A Qualitative Event-based Approach to Continuous Systems Diagnosis

Matthew J. Daigle *Member, IEEE*, Xenofon D. Koutsoukos *Senior Member, IEEE*, and Gautam Biswas *Senior Member, IEEE*

***Abstract*—Fault diagnosis is crucial for ensuring the safe operation of complex engineering systems. Although discrete-event diagnosis methods are used extensively, they do not easily address parametric fault isolation in systems with complex continuous dynamics. This paper presents a novel event-based approach for diagnosis of abrupt parametric faults in continuous systems, based on a *qualitative* abstraction of measurement deviations from the nominal behavior. From a continuous model of the system, we systematically derive dynamic fault signatures expressed as event-based fault models, which are used, in turn, for designing an event-based diagnoser of the system and determining system diagnosability. The proposed approach is applied to a subset of the Advanced Diagnostics and Prognostics Testbed, which is representative of a spacecraft's electrical power system. We present experimental results from the actual testbed, as well as detailed simulation experiments that examine the performance of our diagnosis algorithms under different fault magnitudes and noise levels.**

***Index Terms*—model-based diagnosis, discrete-event systems, electrical power systems**

I. Introduction

FAULT diagnosis is crucial for ensuring the safe operation of complex engineering systems. Faults and degradations need to be quickly identified so corrective actions can be taken and catastrophic situations avoided. Diagnosis approaches can be categorized along several dimensions, such as model-based vs. signal-driven, online vs. offline, and continuous vs. discrete. Discrete-event system (DES) methods are an important framework for event-driven diagnosis in safety-critical systems, since they comprise a well-developed theory that allows for systematic construction of computationally efficient online diagnosers.

Existing DES diagnosers [1]–[4] are designed as extended observers that estimate the system state under nominal and faulty conditions. Although these methods have been used in many practical diagnosis applications [1], [5]–[7], they are very hard to develop for systems with complex continuous dynamics. Quantizing the continuous behavior using a finite set of states and events results in large, nondeterministic models that degrade the performance and increase the computational requirements of the diagnosis algorithms [8]–[10]. In the presence of faults, these models become increasingly complex, and deriving such models for different fault magnitudes may become computationally intractable.

In contrast to traditional DES approaches, this paper presents a novel approach to constructing DES diagnosers for isolating single, abrupt faults in continuous systems, based on a *qualitative* abstraction of the measurement deviations from the nominal behavior. The approach extends TRANSCEND [11], a model-based methodology for fault diagnosis in continuous systems, based on *fault signatures*, a qualitative representation of fault transients. We enhance TRANSCEND by incorporating temporal orderings of measurement deviations as diagnostic information, known as *relative measurement orderings*, which increases the discriminatory power of the measurements, allowing for faster, more efficient fault isolation [12]. Measurement orderings provide advantages for many classes of systems, including electrical systems where an accurate dynamical model can be developed, distributed mechanical systems such as formations of robots [12], and chemical and biological processes with slow dynamics [13]. Further, we formalize the diagnostic information into an event-based framework to enable systematic diagnosability analysis and diagnoser design. We extend preliminary results reported in [14] by developing the diagnoser design through a formal composition operator, introducing diagnosability and showing its relation to the event-based diagnoser, and including a comprehensive case study.

We demonstrate and experimentally verify our diagnosis approach on the Advanced Diagnostics and Prognostics Testbed (ADAPT) [15], deployed at NASA Ames Research Center. ADAPT represents the functionality of a spacecraft's electrical power system, which exhibits complex nonlinear behaviors and is prone to many different faults. Therefore, ADAPT serves as a challenging testbed to verify diagnosis methodologies for electrical power systems. To experimentally validate our approach, we consider a subset of ADAPT that includes a single battery discharging to two DC loads.

The contributions of the paper center on: (*i*) a method for systematically constructing event-based *fault models*, using, for each fault, a finite automaton that captures all possible sequences of measurement deviations, (*ii*) diagnosability analysis of systems and design of event-based diagnosers, (*iii*) a spectrum of diagnoser implementations that trade off space and time efficiency, (*iv*) experimental results on the ADAPT testbed, and (*v*) detailed simulation experiments investigating the effects of sensor noise and different fault magnitudes on

Manuscript received January 24, 2008; revised August 27, 2008. This work was supported in part by NSF grant CNS-0615214, NASA USRA grant 08020-013, and NASA NRA grant NNX07AD12A.

M. Daigle is with the University of California, Santa Cruz, at NASA Ames Research Center, Moffett Field, CA 94035, USA (email: matthew.j.daigle@nasa.gov). X. Koutsoukos and G. Biswas are with the Institute for Software Integrated Systems, Department of Electrical Engineering and Computer Science, Vanderbilt University, 2015 Terrace Place, Nashville, TN 37235, USA (email: xenofon.koutsoukos@vanderbilt.edu; gautam.biswas@vanderbilt.edu).

our diagnosis scheme.

The paper is organized as follows. Section II describes related work in event-based diagnosis. Section III overviews our diagnosis approach. Section IV presents the formulation of our qualitative fault isolation methodology. Section V develops the event-based fault models and formalizes diagnosability. Section VI discusses the event-based diagnoser and its construction, and Section VII describes the spectrum of implementations. Section VIII presents the case study. Section IX concludes the paper.

II. Related Work

We formulate our approach to diagnosis of continuous systems in a DES framework. DES diagnosis methods are based on observing system events and making inferences about the system state. Ideally, a sequence of observable events can be mapped back to a single consistent fault. Most DES approaches construct diagnosers from the system model, which function as extended observers that provide estimates of the system state under both nonfaulty and faulty conditions [1], [3], [4]. Our diagnoser is a special case of traditional DES diagnosers, in that it does not track nominal system behavior, but is focused on isolating faulty conditions by tracking system behavior after fault detection. More importantly, the diagnoser, in contrast to most DES approaches where the event-based models are hand-created, is systematically generated from the continuous model of the system, which greatly reduces the burden of the modeling task.

Applying traditional DES approaches to continuous systems requires abstraction of the continuous dynamics. Timed DES methods [8], [9], [16]–[18] typically employ a quantization of the continuous state-space to produce a DES model of the system. This form of quantization often results in state explosion, and the resulting model is inherently nondeterministic. As a result, the diagnosis algorithms are more complex and less efficient. We propose a qualitative abstraction approach that abstracts the measurements with respect to nominal behavior. Three qualitative states are defined for each measurement: above nominal, at nominal, and below nominal. These states are further refined into magnitude and slope deviations to capture the dynamics of system behavior. Measurement deviations imply the presence of a fault and form the observable event set for our approach.

The proposed abstraction method uses a robust observer based on the continuous model of the system to track nominal behavior [11]. System tracking and fault isolation are separated, so the diagnoser tracks only the faulty behavior as given by the measurement deviations. Therefore, faults can be detected very quickly, unlike in quantization approaches, where the fault detection time will depend on the level of quantization.

Timed event traces in systems can also be modeled using chronicles, which are patterns of event traces that include temporal constraints and represent the possible timed evolutions of the system behaviors. Chronicles capture direct symptom to fault knowledge, so they are very efficient for online diagnosis [19], [20]. As events occur in the system, they are matched against known chronicles to determine which faults are present. From our diagnosis model, we derive fault signatures and measurement orderings. We extract from this information an event-based model of the system that represents only faulty behavior. Like chronicles, the event-based fault models represent direct symptom to fault knowledge. Modeling the timed event traces that result from faults, however, is infeasible for continuous systems with varying fault magnitudes because the number of traces explodes. Using qualitative orderings of measurement deviations avoids this problem.

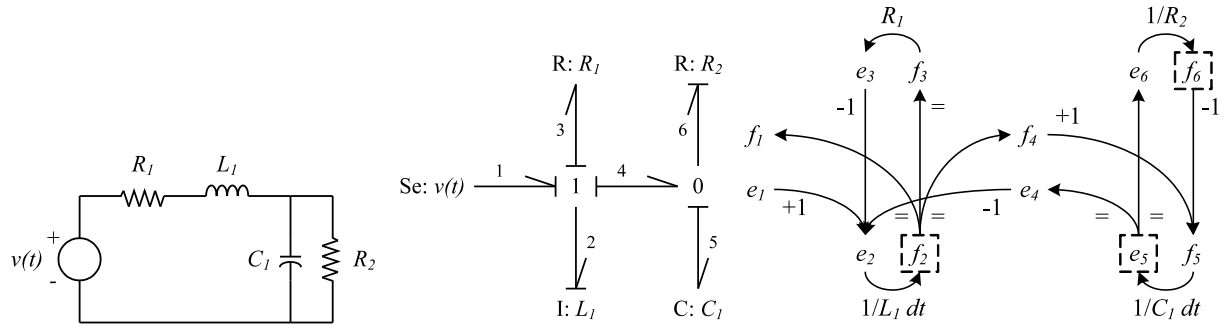
Using temporal orders of measurement deviations is also investigated in [21]–[24], where either time bounds or qualitative orderings for symptom appearance are utilized. These approaches are based on analytical redundancy relations (ARRs), which are difficult to develop for multiplicative faults and nonlinear systems. Our approach can handle both additive and multiplicative faults, but ARR approaches can decouple unknown inputs and disturbances to be robust to their effects [25]. The ARR-based approaches do not address how to obtain the temporal orders, whereas in our approach, the temporal orders are derived systematically from the continuous model. Alternatively, temporal event sequences using qualitative deviational models are developed using process algebras in [26], but a systematic approach to generating the event-based component models or the construction of a diagnoser is not provided.

III. Diagnosis Approach

Our method for diagnosis of single, abrupt, persistent faults in continuous systems extends TRANSCEND [11]. We model systems as bond graphs [27], from which we derive the diagnosis model, the temporal causal graph (TCG). When faults occur, they produce transients that manifest as deviations in measurements from their expected values. These deviations are abstracted to events. The TCG is used to predict possible sequences of measurement deviations that are then matched against observed deviation sequences to isolate faults. Throughout the paper, we illustrate the diagnosis methodology with a circuit example. The schematic, bond graph model, and TCG are shown in Figs. 1a, 1b, and 1c, respectively.

Bond graphs define a domain-independent, energy-based, topological modeling scheme for dynamic systems. They are particularly suitable for diagnosis because they incorporate causal and temporal information required for deriving and analyzing fault transients. Their properties have been exploited in both TCG-based diagnosis [11] and ARR-based approaches [28], [29]. Although in this paper we use bond graphs to model electrical systems as equivalent circuits, bond graphs, and, therefore, our diagnosis approach, can be employed in many other domains [27].

In bond graphs, vertices represent components. Bonds, drawn as half arrows, represent ideal energy connections between the components. Associated with each bond are two variables: *effort* and *flow*, denoted by e_i and f_i, respectively, where i is the bond number, and the product $e_i f_i$ defines the rate of energy transfer through the bond. In the electrical



(a) Schematic. (b) Bond graph. (c) Temporal causal graph.

Fig. 1. Circuit example.

domain, these variables map to voltage and current, respectively. 1-junctions represent series connections (where all f are equal and $\sum e = 0$), and 0-junctions represent parallel connections (where all e are equal and $\sum f = 0$). Other bond graph elements model energy dissipation as resistances (R: R, where $e = Rf$), energy storage as capacitances (C: C, where $\dot{e} = \frac{1}{C}f$) and inductances (I: I, where $\dot{f} = \frac{1}{I}e$), and energy sources as sources of flow (Sf: u, where $f = u$) and effort (Se: u, where $e = u$). The constituent equations of the bond graph elements form a system of equations that describe the continuous behavior of the system, and can be combined into a state-space representation.

Abrupt parametric faults are changes in component parameter values that occur much faster than the time scale of observation [11]. Therefore, they manifest as discontinuities, and we define them as a step change in the parameter value. In the circuit example, faults include increase and decrease in resistance (R_1^+, R_1^-, R_2^+, and R_2^-), capacitance (C_1^+ and C_1^-), and inductance (L_1^+ and L_1^-) values, where the superscript indicates the direction of change in the parameter value.

For both nominal and faulty cases, our model must satisfy conditions for existence and uniqueness of solutions. In bond graphs, the system equations can be computed systematically using *causality*, i.e., the input-output relations on effort and flow variables imposed by the bond graph elements. If the bond graph model has a unique causality assignment, where all energy storage elements can be placed in their integral form, then we obtain a set of ordinary differential equations (ODEs) if the nonlinear functions do not introduce algebraic loops [27]. If the nonlinear functions are smooth, then the ODEs will satisfy the standard Lipschitz conditions from which existence and uniqueness of solutions follow [30]. If causality cannot be assigned uniquely or algebraic loops arise from nonlinear functions, then we obtain a set of differential-algebraic equations, and we assume that they satisfy the corresponding conditions for existence and uniqueness of solutions [31].

Our diagnosis model, the TCG, is derived from the bond graph model of the system [11]. The TCG, which is essentially a signal flow graph with qualitative edge labels, captures the propagation of qualitative fault effects on the measurements. The vertices of the TCG are the system variables. The labeled edges represent the qualitative relationships between the variables, i.e., equality ($=$), direct ($+1$) or inverse (-1) proportionality, integration ($\int dt$, or, in shorthand, simply dt), and parametric relations (e.g. $1/R_1$). The directionality of these edges is determined by causality.

The diagnosis architecture is illustrated in Fig. 2. An *observer*, based on the state-space equations derived from the bond graph model, computes the expected behavior of the system, given the inputs $\mathbf{u}(t)$ and the observed outputs, $\mathbf{y}(t)$. We assume that inputs (which may come from a controller) are known, and do not consider unexpected and unmeasurable changes in the inputs. The difference between observed outputs, $\mathbf{y}(t)$, and expected outputs, $\hat{\mathbf{y}}(t)$, defines the residual, $\mathbf{r}(t)$. Faults will cause the residual values to become nonzero. Nonzero residuals that are statistically significant trigger the *fault detector*, which signals a fault. To accommodate sensor noise and model imperfections, we employ the Z-test [32] to robustly determine if the residual is nonzero using a sliding window technique [33]. Other techniques for fault detection are also applicable [34], [35].

The *symbol generator* abstracts measurement deviations from nominal behavior to corresponding events. They are represented symbolically by qualitative increasing/decreasing values. Like fault detection, symbol generation is performed in a robust manner using the Z-test and sliding windows [33]. These events are used in the *event-based diagnoser* (based on predictions made from the TCG) to formulate the diagnostic hypotheses.

IV. Qualitative Fault Isolation

Abrupt faults generate transients in the dynamic system behavior. Assuming that the system satisfies the conditions for existence and uniqueness of solutions for the nominal and faulty cases, the system output $y(t)$ is continuous and continuously differentiable except at the point of fault occurrence, t_f, so the transient response at $t > t_f$ can be approximated by a Taylor series expansion [36]:

$$y(t) = \sum_{n=0}^{\infty} y^{(n)}(t_f)\frac{(t-t_f)^n}{n!}.$$

If $|y^{(k+1)}|$ is bounded, the Taylor series up to the k^{th} derivative is a good approximation of the true signal $y(t)$ for t close to t_f.

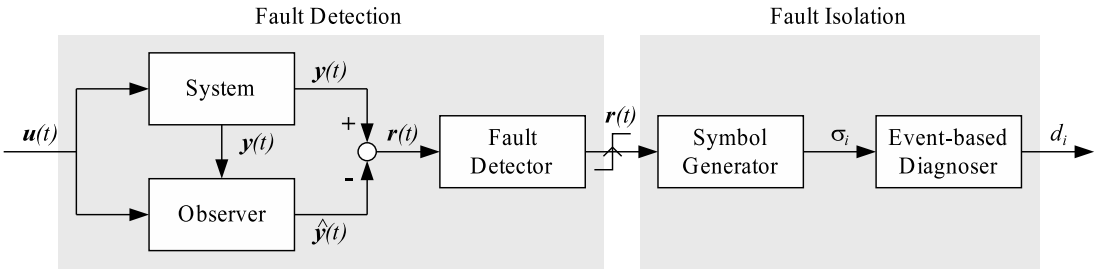


Fig. 2. Diagnosis architecture.

The residual, $r(t) = y(t) - \hat{y}(t)$, can then be approximated by

$$r(t) \quad = \quad \left(\sum_{n=0}^{\infty} y^{(n)}(t_f) \frac{(t-t_f)^n}{n!} \right) - \hat{y}(t).$$

This is the basis for establishing a signature for a fault transient, represented using the magnitude and derivative values of the residual signal. We abstract these magnitude and derivative values using the qualitative values +, −, and 0, which imply an increase, decrease, or no change from the nominal behavior, respectively.

A *fault signature* is defined as the qualitative value of zeroth- through kth-order derivative changes on a residual due to a fault occurrence. Symbol generation extracts two symbols from the deviated signal: (*i*) the observed change at the point of fault occurrence (discontinuity), and (*ii*) the observed first-order change. Since higher-order derivatives eventually manifest as first-order changes that can be detected, we condense the full signatures to the magnitude change and the first nonzero derivative change to reflect the signatures that will be computed using symbol generation, e.g., a seventh-order signature 000−+−+ becomes 0−, and +−+−+−+ becomes +−. The set of possible measurement deviations is then given by {+−, −+, ++, −−, +0, −0, 0+, 0−}. The first symbol represents the direction of abrupt change (the discontinuity at the time of fault occurrence) and the second symbol represents the slope. For +0 and −0, the 0 slope symbol implies that the fault causes a jump but no subsequent change in the measurement. This occurs, for example, with sensor bias faults. Given a measurement m and deviation d, we write the signature as an event using m^d, e.g. m_1^{+-}.

Definition 1: A *fault signature* for a fault f and measurement m is the qualitative magnitude and slope change in m caused by the occurrence of f, and is denoted by $\sigma_{f,m}$. We denote all possible signatures for f and m as $\Sigma_{f,m}$, and denote the set of all fault signatures for fault f as Σ_f, where $\Sigma_f = \bigcup_m \Sigma_{f,m}$.

Because ambiguities may arise in the qualitative arithmetic, we may obtain a signature containing a $*$, which may manifest as either +, −, or 0. So, in general, $\sigma_{f,m}$ may not be unique, and the set $\Sigma_{f,m}$ captures each possibility.

In addition to fault signatures, we also capture the temporal order of measurement deviations, termed *relative measurement orderings* [12], [37], which refer to the intuition that fault effects will manifest in some parts of the system before others. If there are energy storage elements in the path between two measured variables, then the energy storage elements impose a delay in the progression of the transient responses from one measurement to the other [12]. If there are no energy storage elements, the relation between the two transients is algebraic and no delay will be observed. This is based on analysis of the transfer functions from faults to measurements.

Consider a fault parameter f and the variable it immediately affects ν_f. For example, the parameter R_1 immediately affects e_3 (see Fig. 1c). Take two measurements m_1 and m_2. We are interested in the paths which produce the first observable effects on m_1 and m_2. This is determined by the paths of minimum order, i.e., the paths with the minimum number of integrations in the TCG. To illustrate for linear systems, we can characterize the discrete-time transfer functions of these paths for f to m_1, $H_1(z)$, and for f to m_2, $H_2(z)$. Of these paths, if each of m_2's paths passes through m_1 (or a variable algebraically related to m_1), then we can characterize the transfer function $H_2(z)$ as $X(z)H_1(z)$, where $X(z)$ is strictly proper. Therefore m_1 deviates before m_2 for f. More details can be found in [12].

Definition 2: Consider a fault f and measurements m_i and m_j. If f manifests in m_i before m_j then we define a *relative measurement ordering* between m_i and m_j for fault f, denoted by $m_i \prec_f m_j$. We denote the set of all measurement orderings for f as Ω_f.

The fault signatures are systematically derived from the TCG using a forward propagation algorithm to predict qualitative effects of faults on measurements [11]. An extended version of this algorithm computes measurement orderings by analyzing the minimum order paths found during the propagation [38].

We define the set of faults as $F = \{f_1, f_2, \ldots, f_n\}$, and the set of measurements as $M = \{m_1, m_2, \ldots, m_p\}$. For the circuit example, $F = \{R_1^+, R_1^-, R_2^+, R_2^-, C_1^+, C_1^-, L_1^+, L_1^-\}$. The measurement set M includes the current through L_1, the voltage across C_1, and the current through R_2, or $M = \{f_2, e_5, f_6\}$ in the bond graph model. For these faults and measurements, the fault signatures and relative measurement orderings for the circuit system are given in Table I.

For example, consider R_2^+. An increase in R_2 will cause an immediate decrease in f_6. Since all subsequent paths from f_6 to any other observed variable in the system contain some edge with a dt specifier (implying an integration), then deviations in these measurements will only be detected after f_6 deviates. The measured variable e_5 will deviate next with a first-order increase. The change is opposite to the change in f_6 because

TABLE I
FAULT SIGNATURES AND RELATIVE MEASUREMENT ORDERINGS FOR THE CIRCUIT

Fault	f_2	e_5	f_6	Measurement Orderings
R_1^+	0−	0−	0−	$f_2 \prec e_5, f_2 \prec f_6$
R_1^-	0+	0+	0+	$f_2 \prec e_5, f_2 \prec f_6$
R_2^+	0−	0+	−+	$e_5 \prec f_2, f_6 \prec f_2, f_6 \prec e_5$
R_2^-	0+	0−	+−	$e_5 \prec f_2, f_6 \prec f_2, f_6 \prec e_5$
C_1^+	0+	−+	−+	$e_5 \prec f_2, f_6 \prec f_2$
C_1^-	0−	+−	+−	$e_5 \prec f_2, f_6 \prec f_2$
L_1^+	−+	0−	0−	$f_2 \prec e_5, f_2 \prec f_6$
L_1^-	+−	0+	0+	$f_2 \prec e_5, f_2 \prec f_6$

of the -1 specifier in the path, which implies an inverse relationship. The measured variable f_2 will deviate next due to the dt specifier on the path from e_5 to f_2, with a second-order decrease. This will be eventually detected as a first-order change.

V. EVENT-BASED FAULT MODELING

We combine the notion of fault signatures and relative measurement orderings into an event-based framework, where significant measurement deviations are symbolically abstracted to events. For a specific fault, the combination of all fault signatures and relative measurement orderings yields all the possible ways a fault can manifest. Our event set is then the set of possible measurement deviations. We denote each of these possibilities as a *fault trace*.

Definition 3: A *fault trace* for a fault f, denoted by λ_f, is a string of length $\leq |M|$ that includes, for every $m \in M$ that will deviate due to f, a fault signature $\sigma_{f,m}$, such that the sequence of fault signatures satisfies Ω_f.

Consider C_1^+. $\lambda_{C_1^+} = e_5^{-+} f_6^{-+} f_2^{0+}$ is a valid fault trace, but $\lambda_{C_1^+} = f_2^{0+} e_5^{-+} f_6^{-+}$ is not because the measurement deviation sequence does not satisfy $\Omega_{C_1^+}$. Note also that the definition implies that fault traces are of maximal length, i.e., a fault trace includes deviations for all measurements affected by the fault. We group the set of all fault traces into a *fault language*. The *fault model*, defined by a *finite automaton*, concisely represents the fault language.

Definition 4: The *fault language* of a fault $f \in F$ with measurement set M, denoted by L_f, is the set of all fault traces for f.

Definition 5: The *fault model* for a fault $f \in F$ with measurement set M, is the finite automaton that accepts exactly the language L_f, and is given by $\mathcal{L}_f = (Q, q_0, \Sigma, \delta, A)$ where Q is a set of states, $q_0 \in Q$ is an initial state, Σ is a set of events, $\delta : Q \times \Sigma \to Q$ is a transition function, and $A \subseteq Q$ is a set of accepting states.

The finite automata representation allows the composition of the fault signatures and relative measurement orderings into fault models. The possible fault signatures $\Sigma_{f,m}$ can be represented as a finite automaton with event set $\Sigma_{f,m}$, shown in Fig. 3 (left), for the case where $\Sigma_{f,m}$ is a singleton. It consists of only the single event corresponding to the fault signature. In general, multiple edges for each $\sigma_{f,m} \in \Sigma_{f,m}$ are needed going from the first state of the automaton to the final state. This represents the constraint that a measurement's deviation is only observed once. Also, each relative measurement ordering, $m_i \prec_f m_j$, with associated signature sets Σ_{f,m_i} and Σ_{f,m_j}, can be represented as an automaton with event set $\Sigma_{f,m_i} \cup \Sigma_{f,m_j}$, shown in Fig. 3 (right), for the case where Σ_{f,m_i} and Σ_{f,m_j} are singletons. The automaton consists of the associated signatures in the determined ordering. The following lemma formalizes the composition of these automata. (Proofs are given in the Appendix.)

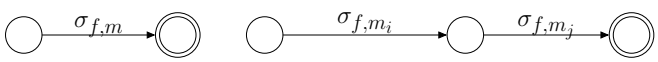
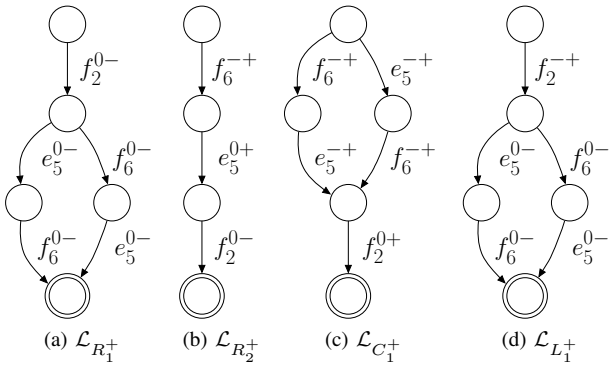


Fig. 3. Fault signature finite automaton representation (left) and relative measurement ordering finite automaton representation (right).



(a) $\mathcal{L}_{R_1^+}$ (b) $\mathcal{L}_{R_2^+}$ (c) $\mathcal{L}_{C_1^+}$ (d) $\mathcal{L}_{L_1^+}$

Fig. 4. Fault models for the faults of the circuit. The fault models for decreases in the parameter values are the same except for a reversal in the signs.

Lemma 1: For fault model for fault f, $\mathcal{L}_f$, is the synchronous product of the individual finite automata for all $\sigma_{f,m} \in \Sigma_f$ and all $m_i \prec_f m_j \in \Omega_f$.

Fig. 4 shows the fault models for the circuit example. For example, take R_1^+. Its orderings specify that f_2 must deviate before e_5 and f_6. Therefore, f_2^{0-} is first, followed by e_5^{0-} and f_6^{0-} in either order.

Ultimately, we would like to be able to make guarantees about the isolation of faults using the event-based diagnoser. To do this, we establish the notions of *distinguishability* and *diagnosability*.

Definition 6: A fault f_i is distinguishable from a fault f_j, denoted by $f_i \nsim f_j$, if f_i always eventually produces effects on the measurements that f_j cannot.

Under our framework, one fault will be distinguishable from another fault if it cannot produce a fault trace that is a prefix[1] (denoted by $\sqsubseteq$) of a trace that can be produced by the other fault. If this is not the case, then when that trace manifests, the first fault cannot be distinguished from the second.

Lemma 2: A fault $f_i \in F$ is *distinguishable* from a fault $f_j \in F$, if there does not exist a pair of fault traces $\lambda_{f_i} \in L_{f_i}$ and $\lambda_{f_j} \in L_{f_j}$, such that $\lambda_{f_i} \sqsubseteq \lambda_{f_j}$.

If a system is diagnosable, i.e., every pair of faults can be distinguished, then we can make guarantees about the unique isolation of every fault in the system. To define this, we first

[1]A fault trace λ_i is a prefix of fault trace λ_j if there is some (possibly empty) sequence of events λ_k that can extend λ_i such that $\lambda_i \lambda_k = \lambda_j$.

define a notion of a system in our framework.

Definition 7: A *system* S is tuple (F, M, L_F), where $F = \{f_1, f_2, \ldots, f_n\}$ is a set of faults, $M = \{m_1, m_2, \ldots, m_p\}$ is a set of measurements, and $L_F = \{L_{f_1}, L_{f_2}, \ldots, L_{f_n}\}$ is the set of fault languages.

Definition 8: A system $S = (F, M, L_F)$ is *diagnosable* if $(\forall f_i, f_j \in F) f_i \neq f_j \implies f_i \nsim f_j$.

If the system is diagnosable, then every pair of faults is distinguishable using the measurements in M. So, each sequence of measurement deviations we observe can be eventually linked to exactly one fault, if measurement deviation events are generated correctly. Hence, we can uniquely isolate all faults of interest. If the fault set is not diagnosable, then ambiguities will remain after fault isolation, i.e., after all possible measurement deviations have been observed.

VI. The Event-based Diagnoser

The goal of the event-based diagnoser is, given a sequence of events from the symbol generation module, to determine which faults are consistent with the observed sequence. We define formally a *diagnosis* and a *diagnoser* in our framework.

Definition 9: A *diagnosis* $d \subseteq F$ is a set of faults that are consistent with the observed measurements.

Definition 10: A *diagnoser* for a fault set F is a tuple $\mathcal{D}_F = (Q, q_0, \Sigma, \delta, A, D, Y)$ where Q is a set of states, $q_0 \in Q$ is an initial state, Σ is a set of events, $\delta : Q \times \Sigma \to Q$ is a transition function, $A \subseteq Q$ is a set of accepting states, $D \subseteq 2^F$ is a set of diagnoses, and $Y : Q \to D$ is a diagnosis map.

A diagnoser is a finite automaton extended by a set of diagnoses and a diagnosis map. A diagnoser takes events as inputs, which, as with fault models, correspond to measurement deviations. From the current state, a measurement deviation event causes a transition to a new state. The diagnosis for that new state represents the set of faults that are consistent with the sequence of events seen up to the current point in time. So, like traditional DES diagnosers, the diagnoser states provide estimates of the system condition, but only after a fault has occurred. As discussed, we assume that nominal behavior is tracked in the continuous domain by an observer.

The accepting states of the diagnoser correspond to a fault isolation result. We say that a diagnoser *isolates* a fault if it accepts all possible valid traces for the fault and the accepting states map to diagnoses containing the fault.

Definition 11: A diagnoser $\mathcal{D}_F$ *isolates* fault $f \in F$ if $\mathcal{D}_F$ accepts all $\lambda_f \in L_f$ and for each $q \in A$ that accepts some λ_f, $f \in Y(q)$.

We also would like to achieve unique isolation of faults, which corresponds to diagnosability. We say that a diagnoser *uniquely isolates* a fault if each accepting state maps to the single fault.

Definition 12: A diagnoser $\mathcal{D}_F$ *uniquely isolates* fault $f \in F$ if $\mathcal{D}_F$ accepts all $\lambda_f \in L_f$ and for each $q \in A$ that accepts some λ_f, $\{f\} = Y(q)$.

Ultimately, we would like to systematically construct a diagnoser for a system S that isolates all $f \in F$. Further, we would like to show that if S is diagnosable, then this diagnoser uniquely isolates all $f \in F$. To do this, we first provide a way to construct a diagnoser for each fault f that isolates f. Then, we provide a composition operator to compose two diagnosers, such that if each diagnoser isolates its own set of faults, the composed diagnoser will isolate the combined set of faults. We then compose the individual diagnosers into a global diagnoser that isolates the complete set of system faults.

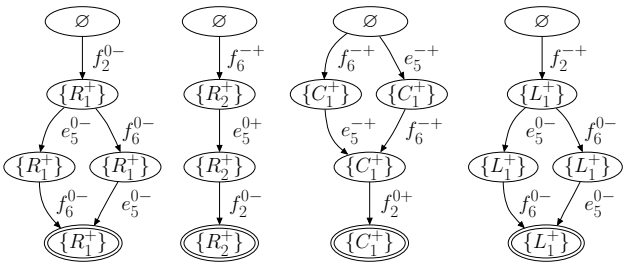


Fig. 5. Diagnosers for the individual faults of the circuit. The diagnosers for decreases in the parameter values are the same except for a reversal in the signs.

First, we construct a diagnoser for each single fault f from $\mathcal{L}_f$. Because the fault model $\mathcal{L}_f$ accepts the fault language L_f, it is easy to show that this diagnoser isolates f. The diagnosers corresponding to the individual faults of the circuit are shown in Fig. 5.

Definition 13: Given f with $\mathcal{L}_f = (Q, q_0, \Sigma, \delta, A)$, $\mathcal{D}^*_{\{f\}}$ is defined as $(Q, q_0, \Sigma, \delta, A, \{\{f\}\}, Y)$, where:

$$Y(q) = \begin{cases} \varnothing, & q = q_0 \\ \{f\}, & \text{otherwise.} \end{cases}$$

Lemma 3: $\mathcal{D}^*_{\{f\}}$ uniquely isolates f.

We next define a composition operator, denoted as Δ. An implementation of Δ is presented in [14]. The Δ composition provides a way to systematically construct the diagnoser for fault set F. It must be defined such that the composed diagnoser captures all valid fault traces for the considered faults, and maps the states to correct diagnoses.

Definition 14: Given the diagnoser for a set of faults F_i, $\mathcal{D}_{F_i} = (Q_i, q_{0,i}, \Sigma_i, \delta_i, A_i, D_i, Y_i)$ and the diagnoser for a set of faults F_j, $\mathcal{D}_{F_j} = (Q_j, q_{0,j}, \Sigma_j, \delta_j, A_j, D_j, Y_j)$, the diagnoser defined by the Δ composition of $\mathcal{D}_{F_i}$ and $\mathcal{D}_{F_j}$ is $\mathcal{D}_{F_i} \Delta \mathcal{D}_{F_j} \triangleq (Q, q_0, \Sigma, \delta, A, D, Y)$, where:

- $Q = Q_i \times Q_j$,
- $q_0 = (q_{0,i}, q_{0,j})$,
- $\Sigma = \Sigma_i \cup \Sigma_j$,
- $\delta((q_i, q_j), \sigma) = (q_{i+1}, q_{j+1})$, where
  - $q_{i+1} = \begin{cases} \delta_i(q_i, \sigma), & \delta_i(q_i, \sigma) \neq \varnothing \\ q_i, & \text{otherwise} \end{cases}$
  - $q_{j+1} = \begin{cases} \delta_j(q_j, \sigma), & \delta_j(q_j, \sigma) \neq \varnothing \\ q_j, & \text{otherwise} \end{cases}$
  - $Y((q_{i+1}, q_{j+1})) =$
    $\begin{cases} Y_i(q_{i+1}) \cup Y_j(q_{j+1}), (q_i, q_j) = q_0 \\ Y((q_i, q_j)) \cap (Y_i(q_{i+1}) \cup Y_j(q_{j+1})) \neq \varnothing, \text{else} \end{cases}$
- $A = \{(q_i, q_j) : q_i \in A_i \vee q_j \in A_j\}$
- $D = 2^{(F_i \cup F_j)}$

Theorem 1: If $\mathcal{D}_{F_i}$ isolates all $f_i \in F_i$, and $\mathcal{D}_{F_j}$ isolates all $f_j \in F_j$, then $\mathcal{D}_{F_i} \Delta \mathcal{D}_{F_j}$ isolates all faults in $F_i \cup F_j$.

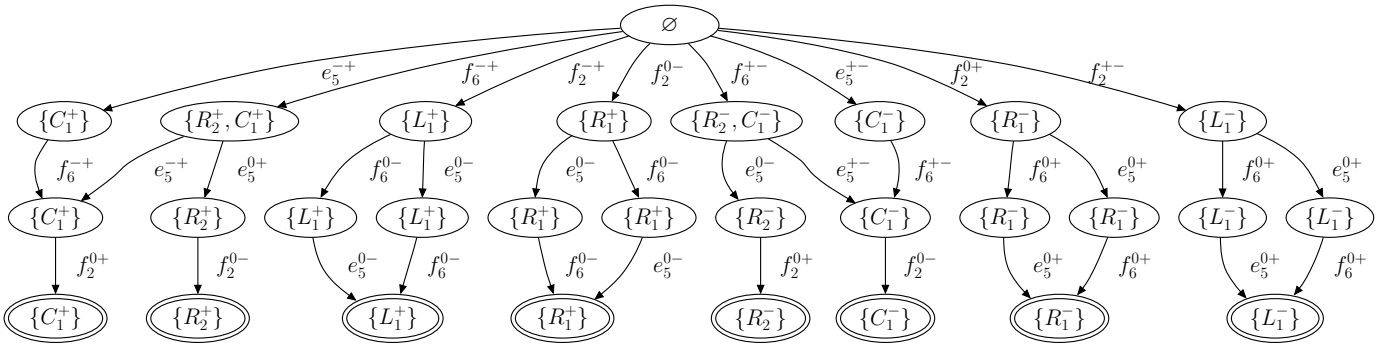


Fig. 6. Event-based diagnoser for the circuit.

The Δ composition is defined to be commutative and associative with respect to isolation, and the theorem shows that this is true, i.e., Δ preserves the isolation property. The order in which the diagnosers are composed does not matter, because at each intermediate step, isolation of the combined fault sets is maintained. Therefore, we can define the global diagnoser as a composition of the individual diagnosers.

Definition 15: For fault set $F = \{f_1, f_2, \ldots, f_n\}$, $\mathcal{D}_F^*$ is defined as $\mathcal{D}_{\{f_1\}}^* \Delta \mathcal{D}_{\{f_2\}}^* \Delta \ldots \Delta \mathcal{D}_{\{f_n\}}^*$.

Corollary 1: The diagnoser $\mathcal{D}_F^*$ isolates all $f \in F$.

Because each $\mathcal{D}_{\{f\}}^*$ isolates f if constructed from $\mathcal{L}_{\{f\}}$ as described, and since Δ preserves the isolation property, then $\mathcal{D}_F^*$ as constructed above isolates all $f \in F$. Further, if the fault set is diagnosable, then this diagnoser guarantees that each fault is uniquely isolated.

Theorem 2: A system $S = (F, M, L_F)$ is diagnosable if and only if $\mathcal{D}_F^*$ uniquely isolates all $f \in F$.

The diagnoser for the circuit example is shown in Fig. 6. We can see that since all accepting states have singleton diagnoses, the system is diagnosable. For example, consider the fault trace $f_6^{-+} e_5^{0+} f_2^{0-}$. For f_6^{-+} occurring as the first deviation, only C_1^+ or R_2^+ could have occurred, given the known fault signatures and measurement orderings. Therefore, the new diagnosis is $\{C_1^+, R_2^+\}$. For e_5^{0+} occurring next, of our current faults, only R_2^+ is consistent, therefore, our new diagnosis is the intersection of $\{C_1^+, R_2^+\}$ and $\{R_2^+\}$, which is $\{R_2^+\}$. At this point we obtain a unique fault hypothesis. The only possible measurement deviation from here is f_2^{0-} which must still be consistent with $\{R_2^+\}$.

VII. Event-based Diagnoser Implementation

The proposed event-based diagnosis framework leads to three different implementations of the event-based diagnoser that trade off space and time complexity.

1) The $\mathcal{D}_F^*$ Implementation: Performing online diagnosis with $\mathcal{D}_F^*$ (Fig. 6) has the best time complexity. At design time, $\mathcal{D}_F^*$ is computed using repeated application of Δ, as discussed. At run-time, the diagnoser needs only to wait for measurement deviations to occur, transition to the next state, and output the associated diagnosis. Using appropriate data structures, these operations can be achieved in constant time.

With a large number of faults and measurements, however, $\mathcal{D}_F^*$ may have poor space complexity. Since $\mathcal{D}_F^*$ must contain all the fault traces for all faults in F, it must capture $O(|F|T)$ traces, where T is the maximum number of traces per fault. This is also the design-time complexity of constructing $\mathcal{D}_F^*$. In the worst case, a fault may have no measurement orderings, thus T is $O(|M|!)$. Therefore, there would be $O(|F||M|!)$ traces and $O(|F||M|!)$ states in the worst case. If T is truly the worst case, however, i.e., a fault allows all possible signatures in any sequence, then the diagnoser would only have $O(|M|!)$ distinct traces to capture, and thus $O(|M|!)$ states. If many temporal orderings exist, then the number of possible fault traces reduces significantly, and $\mathcal{D}_F^*$ will have feasible space requirements. Also, the diagnoser can always be pruned by recursively removing leaf states that have the same diagnosis as their predecessor states, thereby reducing the space requirements further [38].

2) The $\mathcal{D}_{\{f\}}^*$ Implementation: In the $\mathcal{D}_{\{f\}}^*$ implementation, only the individual $\mathcal{D}_{\{f\}}^*$ for each $f \in F$ (Fig. 5) are computed at design time, which is less expensive than computing $\mathcal{D}_F^*$. Each fault may still have, in the worst case, $O(|M|!)$ possible fault traces. The worst case total space requirement is then $O(|F||M|!)$. Again, if many temporal orderings exist, then the space complexity reduces substantially.

Since the global diagnoser $\mathcal{D}_F^*$ must capture all possible traces for each fault, it will have less states than the total number of states combining all the fault models. This occurs because shared prefixes result in combined states in $\mathcal{D}_F^*$. The $\mathcal{D}_{\{f\}}^*$ implementation is more suited to the multiple fault case where $\mathcal{D}_F^*$ contains extra fault traces [39].

In online diagnosis, each diagnoser is traced simultaneously. The hypothesis set, h, is formed by taking the union of the diagnoses in each current state. This operation has time complexity $O(|F|)$. The current diagnosis is formed as the intersection of the hypothesis set and the previous diagnosis (except when the previous diagnosis is $\varnothing$). When a diagnoser becomes blocked, i.e., there is no available event to take from the current state, then it is no longer tracked, because it is no longer consistent with the observed measurement deviations. The current diagnosis can be obtained by taking the union of the diagnoses for the diagnosers that are still active.

3) The $\Sigma_{F,M}/\Omega_{F,M}$ Implementation: If each fault has many measurement orderings, then using either $\mathcal{D}_F^*$ or the set of $\mathcal{D}_{\{f\}}^*$ will be both space-efficient and time-efficient. If few orderings are available, then the diagnosers approach size $O(|M|!)$, therefore, these approaches may not be feasible given the space requirements of the system. The third im-

plementation computes only the fault signatures and relative measurement orderings for each fault at design time (Table I), requiring $O(|F||M|^2)$ space. Alternatively, these can be computed online when a fault is detected, and this operation is polynomial in the size of the TCG [11], [12].

Given a current diagnosis d_{i-1} and an event σ_i occurring, we can check which faults are consistent with σ_i. The hypothesis set h_i consists of those faults. In the $\mathcal{D}^*_{\{f\}}$ implementation, this is determined simply by which diagnosers are still tracking correctly. If $i = 1$, then the new diagnosis d_i is simply h_i. Otherwise, the new diagnosis must be consistent with d_{i-1} and with the new information, i.e., $d_i = d_{i-1} \cap h_i$. Therefore, given d_{i-1}, the new diagnosis can be computed simply as the subset of faults in d_{i-1} consistent with σ_i. This corresponds to only constructing the *path* of $\mathcal{D}^*_F$ relating to the particular fault trace we are observing. In $\mathcal{D}^*_F$, all this work has been done at design time.

In online diagnosis, we form the hypothesis set corresponding to the current measurement deviation by looking through the fault signatures and measurement orderings, and this requires $O(|F||M|^2)$ time. We then compute the new diagnosis, which is a function of the size of the current diagnosis and the current hypothesis set. In the worst case the hypothesis set consists of all faults, so it is $|F|$ in size. A diagnosis can be as large as $|F|$ also. The intersection of the diagnosis and hypothesis set then takes at worst $O(|F|)$ time. In practice, this time complexity is reduced because as measurements deviate, fewer fault hypotheses are being considered.

VIII. Case Study

We demonstrate the proposed diagnosis framework with experiments conducted on the Advanced Diagnostics and Prognostics Testbed (ADAPT) [15] deployed at NASA Ames Research Center. The testbed is functionally representative of a spacecraft's electrical power system, and consists of three subsystems: power generation (battery chargers), power storage (lead-acid batteries), and power distribution (relays, circuit breakers, DC to AC converters, DC and AC loads). For our diagnosis experiments, we consider a subset of ADAPT that involves a battery discharging to two parallel DC loads.

The accuracy of our diagnosis approach is critically dependent on the fault detection and symbol generation processes. Due to model imperfections and sensor noise, the fault detectors have to be tuned to minimize missed detections (false negatives), and false alarms (false positives). A tradeoff exists between these two, because a more sensitive fault detector will get more false positives, but fewer false negatives. Similarly, a less sensitive fault detector will get more false negatives, but less false positives. In our experiments, the fault detectors were empirically tuned to the highest possible sensitivity that avoided false alarms for the observed levels of noise under nominal conditions. To study the performance of the diagnosis algorithms under different fault and noise conditions, we need to perform a large number of experiments. In addition to experiments from the actual testbed, we ran simulation experiments on the Virtual ADAPT testbed [15], [40]. Simulation also allows us to introduce faults that cannot be injected into the actual system safely.

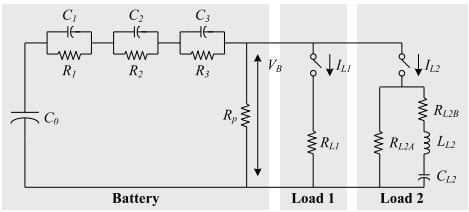


Fig. 7. Electrical circuit equivalent for the battery system.

A. System Modeling

The electrical circuit equivalent of the considered subset of ADAPT is shown in Fig. 7. The battery model describes an electric circuit equivalent based on the model presented in [41], [42]. The charge-holding capacity of the battery is modeled by a large capacitance, C_0. The $R-C$ pairs subtract from the voltage provided by C_0 to obtain the actual provided battery voltage. The resistance parameters, R_1, R_2, and R_3, are nonlinear functions, given by

$$\begin{aligned} R_1 &= R_{10} + A_{11}SOC, \\ R_2 &= -R_{20}\ln(DOC), \\ R_3 &= \frac{R_{30}\exp A_{31}(1-SOC)}{1+\exp A_{32}i(t)}, \end{aligned}$$

where $i(t)$ is the discharge current, SOC is the state of charge, and DOC is the depth of charge. SOC and DOC are computed parameters, given by

$$\begin{aligned} SOC &= 1 - \frac{Q_{max} - q(t)}{K_c C_0^* \left(1 - \frac{\theta(t)}{\theta_f}\right)^{\epsilon}}, \\ DOC &= 1 - \frac{Q_{max} - q(t)}{K_c C_0^* \left(1 - \frac{\theta(t)}{\theta_f}\right)^{\epsilon}} \left(1 + (K_c - 1)\left(\frac{i(t)}{I^*}\right)^{\delta}\right), \end{aligned}$$

where $q(t)$ is the charge in C_0, and $\theta(t)$ is the battery temperature, given by

$$\dot{\theta}(t) = \frac{1}{C_\theta}\left(P_B - \frac{\theta(t) - \theta_a}{R_\theta}\right)$$

where θ_a is the ambient temperature θ_a, and P_B is the power dissipated through the battery resistances. Details of these equations and their parameters may be found in [41].

The selected measurements were the battery voltage, $V_B(t)$, and the currents through the loads, $I_{L1}(t)$ and $I_{L2}(t)$. Multiplicative faults include parameter changes in the battery and the loads. Battery faults include loss of capacity to hold charge represented by a capacitance decrease, C_0^-, and an increase in internal losses, R_1^+. Abrupt battery faults are less likely than incipient faults, but they may produce immediate and significant changes that must be dealt with quickly. Faults in the system loads are represented by increases or decreases in their resistance values, R_{L1} and R_{L2A}. We also consider additive bias faults in the sensors, which produce abrupt changes in the measured values. Sensor faults are labeled by the measured quantity they represent, e.g., V_B^+ represents a bias fault in the battery voltage sensor.

TABLE II
FAULT SIGNATURES AND RELATIVE MEASUREMENT ORDERINGS FOR THE BATTERY SYSTEM

Fault	V_B	I_{L1}	I_{L2}	Measurement Orderings
C_0^-	+−	+−	+−	$\varnothing$
R_1^+	0−	0−	0−	$\varnothing$
R_{L1}^+	0⋆	−+	0⋆	$I_{L1} \prec V_B, I_{L1} \prec I_{L2}$
R_{L1}^-	0⋆	+−	0⋆	$I_{L1} \prec V_B, I_{L1} \prec I_{L2}$
R_{L2A}^+	0⋆	0⋆	−+	$I_{L2} \prec V_B, I_{L2} \prec I_{L1}$
R_{L2A}^-	0⋆	0⋆	+−	$I_{L2} \prec V_B, I_{L2} \prec I_{L1}$
V_B^+	+0	00	00	$V_B \prec I_{L1}, V_B \prec I_{L2}$
V_B^-	−0	00	00	$V_B \prec I_{L1}, V_B \prec I_{L2}$
I_{L1}^+	00	+0	00	$I_{L1} \prec V_B, I_{L1} \prec I_{L2}$
I_{L1}^-	00	−0	00	$I_{L1} \prec V_B, I_{L1} \prec I_{L2}$
I_{L2}^+	00	00	+0	$I_{L1} \prec V_B, I_{L2} \prec I_{L1}$
I_{L2}^-	00	00	−0	$I_{L1} \prec V_B, I_{L2} \prec I_{L1}$

In our bond graph model, causality can be uniquely assigned with all energy storage elements in integral causality, and the nonlinearities (which are smooth functions) do not introduce any algebraic loops. Thus, the system behavior can be expressed as a set of ODEs with a unique solution for both the nominal and faulty cases, so fault signatures and relative measurement orderings are well-defined and can be derived using the TCG generated automatically from the bond graph model. The signatures and orderings for the considered faults are given in Table II. The nonlinearities in the battery introduce ambiguity in the qualitative signatures, and this is denoted by the $\star$ symbol. For example, a signature of 0⋆ may manifest as 0+ or 0−. All possible effects must be included in the fault models. Also note that since the sensors are not part of feedback loops in the system, sensor faults affect only the measurement provided by the sensor. The other measurements are not affected, and so the corresponding fault signatures are denoted by 00, indicating no change in the measurement from expected behavior. If feedback loops are present, the controller can be kept out of the model if its inputs to the system are known. Otherwise, our approach can also deal directly with models that include the controller (e.g., see [12]).

B. Experimental Results

We have performed experiments online on the ADAPT testbed. The event-based diagnoser contained 45 states and 72 transitions, and its pruned version contained 22 states and 26 transitions. In practice, DES diagnosers can easily have thousands of states, which is a main advantage to DES approaches. Therefore, we feel our approach can scale to the full testbed.

To demonstrate the diagnosis approach, we show the results obtained for load faults and a sensor fault. In all experiments, Load 1 is first brought online, followed by Load 2. We inject the fault in the mode where both loads are online. The measurements were sampled at 2 Hz for all the experiments. The nominal behavior of the system is shown in Fig. 8, and this data was used for identification of system parameters shown in Table III. Note that since the circuit representation is an abstraction of actual battery behavior, the R and C values do not correspond to typical values in electric circuits.

TABLE III
IDENTIFIED SYSTEM PARAMETERS

Battery Parameters	
Ambient Temperature	$\theta_a = 22^\circ$C
Thermal Capacitance	$C_\theta = 615.3$ Wh/$^\circ$C
Thermal Resistance	$R_\theta = 0.01^\circ$C/W
Battery Capacitance	$C_0 = 106360$ F
Parasitic Resistance	$R_p = 500\ \Omega$
Maximum Battery Charge	$Q_{max} = 2765360$ C
Electrolyte Freezing Temperature	$\theta_f = -35^\circ$C
Empirical Coefficients	$C_1 = 51.079$ F
	$C_2 = 51.216$ F
	$C_3 = 567.56$ F
	$R_{10} = 0.05582\ \Omega$
	$A_{11} = -0.025025$
	$R_{20} = 0.001847\ \Omega$
	$R_{30} = 0.3579\ \Omega$
	$A_{31} = -2.5315$
	$A_{32} = 0.22208$
	$K_c = 1.33$
	$C_0^* = 270720$ Ah
	$\epsilon = 0.642$
	$\delta = 0.61$
	$I^* = 5$ A
Load Parameters	
Load 1	$R_{L1} = 11.8\ \Omega$
Load 2	$R_{L2A} = 27.696\ \Omega$
	$R_{L2B} = 209.92\ \Omega$
	$C_{L2} = 0.48678$ F
	$L_{L2} = 1.9986$ H

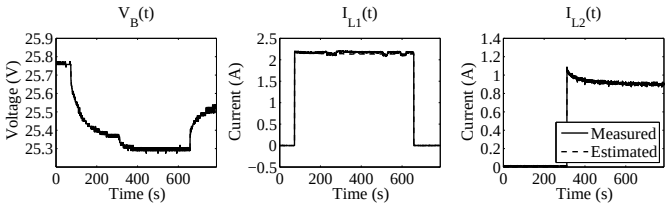


Fig. 8. Nominal system operation.

For the first experiment, a 33% decrease in the Load 1 resistance, R_{L1}^-, is manually injected at 653.0 s by abruptly changing the resistance setting on the load. The measured and estimated outputs are shown in Fig. 9. A partial diagnoser is given in Fig. 10. The decrease in resistance increases the current drawn by the load abruptly, and this change is detected at 653.5 s. Since the slope of the change is not yet known, the possible fault hypotheses are $\{R_{L1}^-, I_{L1}^+, C_0^-\}$. Faults R_{L2A}^+ and R_{L2A}^- are not included, because even though they may cause the current to increase, measurement orderings predict that I_{L2} would have deviated first instead. At 655.0 s, a decrease is detected in $V_B(t)$. Since I_{L1}^+ cannot affect $V_B(t)$, it is dropped. C_0^- is also dropped because it would have increased, and not decreased, the battery voltage. Due to the dynamics of Load 2, the change in $V_B(t)$ is not large enough to cause a change in $I_{L2}(t)$ that can be distinguished from the sensor noise. Even though the full signatures are not known, the partial diagnoser shows that R_{L1}^- must be the only fault. Therefore, the true fault is isolated.

For a second scenario, a 100% increase in the Load 1 resistance, R_{L1}^+, is manually injected at 439.5 s. The measured

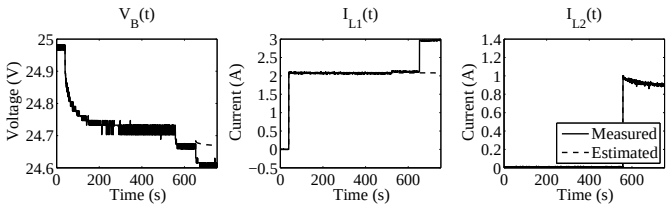


Fig. 9. R_{L1}^{-} fault, where R_{L1} decreases by 33%.

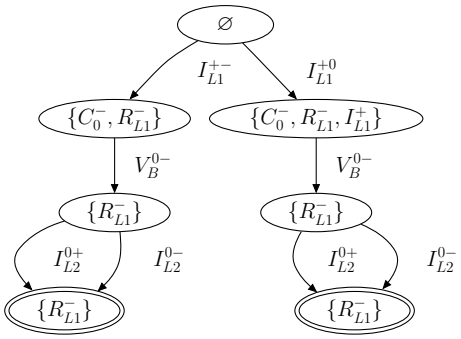


Fig. 10. Partial diagnoser for isolating the R_{L1}^{-} fault.

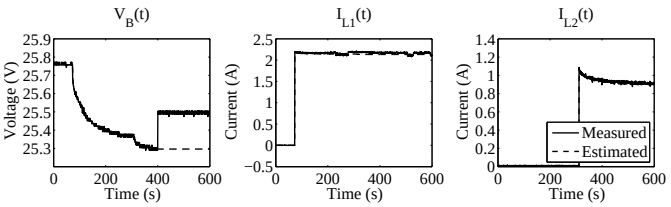


Fig. 13. V_B^{+} fault with bias of 0.2.

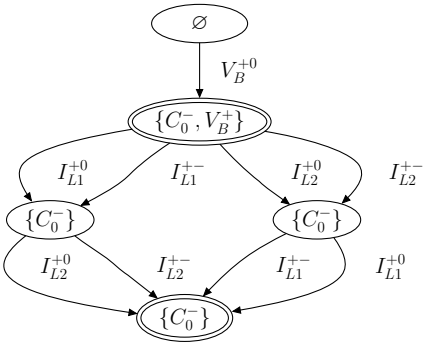


Fig. 14. Partial diagnoser for isolating the V_B^{+} fault.

and estimated outputs are shown in Fig. 11 and Fig. 12, which shows the signals in more detail around the time of fault occurrence. The increase in resistance causes a discontinuous drop in the current, detected at 440.0 s. Since the slope has not yet been computed, the possible fault candidates are $\{R_{L1}^+, I_{L1}^-, R_1^+\}$. Again, faults R_{L2A}^+ and R_{L2A}^- are not included, because measurement orderings predict that I_{L2} would have deviated first instead. At 441.0 s, an increase is detected in $V_B(t)$. Since I_{L1}^- cannot affect $V_B(t)$, it is dropped. R_1^+ is also dropped because it would have decreased, and not increased, the battery voltage. Independent of how $I_{L2}(t)$ deviates, the diagnoser ends up in a state that isolates R_{L1}^+.

In a third experiment, a positive bias of 0.2 V is injected into the voltage sensor at 400.0 s by spoofing the real sensor data in software. The measured and estimated outputs are shown in Fig. 13. A partial diagnoser is shown in Fig. 14. The fault detector reports an increase in battery voltage at 400.0 s. The fault candidates generated are $\{V_B^+, C_0^-\}$, since no other fault can cause an increase in $V_B(t)$ as the first deviation. At 407.5 s, the slope is computed to be 0, under the assumption that the window is large enough to distinguish zero from nonzero slopes. So, the diagnosis remains $\{V_B^+, C_0^-\}$. Because no further measurements deviate, C_0^- cannot be eliminated. This is demonstrated by the partial diagnoser and predicted by diagnosability analysis. After V_B^{+0} occurs, the diagnoser is in an accepting state (because it corresponds to a fault trace of V_B^+), and their are multiple faults remaining. Therefore, if no further measurements deviate, the faults cannot be distinguished.

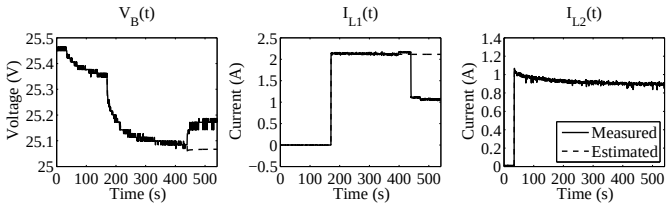


Fig. 11. R_{L1}^{+} fault, where R_{L1} increases by 100%.

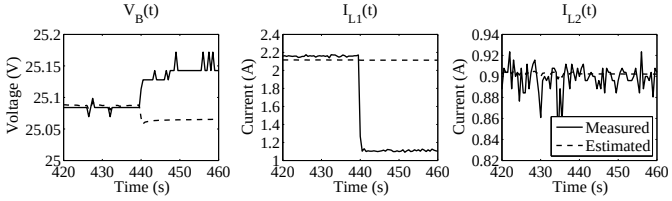


Fig. 12. Detailed plot of R_{L1}^{+} fault.

C. Simulation Results

In the following simulation experiments, we considered different fault magnitudes and different levels of sensor noise to investigate the robustness and sensitivity of our fault detection and isolation scheme. We used a zero-mean Gaussian noise model, and the noise level was reflected in the variance. The three noise levels (N_0, N_1, N_2) reflect no noise, the observed noise magnitudes of the testbed, and double the observed noise. These values were selected as 0, 4×10^{-4}, and 8×10^{-4} for the voltage sensor, and 0, 4×10^{-5}, and 8×10^{-5} for the current sensors.

Due to model imperfections and sensor noise, the fault detectors must be tuned to (*i*) minimize missed detections (false negatives), and (*ii*) minimize false alarms (false positives). Similarly, the symbol generators must be tuned to achieve the same performance metrics. In our experiments, the fault detectors and symbol generators were tuned to the highest possible sensitivity that would avoid false alarms for noise level N_1 under nominal conditions. With the particular noise levels and fault magnitudes chosen, no false positives or false negatives

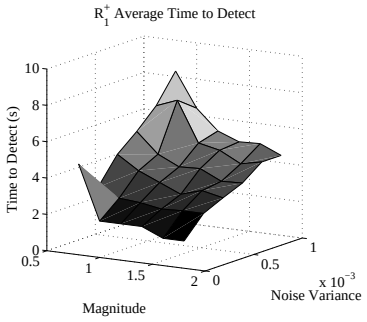


Fig. 15. Average time to detect for R_1^+ with varying noise variance and fault magnitude.

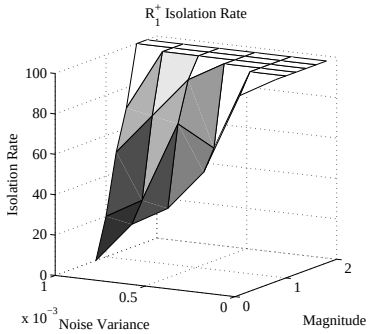


Fig. 16. Isolation rate for R_1^+ with varying noise variance and fault magnitude. The fault is considered successfully isolated if it is in the final list of faults returned by the diagnoser.

occurred in the fault detection. Since the threshold is computed as a function of the signal variance [33], false alarms are avoided even for higher levels of noise than expected. Other diagnosis approaches that transform noisy, continuous signals into some abstraction that facilitates diagnostic reasoning must tune parameters of those transformations as well.

The diagnosis results are summarized in Table IV. For the sensor faults, the magnitude is given as an additive bias in V or A. For the process faults, the faulty parameter value is given by its nominal value multiplied by the given factor, e.g., a factor of 1.10 increases R_{L1} by 10% of its nominal value of 11.8 Ω. Ten experiments were performed for each fault, magnitude, and noise level. The table presents the average results over these runs. In each of the scenarios, the time of fault injection, t_f, was set at 500 s. The times for detection and isolation are denoted by t_d and t_i, respectively. In some cases the true fault cannot be uniquely isolated, so t_i represents the time at which the fault candidate list stopped reducing. We report on the average times to detect and isolate, the average size of the final fault candidate list F_n, and the percentage of times the true fault was in F_n.

The results show that the sensor noise and fault magnitude can have a significant effect on time to fault detection. Fig. 15 shows the average time to detect as a function of the variance in the sensor noise and the fault magnitude for R_1^+. For smaller fault magnitudes and a lower signal to noise ratio, it takes longer for the effects of the fault to be identified in relation to the noise band. Therefore, reliable detection takes longer. As shown in Table IV, faults are detected faster when magnitudes are larger, because a shorter interval is needed to determine that the mean of the residual is statistically outside of the computed signal variance. Fault detection times also improve with lower noise, because the deviations caused by a fault are more clearly differentiated from the noise. Similar results were obtained for the other faults. For I_{L1}^+ and I_{L2}^+ with a magnitude of $+\,0.1$, and R_{L2A}^- with a magnitude of $\times\,0.95$ and $\times\,0.90$, the fault is always detected after one sample (0.5 s) with no noise, but in some experiments with the noise, the noise worked in favor of fault detection and detection at the point of fault occurrence was obtained.

Sensor noise and fault magnitude can also affect the isolation results. If an incorrect symbol is generated, then the true fault may be eliminated as a candidate. This situation is shown well by the experiments with R_1^+. Fig. 16 shows the isolation rates for this fault as a function of fault magnitude and sensor noise. When the fault magnitude was large enough, the symbols were correctly generated and the fault correctly isolated, even for the highest levels of noise. However, as fault magnitude decreased and noise increased, the wrong fault was sometimes isolated. R_1^+ produces a first-order change on the voltage. If this change is detected early, then the transient can be observed and the signature correctly computed as $0-$. If the change is detected after the initial transient, when the voltage has already reached a steady state, then the slope is observed to be zero, so -0 is computed, thus isolating V_B^- as the fault. The incorrect signature was more likely to be computed when the fault magnitude was low and sensor noise was high, because detection of the change in $V_B(t)$ is more likely to occur after the transient. Knowledge of expected fault magnitudes and noise levels can help tune the fault detector parameters to correctly compute these features. For the other faults, this problem did not occur except for a few cases with V_B^+ and V_B^- when high noise caused the slope computation to be $+$ or $-$ instead of 0, which can be viewed as a false alarm in the slope computation. This is also the explanation for the decrease in $|F_n|$ and the $f \in F_n$ percentages for some of these scenarios, e.g., the correct fault, V_B^+, gets eliminated from F_n when the slope is incorrectly determined to be $-$.

IX. Conclusions

We have presented an event-based modeling and diagnosis methodology applied to parametric faults in continuous systems and demonstrated its application to an electrical power system testbed. The main issue in applying DES approaches is creating a system model that captures all relevant system behavior. Quantization-based abstractions create large, nondeterministic models. On the other hand, our qualitative abstraction approach systematically creates event-based models of faulty system behavior given a continuous model of the system, which can be used to develop an event-based diagnoser and determine diagnosability of the system. The automatic model construction contrasts to most current DES approaches, where models are created by hand. The qualitative abstraction enables a diagnosis approach that applies well to continuous systems. The approach was applied to ADAPT, which is a complex electrical power system. Detailed simulation experiments ex-

TABLE IV
ADAPT EXPERIMENTS WITH DIFFERENT FAULT MAGNITUDES AND NOISE LEVELS

Faults		Performance Parameters											
Fault	Magnitude	$t_d - t_f$ (s)			$t_i - t_f$ (s)			$\lvert F_n \rvert$			$f \in F_n$ (%)		
		N_0	N_1	N_2	N_0	N_1	N_2	N_0	N_1	N_2	N_0	N_1	N_2
V_B^+	+ 0.10	0.50	1.20	1.95	0.50	1.20	3.65	2.00	1.40	1.80	100	70	80
	+ 0.20	0.00	0.40	0.80	0.00	0.40	1.45	2.00	2.00	1.90	100	100	90
	+ 0.40	0.00	0.00	0.35	0.00	0.00	1.65	2.00	2.00	1.80	100	100	80
V_B^-	− 0.10	0.50	1.25	1.80	17.00	17.75	18.30	1.00	1.00	1.00	100	100	100
	− 0.20	0.00	0.50	0.95	16.50	17.00	21.00	1.00	1.00	0.90	100	100	80
	− 0.40	0.00	0.05	0.30	16.50	16.55	14.8	1.00	1.00	1.00	100	100	80
I_{L1}^+	+ 0.10	0.50	0.20	0.45	0.50	0.20	0.45	3.00	3.00	3.00	100	100	100
	+ 0.20	0.00	0.00	0.00	0.00	0.00	0.00	3.00	3.00	3.00	100	100	100
	+ 0.40	0.00	0.00	0.00	0.00	0.00	0.00	3.00	3.00	3.00	100	100	100
I_{L1}^-	− 0.10	0.00	0.35	0.50	17.50	17.85	18.00	2.00	2.00	2.00	100	100	100
	− 0.20	0.00	0.00	0.00	17.50	17.50	17.50	2.00	2.00	2.00	100	100	100
	− 0.40	0.00	0.00	0.00	17.50	17.50	17.50	2.00	2.00	2.00	100	100	100
I_{L2}^+	+ 0.10	0.50	0.20	0.45	18.00	17.70	17.95	3.00	3.00	3.00	100	100	100
	+ 0.20	0.00	0.00	0.00	17.50	17.50	17.50	3.00	3.00	3.00	100	100	100
	+ 0.40	0.00	0.00	0.00	17.50	17.50	17.50	3.00	3.00	3.00	100	100	100
I_{L2}^-	− 0.10	0.00	0.35	0.50	17.50	17.85	18.00	2.00	2.00	2.00	100	100	100
	− 0.20	0.00	0.00	0.00	17.50	17.50	17.50	2.00	2.00	2.00	100	100	100
	− 0.40	0.00	0.00	0.00	17.50	17.50	17.50	2.00	2.00	2.00	100	100	100
C_0^-	× 0.99	0.00	0.45	0.60	2.00	2.80	13.00	1.00	1.00	1.00	100	100	100
	× 0.98	0.00	0.00	0.10	1.00	1.00	1.05	1.00	1.00	1.00	100	100	100
	× 0.96	0.00	0.00	0.00	0.50	0.50	0.55	1.00	1.00	1.00	100	100	100
R_1^+	× 2.00	2.00	4.05	7.20	4.50	15.25	22.05	1.00	1.00	1.00	100	60	30
	× 2.20	2.00	3.55	5.00	4.50	10.65	17.50	1.00	1.00	1.00	100	100	90
	× 2.40	2.00	3.05	4.75	4.00	8.25	11.75	1.00	1.00	1.00	100	100	100
R_{L1}^+	× 1.05	0.50	0.30	0.50	18.00	17.80	18.00	2.00	2.00	2.00	100	100	100
	× 1.10	0.00	0.00	0.00	17.50	17.50	17.50	2.00	2.00	2.00	100	100	100
	× 1.20	0.00	0.00	0.00	8.50	17.50	17.50	1.00	2.00	2.00	100	100	100
R_{L1}^-	× 0.95	0.00	0.05	0.40	0.00	0.05	0.4	3.00	3.00	3.00	100	100	100
	× 0.90	0.00	0.00	0.00	0.00	0.00	0.00	3.00	3.00	3.00	100	100	100
	× 0.80	0.00	0.00	0.00	3.50	0.00	0.00	1.00	3.00	3.00	100	100	100
R_{L2A}^+	× 1.05	1.00	0.95	1.40	18.50	18.45	18.90	2.00	2.00	2.00	100	100	100
	× 1.10	0.50	0.50	0.50	18.00	18.00	18.00	2.00	2.00	2.00	100	100	100
	× 1.20	0.00	0.00	0.10	17.50	17.50	17.60	2.00	2.00	2.00	100	100	100
R_{L2A}^-	× 0.95	1.00	0.90	0.95	18.50	18.40	18.45	3.00	3.00	3.00	100	100	100
	× 0.90	0.50	0.15	0.45	18.00	17.65	17.95	3.00	3.00	3.00	100	100	100
	× 0.80	0.00	0.00	0.00	17.50	17.50	17.50	3.00	3.00	3.00	100	100	100

amined the effects of fault magnitude and sensor noise on the robustness of the approach. If symbol generation is correct, then the true fault is always included in the final candidate list. The approach can easily be coupled with fault identification methods developed in previous work [33] to complete the diagnosis.

An important practical issue in applying this approach is to ensure correct detection of the signatures and measurement orderings. Measurement orderings are more reliable for systems with slow dynamics relative to the sampling frequency of the sensors. Correct detection is also a function of the amount of sensor noise and the reliability of the fault detectors. The fault detectors must be tuned to have similar sensitivity relative to each other, so that deviations are detected in a timely manner and measurement orderings are not violated. Still, we have demonstrated the practicality of our approach by applying it to real systems in the electrical domain, described here, and in the robotics domain, described in [12]. Additional discussion of practical issues can be found in [38]. In future work, we will develop more robust solutions using a stochastic framework.

APPENDIX

PROOF OF LEMMA 1

Proof: Since the synchronous product must accept fault traces that obey all individual ordering constraints and includes all measurement deviation events for the fault, it accepts all valid measurement deviation sequences, i.e., all $\lambda \in L_f$, and no others. ■

PROOF OF LEMMA 2

Proof: Assume f_i is not distinguishable from f_j, i.e., $f_i \sim f_j$. Then by definition, there must exist some maximal sequence of effects on the measurements by f_i that f_j can also produce. Fault traces capture these effects, and are by definition maximal. Therefore, there must exist some fault trace for f_i, i.e., some $\lambda_{f_i} \in L_{f_i}$, and some sequence of measurement deviations produced by f_j that is not distinct from λ_{f_i}. Since the possible sequences of measurement deviations produced by f_j is $\{\lambda : \lambda \sqsubseteq \lambda_{f_j}, \lambda_{f_j} \in L_{f_j}\}$, then λ_{f_i} must be a prefix of some fault trace $\lambda_{f_j} \in L_{f_j}$. Therefore, if $f_i \sim f_j$ then there exits some $\lambda_{f_i} \in L_{f_i}$ and $\lambda_{f_j} \in L_{f_j}$ such that $\lambda_{f_i} \sqsubseteq \lambda_{f_j}$. By the contrapositive, if there does not exist $\lambda_{f_i} \in L_{f_i}$ and $\lambda_{f_j} \in L_{f_j}$ such that $\lambda_{f_i} \sqsubseteq \lambda_{f_j}$, then $f_i \nsim f_j$. ∎

PROOF OF LEMMA 3

Proof: $\mathcal{D}^*_{\{f\}}$ extends $\mathcal{L}_f$ by defining D and Y. Therefore, by definition of $\mathcal{L}_f$, $\mathcal{D}^*_{\{f\}}$ must accept all $\lambda_f \in L_f$. By definition of Y, $Y(q)$ for all $q \in A$ must map to $\{f\}$, since $q_0 \notin A$. So, $\mathcal{D}^*_{\{f\}}$ uniquely isolates f. ∎

PROOF OF THEOREM 1

Proof: Assume $\mathcal{D}_{F_i}$ isolates all $f_i \in F_i$, and $\mathcal{D}_{F_j}$ isolates all $f_j \in F_j$. Then for some fault $f_i \in F_i$ with some trace $\lambda_{f_i} \in L_{f_i}$, $\mathcal{D}_{F_i}$ must accept λ_{f_i}, and this corresponds to some $q_i \in A_i$, where $f_i \in Y_i(q_i)$. The first event in λ_{f_i} corresponds to a state in Q, by definition of δ, and the state maps to a diagnosis containing f_i by definition of Y. For some prefix λ of λ_{f_i}, there is a corresponding state $q_i \in Q_i$ where $f_i \in Y_i(q_i)$, given that $\mathcal{D}_{F_i}$ isolates f_i. By the same logic, $\lambda\sigma \sqsubseteq \lambda_{f_i}$, corresponds to a state $q_{i+1} \in Q_i$ where $f_i \in Y_i(q_{i+1})$. If λ corresponds to a state $(q_i, *) \in Q$ with $f_i \in Y((q_i, *))$, then by definition of δ, $\lambda\sigma$ corresponds to a state $(q_{i+1}, *) \in Q$, and by definition of Y, $f_i \in Y((q_{i+1}, *))$ since $f_i \in Y((q_i, *))$ and $f_i \in Y_i(q_{i+1})$. By induction, λ_{f_i} corresponds to a state in Q and in A since λ_{f_i} is also accepted by $\mathcal{D}_{F_i}$, and the corresponding accepting state contains f_i in its diagnosis. Since λ_{f_i} was general, the composed diagnoser isolates f_i. Since f_i was general, the diagnoser isolates all $f \in F_i$. The same reasoning applies for all $f \in F_j$. Therefore, $\mathcal{D}_{F_i} \Delta \mathcal{D}_{F_j}$ isolates all $f \in F_i \cup F_j$. ∎

PROOF OF THEOREM 2

Proof: Assume some $f_i \in F$ with fault trace $\lambda_{f_i} \in L_{f_i}$. $\mathcal{D}^*_F$ accepts λ_{f_i} and for corresponding accepting state q^a, $f_i \in Y(q^a)$ by Corollary 1 and the definition of isolation. Since F is diagnosable, there is no $f_j \in F$ with fault trace $\lambda_{f_j} \in L_{f_j}$ where $\lambda_{f_i} \sqsubseteq \lambda_{f_j}$. Therefore, $f_j \notin Y(q^a)$. So, $Y(q^a) = \{f_i\}$ and $\mathcal{D}^*_F$ uniquely isolates each $f \in F$. Assume that $\mathcal{D}^*_F$ uniquely isolates each $f \in F$. Then each possible fault trace λ_{f_i} has an associated accepting state q^a, where $Y(q^a) = \{f_i\}$. Thus, there cannot be some $\lambda \sqsubseteq \lambda_{f_j}$ for $f_i \neq f_j$ that can reach q_a, otherwise $f_j \in Y(q^a)$. Therefore, $f_i \nsim f_j$, so F is diagnosable. Thus S is diagnosable if and only if $\mathcal{D}^*_F$ uniquely isolates each $f \in F$. ∎

REFERENCES

[1] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "Failure diagnosis using discrete-event models," *IEEE Transactions on Control Systems Technology*, vol. 4, no. 2, pp. 105–124, Mar. 1996.

[2] ——, "Diagnosability of discrete-event systems," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1555–1575, Sept. 1995.

[3] S. H. Zad, R. Kwong, and W. Wonham, "Fault diagnosis in discrete-event systems: framework and model reduction," *IEEE Transactions on Automatic Control*, vol. 48, no. 7, pp. 1199–1212, July 2003.

[4] S. Jiang and R. Kumar, "Failure diagnosis of discrete-event systems with linear-time temporal logic specifications," *IEEE Transactions on Automatic Control*, vol. 49, no. 6, pp. 934–945, June 2004.

[5] J. Kurien, X. Koutsoukos, and F. Zhao, "Distributed diagnosis of networked embedded systems," in *Proceedings of the 13th International Workshop on Principles of Diagnosis (DX-02)*, Semmering, Austria, May 2002, pp. 179–188.

[6] A. Benveniste, E. Fabre, S. Haar, and C. Jard, "Diagnosis of asynchronous discrete-event systems: a net unfolding approach," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 714–727, May 2003.

[7] V. Chandra, Z. Huang, and R. Kumar, "Automated control synthesis for an assembly line using discrete event system control theory," *IEEE Trans. on Systems, Man and Cybernetics, Part C*, vol. 33, no. 2, pp. 284–289, May 2003.

[8] J. Lunze, "Diagnosis of quantized systems based on a timed discrete-event model," *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, vol. 30, no. 3, pp. 322–335, 2000.

[9] J. Lunze and J. Shröder, "Sensor and actuator fault diagnosis of systems with discrete inputs and outputs," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 2, pp. 1096–1107, Apr. 2004.

[10] X. Koutsoukos, P. Antsaklis, J. Stiver, and M. Lemmon, "Supervisory control of hybrid systems," *Proceedings of IEEE*, vol. 88, no. 7, pp. 1026–1049, July 2000.

[11] P. Mosterman and G. Biswas, "Diagnosis of continuous valued systems in transient operating regions," *IEEE Transactions on Systems, Man and Cybernetics, Part A*, vol. 29, no. 6, pp. 554–565, 1999.

[12] M. J. Daigle, X. D. Koutsoukos, and G. Biswas, "Distributed diagnosis in formations of mobile robots," *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 353–369, Apr. 2007.

[13] I. Roychoudhury, G. Biswas, and X. Koutsoukos, "Designing distributed diagnosers for complex continuous systems," *IEEE Transactions on Automation Science and Engineering*, to appear.

[14] M. Daigle, X. Koutsoukos, and G. Biswas, "Fault diagnosis of continuous systems using discrete-event methods," in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 2626–2632.

[15] S. Poll, A. Patterson-Hine, J. Camisa, D. Nishikawa, L. Spirkovska, D. Garcia, D. Hall, C. Neukom, A. Sweet, S. Yentus, C. Lee, J. Ossenfort, I. Roychoudhury, M. Daigle, G. Biswas, X. Koutsoukos, and R. Lutz, "Evaluation, selection, and application of model-based diagnosis tools and approaches," in *AIAA Infotech@Aerospace 2007 Conference Proceedings*, May 2007.

[16] Y.-L. Chen and G. Provan, "Modeling and diagnosis of timed discrete event systems – a factory automation example," in *Proceedings of the American Control Conference*, June 1997, pp. 31–36.

[17] S. H. Zad, R. H. Kwong, and W. M. Wonham, "Fault diagnosis in timed discrete-event systems," in *Proceedings of the 38th Conference on Decision and Control*, Dec. 1999, pp. 1756–1761.

[18] S. Tripakis, "Fault diagnosis for timed automata," in *Formal Techniques in Real Time and Fault Tolerant Systems (FTRTFT02)*, ser. Lecture Notes in Computer Science, vol. 2469. Springer, 2002, pp. 205–221.

[19] C. Dousson, "Alarm driven supervision for telecommunication network: Ii – on-line chronicle recognition," *Annales des Telecommunications*, vol. 51, no. 910, pp. 501–508, 1996.

[20] M.-O. Cordier and C. Dousson, "Alarm driven monitoring based on chronicles," in *Proceedings of the 4th Symposium on Fault Detection Supervision and Safety for Technical Processes (Safeprocess 2000)*, June 2000, pp. 286–291.

[21] J. M. Kościelny, "Fault isolation in industrial processes by the dynamic table of states method," *Automatica*, vol. 31, no. 5, pp. 747–753, 1995.

[22] J. M. Kościelny and K. Zakroczymski, "Fault isolation method based on time sequences of symptom appearance," in *Proceedings of IFAC SafeProcess 2000*, 2000.

[23] V. Puig, J. Quevedo, T. Escobet, and B. Pulido, "On the integration of fault detection and isolation in model-based fault diagnosis," in *Proceedings of the 16th International Workshop on Principles of Diagnosis (DX-05)*, 2005, pp. 227–232.

[24] V. Puig, F. Schmid, J. Quevedo, and B. Pulido, "A new fault diagnosis algorithm that improves the integration of fault detection and isolation," in *Proceedings of the 44th IEEE Conference on Decision and Control*, Dec. 2005, pp. 3809–3814.

[25] J. Gertler, *Fault Detection and Diagnosis in Engineering Systems*. New York: Marcel Dekker, 1998.

[26] L. Console, C. Picardi, and M. Ribaudo, "Process algebras for systems diagnosis," *Artificial Intelligence*, vol. 142, no. 1, pp. 19–51, 2002.

[27] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *Systems Dynamics: Modeling and Simulation of Mechatronic Systems*. New York: John Wiley & Sons, Inc., 2000.

[28] A. Samantaray, K. Medjaher, B. O. Bouamama, M. Staroswiecki, and G. Dauphin-Tanguy, "Diagnostic bond graphs for online fault detection and isolation," *Simulation Modelling Practice and Theory*, vol. 14, pp. 237–262, 2006.

[29] A. K. Samantaray and B. O. Bouamama, *Model-based process supervision: a bond graph approach*. Springer London, 2008.

[30] M. Hirsch and S. Smale, *Differential Equations, Dynamical Systems, and Linear Algebra*. Academic, 1974.

[31] E. Griepentrog and R. März, *Differential-algebraic equations and their numerical treatment*. Teubner, 1986.

[32] R. E. Kirk, *Statistics: An Introduction*. Fort Worth: Harcourt Brace, 1999.

[33] G. Biswas, G. Simon, N. Mahadevan, S. Narasimhan, J. Ramirez, and G. Karsai, "A robust method for hybrid diagnosis of complex systems," in *Proceedings of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes*, June 2003, pp. 1125–1131.

[34] M. Basseville and I. Nikiforov, *Detection of Abrupt Changes - Theory and Application*. Prentice-Hall Inc., 1993.

[35] M. Djeziri, R. Merzouki, B. Bouamama, and G. Dauphin-Tanguy, "Robust fault diagnosis by using bond graph approach," *IEEE/ASME Transactions on Mechatronics*, vol. 12, no. 6, pp. 599–611, Dec. 2007.

[36] E.-J. Manders, S. Narasimhan, G. Biswas, and P. Mosterman, "A combined qualitative/quantitative approach for fault isolation in continuous dynamic systems," in *SafeProcess 2000*, vol. 1, Budapest, Hungary, June 2000, pp. 1074–1079.

[37] M. Daigle, X. Koutsoukos, and G. Biswas, "Relative measurement orderings in diagnosis of distributed physical systems," in *43rd Annual Allerton Conference on Communication, Control, and Computing*, Sept. 2005, pp. 1707–1716.

[38] M. J. Daigle, "A qualitative event-based approach to fault diagnosis of hybrid systems," Ph.D. dissertation, Vanderbilt University, 2008.

[39] M. Daigle, X. Koutsoukos, and G. Biswas, "A qualitative approach to multiple fault isolation in continuous systems," in *Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence*, 2007, pp. 293–298.

[40] M. Daigle, I. Roychoudhury, G. Biswas, and X. Koutsoukos, "Efficient simulation of component-based hybrid models represented as hybrid bond graphs," in *Hybrid Systems: Computation and Control*, ser. LNCS. Springer-Verlag, 2007, vol. 4416, pp. 680–683.

[41] M. Ceraolo, "New dynamical models of lead-acid batteries," *IEEE Transactions on Power Systems*, vol. 15, no. 4, pp. 1184–1190, Nov. 2000.

[42] S. Barsali and M. Ceraolo, "Dynamical models of lead-acid batteries: Implementation issues," *IEEE Transactions on Energy Conversion*, vol. 17, no. 1, pp. 16–23, Mar. 2002.

Matthew J. Daigle (S'07–M'08) received the B.S. degree in Computer Science and Computer and Systems Engineering from Rensselaer Polytechnic Institute, Troy, NY, in 2004, and the M.S. and Ph.D. degrees in Computer Science from Vanderbilt University, Nashville, TN, in 2006 and 2008, respectively.

From September 2004 to May 2008, he was a Graduate Research Assistant with the Institute for Software Integrated Systems and Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN. During the summers of 2006 and 2007, he was an intern with Mission Critical Technologies, Inc., at NASA Ames Research Center. Since June 2008, he has been with the University of California, Santa Cruz, at NASA Ames Research Center. His current research interests include physics-based modeling, model-based diagnosis and prognosis, and hybrid systems.

Dr. Daigle is a recipient of the 4.0 Award and Ricketts Prize from Rensselaer Polytechnic Institute, and a University Graduate Fellowship from Vanderbilt University. He is a member of the IEEE.

Xenofon D. Koutsoukos (S'95–M'00–SM'07) received the Diploma in electrical and computer engineering from the National Technical University of Athens, Athens, Greece, in 1993, M.S. degrees in electrical engineering and applied mathematics and the Ph.D. degree in electrical engineering from the University of Notre Dame, Notre Dame, IN, in 1998 and 2000, respectively.

From 2000 to 2002, he was a member of Research Staff with the Xerox Palo Alto Research Center, Palo Alto, CA, working in the Embedded Collaborative Computing Area. Since 2002, he has been with the Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, where he is currently an Assistant Professor and a Senior Research Scientist in the Institute for Software Integrated Systems. His research interests include hybrid systems, real-time embedded systems, sensor networks, and cyber-physical systems. He currently serves as Associate Editor for the ACM Transactions on Sensor Networks and for Modelling Simulation Practice and Theory.

Dr. Koutsoukos is a senior member of IEEE and a member of ACM. He was the recipient of the National Science Foundation CAREER Award in 2004.

Gautam Biswas (S'78–M'82–SM'91) is a Professor of Computer Science and Computer Engineering in the EECS Department and a Senior Research Scientist at the Institute for Software Integrated Systems (ISIS) at Vanderbilt University. He has a Ph.D. degree in Computer Science from Michigan State University in E. Lansing, MI.

Prof. Biswas 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 has worked on fault-adaptive control of fuel transfer systems for aircraft, and Advanced Life Support systems for NASA. He has also initiated new projects in distributed monitoring and diagnosis and prognosis and health management of complex systems. In other research projects, he is involved in developing simulation-based environments for learning and instruction and planning and scheduling algorithms for distributed real-time environments. His research has been supported by funding from NASA, NSF, DARPA, and ONR.

Dr. Biswas is an associate editor of the IEEE Transactions on Systems, Man, and Cybernetics, Part A. He has served on the Program Committee of a number of conferences. He is a senior member of the IEEE Computer Society, ACM, AAAI, and the Sigma Xi Research Society.