1/p$ label indicates that ν_2 is proportional to ν_1/p implying that a decrease in p would result in an increase in ν_2. Temporal relations, introduced through capacity and inertia elements, are represented by ∂t in the label. If the edge label is $p\partial t$, this implies that ν_2 is directly proportional to p and $\int \nu_1 \partial t$. Therefore, an increase in ν_1 would cause an increase in the derivative of ν_2. $\partial t/p$ indicates that ν_2 is proportional to $1/p \int \nu_1 \partial t$. Similar meanings hold for the relations that include a parameter and a negative sign [$-p$, $-1/p$, $-p\partial t$, and $-\partial t/p$]. The algorithm to derive the TCG from HBG models is presented in [15].

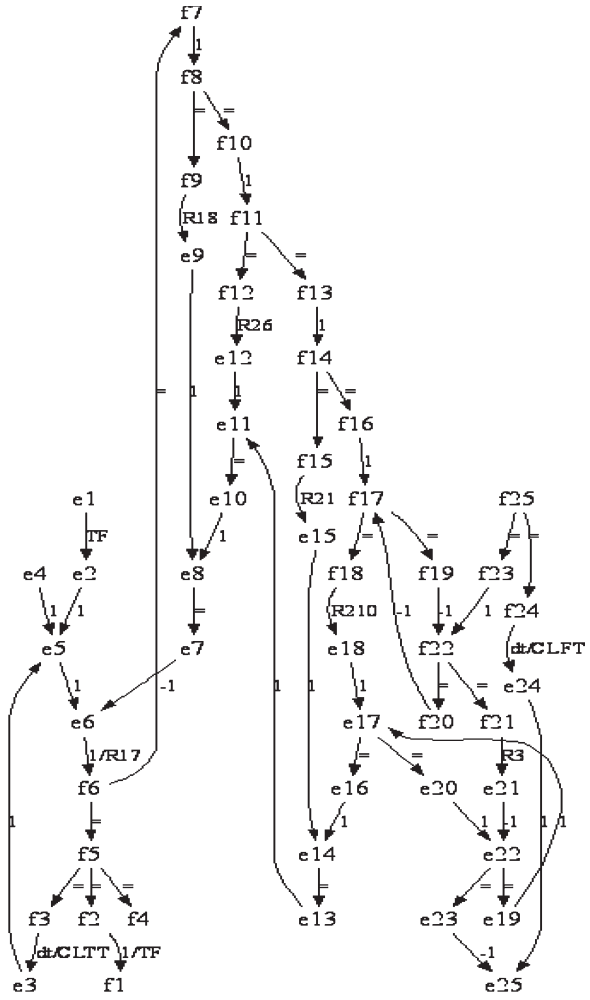

Fig. 3. TCG for HBG in Fig. 2.

The TCG model derived from the partial HBG model in Fig. 2 is illustrated in Fig. 3. The ei and fi variables correspond to pressure and flow rates at various locations in the system. The component parameters, CLFT, CLTT, R1, R2, and TY, appear on the edges. They modulate the corresponding pressure-flow relations in the system.

III. MBD ARCHITECTURE

The diagnosis scheme, illustrated in Fig. 4, includes a hybrid observer that tracks the continuous-system-behavior and discrete-mode changes. When the observer output shows statistically significant deviations from observed behavior, the fault detector triggers the fault-isolation scheme. We follow the International Federation of Automatic Control Technical Committee SAFEPROCESS [7] definition of a fault as "an unpermitted deviation of at least one characteristic property or parameter of the system from acceptable, usual, or standard conditions." This paper focuses on persistent abrupt changes in the component parameter values, which correspond to multiplicative faults.

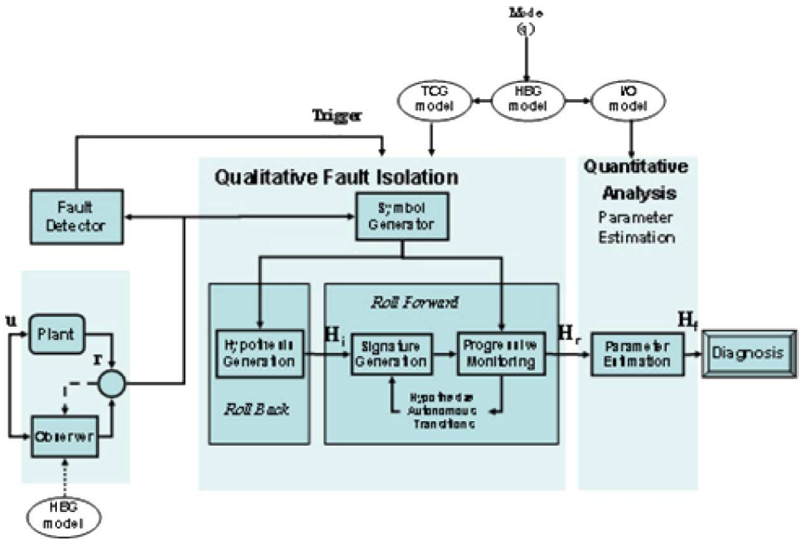

Fig. 4. Fault isolation and identification architecture.

The component parameters of diagnostic interest are in one-to-one correspondence with the parameters of the HBG model. Furthermore, we make the single-fault assumption. As discussed earlier, the hybrid nature of the system complicates the tracking and diagnosis tasks, because mode transitions cause model switching, and this has to be included in online behavior tracking and fault isolation.

This approach to online FDI is novel and extends our earlier work [6], [8] on combined qualitative and quantitative fault isolation in continuous systems. The qualitative approach overcomes limitations of quantitative schemes, such as convergence and accuracy problems in dealing with complex nonlinearities, and lack of precision of parameter values in system models. The significant reduction in computational complexity also makes online fault isolation feasible in the hybrid framework. The qualitative-reasoning scheme is fast and effective, but it has limited discriminatory ability. To uniquely identify the true fault candidate, we employ a quantitative parameter-estimation scheme that computes the deviated parameter value. In the rest of this section, we describe each of the components of the diagnosis scheme in greater detail.

A. Tracking and Fault Detection

The hybrid observer uses an extended Kalman filter (EKF) scheme [17] to track continuous behavior in individual modes of operation. A hybrid automaton derived from the individual-guard conditions for junction switching defines the mode transitions. To track continuous behavior interspersed with discrete mode changes, the hybrid observer must: 1) execute controlled and autonomous mode changes. This requires the following operations to be performed online: a) switching models and b) deriving the initial system state in the new mode after the transition. 2) Track continuous behavior in individual modes of operation. The EKF is designed to track nonlinear system behavior [17]. Model uncertainty and measurement noise are implemented as white uncorrelated Gaussian distributions with zero mean.

The state-space model in mode q is defined by the state equation

$$\underline{\chi}_{k+1} = F_q(\underline{\chi}_k)\underline{\chi}_k + G_q(\underline{\chi}_k)\underline{u}_{k+1} + \underline{w}_k$$
$$\underline{y}_{k+1} = C_q(\underline{\chi}_k)\underline{\chi}_{k+1} + D_q(\underline{\chi}_k)\underline{u}_{k+1} + \underline{\nu}_{k+1}$$

where $\underline{w}$ is distributed $N(0, Q)$ and $\underline{\nu}$ is distributed $N(0, R)$. Q and R are process and measurement noise-covariance matrices, and $\underline{w}_k$ incorporates the $\Delta F_q \times \underline{x}_k$ term that captures modeling errors in the system. In this paper, the Q and R matrices are determined empirically.

Mode-change calculations are based on the system mode at time step k (q_k) and the continuous state of the system $\underline{x}_k$. Discrete controller signals to the plant and autonomous mode changes computed from the estimated state vector $\underline{x}_k$ at time-step k causes a mode transition from q_k to q_{k+1} at time-step $k+1$. Several transition functions may be triggered simultaneously. They are combined to derive the current-system mode and the state-equation model for that mode.

For complex systems with a large number of switching elements (e.g., valves and pumps), the number of possible modes may be very large, and it is infeasible to precompute the complete hybrid-automaton model. We have developed an innovative scheme for dynamically generating the state equations for the new mode from the HBG model when mode transitions occur. This includes an efficient symbolic solver that constructs the state equations from equation fragments. The equation fragments correspond to constituent equations that define component behavior and junction relations. When switching occurs, sets of equation fragments are deactivated and others are activated. The new state equations, i.e., the matrices F_q, G_q, C_q, and D_q are then derived incrementally.

The online observer algorithm for tracking hybrid behavior is outlined below (details appear in [15]).

Step 1) Initialize Kalman gain matrix and derive the constituent element and junction equations for all BG elements in symbolic form.

Step 2) Derive model for initial mode and when mode changes occur by identifying the set of junctions that are on and deriving the state-space matrices from the active equation fragments.

Step 3) Recalculate parameter values to accommodate nonlinearities in the model (time-varying physical parameters).[1] Typically, only a few parameters are time varying, so these calculations do not consume a lot of computational resources.

Step 4) Execute a step of the EKF by updating the estimated state vector, the Kalman gain matrix, and the estimated output vector. Junction switch guard conditions are checked with the updated values of the state variables. When mode changes occur, find the new set of active junctions and repeat step 2). Modulated parameters may necessitate iterative calculations to find the fixed point of the corresponding modulating functions.

Fault detection compares the observer estimates of the output variables $\hat{\underline{y}}_k$ with the observed $\underline{y}_k$. Noise suppression is performed with an appropriate FIR filter. For this paper, we have used a simplistic filter of the form $\underline{y}_k = (1/N)\sum \underline{y}_{k-n}$ for smoothing the measured output signal. Please note that this filter may not work in all cases, however, any other appropriately designed filter can replace the aforementioned filter without affecting the rest of the algorithm. The residual at time-step k is computed as the difference between the smoothed measurement $\underline{y}_k$ and the observer-estimated $\hat{\underline{y}}_k$, i.e., $\underline{r}_k = \underline{y}_k - \hat{\underline{y}}_k$. The fault-detection scheme utilizes a statistical hypothesis-testing scheme for reliable fault detection with low false-alarm rates [15].

B. Fault Isolation and Identification

Our fault-isolation and identification scheme is presented in Fig. 4. For hybrid systems, discontinuous changes in measured variables can only occur at the point of failure or at points at which discrete mode changes occur in the plant behavior. At all other time points, the plant behavior is continuously differentiable. We take advantage of this fact when analyzing the residual vector $(\underline{r}_k)$. Since the residual is continuously differentiable, the diagnosis task involves continuous residual analysis within modes and switching the system model when mode changes are detected or hypothesized. Continuous residual analysis using qualitative-reasoning methods has been described in earlier work [6], [8], and is described briefly in Section III-B1. The hybrid-diagnosis algorithm extends the continuous-residual-analysis method to account for mode changes as summarized below.

1) A fast (performed in a single time step) qualitative roll-back process that takes into account that the fault may have happened in a previous mode of the system.
2) A quick (performed in a single time step) qualitative roll-forward process that extends signature generation and progressive monitoring across mode changes.
3) Continued progressive monitoring to refine the fault-candidate set allowing for mode changes followed by parameter estimation using least square estimation to isolate and identify the faulty parameter.

1) Residual Analysis and Diagnosis of Continuous Systems: We define a fault as an abrupt change in a parameter value, i.e., $\langle p, \lambda \rangle$, where p is the parameter and $\lambda = \pm$ indicates the deviation of the parameter value from normal. The Taylor series expansion characterizes the continuous residual signal transient about the point of failure, i.e.,

$$r(t) = r_{t_{\text{fo}}} + r'_{t_{\text{fo}}}(t - t_{\text{fo}})/1! + \cdots + r^k_{t_{\text{fo}}}(t - t_{\text{fo}})^k/k! + R_k$$

where t_{fo} is the time point of fault occurrence and $t > t_{\text{fo}}$. R_k is a remainder term based on higher order derivatives of r. For most well-behaved functions, this series converges, i.e., $R_k \to 0$ as $k \to \infty$. In particular, if $|r^{(k+1)}|$ is bounded, the Taylor series expansion is a good approximation of the true signal r_t when t is close to t_{fo}. Note that r_t corresponds to the effects of an abrupt change in the parameter value at $t = t_{\text{fo}}$. This representation forms the basis for our characterization of the fault transient by a set of fault signatures. Each fault signature, associated with a measured variable, is defined as a tuple made up of the magnitude change and derivatives of the signal transient. A unique contribution of our previous work is the ability to define and analyze dynamic fault signatures in a qualitative framework [18].

Definition 1—Fault Signature: The fault signature $\text{fs}_h(y^j)$ describes the effects of a fault $\langle p, \lambda \rangle$ on a measured variable y^j, as coefficients of magnitude and higher order derivatives of the Taylor series of the corresponding residual signal r^j computed at the time point of failure.

The fault signature is generated from the TCG model. Comparing the fault signature against observed residuals forms the basis for fault isolation [6]. We express the symbolic magnitude and slope of the measured residual as $\rho(r)$ and $\rho'(r)$, respectively. The elements of the two vectors can take on values 0 (normal), +(above nominal), and −(below nominal).

Definition 2—Qualitative Fault Signature: The qualitative fault signature $\text{qfs}_h(y^j)$ describes the effects of a fault $\langle p, \lambda \rangle$ on a measured variable y^j, as the sign of the magnitude and higher order derivatives of the coefficients of the corresponding fault signature computed at the point of failure.

A more formal description of the residual-generation and analysis scheme and the discussion of the computation of the symbolic residuals is presented in the study in [8]. The qualitative fault signature is exploited to develop a three-step diagnosis algorithm: 1) hypothesis generation; 2) signature generation; and 3) hypothesis refinement (we briefly describe these three steps below and refer the reader to [6] for further details of these algorithms).

Step 1) *Hypothesis Generation:* The first step in the qualitative fault-isolation algorithm, hypothesis generation, applies the backpropagation algorithm [6] to implicate parameters in the model that are consistent with the set of nonzero residuals $(\{\rho^k\} \subseteq \rho : \rho^k = \pm)$.

[1] When parameter values are constant, a one-time value substitution is made at the first time step in the new mode.

This algorithm uses the TCG model to generate fault hypotheses defined as the two-tuple $\langle p, \lambda \rangle$.

Step 2) *Signature Generation:* Starting with the hypothesis set, $H = \{h = \langle p, \lambda \rangle\}$, we generate qualitative fault signatures, qfs_h for each hypothesis, $h \in H$ and all measured variables using a forward-propagation algorithm [6] on the TCG. The result is an assignment of $+$, 0, $-$, and ? (unknown) symbols for the zeroth, first, and higher order derivatives for all the plant variables. The propagation algorithm terminates when the signatures of a prespecified derivative level (this is typically chosen to be the order of the system). The relation between the derivative level, set of measured variables, and system diagnosability is presented in [19].

Step 3) *Hypothesis Refinement:* This process uses a progressive-monitoring scheme to compare the qualitative fault signatures of all the hypothesized faults against the symbols extracted from the measured signal using the signal-to-symbol transformation algorithm [6]. When inconsistencies occur, the corresponding fault hypothesis is dropped.

In noisy signals with complex dynamics, the second and higher order derivatives of a signal cannot be measured reliably. The hypotheses-refinement algorithm uses the progressive-monitoring scheme [6] to track the symbolic form of the magnitude and slope of the measured signals, ρ_h and ρ'_h, and compares them to the corresponding fault signatures. The first change in a measured signal ($\pm$ value) is compared to the first nonzero symbol in the corresponding fault signature. A mismatch results in the fault hypothesis being dropped from the hypothesis set; otherwise, the monitoring process continues. If continued monitoring produces a mismatch, higher order derivatives of the signature are propagated down to continue to match the current magnitude and slope values of the measured signal. If the match succeeds, the hypothesis is retained and the monitoring progresses, otherwise the hypothesis is dropped. The correctness of the progressive-monitoring scheme has been established using the Taylor series expansion of the measured signal [8].

2) Extending the Diagnosis Algorithms to Hybrid Systems: Two new situations have to be dealt with for hypothesis generation in the context of hybrid diagnosis: 1) Given the time of fault detection (t_{fd}) = Time of fault occurrence (t_{fo}) + Response time of fault detector (Δt_{fd}) [15], mode changes may occur in the system during the detection delay interval, $[t_{\text{fo}}, t_{\text{fd}}]$. The autonomous mode changes in this interval cannot be predicted correctly by the observer, because it lacks a correct system model after the fault occurs. Hence, the observer-estimated mode trajectory, i.e., the predicted system-mode sequence $\hat{Q}\ [t_{\text{fo}}, t_{\text{fd}}])$ may be different from the actual mode trajectory $(Q[t_{\text{fo}}, t_{\text{fd}}])$. This also implies that the current mode predicted by the observer may be different from the true current system mode. This issue needs to be resolved when generating the initial fault hypotheses. 2) After a mode change, the system model changes. This implies that the fault signature linking a parameter to the measured variables may change, and the qualitative residual vector in the current mode may not be the same as the residual vector in the mode in which the fault occurred, i.e., $\rho^k_{t_{\text{fd}}} \neq \rho^k_{t_{\text{fo}}}$. Therefore, we need a scheme to map $\rho^k_{t_{\text{fd}}}$ back through the mode changes that have occurred in $[t_{\text{fo}}, t_{\text{fd}}]$ to obtain $\rho^k_{t_{\text{fo}}}$. Similarly, the signature-generation algorithm applied to hybrid systems has to deal with two issues: 1) The system mode and, therefore, the model of the system may change during diagnostic analysis. Therefore, signatures have to be regenerated online as mode transitions occur. 2) Mode changes can cause discrete changes in system variables, and these changes can affect the fault signature. When analyzing mismatches between the fault signature qfs_h and symbolic value of the residuals $\underline{\rho}$, hybrid progressive monitoring has to distinguish between the situations when: 1) h is not the true fault, therefore, h should be dropped from the hypothesis set and 2) the hypothesized current mode $q_{\text{h current}}$ is incorrect, because autonomous mode changes have not been correctly predicted. In this case, do not eliminate h but find the true current mode of the plant. In the next two sections, we extend the hypothesis generation and refinement algorithms to address the above issues.

3) Hybrid Hypothesis Generation: Given that the fault may have occurred in a mode prior to the current mode hypothesized by the observer, the hybrid-hypothesis-generation algorithm reasons backwards, envisioning past system modes, and hypothesizing the fault may have occurred in one of the past modes. This backward reasoning is the roll-back process. An approach to roll back would start in the current observer-estimated mode and use the inverse of the preoperator defined in [20] repeatedly to determine all possible modes that the plant could have been in prior to the current time. This approach makes the computational complexity of the algorithm exponential in the number of nodes. However, Assumption 1 helps to significantly reduce the complexity of the backward search.

Assumption 1—Accurate Mode Tracking Under Nominal Conditions: The hybrid observer-estimated mode trajectory coincides with the actual system-mode trajectory until time-step k when the fault occurs, i.e., $\hat{Q}[0, k] = Q[0, k]$, $k \leq t_{\text{fo}}$.

Lemma 1—Backpropagation Across Modes: The mode in which the fault occurs q_f must belong to the observer-estimated mode trajectory, i.e., $q_f \in \hat{Q}[0, k]$, $k \leq t_{fo})$.

Proof: Applying Assumption 1, we can say that at the time of fault occurrence t_{fo}, the observer-estimated mode has to be the same as the actual system mode, i.e., $\hat{q}_f = q_f$. Therefore, $q_f \in \hat{Q}[0, t_{\text{fo}}]$. We can break up the observer-estimated mode trajectory at time-step k into two parts: $\hat{Q}[0, k] = \hat{Q}[0, t_{\text{fo}}] \cup \hat{Q}[t_{\text{fo}}, k]$. Since $q_f \in \hat{Q}[0, t_{\text{fo}}]$, it follows that $q_f \in \hat{Q}[0, k]$.

Lemma 1 informs us that the hypothesis-generation algorithm only needs to consider the observer-estimated mode trajectory. Since the mode in which the fault occurred is unknown, a roll-back process has to be incorporated into the hypothesis-generation algorithm.

Assumption 2—n-Diagnosable: The system is assumed to be n-diagnosable for some constant n.

This implies that the measurement deviations caused by the fault will be detected within n mode transitions from the

occurrence of the fault. The value of n is determined by offline analysis. If mode changes do not occur in fast succession, n is typically set to two. Assumption 2 states that we need to go back at most n modes in the observer-estimated mode trajectory to get to the mode in which the fault occurred.

How do we handle the problem of possible changes in the qualitative residual, ρ_t^k, $t_{\text{fo}} \leq t \leq t_{\text{fd}}$, due to mode changes? We define the qualitative inverse of the reset function γ^{-1} associated with the mode changes in the HBG. The reset function γ specifies the change in the state variables because of a mode change [12]. In our HBG modeling work [14], the reset function is defined in terms of state variables. If a mode change occurs at time-step k, the reset function computes the state vector in the new mode as $\underline{x}_k^- = \gamma(\underline{x}_{k-1}, q_{k-1}, q_k)$. The qualitative inverse of this function specifies how symbols get reset when going back from mode q_k to q_{k-1}, $\rho(\underline{x}_{k-1}) = \gamma^{-1}(\rho(\underline{x}_k), q_{k-1}, q_k)$, where $\rho(\underline{x}_{k-1})$, and $\rho(\underline{x}_k)$ represent the qualitative values for the deviations in the state variables at time $k-1$ and k, respectively.

Assumption 3—Inverse Reset Function: The reset function γ is assumed to be invertible.

As we go back in the observer-estimated mode trajectory, we apply the corresponding qualitative inverse reset function γ^{-1} on the symbols for the state variables in mode q_k to determine the symbols to be backpropagated in the previous mode q_{k-1}. In some cases, the inverse reset function may return more than one symbol due to the ambiguity in qualitative arithmetic. In such cases, all the symbols have to be backpropagated in the previous mode to generate candidates.

Assumption 4—Transients in State Variables: The slope of the transients in the state variables do not change sign within a mode between the time of fault occurrence and fault detection.

It is sufficient to backpropagate the symbols generated for the state variables in mode q_k to mode q_{k-1}. Assumption 4 states that these symbols cannot have changed sign.

The hybrid-hypothesis-generation algorithm using the roll-back process is summarized as Algorithm 1. Given the observer-estimated mode trajectory $\hat{Q} = [\hat{q}_1, \hat{q}_2, \ldots, \hat{q}_k]$, the backpropagation algorithm $B(q, \rho^k)$ [6] is applied to generate the hypotheses set $P = \{\langle p, \lambda \rangle\}$ in each of the previous modes, $\hat{q}_{k-1}, \hat{q}_{k-2}, \ldots, q_{k-n+1}$ when the system is assumed to be n-diagnosable. During this roll-back process, the qualitative inverse of the reset function (γ^{-1}) is applied to get the initial deviations in each mode. $\rho(\underline{x})$ is similarly derived for the state vector $\underline{x}$, i.e., γ^{-1} derives $\underline{x}_{k-1}$ from $\underline{x}_k$. The algorithm returns the hypothesis set

$$H = \{h_1, h_2, \ldots h_n\}; h_i = \langle q_i, p_i, \lambda_i \rangle, k - n + 1 \leq i \leq k.$$

Algorithm 1: Hybrid Hypothesis Generation
$H = \Phi, q = \hat{q}_k$
Find fault hypotheses in fault detected mode
$P = B(q, \{\rho^k\})$
$H = H \cup \langle P, q \rangle$
$\rho(\underline{x}_{k-1}) = \gamma^{-1}(\hat{\rho}(\underline{x}_k), \hat{q}_k, \hat{q}_{k-1})$
Roll-back n steps and create new hypotheses in each mode
For $i = 1 : n-1$
  $q = \hat{q}_{k-i}$
  $P = B(q, \rho(\underline{x}_{k-i}))$
  $H = H \cup \langle P, q \rangle$
  $\rho(\underline{x}_{k-i}) = \gamma^{-1}(\hat{\rho}(\underline{x}_{k-i+1}), \hat{q}_k, \hat{q}_{k-i+1})$
EndFor
Return H

4) Hybrid-Hypothesis Refinement: The hybrid signature generation and the progressive-monitoring algorithms for hybrid diagnosis are presented below.

Hybrid Signature Generation: The effects of roll back applied in the hypothesis-generation process have to be reversed to ensure that the fault signatures generated for progressive monitoring correspond to the current mode of system operation. This requires a quick roll-forward process, where we start from the mode in which the fault is hypothesized and work forward through possible mode changes from the point of fault occurrence t_{fo} to the current time t. After fault occurrence, the system model cannot predict autonomous mode changes correctly. This implies that the roll-forward process needs to consider all possible mode-change sequences. However, Lemma 2 helps establish that the order in which the mode transitions occur does not impact the fault signature generated in the hypothesized current mode.

Lemma 2—Sequence of Mode Transitions: Any permutation of the k discrete-events $S = [\sigma_1, \sigma_2, \ldots, \sigma_k]$, $\sigma_i \in \Sigma$ applied to mode q_i will transition the hybrid system to the same mode q_{i+k}.

Each mode q_i of the system is defined by the state of the switched junctions (on or off) of the HBG model of the system and not the order in which the current values were attained. For each hypothesized fault $h_i = \langle q_i, p_i, \lambda_i \rangle$, all the controlled transitions from S are applied sequentially, starting from q_i to arrive at the hypothesized current mode $\hat{q}_i(t)$. Progressive monitoring is initiated from this mode, and Lemma 2 implies that any autonomous transitions that occurred between the controlled transitions can be applied later, i.e., it is not required to apply mode changes in the exact order in which they occurred to continue progressive monitoring in later modes. The qualitative form of the reset function γ is applied simultaneously to determine the change in the state vector at the point of mode transition, i.e., $\underline{x}_k^- = \gamma(\underline{x}_{k-1}, q_{k-1}, q_k)$. Assumption 4 clarifies that γ is applied to the first nonzero symbol in the signature of the state variables to derive the initial symbols in the next mode.

The assumed mode $\hat{q}_i(t)$ is the true current mode only if no autonomous mode changes have occurred during the $[t_{\text{fo}}, t]$ time interval. The progressive-monitoring algorithm for continuous systems [6] starts with $\hat{q}_i(t)$ as the assumed current mode, and fault signatures qfs_h generated using the TCG model in this mode are compared against the measured symbols $\{\rho^k\}$.

The hybrid-signature-generation algorithm, summarized as Algorithm 2, starts from the hypothesized fault mode q_h for each fault hypothesis $h \in H$ applies all controlled transitions observed in the interval $[t_{\text{fo}}, t]$ and returns the qualitative fault signature (qfs_h) for all faults h. $\rho(x)$ represents the initial deviations in the state variable, and $\Theta(\rho(\underline{x}))$ is the first nonzero derivative symbol for state variable $\underline{x}$. δ is the mode-transition function specified in the system HBG model. F represents

the forward-propagation (signature generation) algorithm presented in [6]. Further savings can be obtained during roll forward even before we get to progressive monitoring.

Algorithm 2: Hybrid Signature Generation

For $\forall h \in H$
  $\text{qfs}_k = F(q_k, \lambda_k)$
  For $\forall \sigma_i \in \Sigma_C$
    $\alpha = \Theta(\rho(\underline{x}))$
    $q_{\text{k current}} = \delta(q_k, \sigma_i)$
    $\text{qfs}_k = F(\lambda_k, q_{\text{k current}}) \cup F(\alpha, q_{\text{k current}})$
  EndFor
EndFor

Assumption 5—Discontinuity Detection: The occurrence of a discontinuity in a measured signal will be detected by the fault detector in the mode in which it occurs.

Assumption 5 implies that if a fault causes a discontinuity in a measured signal, the fault will be detected in the mode of fault occurrence. In other words, if a discontinuity is detected in a residual signal, the roll-back and roll-forward steps for hybrid diagnosis need not be applied. Furthermore, if a fault is detected in a mode that is different from the one in which it occurred, this fault cannot cause a discontinuous change in the measured variables. Therefore, we look at hypotheses, h generated in previous modes. If they predict a discontinuity in a measured variable in the hypothesized mode of occurrence q_h, then h is dropped. The reasoning is that if h is indeed the true fault, then the resultant discontinuity in the measured variable in mode q_h would have been detected immediately. For all remaining hypotheses, we start the hybrid-progressive-monitoring scheme described next.

Hybrid Progressive Monitoring: As discussed earlier, progressive monitoring is applied in continuous modes of system operation to determine if predicted fault signatures match the deviations observed in the measured residuals. A match implies support for the hypotheses, whereas a mismatch may imply the hypothesis should be dropped or the hypothesized mode does not match the system mode.

In case of a mismatch, we consider all the possible autonomous events in the plant Σ_A and spawn $|\Sigma_A|$ new fault hypotheses H_{new} (one for each autonomous transition in $\sigma_a \in \Sigma_A$) to generate the updated hypothesis set $H = H \cup H_{\text{new}} - h$. Each hypothesis $h_a \in H_{\text{new}}$ is formed from $h = \langle p, \lambda, q \rangle$ by applying the autonomous event $\sigma_a \in \Sigma_A$ to q, i.e., derive $q_a = \delta(q, \sigma_a)$ and $h_a = \langle p, \lambda, q_n \rangle$. For each new fault hypothesis h_a, the hybrid-signature-generation algorithm is rerun in the hypothesized current mode q_a to generate a new set of qualitative fault signatures for the hypothesis. The progressive-monitoring scheme is applied to the modified hypotheses set H.

If a mismatch persists for any of the new hypotheses, we hypothesize more autonomous transitions. For example, if the signature for h_a does not match the measurements, then we spawn $|\Sigma_{A-a}|$ new hypotheses, where Σ_{A-a} is the set of all possible autonomous transitions that do not include a. Also, if a controlled event (σ_{new}) occurs during the progressive monitoring scheme, we rehypothesize the current mode for each fault hypothesis by applying this controlled-mode change $(q_{\text{h current}} = \delta(q_h, \sigma_{\text{new}}))$ and recompute the fault signatures before continuing the progressive monitoring. Algorithm 2 is applied to all the new hypotheses, and progressive monitoring continues. To reduce the exponential growth in hypotheses, we use the following scheme when spawning new hypotheses. For each hypothesis h, we keep a list of autonomous transitions hypothesized so far, $\hat{\Sigma}_h$. When spawning new hypotheses, we consider only autonomous transitions in the set $\Sigma_A - \hat{\Sigma}_h$.

For two hypotheses, $h_1 = \langle p_1, \lambda_1, q_1 \rangle$ and $h_1 = \langle p_2, \lambda_2, q_2 \rangle$ with $p_1 = p_2$, $\lambda_1 = \lambda_2$, $q_1 = q_2$ and $q_{h_{1_{\text{current}}}} = q_{h_{2_{\text{current}}}}$, then only one of the two hypotheses is considered for future analysis of fault candidates. In other words, if the starting points for the two hypotheses are the same and the current mode reached when performing tracking for hypothesis refinement is the same, the sequence of mode changes executed to get to this point is irrelevant. The justification is similar to the one used to establish Lemma 2.

The hypothesizing of autonomous events is terminated when the number of events (autonomous and controlled, i.e., $|\Sigma_C| + |\hat{\Sigma}_h|$) between the mode in which the fault is hypothesized q, and the hypothesized current mode $q_{\text{h current}}$ in the behavior trajectory for any hypothesis h exceeds n, the diagnosability of the system. If a mismatch exists at this point, then h is dropped from the hypotheses set.

The hybrid-progressive-monitoring algorithm is presented as Algorithm 3. σ_c is a controlled event that occurs during progressive monitoring. α, ρ, Θ, and F are as defined for the hybrid-signature-generation algorithm. $P(h)$ represents the results of continuous-mode progressive monitoring on hypothesis h.

Algorithm 3: Hybrid Progressive Monitoring

For $\forall h \in H$, $h = \{p_h, \lambda_h, q_h\}$
  If Σ_c
    $q_{\text{current}} = \delta(q_h, \Sigma_c)$
    $\text{qfs}_h = F(p_h, \lambda_h, q_{\text{h current}})$
    $\alpha = \Theta(\rho(x))$
    $\text{qfs}_h = F(p_h, \lambda_h, q_{\text{h current}}) \cup F(\alpha, \lambda_h, q_{\text{h current}})$
  EndIf
  If $(\neg P(h))$
    If $(|q_{\text{h current}} - q_h| = n)$
      $H = H - h$
    Else
      $H = H - h$
      For $\sigma_a \in \Sigma_a$
        $q_{\text{h current}} = \delta(q_h, \sigma_a)$
        $\text{qfs}_h = F(p_h, \lambda_h, q_{\text{h current}})$
        $\alpha = \Theta(\rho(\underline{x}))$
        $\text{qfs}_h = F(p_h, \lambda_h, q_{\text{h current}}) \cup F(\alpha, \lambda_h, q_{\text{h current}})$
      EndFor
    EndIf
  EndIf
EndFor

5) Quantitative Fault Isolation and Identification: We use a least squares parameter-estimation technique to further refine the fault hypothesis set and to estimate the fault magnitude. This helps overcome the limited discriminatory capabilities of the qualitative progressive-monitoring scheme [8]. Furthermore, determining the parameter magnitude is essential to implementing fault-adaptive control strategies.

The standard least squares estimate for parameter estimation is expressed in input–output form as

$$\Phi\Theta = Z, Z = \begin{bmatrix} y_{k+1} \\ y_{k+2} \\ \vdots \\ y_{k+n} \end{bmatrix} \Phi = \begin{bmatrix} u_{k+1} & y_{k+1} \\ u_{k+2} & y_{k+2} \\ \vdots & \vdots \\ u_{k+n} & y_{k+n} \end{bmatrix}$$

where θ denotes the entire parameter vector for the system and Z and Φ represent matrices that capture the measured (output) variable vector $(\underline{y})$ and input vector $(\underline{u})$. The least squares estimate of the parameter value is $\theta_{\text{est}} = (\Phi^T\Phi)^{-1}\Phi^T Z$ [1].

The single-fault assumption implies only one parameter of the model is unknown. This helps reduce the complexity of the estimation task to single-parameter estimation and helps mitigate the need for the least square estimation technique to have persistent excitation of signals for accurate estimation [3]. At the point where we switch from qualitative progressive monitoring to parameter estimation, there is a hypothesized current mode $q_{\text{h current}}$ associated with each fault hypothesis $h \in H$. For each hypothesis h, the symbolic form of the parameterized input–output model of the system in derived in the corresponding mode $q_{\text{h current}}$ from the HBG model of the system using Mason's gain rule [15], [16]. Except for the parameter associated with the hypothesis p_h, the nominal numeric values for all other parameters are substituted, leaving a system of input–output equations with one unknown variable.

Assumption 6—First-Order Relation Between Input–Output Model Parameters and HBG Parameter: The parameters of the IOE model $\underline{\theta}$ are first-order polynomial functions of any one parameter p of the HBG model. We express $\underline{\theta}$ as $\underline{K}_1 p + \underline{K}_0$, where $\underline{K}_1$ and $\underline{K}_0$ are vectors. Reformulating the least squares problem, we obtain the optimal estimate for the faulty parameter p as

$$p = \frac{K_1^T\Phi^T Z - K_1^T\Phi^T K_0 - K_0^T\Phi^T K_1 + Z^T\Phi K_1}{2 \times K_1^T\Phi^T K_1}.$$

This technique is applied to each of the fault hypotheses $h \in H$. The accumulated input and output data from the plant form the matrices Φ and Z. At each new time step, an additional row of data is added to Φ and Z.

If a controlled mode change, $\sigma_c \in \Sigma_c$ occurs during the parameter-estimation process, we have to restart the parameter-estimation algorithm in the new mode for all $h \in H$. This involves recalculating $\underline{K}_0$ and $\underline{K}_1$ in the new mode $\delta(q_{\text{h current}}, \sigma_c)$. The Φ and Z matrices are reset, and the data are accumulated in the new mode. Since autonomous mode transitions cannot be predicted after the fault occurrence, we make one further assumption:

Assumption 7—No Autonomous Changes During Parameter Estimation: No autonomous mode changes occur in the plant during the parameter-estimation process.

The hybrid-parameter-estimation algorithm is implemented to run as a batch process, but in the future, to make the computation more efficient, we will look at schemes for recursive parameter-estimation [21] and subspace [22] based methods. Once the estimates for each $h \in H$ have been computed, we use the estimated values to compute the mean-square error for $y - \hat{y}$. The parameter h with the least estimated error is assumed to be the true fault.

6) Space and Time Complexity for Hybrid Diagnosis: For the hybrid-hypotheses-generation algorithm, we need to store the mode trajectory, the TCG for the mode that the backpropagation is being performed, and the symbols and candidates generated in that mode. The mode trajectory can be represented as n integers, where the system is assumed to be n-diagnosable. The TCG requires $\nu + 2e$ integers, where v is the number of vertices and e is the number of edges. Since we need to store symbols for all variables in the TCG and this is equal to the number of vertices ν in the TCG, we need 2ν integers. Finally, each candidate can be stored as an integer value. The number of candidates in each mode in the worst case would be the twice the number of parameters p in the TCG. The total number of candidates can be $n \times 2 \times p$. Adding up the components, the total space complexity for the hybrid hypotheses generation is computed as a polynomial function $P(n, \nu, e, p)$.

For the time complexity, two parts need to be considered, backpropagation within a mode and across modes. Since the backpropagation is a traversal through the TCG, its time complexity is give by a polynomial function $P(\nu + e)$. The roll-back process evaluates the qualitative inverse reset function for each state variable. Since this expression is assumed to be a linear invertible polynomial function, the time complexity of the roll-back process is $P(|\underline{x}|)$, where $|\underline{x}|$ is the number of state variables. Adding up, the time complexity of the hybrid-hypotheses-generation algorithm is a polynomial function $P(|\underline{x}|, \nu, e)$.

The space and time complexities of the hybrid hypotheses refinement are the sum of the corresponding complexities for the signature-generation and progressive-monitoring algorithms. Signature generation is the inverse of the hybrid-hypothesis-generation algorithm except that the TCG is traversed forward d times, where d is the specified order of the signature. Hence, the space complexity of signature generation is given by a polynomial function $P(n, \nu, e, p, d)$, where n is the diagnosability of the system, ν is the number of vertices, e is the number of edges, and p is the number of parameters in the TCG. Similarly, the time complexity of the signature-generation algorithm is given by $P(|\underline{x}|, \nu, e, d)$.

The time complexity of the progressive-monitoring algorithm is given by the polynomial function $P(|\underline{y}|, d)$, where $|\underline{y}|$ is the number of output variables and d is the order of the generated signature. There is no additional space requirement for the progressive-monitoring algorithm. Therefore, the total space complexity for the hybrid-hypotheses-refinement algorithm is $p(n, \nu, e, p, d)$, and the time complexity is $P(|\underline{x}|, |\underline{y}|, \nu, e, d)$.

In order to run the parameter estimation, we need to store the Z, Φ, K_0, and K_1 matrices. The size of Z matrix is $|\underline{y}| \times n$, where $|\underline{y}|$ is the number of measured variables and n is the size of the regression window. The size of the Φ matrix is $(|\underline{y}| + |\underline{u}|) \times n$, where $|\underline{u}|$ is the number of input variables. Since each element of these matrices is a real number, the space complexity of the parameter estimation is $P(|\underline{u}|, |\underline{y}|, |K_0|, |K_1|, n)$ real numbers. The time complexity is governed by the revised equation for parameter estimation, which involves 16

TABLE II
FUEL-SYSTEM EXPERIMENTS WITH DIFFERENT FAULT MAGNITUDES AND NOISE LEVELS

Faults	Performance Parameters								
Fault Type	Fault Magnitude	Fault Detection Time (seconds)		Fault Isolation Time (seconds)		Initial/Final Candidate Set number/number		Parameter Estimation Error (percent)	
		2%	3%	2%	3%	2%	3%	2%	3%
LTT-Pump Efficiency Drop	33%	422	555	225	398	14/3	13/4	2.19	5.43
	60%	182	183	144	240	13/4	13/4	1.28	1.79
	80%	134	134	124	197	13/4	13/5	0.88	1.49
RWT-Pump Efficiency Drop	33%	117	285	170	211	13/4	13/4	2.15	6.11
	60%	83	93	139	183	13/4	13/4	1.52	1.67
	80%	5	5	55	106	13/3	13/4	0.68	0.68
RLCV-Block (Valve)	× 1.5	63	65	97	103	25/2	25/2	0.62	0.5
	× 1.75	51	63	58	86	25/2	23/1	0.28	0.46
	× 2.0	51	52	46	79	23/1	25/2	0.2	0.2
Leg21 Pipe Block	× 1.5	99	100	136	350	14/3	14/3	1.58	1.65
	× 1.75	95	95	90	303	14/2	14/3	0.78	1.57
	× 2.0	93	93	76	202	14/2	14/2	0.19	0.34

operations (one assignment, one inverse, three additions or subtractions, and 11 multiplications). Since the operands involved in this operation are of $P(|\underline{u}|, |\underline{y}|, |K_0|, |K_1|, n)$ size, the time complexity of the parameter estimation is also $P(|\underline{u}|, |\underline{y}|, |K_0|, |K_1|, n)$ operations.

IV. EXPERIMENTAL RESULTS

We demonstrate the effectiveness of the hybrid-diagnosis approach by conducting a set of experiments on a real example—the fuel-transfer system of a fighter aircraft illustrated in Fig. 1 and described in Section II. Fuel transfer from the wing and transfer tanks to the feed tanks was based on a controller sequence provided by the system designers. Table II illustrates the results of a set of diagnosis experiments that we ran for the faults described in Section II. In this paper, we varied the fault size and amount of measurement noise in the signal. In designing the experiments, we had to set parameters for the Kalman filter, fault detector, and symbol generator (a detailed description of these parameters and their values are presented in [23]). A high-fidelity simulator was used to generate the data for the experimental runs, and measurement noise was added to the simulated data. Ten runs were conducted for each noise level and fault size and the mean values of the detection and isolation times, the candidates generated by qualitative fault isolation, and the parameter-value error after least squares estimation are reported in the table. The results indicate that as the noise levels in the measurements increase and the fault magnitudes become smaller, the time to detection, isolation, and identification (i.e., parameter estimation) increase and the parameter-estimation error increases.

To provide a more detailed illustration of the working of the diagnosis algorithm, we list important steps from the point of fault detection to the end of parameter estimation. The particular scenario deals with a partial block that occurs in the connecting pipe from the manifold to the LFT, i.e., R21.

The fault, represented as a doubling of the pipe resistance, occurs at time step $= 350$. In this paper and other experiments, the pressures at the output of the six tanks plus the pressure at the transfer manifold are the measured variables. In the first 300 time steps, the LTT and RWT paths and, then, the LTT and RTT pumps are activated (see Table III) resulting in transfer of fuel from all four tanks to the two feed tanks through the common pressure manifold. The fault occurs at time-step 350, but because of the noise levels, our statistical detection technique detects the fault with sufficient confidence at time-step 481 (measurement noise with zero mean and variance $=$ 3% of mean signal value was introduced in this experiment). In between, a mode change is triggered by the controller, i.e., the RTT path is turned OFF (this is illustrated in Table III and Fig. 5). The TCG log appears on the left bottom of the screen in Fig. 6. In the transfer-manifold-pressure plot (top right), the deviation of the observer-estimated value (solid line) from the sensor measurement (dots) can be observed. The TCG log shows the initial list of 14 probable fault candidates generated using the TCG model. At the next time step, the reported discontinuity in the transfer-manifold pressure reduces the number of candidates to ten. Four of the initial candidates whose fault signatures are inconsistent with the discontinuity are dropped. Note that the roll-back and roll-forward mechanisms have to be applied because the fault is detected only after a mode change has occurred in the system. This is not known to the diagnoser, so it starts with multiple candidates from different modes. Many of these hypothesized candidates are eliminated when their signatures do not match the observed behaviors.

TABLE III
DETAILED TRACE OF DIAGNOSIS ALGORITHM FOR BLOCKING FAULT IN LEG 21

	Event	Fault Set
0-320	Number of mode changes Left & Right Wing and Transfer Tanks active	none
350	Fault Introduced in Simulation Resistance in Leg21 (Leg21.R) doubles	
477	Right Transfer Tank Pump off; All others on	
481	Fault Detected XMP: observed value + (above normal)	14 candidates
482	XMP = (+,0) : discontinuous change	11 candidates
529	Left Feed Tank Pressure = (0,−)	LeftLCV.R+, Leg21.R+, Leg210.R+, Leg26.R+
548	Left Wing Tank Pressure = (0,+)	LeftLCV.R+, Leg21.R+, Leg210.R+
557	Right Wing Tank Pressure = (0,+)	LeftLCV.R+, Leg21.R+, Leg210.R+
598	Right Transfer Tank Pump on; All pumps on	LeftLCV.R+, Leg21.R+, Leg210.R+
668	Right Transfer Tank Pump off; All others on	LeftLCV.R+, Leg21.R+, Leg210.R+
689	Mode change: Left Wing Tank empty	LeftLCV.R+, Leg21.R+, Leg210.R+
749	Mode change: Right Wing Tank empty	LeftLCV.R+, Leg21.R+, Leg210.R+
788	Parameter Estimation Started for [LeftLCV.R+ Leg210.R+ Leg21.R+]	LeftLCV.R error = 59.7; Leg210.R error = 24596 Leg21.R error = 51.5; Fault: Leg21.R – resistance change 99.9%
789	Right Transfer Tank Pump on	

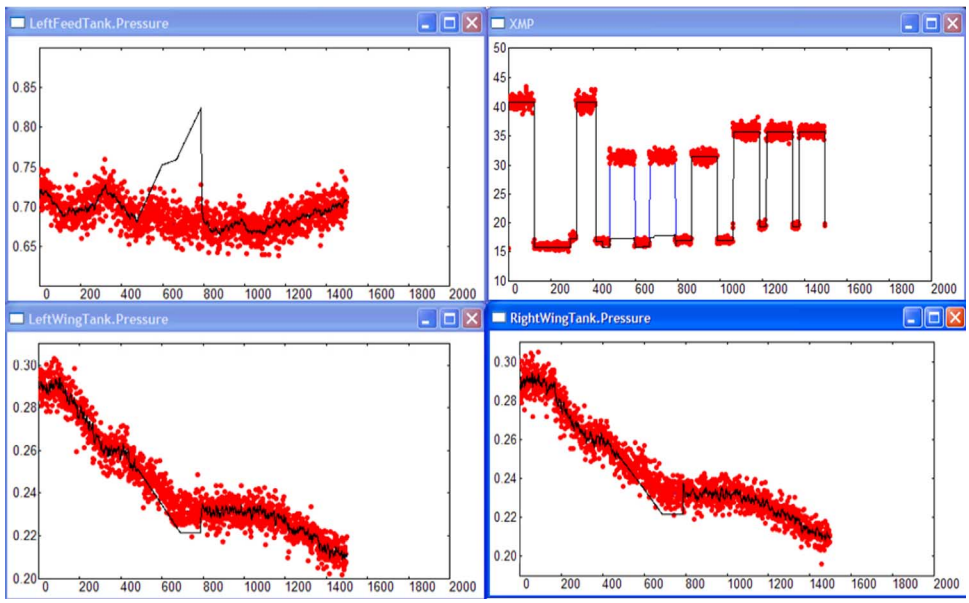

Fig. 5. Tracking system behavior before fault (up to time-step 350), from fault detection to fault estimation (time-steps 451–788), and beyond (up to time-step 1400).

At this stage, no more reduction is possible in the candidate set. The qualitative analysis is terminated, and the parameter-estimation algorithm is invoked at time-step 788 with three fault hypotheses. For each of the hypotheses in the list, the algorithm computes a parameter value that gives the least prediction error. The hypothesis with the smallest prediction error among all of the hypotheses is established as the fault candidate. The observer resets the value of the fault candidate parameter to its new value and restarts tracking of the faulty system. The results of the parameter-estimation scheme are shown in row 788 of

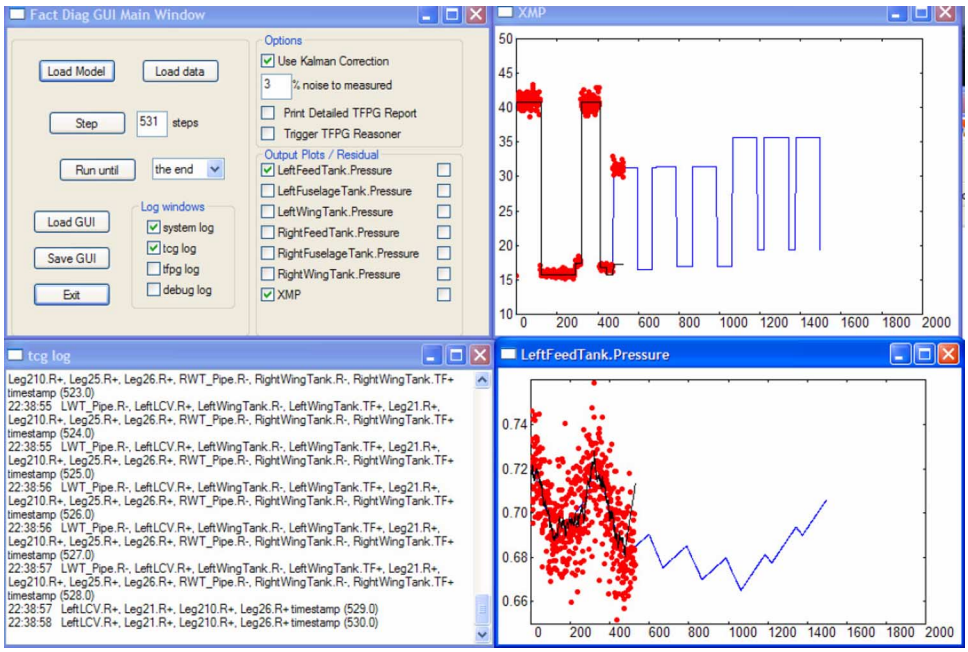

Fig. 6. Fault hypotheses at time-step 530. Manifold pressure (XMP) deviation detected at step 481, and LFT pressure deviation detected at step 529.

TABLE IV
FUEL-SYSTEM DIAGNOSABILITY

	WTP	WTR	TTP	TTR	SPR	FTP	FTR
WTP	-	No	Yes	Yes	Yes	Yes	Yes
WTR	No	-	Yes	Yes	Yes	Yes	Yes
TTP	Yes	Yes	-	No	Yes	Yes	Yes
TTR	Yes	Yes	No	-	Yes	Yes	Yes
SPR	Yes	Yes	Yes	Yes	-	Yes	Yes
FTP	Yes	Yes	Yes	Yes	Yes	-	No
FTR	Yes	Yes	Yes	Yes	Yes	No	-

WTP = Wing Tank Pump Efficiency; WTR = Wing Tank Resistance;
TTP = Fuselage Transfer Tank Pump Efficiency;
TTR = Fuselage Transfer Tank Resistance
SPR = Switched Pipe Resistance; FTP = Feed Tank Pump Efficiency
FTR = Feed Tank Resistance (FTR);
Yes = Diagnosable; No = Not Diagnosable

Table III. The candidate, pipe resistance for leg 21 (Leg21.R), shows the least error. Its value is set to the estimated value (twice its original value), and after this change, the observer again tracks the faulty system very well. This is illustrated in Fig. 5, where after fault-occurrence measured system behavior (band of dots) deviates considerably from expected system behavior (dark line between dots). But after time-step 788, the observer model is updated with the faulty parameter value (Leg21.R), and the observer correctly tracks the expected faulty system behavior. Note that the observer is turned OFF after the fault is detected to facilitate the isolation task.

Table IV summarizes the different fault classes that can be distinguished by our diagnosis algorithms. A Yes in row i and column j indicates that fault class i can be distinguished from fault class j. A No indicates that the current controller sequence and set of measurements are not sufficient to distinguish between the pair in question. From the table, we see that we cannot distinguish between tank-pump faults and tank-outlet-pipe resistance faults. However, this is true only for a pump-efficiency (TF$-$) decrease and pipe-resistance increase (R$+$). Since the pump efficiency cannot increase (no T_F+ fault), the pipe resistance decrease R$-$ (i.e., leaks) can be uniquely identified. All other classes of faults can be distinguished from one another. In some cases, the isolation may be achieved only after a mode change occurs.

In another work, we have applied this approach to advanced life-support-system subsystems, such as the water-recovery system for fault isolation, identification, and fault-adaptive control. A real-time version of our approach has also been applied to detect actuator faults in an R-MAX helicopter systems for a DARPA project demonstration [24]–[26]. In all of the examples, the system behavior includes complex nonlinearities and number of controlled- and autonomous-mode changes. The TCG used for qualitative diagnosis had nodes and edges in the order of hundreds. In the R-MAX demonstration, our diagnosis algorithm could detect and isolate actuator stuck-at faults in hard real time.

V. CONCLUSION

In this paper, we have presented an integrated approach for solving the tracking, fault detection, isolation, and identification tasks for hybrid systems. The novel contribution of this paper is the extension of continuous system MBD techniques to hybrid systems. These include the hybrid observer that combines an EKF and hybrid-automaton hybrid fault isolation that combines roll back and roll forward using qualitative-analysis schemes and least squares-based parameter estimation to complete fault isolation and identification. A key issue that we have demonstrated to the MBD community (DX) is the use of qualitative-reasoning techniques for robust diagnosis starting from raw measured data obtained from sensors. This is in contrast to lot of work within the DX community that assumes

symbolic data is already available for diagnostic analysis (e.g., see http://www.ksl.stanford.edu/dx/, http://www.laas.fr/DX04/). The key to developing the end-to-end system, starting with raw sensor data to quantitative parameter estimation was achieved by the use of statistical techniques for fault detection and symbol generation and innovative schemes that combine qualitative and quantitative methods that can be applied for online analysis.

This paper is motivated by the requirements of the fault-accommodation task in embedded systems, where diagnosis is performed online during system operation. The application of hybrid-diagnosis techniques to fault-adaptive control of embedded systems is presented elsewhere [27]. We have also demonstrated through time- and space-complexity analysis that our algorithms can be applied to online analysis in resource-constrained environments. In future work, we will look for situations where the switch from the qualitative fault isolation to quantitative parameter estimation can be performed in a more efficient manner.

ACKNOWLEDGMENT

The authors would like to thank E. Manders, G. Karsai, G. Simon, T. Szemethy, and N. Mahadevan for their help in designing and implementing components of the system. The authors would also like to thank the reviewers for their valuable comments. They have helped in significantly improving the quality of the presentation.

REFERENCES

[1] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker, 1988.
[2] W. Hamscher, L. Console, and J. De Kleer, *Readings in Model-Based Diagnosis*. San Mateo, CA: Morgan Kaufmann, 1992.
[3] P. M. Frank, A. Garcia, and B. Koppen-Seigler, "Modeling for fault detection versus modeling for control," *Math. Comput. Simul.*, vol. 53, no. 4–6, pp. 259–271, Oct. 2000.
[4] M. Sampath, R. Sengupta, S. Lafortune, and K. Sinnamohideen, "Failure diagnosis using discrete-event models," *IEEE Trans. Control Syst. Technol.*, vol. 4, no. 2, pp. 105–124, Mar. 1996.
[5] J. Lunze and J. Schroder, "Sensor and actuator fault diagnosis of systems with discrete inputs and outputs," *IEEE Trans. Syst., Man, Cybern.*, vol. 34, no. 2, pp. 1096–1107, Apr. 2004.
[6] P. J. Mosterman and G. Biswas, "Diagnosis of continuous valued systems in transient operating regions," *IEEE Trans. Syst., Man, Cybern.*, vol. 29, no. 6, pp. 554–565, Nov. 1999.
[7] R. J. Patton, P. M. Frank, and R. N. Clark, *Issues of Fault Diagnosis for Dynamic Systems*. London, U.K.: Springer-Verlag, 2000.
[8] E. Manders, S. Narasimhan, G. Biswas, and P. J. Mosterman, "A combined qualitative/quantitative approach to efficient fault isolation in complex dynamic systems," in *Proc. Symp. SafeProcess*, Budapest, Hungary, 2000, pp. 512–517.
[9] M. W. Hofbaur and B. C. Williams, "Mode estimation of probabilistic hybrid systems," in *Proc. 5th Int. Workshop HSCC*, Stanford, CA, 2002, pp. 253–266.
[10] R. Dearden and D. Clancy, "Particle filters for real-time fault detection in planetary rovers," in *Proc. 13th Int. Workshop DX*, Semmering, Austria, 2002, pp. 1–6.
[11] X. Koutsoukos, F. Zhao, H. Haussecker, J. Reich, and P. Cheung, "Fault modeling for fault monitoring and diagnosis of sensor-rich hybrid systems," in *Proc. IEEE CDC*, Orlando, FL, 2001, pp. 793–801.
[12] R. Alur *et al.*, *Hybrid Automata—An Algorithmic Approach to Specification and Verification of Hybrid Systems*. Berlin, Germany: Springer-Verlag, 1994, pp. 209–229.
[13] D. Karnopp, D. L. Margolis, and R. C. Rosenberg, *System Dynamics: Modeling and Simulation of Mechatronic Systems*, 3rd ed. New York: Wiley, 2000.
[14] P. J. Mosterman and G. Biswas, "A theory of discontinuities in physical system models," *J. Franklin Inst.*, vol. 335, no. 3, pp. 401–439, Apr. 1998.
[15] S. Narasimhan, "Model-based diagnosis of hybrid systems," Ph.D. dissertation, Dept. Electr. Eng. Comput. Sci., Vanderbilt Univ., Nashville, TN, 2002.
[16] S. J. Mason, "Feedback theory: Some properties of signal flow graphs," *Proc. IRE*, vol. 41, no. 9, pp. 1144–1156, Sep. 1953.
[17] A. Gelb, *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1979.
[18] P. J. Mosterman and G. Biswas, "Behavior generation using model switching: A hybrid bond graph modeling technique," *Trans. Soc. Simul.*, vol. 27, no. 1, pp. 177–182, 1995.
[19] S. Narasimhan, P. J. Mosterman, and G. Biswas, "A systematic analysis of measurement selection algorithms for fault isolation in dynamic systems," in *Proc. 9th Int. Workshop DX*, Cape Cod, MA, 1998, pp. 94–101.
[20] X. Koutsoukos *et al.*, "Supervisory control of hybrid systems," *Proc. IEEE*, vol. 88, no. 7, pp. 1026–1049, Jul. 2000.
[21] I. Landau, "Unbiased recursive identification using model reference adaptive techniques," *IEEE Trans. Autom. Control*, vol. AC-21, no. 2, pp. 194–202, Apr. 1976.
[22] M. Viberg, "Subspace-based methods for identification of linear time-invariant systems," *Automatica*, vol. 31, no. 12, pp. 1835–1851, Dec. 1995.
[23] G. Biswas *et al.*, "A robust method for hybrid diagnosis of complex systems," in *Proc. 5th IFAC Symp. SafeProc*, Washington, DC, 2003, pp. 1125–1130.
[24] S. Abdelwahed *et al.*, "Online hierarchical fault-adaptive control of advanced life support systems," presented at the Int. Conf. Environmental Systems, Colorado Springs, CO, 2004, Paper 2004-01-2441.
[25] G. Biswas *et al.*, "Online model-based diagnosis to support autonomous operation of advanced life support system," *Habitat.: Int. J. Human Support Res.*, vol. 10, no. 1, pp. 21–38, 2004.
[26] G. Karsai and G. Biswas, "Fault-adaptive control technology," in "DARPA SEC Final Report," Vanderbilt Univ., Nashville, TN, 2004. (Agreement: F33615-99-C-3611).
[27] G. Karsai, G. Biswas, S. Narasimhan, T. Pasternak, and T. Szemethy, "Towards fault-adaptive control of complex dynamic systems," in *Software-Enabled Control: Information Technologies for Dynamical Systems*, T. Samad and G. Balas, Eds. Piscataway, NJ: IEEE Press, 2002, pp. 347–368.

Sriram Narasimhan received the B.E. degree in computer science and the M.S. degree in economics from the Birla Institute of Technology and Science, Pilani, India, in 1995 and the M.S. degree in computer science and the Ph.D. degree in electrical engineering and computer science from Vanderbilt University, Nashville, TN, in 1998 and 2002, respectively.

From 2002 to 2005, he was a Computer Scientist with the QSS Group Inc., working as a Contractor with the NASA Ames Research Center. He has worked as a Summer Intern and a Consultant at the Palo Alto Research Center (formerly known as Xerox PARC), where he developed a model-based diagnosis engine for copiers. He is currently a Project Scientist with the University of California, Santa Cruz, working as a Contractor at the NASA Ames Research Center, Moffett Field, CA, where he is Technical Lead for the project to develop the HyDE model-based diagnostic engine. His current research interests include model-based diagnosis and recovery for hybrid and stochastic systems. He has coauthored several papers in these areas.

Gautam Biswas (S'82–M'83–SM'91) received the Ph.D. degree in computer science from the Michigan State University, East Lansing.

He is currently a Professor of computer science and engineering and a Senior Research Scientist with the Institute for Software Integrated Systems in the Electrical Engineering and Computer Sciences Department, Vanderbilt University, Nashville, TN. He conducts research in intelligent systems with primary interests in hybrid modeling, simulation, and analysis of complex embedded systems and their applications to diagnosis and fault-adaptive control. As part of this work, he is working on fault-adaptive control of fuel-transfer systems for aircraft and unmanned combat air vehicles and advanced life-support systems for NASA. He is also initiating new projects in distributed monitoring and condition-based maintenance. In other research projects, he is also involved in developing simulation-based environments for learning and instruction. His research is currently supported by funding from the National Aeronautics and Space Administration, the National Science Foundation, and Boeing PhantomWorks. He is an Associate Editor for the *Journal of Applied Intelligence*.

Dr. Biswas is an Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS. He has served on the Program Committee of a number of conferences. He is a member of the Association for Computing Machinery, AAAI, and the Sigma Xi Research Society.