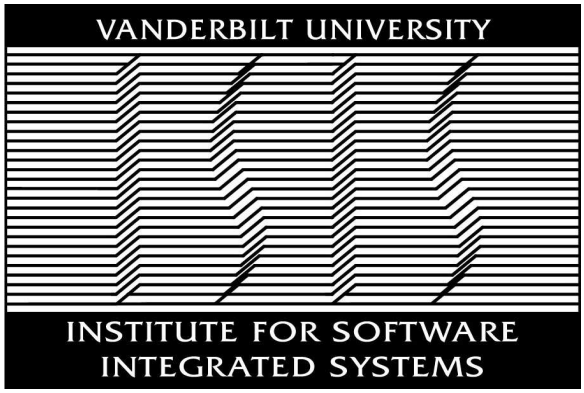


Institute for Software Integrated Systems
Vanderbilt University
Nashville, Tennessee, 37235

An Integrated Approach to Parametric and Discrete Fault Diagnosis in Hybrid Systems

Matthew Daigle, Xenofon Koutsoukos, and
Gautam Biswas

TECHNICAL REPORT

ISIS-07-815

An Integrated Approach to Parametric and Discrete Fault Diagnosis in Hybrid Systems

Matthew Daigle, Xenofon Koutsoukos, and Gautam Biswas

Institute for Software Integrated Systems (ISIS)
Department of Electrical Engineering and Computer Science
Vanderbilt University, Nashville, TN 37235, USA
matthew.j.daigle,xenofon.koutsoukos,gautam.biswas@vanderbilt.edu

Abstract. Fault diagnosis is crucial for ensuring the safe operation of complex engineering systems. These systems are typically hybrid in nature, therefore, model-based diagnosis requires hybrid system models. Previous work in hybrid systems diagnosis, however, has focused either on parametric or discrete faults. We present an integrated approach for diagnosis of both parametric and discrete faults in hybrid systems that encompasses a compact hybrid systems modeling approach and an efficient qualitative fault isolation scheme. Experimental results from a case study performed on a complex electrical power system demonstrate the effectiveness of the approach.

1 Introduction

Fault diagnosis is crucial for ensuring the safe operation of complex engineering systems. Faults and degradations need to be quickly identified so that corrective actions can avoid catastrophic situations. Most real-world, embedded systems are hybrid in nature. In such systems, a discrete abstraction may be adequate for model-based diagnosis [1]. Purely discrete techniques, however, are inadequate for systems that combine complex continuous and discrete behaviors [2]. Hybrid models have to be employed for correct tracking and diagnosis.

The Advanced Diagnostics and Prognostics Testbed (ADAPT) [3], deployed at NASA Ames Research Center, is functionally representative of a spacecraft's electrical power system. Over fifty relays and circuit breakers configure the system into different modes of operation. Therefore, the system behavior is naturally hybrid. Parametric faults, such as changes in resistance and inductance values, can occur in the components. Discrete faults, such as relays becoming stuck, may also occur. In such systems, it is important to address both types of faults in a comprehensive, unified framework.

Very little previous work, however, has addressed a combined approach to hybrid diagnosis. The approach of [4] is suitable only for simple hybrid automata and a particular application. The approach of [5] uses the parity relations approach, which is difficult to apply to nonlinear systems with multiplicative faults. Methods have addressed either discrete fault [1,6–11] or parametric fault diagnosis [12–14]. In discrete approaches, fault modes are added to the nominal system

model for each discrete fault. Diagnosability of hybrid systems has been investigated in this framework in [15, 16]. This style of modeling typically reduces the discrete diagnosis task to a mode estimation problem [6–9]. These approaches use a combination of continuous state and mode observers [6, 7] or particle filters [8,9]. Others employ reasoners to guarantee that only modes consistent with the observations are tracked using consistency-based approaches [10,11]. To handle parametric faults under this scheme, a discrete abstraction can be generated via quantization [17], but this results in large, nondeterministic models that do not include fault transient information, which is critical to quick diagnosis of parametric faults [18].

Parametric faults have also been addressed in hybrid systems. In [12], qualitative techniques produce parametric fault candidates assuming only controlled mode changes. The approach of [13, 14] also performs qualitative isolation, but includes reasoning to handle both controlled and autonomous mode changes that occur after fault occurrence. Discrete faults can be captured at a very detailed level as parameter changes, but this produces highly nonlinear, high order models that are difficult to simulate and analyze efficiently.

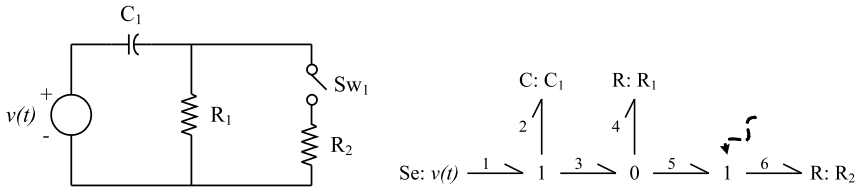
In contrast, we present an integrated model-based approach to diagnosing both parametric and discrete faults in hybrid systems. This extends our previous work in diagnosis of parametric faults in hybrid systems [13, 14, 18] by including discrete faults, resulting in a unified hybrid diagnosis methodology. We establish a compact, integrated hybrid modeling framework, using hybrid bond graphs [19], that can represent both parametric and discrete faults. We apply our approach using a case study on a real hybrid system, ADAPT, and demonstrate our techniques with experiments performed on the actual testbed.

2 Modeling Hybrid Systems

2.1 Hybrid Bond Graphs

We develop component-based models of hybrid physical systems using hybrid bond graphs (HBGs) [19]. HBGs extend bond graphs [20], which define an energy-based, multi-domain, 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. HBGs can also be transformed to hybrid automata [13].

Throughout the paper, we will illustrate the diagnosis methodology with a circuit example. The schematic and HBG are shown in Figs. 1(a) and 1(b), respectively. In bond graphs, vertices represent components, and bonds, drawn as half arrows, represent ideal energy connections between them. 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 $e_i \times f_i$ defines the rate of energy transfer through the bond. In the electrical 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



(a) Schematic. (b) Hybrid bond graph.

Fig. 1. Switched circuit example.

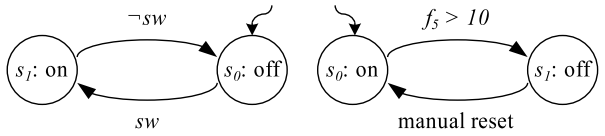
all e are equal and $\sum f = 0$). Other bond graph elements model energy dissipation as resistances (R, where $e = Rf$), energy storage as capacitances (C, where $\dot{e} = \frac{1}{C}f$) and inductances (I, where $\dot{f} = \frac{1}{I}e$), and energy sources as sources of flow (Sf, where $f = u$) and effort (Se, where $e = u$). The constituent equations of the bond graph elements form a set of differential algebraic equations that describe the continuous system behavior.

Hybrid bond graphs [19] extend bond graphs by introducing *controlled junctions*, denoted in Fig. 1(b) by the dashed arrow. Controlled junctions act as ideal switches, enabling a junction to be in either the on or off mode. Off 1-junctions behave as sources of zero flow, so they impose $f = 0$ on all their bonds. Similarly, off 0-junctions act as sources of zero effort. When on, controlled junctions behave as normal junctions. In the circuit example, the controlled junction models the switch. When the switch is closed, the junction is on, and when the switch is open, the junction is off, forcing the current through R_2 to be zero.

The switching behavior of a controlled junction is defined by a *control specification* (CSPEC), modeled as a finite automaton [14, 19]. A CSPEC defines a finite number of states. The state transitions may be attributed to controlled or autonomous events. The output of the CSPEC determines whether the junction is on or off. So, the system mode is defined implicitly by the individual modes of all the controlled junctions, providing a concise representation of the hybrid system model. A single mode change may correspond to multiple junctions switching mode. Therefore, events may be shared over different CSPECs. Given an event e and the current system mode q, the new system mode q' is given by $q' = \delta(e, q)$, where the system mode transition function δ simply applies e to all CSPECs, and obtains the new CSPEC output, i.e., the junction mode, for each controlled junction.

Associated with each system mode q is a continuous bond graph. The computational model for each mode (e.g., state-space equations or signal flow graphs) can be derived systematically [20]. The computational model for a new mode can also be automatically generated from the previous mode efficiently [21]. This type of modeling framework offers significant advantages for large hybrid systems like ADAPT, because it avoids preenumeration of system modes.

Example CSPECs for the circuit are given in Fig. 2. Events are generated when Boolean predicates derived from system variables evaluate to true. The



(a) Relay CSPEC. (b) Circuit breaker CSPEC.

Fig. 2. Circuit switch CSPECs.

CSPEC for a relay controlled by switching signal sw is shown in Fig. 2(a). For a circuit breaker (Fig. 2(b)), the junction behavior is autonomous. It is initially in s_0, where the junction is on. When the current through the circuit breaker, f_5, exceeds the value 10, the CSPEC transitions to s_1, where the junction is off.

2.2 Modeling Faults

We focus on the diagnosis of single, abrupt, persistent faults in hybrid systems. We classify hybrid system faults into two categories, (*i*) *parametric faults*, and (*ii*) *discrete faults*. Parametric faults cover partial failures or degradations in system components. Discrete faults are associated with switching in components.

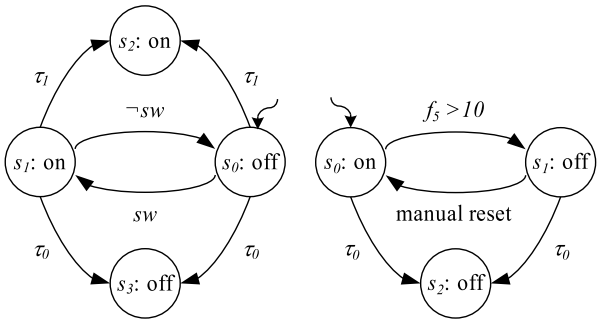
Definition 1 (Parametric Fault) *A* parametric fault *is an unexpected change in the value of a system parameter in the model.*

System components appear as HBG model parameters, so can model faults that affect system behavior. Abrupt parametric faults are defined as a step change in a component parameter value. In the circuit example, parametric faults may include increase and decrease in resistance (R_1 or R_2) and capacitance (C_1) values.

Definition 2 (Discrete Fault) *A* discrete fault *is a discrepancy between the actual and expected mode of a switching element in the model.*

Discrete faults in the circuit include switch malfunctions. For example, the switch may be commanded to close, but remain stuck open. Also, it may unexpectedly open or close without a command. In HBGs, mode changes are modeled using controlled junctions, so discrete faults are captured as unexpected changes in junction mode. Because the junction mode is determined using a CSPEC, we introduce new unobservable fault events in the CSPEC and link discrete faults to them. Mode changes in components may correspond to many junctions changing mode, so these fault events may be shared among the different CSPECs of the component. The linking of discrete faults to fault events in the CSPEC gives, as with parametric faults, a one-to-one mapping between model entities and faults. This leads to the following definition of the CSPEC to include discrete faults.

Definition 3 (Control Specification) *A* control specification *is a tuple* $\mathcal{M} = (S, E, t, o, s_0)$*, where* S *is the finite set of states,* $E = E_o \cup E_u$ *is the set of*



(a) Relay CSPEC. (b) Circuit breaker CSPEC.

Fig. 3. Circuit switch extended CSPECs.

observable and unobservable (fault) events, $t : S \times E \rightarrow S$ *is the transition function,* $o : S \rightarrow \{\text{on,off}\}$ *is the output function, and* $s_0 \in S$ *is the initial state.*

The extended example CSPECs for the circuit are given in Fig. 3. For the relay CSPEC (Fig. 3(a)), we introduce fault events τ_0 and τ_1. When τ_1 occurs, the CSPEC moves to s_2, where the junction is stuck on. If the junction was previously off, then this fault manifests in the measurements immediately, i.e., the switch closes by itself. Otherwise, it will only manifest when *sw* becomes true, i.e., the switch becomes stuck closed. The case is similar for the τ_1 event. For the circuit breaker CSPEC (Fig. 3(b)), only the stuck off fault, τ_0, is appropriate, and the behavior is similar. The circuit breaker may open due to the current limit being exceeded, which is nominal behavior, or may open due to a fault, i.e., it opens when the current limit has not yet been exceeded.

3 Hybrid Diagnosis Approach

3.1 Diagnosis Architecture

Our method for integrated diagnosis of parametric and discrete faults in hybrid systems extends the Hybrid Transcend [14] approach for diagnosing single, abrupt, parametric faults in hybrid systems (see Fig. 4). The diagnosis is based on analysis of fault transients in the residual signal [18]. When faults occur, they produce deviations in measurements from their expected values. Our diagnosis model expresses these deviations as fault signatures, which are matched against observed deviations to isolate faults. A hybrid observer computes the expected behavior of the plant. The observer, a switched extended Kalman filter, tracks the continuous behavior in individual modes of operation. An accompanying automata scheme implements the CSPECs, and executes controlled and autonomous mode changes [14].

The difference between observed outputs, $\mathbf{y}(t)$, and expected outputs, $\hat{\mathbf{y}}(t)$, defines the residual, $\mathbf{r}(t)$. Faults will cause statistically significant differences

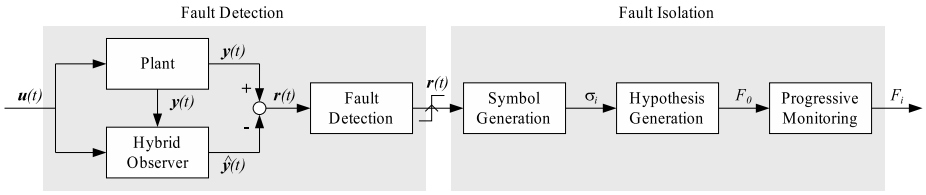


Fig. 4. Diagnosis architecture.

between the observed and expected outputs. A parametric fault increases or decreases a parameter value. A discrete fault results in a change in the system mode. In either case, a difference will manifest between expected and observed behavior. This triggers the fault detector, which determines if the observed differences are statistically significant, indicating a fault. Due to sensor noise and model imperfections, we employ a statistical significance test to robustly determine if the residual is nonzero using a sliding window technique [22].

Fault isolation begins when a fault is detected. *Symbol generation* symbolically abstracts the residual signals of the deviating measurements into qualitative features. *Hypothesis generation* produces the initial fault candidate set F_0 from the initial deviation. *Progressive monitoring* prunes the fault set as further deviations occur by dropping inconsistent candidates. The following subsections describe these steps in more detail.

3.2 Qualitative Fault Isolation

Fault Signatures The fault transients caused by abrupt faults are represented as symbolic predictions for qualitative fault isolation. Assuming that the system output is continuous and continuously differentiable within a mode except at the points of fault occurrence[1], the transient response after fault occurrence can be approximated by a Taylor series expansion. Measurement transients are described using the magnitude and the derivative values of the residual signal [18]. This is the basis for establishing a signature for a fault transient. TRANSCEND abstracts these signatures 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 measurement residual due to the occurrence of a fault. Only magnitude and slope of a signal can be reliably measured, so we condense higher order signatures to the magnitude change symbol and the first nonzero derivative change, e.g., `000-+-+` becomes `0-`, and `+-+-+-+` becomes `+-`. The first symbol represents the immediate direction of abrupt change (a discontinuity) and the second symbol represents the slope. For `+0` and `-0`, the `0` slope symbol implies that the fault will cause a jump but no subsequent change in the slope. This will occur for some discrete faults and some sensor faults (e.g. sensor

[1] We assume that parametric faults do not occur at the same time as a mode change.

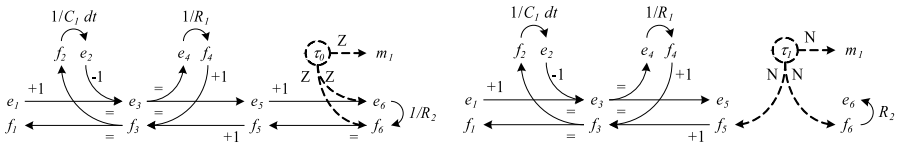
bias). We omit signatures of `++` and `--` because they represent physically unstable systems. Qualitative arithmetic may result in ambiguities in the signatures, denoted by `*`. Details may be found in [18].

We augment fault signatures to include information directly indicating discrete faults. Because discrete faults cause junctions to change mode, they cause some variable values to go from nonzero to zero (for a junction turning off) or go from zero to nonzero (for a junction turning on). For example, if the circuit switch is expected to be on, but is off, then we will observe the current f_6 through R_2 go to zero. If it is expected to be off, but is on, we will observe f_6 go to a nonzero value. Measuring variables affected in this manner provides additional discriminatory information, because parametric faults are unlikely to cause this behavior. If the expected switch state is correct and a fault in R_2 occurs, then we will not observe this behavior because a finite change in value cannot force zero to nonzero or nonzero to zero behavior in f_6. Therefore, we include additional symbols `N`, `Z`, and `X`, implying zero to nonzero, nonzero to zero, or no discrete change behavior in the measurement from the estimate.

Definition 4 (Fault Signature) *A* fault signature *for a fault f and measurement m is the qualitative effect of the occurrence of f on m, and is denoted by $\sigma_{f,m} = (s_1 s_2, s_3)$, where $s_1, s_2 \in \{$`+`,`0`,`-`,`*`$\}$, and $s_3 \in \{$`N`,`X`,`Z`,`*`$\}$. We denote the set of all fault signatures for fault f in mode q as $\Sigma_{f,q}$.*

In symbol generation, we extract symbolic features from the measured change in a residual that are matched to predicted fault signatures. When a significant deviation is detected on a residual, these symbols are computed and associated with the measurement, and this forms the observed fault signature. The symbols are derived by computing the initial direction of change and the successive slope of the change using a statistical significance test and a sliding window method [22]. We compute the discrete change symbol using the same techniques. After a fault is detected, we compute the means of the observation and the estimate over a small window (e.g., 5 samples). We then determine whether each signal belongs to a distribution with zero mean. If the estimate is nonzero and the measurement is zero, we report `Z`, and if the estimate is zero and the measurement is nonzero, we report `N`, otherwise, we report `X`.

Temporal Causal Graphs We compute predicted fault signatures using the temporal causal graph (TCG), derived from the bond graph of a given mode of the system [18]. The TCG captures the dynamics of the system, therefore can be used to predict the qualitative effects of faults 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 (dt), and parametric relations (e.g. $1/R_1$). The directionality of these edges is determined by *causality*, i.e., the input-output relations of the bond graph elements. Causality determines whether effort or flow is being imposed by a bond on an element and is derived automatically [20].



(a) TCG for the on mode. (b) TCG for the off mode.

Fig. 5. TCGs for the switched circuit.

We augment the TCG to capture the effect of discrete faults on the system variables by creating a new vertex in the TCG for each discrete fault event. We create new edge types linked to the appropriate junction variables. We also introduce the junction mode variable m_i for each controlled junction i, which the discrete fault will also affect. The discrete fault event is added to the TCG and edges made to the junction variables if it is possible from the last known mode that the discrete fault could have occurred, given the logic of the junction's CSPEC. The new edge labels are Z, causing a variable value to go to zero, and N, causing a variable value to go nonzero.

In the circuit example, consider the mode where the relay is closed. To correctly associate the fault event with the junction variables, we need to determine which variables are immediately affected by the change in the junction mode. The closed switch creates a configuration where a voltage is imposed on R_2, determining the current flow. An open switch, however, imposes zero current on R_2, therefore determining its voltage. This is directly derived by causal analysis of the HBG. Each 1-junction (0-junction) that is on has a flow (effort) determining bond. When a junction turns off as the result of a fault, the determining flow or effort will be immediately affected because it goes to zero. The other variable of the bond will, if not tied to a source, also be immediately affected, because a resistor element must absorb the causal change, and these variables are algebraically related. Further, since the overall resistance of the components connected to that bond is positive, the effort and flow always change in the same direction. In the circuit, this relates to the current and voltage through R_2. It is necessary to relate this to the voltage, because otherwise the effect of the discrete fault on that variable may not be correctly predicted. The TCG for this mode is shown in Fig. 5(a). The τ_0 fault will immediately affect e_6, f_6, and m_1.

When a switch is expected to be off, a nonzero current can only be explained by the switch turning on. In the HBG, if the junction is off, then its flows (for a 1-junction) or efforts (for a 0-junction) will no longer be zero, so these variables will be affected at the point of failure. The TCG in this mode is shown in Fig. 5(b). The τ_1 fault will immediately affect f_5, f_6, and m_1.

3.3 Hypothesis Generation

Symbol generation produces a qualitative value for the initial measurement deviation. The goal of hypothesis generation is, given this initial deviation σ_m, and

Algorithm 1 $F_0 \leftarrow$ `GenerateHypotheses`$(\sigma_m, \hat{q}(t_d))$

$F_0 \leftarrow \varnothing$
$Q_{t_f} \leftarrow$ `RollBack`$(\hat{q}(t_d))$
for all $\hat{q}(t_f^-) \in Q_{t_f}$ **do**
  $F_q \leftarrow$ `PropagateBackward`$(\sigma_m, \hat{q}(t_f^-))$
  for all $f \in F_q$ **do**
    $F_0 \leftarrow F_0 \cup \{(f, \delta(e_f, q(t_f^-))) : f \in F_q\}$

the hypothesized system mode at the time of fault detection $\hat{q}(t_d)$, to produce a consistent set of fault candidates. The procedure is shown as Algorithm 1.

At the time of fault occurrence, t_f, the system is in some mode $q(t_f^-)$. For parametric faults, $q(t_f^-) = q(t_f^+)$, but for discrete faults, a mode change is induced, i.e., $q(t_f^-) \neq q(t_f^+)$. For a discrete fault event e_f, $q(t_f^+) = \delta(e_f, q(t_f^-))$. For a parametric fault f, we take e_f to be $\varnothing$. When the fault is detected at $t_d \geq t_f$, the system is in some mode $q(t_d)$. Due to fault detector delays, it may be the case that $q(t_f^+) \neq q(t_d)$, due to controlled or autonomous mode changes.

Therefore, we need to determine the possible modes of fault occurrence, Q_{t_f}, to generate consistent hypotheses. We assume that we can accurately track the system mode under nominal conditions. Therefore, the following lemma holds[2].

Lemma 1. *The true mode of fault occurrence, $q(t_f^-)$, belongs to the estimated mode history, and can be derived from the estimated system mode at the time of fault detection, $\hat{q}(t_d)$.*

Assuming that no more than n mode changes occurred between t_f and t_d, `RollBack` performs a backward mode search starting from $\hat{q}(t_d)$ to produce a set of possible modes, Q_{t_f}, in which the fault may have occurred [14]. Discontinuities should be detected at the point of fault occurrence, so, if a discontinuity is observed, $Q_{t_f} = \{\hat{q}(t_d)\}$. Since discrete faults explicitly set variables to zero or set variables from zero, then if one of these variables is measured, a discontinuity will be detected and roll back is not performed. Therefore, measuring these variables will make roll back more efficient when discrete faults occur.

Given a hypothesized system mode after fault detection, $\hat{q}(t_f^-) \in Q_{t_f}$, and given a measurement deviation, σ_m, and using the TCG, `PropagateBackward` starts from the observed measurement deviation and maps it back to possible changes in variables and parameter values, creating a set of fault candidates, F_q [18]. This includes the discrete fault events. For discrete faults, we do not need to produce $\hat{q}(t_f^+)$ to generate this set, because all we need to do is link back to some change in a junction variable that can be affected by a change in junction state. So, we can generate both parametric and discrete faults as candidates using the TCG for $\hat{q}(t_f^-)$, thus improving the efficiency of the approach.

For example, consider that a decrease in the current through R_2, or f_6^-, is observed when the switch was expected to be closed. We denote this mode as

[2] This is a revised form of Lemma 1 in [14].

Algorithm 2 $F_{i+1} \leftarrow$ `RefineHypotheses`(σ_m, F_i)

```
F_{i+1} ← ∅
for all (f, q) ∈ F_i do
  if σ_m ∈ Σ_{f,q} then
    F_{i+1} ← F_{i+1} ∪ {(f, q)}
  else
    Q' ← RollForward(q)
    for all q' ∈ Q' do
      GenerateSignatures(f, q')
      if σ_m ∈ Σ_{f,q'} then
        F_{i+1} ← F_{i+1} ∪ {(f, q')}
```

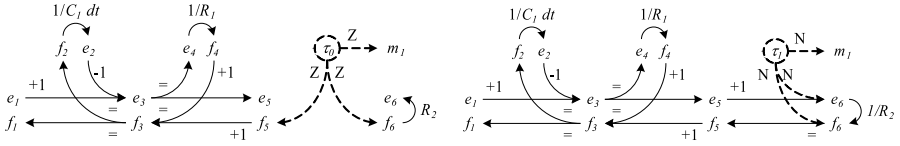
q_1 and the open mode as q_0. Since there are no autonomous transitions that may occur, the only possible $\hat{q}(t_f^-) \in Q_{t_f}$ is that where the switch is closed. The TCG for this mode is shown in Fig. 5(a). The decrease in f_6 can be explained by the fault event τ_0, given f_6 is positive. It can also be explained by a fault in the sensor of f_6 ($S_{f_6}^-$), R_2^+, or e_6^-. Propagating backwards further can link the change in e_6^- to C_1^- and R_1^-. Thus $F_0 = \{(\tau_0, q_0), (S_{f_6}^-, q_1), (R_2^+, q_1), (C_1^-, q_1), (R_1^-, q_1)\}$.

3.4 Progressive Monitoring

Hypothesis generation produces the set of fault candidates consistent with the initial measurement deviation. Progressive monitoring prunes this set as additional measurements deviate using a hypothesis refinement algorithm. When controlled events occur, we update the fault hypotheses, i.e., for event e, (f, q) is replaced in F_i with $(f, \delta(e, q))$. For each candidate (f, q), we make predictions about future measurement deviations in the form of fault signatures, $\Sigma_{f,q}$. Candidates whose predictions are consistent with an observed deviation are retained, and inconsistent candidates are dropped. The hypothesis refinement procedure is shown as Algorithm 2.

To track our fault candidates against new measurement deviations, we need to first predict what these deviations will be in the form of fault signatures. For hypothesized fault f in mode q, and using the TCG, `GenerateSignatures` performs a forward propagation of the fault effects to the measured variables, producing the fault signatures for all measurements of a fault, $\Sigma_{f,q}$ [18]. Note that Algorithm 1 applies the mode change induced by the fault. This is important for discrete faults, because this is the correct mode in which to make predictions. For the circuit, the TCGs for the fault-induced modes are given in Fig. 6(a) for the τ_0 fault and in Fig. 6(b) for the τ_1 fault.

A parameter change propagates increase or decrease values along the edges, and a discrete change of `X` to all variables. The increase or decrease propagation for discrete faults depends on whether a junction turns on or off. If the fault results in a junction turning off, then a decrease is propagated if the junction variable was last known to be positive, otherwise an increase is propagated. We propagate `Z` to the immediate variables. The estimated variable sign is needed



(a) TCG for the τ_0 fault-induced mode. (b) TCG for the τ_1 fault-induced mode.

Fig. 6. TCGs for the fault-induced modes.

for parametric faults as well, because an increase or decrease in a parameter value will have a different effect if the immediately affected variable is positive or negative. If the fault results in a junction turning on, then, in general, we do not know whether the junction variables will become positive or negative. `PropagateBackward` gives us this information. For example, consider the τ_1 fault for the circuit. Consider that we observe e_2^+. When we propagate backward (Fig. 5(b)), we map e_2^+ to f_5^+, which in turn implicates the τ_1 fault. So in this case, if τ_1 is the actual fault, it will cause f_5^+. Since at a 1-junction the flows are equal, the `+` change should be propagated forward to f_6 (Fig. 6(b)). As discussed in Section 3.2, f_6 and e_6 should change in the same direction, so we also know what to propagate to e_6. We propagate `N` to the junction flow (effort) for a 1-junction (0-junction), and also to the effort (flow) if it is equal to the flow (effort) by a gain, as in Fig. 6(b). Propagation of `Z` or `N` only continues along edges labeled with = or some parameter p. Otherwise, the change in the variable will not exhibit this behavior, and `X` is propagated along the remaining edges.

For example, consider the τ_0 fault in the circuit, where m_1, f_6, and e_2 are measured. The fault-induced mode, $\hat{q}(t_f^+)$, is given by the TCG in Fig. 6(a). Propagation begins at the fault event τ_0. This results in an immediate change in m_1, resulting in a signature of `(-0,Z)`, and an immediate decrease in f_6, assuming f_6 was positive, resulting in `(-*,Z)`[3]. It will also cause a decrease in f_5, which will propagate to the rest of the system. The `Z` symbol is also propagated to the immediate variables. This propagation stops at f_5, because f_5 is the sum of f_6 and f_4. The residual for e_2 will exhibit a first-order change, due to the integration along the edge labeled with $1/C_1 dt$, yielding `(0-,X)`.

From our current set of candidates, F_i, we generate a new set, F_{i+1}, when a new deviation is detected. Controlled mode changes may occur between t_i and t_{i+1}, so the changes can be applied to produce the new nominal reference, and generate a new set of associated fault signatures for the new mode. Because autonomous mode changes may have also occurred, they must be accounted for when generating F_{i+1}. For a given $(f, q) \in F_i$, we check if the new deviation is consistent with the predicted fault signatures. If so, it is retained. If not, it is dropped. The inconsistency, however, may be because autonomous mode changes

[3] Even though the TCG does not predict the `*`, it is necessary to include. The slope in the new mode does not change, but the slope in the estimated mode may change, so the residual may have a nonzero slope.

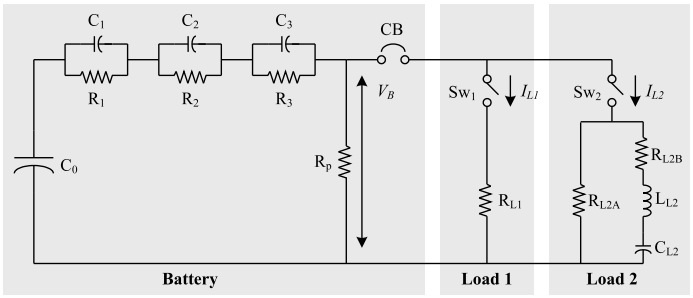


Fig. 7. Electrical circuit equivalent for the battery system.

(which can no longer be correctly predicted) have occurred, so the mode q is not correct. Assuming no more than n autonomous mode changes have occurred, `RollForward` performs a forward mode search to produce a set of new modes Q' from q that may be the correct mode [14]. For each (f, q'), where $q' \in Q'$, we generate signatures for f in mode q', and retain (f, q') if the prediction is consistent or drop it if not.

Progressive monitoring performs the qualitative fault isolation. Backward and forward propagation are polynomial in the size of the TCG [18]. The roll back and roll forward algorithms are polynomial in the maximum number of expected mode changes during diagnosis, n [14]. Adding discrete faults only introduces more fault candidates, so it does not change the worst-case time or space complexity of the diagnosis algorithms. The discrete change symbol will often help to distinguish between discrete and parametric faults, but if multiple discrete and parametric fault candidates remain after progressive monitoring, then a quantitative fault identification algorithm can be used to resolve the ambiguity and determine the true system fault and its magnitude [14].

4 Case Study

4.1 Experimental Setup

We illustrate our integrated approach to parametric and discrete fault diagnosis on ADAPT [3] deployed at NASA Ames Research Center. The testbed is functionally representative of a spacecraft's electrical power generation, storage, and distribution subsystems. It has a large number of modes, over 100 sensors, and over 170 possible faults. For our diagnosis experiments, we consider a subset of ADAPT that involves a battery discharging to two parallel DC loads as shown in Fig. 7. The selected sensors measure the battery voltage, $V_B(t)$, the currents through the loads, $I_{L1}(t)$ and $I_{L2}(t)$, and the mode of the circuit breaker, $M_{CB}(t)$. A sampling rate of 2 Hz is used in all the experiments.

Fault	$V_B(t)$	$I_{L1}(t)$	$I_{L2}(t)$	$M_{CB}(t)$
C_0^-	(+*,X)	(+*,X)	(+*,X)	(00,X)
R_1^+	(0-,X)	(0-,X)	(0-,X)	(00,X)
R_{L1}^+	(0*,X)	(-*,X)	(0*,X)	(00,X)
R_{L1}^-	(0*,X)	(+*,X)	(0*,X)	(00,X)
R_{L2A}^+	(0*,X)	(0*,X)	(-*,X)	(00,X)
R_{L2A}^-	(0*,X)	(0*,X)	(+*,X)	(00,X)
$Sw_1.off$	(0*,X)	(-*,Z)	(0*,X)	(00,X)
$Sw_2.off$	(0*,X)	(0*,X)	(-*,Z)	(00,X)
I_{L1}^+	(00,X)	(+0,*)	(00,X)	(00,X)
I_{L1}^-	(00,X)	(-0,*)	(00,X)	(00,X)

Table 1. Fault Signatures for the Battery System with Both Loads Online

4.2 Modeling Faults

The battery is modeled by an electric circuit equivalent [23]. The battery capacitance is modeled using a large capacitance, C_0. Other parameters model nonlinear, dissipative behaviors. Battery faults include loss of charge represented by a capacitance decrease, C_0^-, and internal resistance increase, R_1^+. In the loads, faults affect the resistance values R_{L1} and R_{L2A} which can increase or decrease. In the sensors, we consider bias faults which cause 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. For discrete faults, we consider faults in the relays, Sw_1 and Sw_2, and the circuit breaker, CB.

Selected fault signatures for the system mode with both loads online are given in Table 1. The nonlinearities in the battery introduce ambiguity in the qualitative signatures, denoted by the * symbol. Also note that since the sensors do not feed back into the system, sensor faults affect only the measurement provided by the sensor. The other measurements are not affected, and the corresponding signature is denoted by 00, indicating no change in the measurement from expected behavior. Sensor faults are allowed to produce discrete changes, therefore their discrete change symbol is given by *.

4.3 Experimental Results

We investigate the diagnosis of software-injected discrete faults and a manually-injected load fault in the actual system. We will denote the system mode by q_{ijk} where i is the mode of Sw_1, j is the mode of Sw_2, and k is the mode of CB.

We first investigate an unexpected switch fault. At 375.5 s, Sw_1 opens without a command. The measured and estimated outputs are shown in Fig. 8. As a result, $I_{L1}(t)$ goes immediately to zero, and $V_B(t)$ increases as a result of less current being drawn. The fault is detected at 376.0 s, and the symbol generator reports a decrease in $I_{L1}(t)$. The only possible mode of fault occurrence is q_{111}, so $F_0 = \{(I_{L1}^-, q_{111}), (R_1^+, q_{111})), (R_{L1}^+, q_{111})), (R_{L2A}^+, q_{111})), (R_{L2A}^-, q_{111})),$

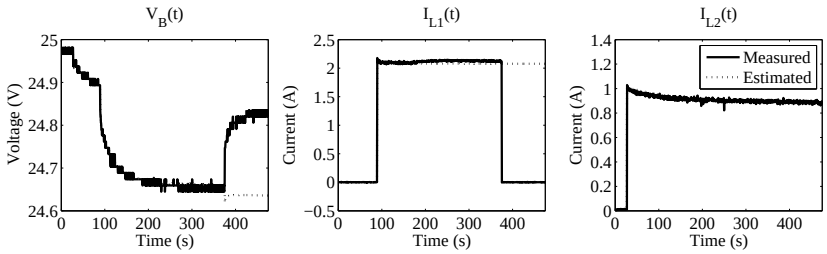


Fig. 8. Sw_1 opens.

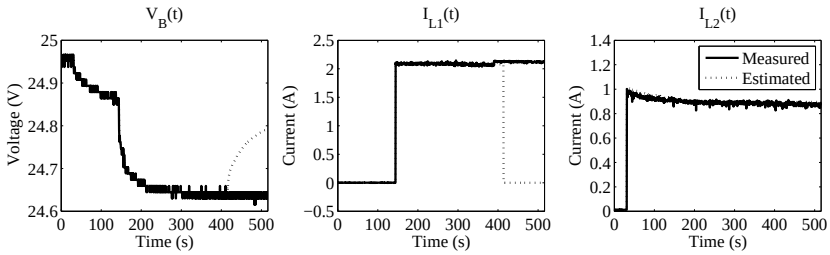


Fig. 9. Sw_1 gets stuck closed.

$(Sw_1.off, , q_{011})\}$. At 376.5 s, the symbol generator reports an increase in $V_B(t)$, thus eliminating the sensor fault as a candidate from F_1. Because M_{CB} does not change, and we assume single faults, the autonomous mode change of the circuit breaker could not have occurred. At 378.5 s, the symbol generator reports I_{L1} `Z` because $\hat{I}_{L1}(t)$ is nonzero, but $I_{L1}(t)$ is zero. Since parametric faults cannot cause this behavior, $Sw_1.off$ is correctly identified as the fault.

We now investigate a stuck switch fault. At 414.0 s, Sw_1 is commanded off but remains closed. The measured and estimated outputs are shown in Fig. 9. Therefore, the estimated system mode is q_{011} but the actual system mode is q_{111}, and $\hat{I}_{L1}(t)$ goes to zero, while $I_{L1}(t)$ remains nonzero. The fault is detected at 416.0 s, and the symbol generator reports that $I_{L1}(t)$ has increased. Because the expected mode is q_{011}, the only reason for the current to deviate is due to a discrete fault or a sensor fault, i.e., $F_0 = \{(I_{L1}^-, q_{011}), (Sw_1.on, q_{111})\}$. At 418.5 s, the symbol generator reports $I_{L1}(t)$ `N`, because the measurement went nonzero with respect to the estimate. Because sensor faults are also allowed to cause discrete behavior, both faults are retained in F_1. At 419.5 s, we observe a decrease in $V_B(t)$, and since I_{L1}^- cannot cause this, $Sw_1.on$ is isolated as the true fault. Again, no autonomous mode changes need to be considered.

We now investigate a parametric fault in Load 1. A 33% decrease in the Load 1 resistance, R_{L1}^-, is injected at 417.0 s. The measured and estimated outputs are shown in Fig. 10. The decrease in resistance increases I_{L1} abruptly, and is detected at 417.0 s, resulting in $F_0 = \{(I_{L1}^+, q_{111}),\ (Sw_2.off, q_{101}),\ (C_0^-, q_{111}),\ (R_{L1}^-, q_{111}),\ (R_{L2A}^+, q_{111}),\ (R_{L2A}^-, q_{111})\}$. The first order change due to the fault is

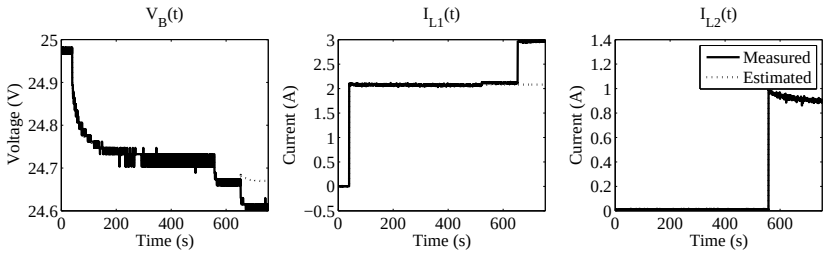


Fig. 10. R^-_{L1} fault with magnitude of 33%.

compensated by the battery, and at 422.0 s, the slope of the change in I_{L1} is determined to be zero, reducing the candidate set to $F_1 = \{(I^+_{L1}, q_{111}), (C^-_0, q_{111}), (R^-_{L1}, q_{111})\}$ because now the immediate change is known to be a discontinuity. Again, the lack of change in M_{CB} rules out an autonomous transition of CB. At 433.0 s, the symbols generator reports a decrease in $V_B(t)$. Since I^+_{L1} cannot produce a deviation in this measurement, it is dropped, and since C^-_0 would have caused an increase, it is also dropped, and R^-_{L1} is correctly isolated.

Other faults were injected in the hardware or in simulation. We have performed many additional experiments, and have found that qualitative fault isolation typically reduces the candidate set to a single candidate within a few measurement deviations. In some cases, especially those involving sensor faults, multiple candidates remained after qualitative fault isolation, and a quantitative fault identification module would be necessary to resolve the ambiguities.

5 Conclusions

We presented an integrated parametric and discrete fault diagnosis approach for hybrid systems. Deviations in expected behavior are abstracted to perform qualitative fault isolation. Both parametric and discrete faults are included in the diagnosis model so that their effects can be predicted using our qualitative algorithms. Our methodology for predicting the effects of discrete faults can also be applied to predicting the effects of autonomous mode changes, which can be used within a larger mode estimation scheme to generate only consistent mode branchings from the current mode. We presented a case study for hybrid diagnosis on the ADAPT system, with experimental results that demonstrate the effectiveness of the approach to a real, complex hybrid system. Future work will incorporate a fault identification module to handle both parametric and discrete faults, and formally incorporate the use of discrete mode sensors for switching elements to more efficiently handle the roll back and roll forward processes.

Acknowledgments

This work was supported in part by grants NSF-NASA USRA 08020-013, NASA NRA NNX07AD12A, NSF CNS-0615214, and NSF CNS-0347440.

References

1. Williams, B.C., Nayak, P.P.: A model-based approach to reactive self-configuring systems. In: Proc. of AAAI-96, AAAI Press (1996) 971–978
2. Goodrich, C., Kurien, J.: Continuous measurements and quantitative constraints — challenge problems for discrete modeling techniques. In: Proc. of iSAIRAS-2001. (2001)
3. Poll, S., Patterson-Hine, A., Camisa, J., Nishikawa, D., Spirkovska, L., Garcia, D., Hall, D., Neukom, C., Sweet, A., Yentus, S., Lee, C., Ossenfort, J., Roychoudhury, I., Daigle, M., Biswas, G., Koutsoukos, X., Lutz, R.: Evaluation, selection, and application of model-based diagnosis tools and approaches. In: AIAA Infotech@Aerospace 2007 Conference Proceedings. (May 2007)
4. Zhao, F., Koutsoukos, X., Haussecker, H., Reich, J., Cheung, P.: Monitoring and fault diagnosis of hybrid systems. IEEE Trans. on Systems, Man, and Cybernetics, Part B **35**(6) (2005) 1225–1240
5. Cocquempot, V., El Mezyani, T., Staroswiecki, M.: Fault detection and isolation for hybrid systems using structured parity residuals. In: Proceedings of the 5th Asian Control Conference. (2004) 1204–1212
6. Hofbaur, M., Williams, B.: Mode estimation of probabilistic hybrid systems. In: Hybrid Systems: Computation and Control. Volume 2289 of LNCS. Springer-Verlag (2002) 253–266
7. Wang, W., Li, L., Zhou, D., Liu, K.: Robust state estimation and fault diagnosis for uncertain hybrid nonlinear systems. Nonlinear Analysis: Hybrid Systems **1**(1) (March 2007) 2–15
8. Koutsoukos, X., Kurien, J., Zhao, F.: Estimation of distributed hybrid systems using particle filtering methods. In: Hybrid Systems: Computation and Control. Volume 2623 of LNCS. Springer (2003) 298–313
9. Dearden, R., Clancy, D.: Particle filters for real-time fault detection in planetary rovers. In: Proc. of the 12th Int. Workshop on Principles of Diagnosis. (2001) 1–6
10. Benazera, E., Travé-Massuyès, L., Dague, P.: State tracking of uncertain hybrid concurrent systems. In: Proc. of the 13th Int. Workshop on Principles of Diagnosis. (2002) 106–114
11. Narasimhan, S., Brownston, L.: HyDE — a general framework for stochastic and hybrid model-based diagnosis. In: Proc. of the 18th Int. Workshop on Principles of Diagnosis. (May 2007) 162–169
12. McIlraith, S.A., Biswas, G., Clancy, D., Gupta, V.: Hybrid systems diagnosis. In: Hybrid Systems: Computation and Control. Volume 1790 of LNCS. Springer (2000) 282–295
13. Narasimhan, S., Biswas, G.: An approach to model-based diagnosis of hybrid systems. In: Hybrid Systems: Computation and Control. Volume 2289 of LNCS. Springer-Verlag (2002) 308–322
14. Narasimhan, S., Biswas, G.: Model-based diagnosis of hybrid systems. IEEE Trans. on Systems, Man and Cybernetics, Part A **37**(3) (May 2007) 348–361
15. Fourlas, G.K., Kyriakopoulos, K.J., Krikelis, N.J.: Fault diagnosis of hybrid systems. In: Proc. of the 2005 IEEE Int. Symp. on Intelligent Control. (June 2005) 832–837
16. Di Benedetto, M.D., Di Gennaro, S., D'Innocenzo, A.: Diagnosability verification for hybrid automata. In: Hybrid Systems: Computation and Control. Volume 4416 of LNCS. Springer (2007) 684–687

17. Lunze, J., Shröder, J.: Sensor and actuator fault diagnosis of systems with discrete inputs and outputs. IEEE Transactions on Systems, Man, and Cybernetics, Part B **34**(2) (April 2004) 1096–1107
18. Mosterman, P., Biswas, G.: Diagnosis of continuous valued systems in transient operating regions. IEEE Transactions on Systems, Man and Cybernetics, Part A **29**(6) (1999) 554–565
19. Mosterman, P.J., Biswas, G.: A theory of discontinuities in physical system models. Journal of the Franklin Institute **335B**(3) (January 1998) 401–439
20. Karnopp, D.C., Margolis, D.L., Rosenberg, R.C.: Systems Dynamics: Modeling and Simulation of Mechatronic Systems. John Wiley & Sons, Inc., New York (2000)
21. Daigle, M., Roychoudhury, I., Biswas, G., Koutsoukos, X.: Efficient simulation of component-based hybrid models represented as hybrid bond graphs. In: Hybrid Systems: Computation and Control. Volume 4416 of LNCS. Springer-Verlag (2007) 680–683
22. Biswas, G., Simon, G., Mahadevan, N., Narasimhan, S., Ramirez, J., Karsai, G.: A robust method for hybrid diagnosis of complex systems. In: Proc. of the 5th Symposium on Fault Detection, Supervision and Safety for Technical Processes. (June 2003) 1125–1131
23. Ceraolo, M.: New dynamical models of lead-acid batteries. IEEE Transactions on Power Systems **15**(4) (November 2000) 1184–1190