On Discrete Event Diagnosis Methods for Continuous 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
Email: matthew.j.daigle,xenofon.koutsoukos,gautam.biswas@vanderbilt.edu

***Abstract*— Fault detection and isolation is a key component of any safety-critical system. Although diagnosis methods based on discrete event systems have been recognized as a promising framework, they cannot be easily applied to systems with complex continuous dynamics. This paper presents a novel approach for discrete event system diagnosis of continuous systems based on a qualitative abstraction of the measurement deviations from the nominal behavior. We systematically derive a diagnosis model, provide diagnosability analysis, and design a diagnoser. Our results show that the proposed approach is easily applicable and can be used for online diagnosis of abrupt faults in continuous systems.**

I. Introduction

Fault detection and isolation (FDI) is a key component of any safety-critical system. When faults and degradations occur, it is important to quickly identify them so corrective actions can be taken in a timely manner and catastrophic situations can be avoided. There are several frameworks to FDI that can be categorized along several dimensions, including model-based vs. signal-driven, online vs. offline, and continuous vs. discrete. Discrete event system (DES) methods have been recognized as a promising framework due to the significance of event-driven models in safety-critical systems, the well-developed theory that allows systematic construction of a diagnostic system, and the computational efficiency that enables online diagnosis for large systems.

Existing DES diagnosis methods [1]–[3] are usually based on a detailed, automata-based model capturing both the nominal and faulty system behavior. If such a model exists, these approaches can be easily applied for diagnosis of abrupt faults represented by unobservable events. Although discrete event systems can readily model many practical applications [1], [4]–[6], they are not well-suited to capture complex continuous dynamics. Abstracting continuous dynamics requires quantization of the continuous state space that results in large, nondeterministic models [7], [8]. Further, even if it is reasonable to abstract the nominal continuous behavior, developing DES models for faulty behavior is very challenging. Faults in continuous dynamic systems are represented by changes in the system parameters, and therefore, quantization techniques must consider a high-dimensional state space and often complex nonlinear dynamics.

This paper presents a novel approach for DES diagnosis of continuous systems based on a *qualitative* abstraction of the measurement deviations from the nominal behavior. We describe a systematic method for generating a discrete event model of the system representing the faulty behaviors. The approach is based on the TRANSCEND [9] methodology for model-based, qualitative fault diagnosis in continuous systems. Starting with the system model, we systematically derive a diagnosis model, extract diagnostic information, and build a diagnoser.

Specifically, the contribution of the paper is threefold: (*i*) we systematically construct a labeled transition system capturing the *fault language*, which, for each fault, describes all possible sequences of measurement deviations, (*ii*) we analyze the diagnosability of the system and design an event-based diagnoser, and (*iii*) we describe an implementation that improves the computational efficiency of the diagnoser. Diagnosis of component faults in an electric circuit is used throughout the paper to illustrate the approach.

Our approach to diagnosis of continuous systems exploits the qualitative form of the fault transient created by abrupt deviations in component parameter values as well as the temporal ordering of measurement deviations, thereby generating event sequences [10], [11]. Diagnosis in discrete event systems is concerned with diagnosing system failures based on sequences of observed events. Therefore, there is a direct link between our diagnosis approach and DES approaches. Section II clarifies the connection between traditional DES diagnosis methods and our proposed work. Section III presents our modeling and analysis approach, Section IV describes the design of the diagnoser, and Section V concludes the paper.

II. Related Work

We formulate our approach to diagnosis of continuous systems into a DES framework. DES diagnosis methods are based on observing system events and making inferences about the system state. The basic idea is that the occurrence of a fault will generate a unique sequence of observable events that will establish the presence of the fault. The seminal work of [1], [2] describes an event-based DES diagnosis framework. A diagnoser based on the system model functions as an extended observer that provides estimates of the system state under nonfaulty and faulty conditions. In [3], the authors use a state-based approach, so the diagnoser determines the system

condition, rather than which failure events have occurred.

To apply DES diagnosis approaches to continuous systems, the system models must be abstracted in some way. One method is to create a timed DES model. Such models typically include an additional observable event representing the tick of a global clock [12], [13]. Diagnosis of timed DES has been investigated in [12] as an extension of [1] and in [13] as an extension of [3]. Alternatively, a timed automaton model of the system can be used for diagnosis [14]. The approach of [7] develops the abstracted timed DES model through quantization. The continuous state space is partitioned and events defined for crossings of those partitions. This approach is limited to discrete inputs, discrete measurements, and discrete faults. Coarse quantizations lose information which may be valuable to diagnosis, but fine-grained quantizations suffer from state explosion. To use the quantization approach, faults have to be quantized according to their magnitude and other characteristics. If faults are possible at any state of the system (as is usually the case), then the number of states grows even more. Furthermore, the resulting model is, except in trivial cases, nondeterministic which degrades the performance and increases the computational requirements of diagnosis algorithms.

We instead propose a qualitative abstraction approach, where we model only the faulty behavior relevant to diagnosis. Three qualitative states are defined for each measurement: above nominal, at nominal, and below nominal. Measurement deviations directly indicate the presence of a fault and form the event set of our approach. The diagnoser therefore does not need to track the nominal behavior, as with quantization approaches, but only track the fault behavior as given by the measurement deviations. Nominal behavior is instead computed with an observer based on the system model [9] The observer uses a continuous model of the system, so does not further abstract the model via quantization. The tasks of tracking nominal behavior and fault isolation are separated so that the diagnoser is concerned only with faulty behavior.

In existing DES diagnosis approaches [1]–[3], [7], [12], [13], the discrete event model of the system and all its faulty behavior are assumed to be given. We assume our continuous model of the system is given, where faults are represented as parameter changes in the nominal model of the system. As a result, the system model represents both nominal and faulty behavior in a very concise way. From this model, we systematically derive the diagnosis model [9] to generate fault signatures and measurement orderings, and extract from this information a discrete event model of the system with respect to faulty behavior. This greatly reduces the burden of the modeling task, as well as providing a systematic framework for deriving the faulty behavior. Unlike quantization approaches, our approach is not dependent on fault magnitude because we are only concerned with the qualitative form of the measurement deviations.

Because we are working under an event-based framework, the notions of fault traces, fault languages, distinguishability, and diagnosability that we define bear a resemblance to those defined in the DES literature, including [15]. The notion of using temporal orders of measurement deviations to help discriminate faults is also investigated in [16], [17], however it is based on analytical redundancy relations, which are difficult to develop for nonlinear systems and multiplicative faults. The approach also does not address how to obtain this information, whereas in our approach, it is derived systematically from the system model.

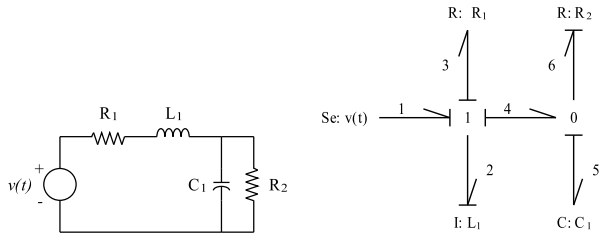
III. A Diagnosis Approach Based on Qualitative Abstraction

While certainly applicable, the utility of explicit discrete event approaches to diagnosis of continuous systems seems limited. To achieve diagnostic precision, the system quantization may have to be very fine-grained. The nondeterminism of the model and its large size lead to large diagnosers with high computational requirements. We instead abstract continuous system behaviors with respect to their deviations from the nominal system behavior. These are represented as qualitative measurement deviations in a way that captures the dynamics associated with the deviations after fault occurrence, resulting in compact fault models.

Our approach is based on the TRANSCEND [9] methodology. TRANSCEND is a model-based approach to diagnosis of continuous systems. Starting with a bond graph model [18] of the system, a temporal causal graph (TCG) is derived, which captures the propagation of fault effects as deviations in the system variables. These effects are captured qualitatively as *fault signatures* of the observed measurements [9] and temporally as *relative measurement orderings* [10].

Throughout the paper we will illustrate the diagnosis method using a circuit example. Fig. 1(a) gives the schematic, and the associated bond graph is shown in Fig. 1(b). We assume that our input voltage, $v(t)$, is constant and positive. The derived TCG is given in Fig. 2. In our framework, a fault is an abrupt change in a parameter value of the system model, and measurement deviations are transients due to the fault. We also assume that only one fault occurs in the system at a time. The set of faults is assumed to be $F = \{R_1^+, R_1^-, R_2^+, R_2^-, C_1^+, C_1^-, L_1^+, L_1^-\}$, where the superscript indicates the direction of change of the parameter value. We define the measurement set as 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.

Faults signatures provide the discriminatory information in TRANSCEND. A fault signature is the symbolic manifestation of a measurement deviation, represented as the qualitative effect of a fault on a measurement. It represents the qualitative value of zeroth- through kth-order derivative changes on a measurement due to the occurrence of a fault. Because only magnitude and slope can be reliably measured, we condense the signatures to the magnitude change symbol and the first nonzero derivative change, e.g., $000{-}{+}{-}{+}$ becomes $0{-}$, and ${+}{-}{+}{-}{+}{-}{+}$ becomes ${+}{-}$. We can do this because



(a) Schematic. (b) Bond graph.

Fig. 1. Circuit example.

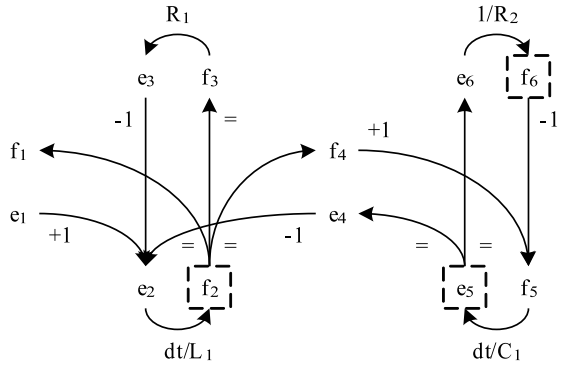


Fig. 2. Temporal causal graph for the circuit.

higher-order changes will eventually manifest as first-order changes. Furthermore, higher order effects after the first change provide no additional discriminatory value [19]. Therefore, a fault signature for measurement m will be an element of the set $\Sigma_m \triangleq \{m^{+-}, m^{-+}, m^{0+}, m^{0-}\}$. The superscript indicates the observed deviation. The first symbol represents the immediate direction of abrupt change (a discontinuity) and the second symbol represents the direction of change.

When two paths of the same derivative order in the TCG exist from a fault to a measurement and give opposite qualitative propagations, the derivative effect cannot be determined, resulting in a signature of $0*$. Such a signature can manifest as $0+$ or $0-$. So, in general, $\sigma_{f,m}$ may not be unique.

Definition 1 (Fault Signature): A *fault signature* for a fault f and measurement m is the qualitative effect of f on m, given f has occurred, and is denoted by $\sigma_{f,m} \in \Sigma_{f,m}$, where $\Sigma_{f,m} \subseteq \Sigma_m$. We denote the set of all fault signatures for fault f as Σ_f.

Relative measurement orderings define, with respect to a given fault, a partial order of measurement deviations. These are also predicted using the TCG based on common temporal subpaths [10].

Definition 2 (Relative Measurement Ordering): 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 as $m_i \prec_f m_j$. We denote the set of all measurement orderings for fault f as Ω_f.

The fault signatures and relative measurement orderings for the circuit system are given in Table I. For example, consider L_1^-. A decrease in L_1 will cause an immediate increase in f_2, because of the inverse relation implied in the TCG. Since all subsequent paths from f_2 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_2 deviates. Either e_5 or f_6 may deviate next. It cannot be determined which will deviate first because the path from e_5 to f_6 contains no integrations. The changes in these measurements will not be abrupt because of the integration in the path from L_1 to the measurement, and the direction of change will be opposite that of f_2 because the -1 specifier in the path from f_2 to e_5 and f_6 indicates an inverse proportionality relationship.

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^-	$+-$	$0+$	$0+$	$f_2 \prec e_5, f_2 \prec f_6$

We combine the notion of fault signatures and relative measurement orderings into an event-based framework. Essentially, for a specific fault, the combination of all fault signatures and relative measurement orderings yields all the possible ways a fault can manifest. We denote one of these possibilities as a *fault trace*.

Definition 3 (Fault Trace): 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 order of fault signatures satisfies Ω_f.

As an example, consider C_1^+. A valid fault trace is $\lambda_{C_1^+} = e_5^{-+} f_6^{-+} f_2^{0+}$, but, $\lambda_{C_1^+} = f_2^{0+} e_5^{-+} f_6^{-+}$ is not because this sequence of events does not satisfy $\Omega_{C_1^+}$. We group the set of all fault traces into a *fault language*.

Definition 4 (Fault Language): The *fault language* of a fault f, denoted by L_f, is the set of all fault traces for f.

This language can be represented concisely by a *labeled transition system* (LTS).

Definition 5 (Labeled Transition System): A *labeled transition system* is a tuple $\mathcal{L} = (Q, q_o, \Sigma, \rightarrow)$ such that: Q is a set of states, $q_o \in Q$ is an initial state, Σ is a set of labels, and $\rightarrow \subseteq Q \times \Sigma \times Q$ is a transition relation.

To systematically construct the LTS representation of a fault language, we can represent each fault signature and each relative measurement ordering as a LTS, and then compose all this information. Each fault signature $\sigma_{f,m}$ can be represented as a LTS, shown as the first LTS in Fig. 3. It consists of only the single event corresponding

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

to the fault signature[1]. Also, each relative measurement ordering, $m_i \prec_f m_j$, with associated signatures σ_{f,m_i} and σ_{f,m_j} can be represented as a LTS, shown as the second LTS in Fig. 3. It consists of the two associated signatures in the determined ordering.

Lemma 1: The LTS representation of a fault language L_f for fault f, denoted by $\mathcal{L}_f$, is the synchronous product of the individual LTS for all $\sigma_{f,m} \in \Sigma_f$ and all $m_i \prec_f m_j \in \Omega_f$, where the alphabets for each LTS are taken to be the events contained in the LTS.

Proof: Because the synchronous product must obey all the individual ordering constraints and will include all measurement deviation events for the fault, then it will produce all possible measurement deviation sequences for the fault, and only those. ■

Lemma 2 (Distinguishability): A fault f_i is *distinguishable* from a fault f_j, denoted by $f_i \nsim f_j$, if $(\forall\, \lambda_{f_i} \in L_{f_i}, \lambda_{f_j} \in L_{f_j})\ (\neg\exists\ \lambda)\ \lambda_{f_i}\lambda = \lambda_{f_j}$.

Proof: Two faults are distinguishable if it is not possible for them to manifest in the measurements in the same way. Since a fault language represents all possible measurement deviation sequences for a particular fault, then if one fault exhibits a trace that is a substring of another fault, then the faults cannot be distinguished. Otherwise, they cannot manifest in the same way and are distinguishable. ■

Depending on how they actually manifest in the system however, two faults which are indistinguishable may be discriminated if fault f_i occurs and manifests in a way that it is not possible for fault f_j to manifest, i.e, $\lambda_{f_i} \notin L_{f_j}$. Distinguishability is, therefore, a conservative notion. To design diagnosers, we look for the notion of *diagnosability*, based on the notion of distinguishability.

Lemma 3 (Diagnosability): A system is *single fault diagnosable* if $(\forall f_i, f_j \in F) f_i \neq f_j \implies f_i \nsim f_j$.

Proof: A system is diagnosable if each possible fault trace is consistent with a unique fault. If two faults are distinguishable, then they cannot manifest in the same way. Therefore, if all pairs of faults are distinguishable, then a given fault trace cannot be consistent with the more than one fault. Therefore the fault trace corresponds to a unique fault, so the system is diagnosable. ■

IV. Diagnoser Design

A. Diagnosis Algorithm

The notion of diagnosability is used in building correct diagnosers. To guarantee unique diagnosis of every fault, a system must be diagnosable. We now describe a method to systematically create such a diagnoser, but first, we define formally a *diagnosis* and a *diagnoser*.

[1]If $\sigma_{m,f}$ is not unique, multiple edges for each possibility are needed going from the first state of the LTS to the final state.

Algorithm 1 $\mathcal{D} \leftarrow$ `CreateDiagnoser`$(\mathcal{D}_1, \mathcal{D}_2)$

```
Q ← ∅, δ ← ∅, D ← ∅, Σ ← Σ1 ∪ Σ2
q_o ← (q_o1, q_o2), Y(q_o) ← ∅, Q_pend ← {q_o}
while Q_pend ≠ ∅ do
  (q1, q2) ← pop(Q_pend)
  for all σ_m ∈ Σ do
    if m ∉ H((q1, q2)) then
      if δ1(q1, σ_m) and δ2(q2, σ_m) then
        q' ← (δ1(q1, σ_m), δ2(q2, σ_m))
        h ← Y(δ1(q1, σ_m)) ∪ Y(δ2(q2, σ_m))
      else if δ1(q1, σ_m) then
        q' ← (δ1(q1, σ_m), q2)
        h ← Y(δ1(q1, σ_m))
      else if δ2(q2, σ_m) then
        q' ← (q1, δ2(q2, σ_m))
        h ← Y(δ2(q2, σ_m))
      else
        q' ← ∅
        h ← ∅
      if q' ≠ ∅ then
        if Y((q1, q2)) = ∅ then
          d ← h
        else
          d ← Y((q1, q2)) ∩ h
        if d ≠ ∅ then
          Q ← Q ∪ {q'}
          H(q') ← H((q1, q2)) ∪ {m}
          δ((q1, q2), σ_m) ← q'
          D ← D ∪ {d}
          Y(q') ← d
          if q' ∉ Q_pend then
            push(Q_pend, q')
```

Definition 6 (Diagnosis): A *diagnosis* $d \subseteq F$ is a set of faults consistent with the observations.

Definition 7 (Diagnoser): A *diagnoser* is a tuple $\mathcal{D} = (Q, q_o, \Sigma, \delta, D, Y)$ such that: Q is a set of states, $q_o \in Q$ is an initial state, Σ is a set of labels, $\delta \subseteq Q \times \Sigma \times Q$ is a transition relation, $D \subseteq C$ is a set of diagnoses, and $Y : Q \to D$ is a diagnosis map.

A diagnoser is a LTS extended by a set of diagnoses and a diagnosis map. Similar to the LTS of a fault, the labels correspond to measurement deviations. Associated with the states are diagnoses, i.e., the set of possible faults for the measurement deviations seen thus far.

The diagnoser construction procedure is shown as Algorithm 1. Diagnosers are constructed by incrementally composing smaller diagnosers, i.e., a diagnoser for a set of faults F_i is composed with a diagnoser for a set of faults F_j to create a new diagnoser for the set of faults $F_i \cup F_j$. Initially, we begin with diagnosers for singleton fault sets. These are constructed using the individual fault models. For a single fault f, we augment $\mathcal{L}_f$ to form $\mathcal{D}_f$ by constructing the diagnosis map as mapping every state except the initial state to $\{f\}$. The initial state is mapped to the empty diagnosis $\varnothing$, because until a measurement deviation is observed, we assume the system is operating nominally. The diagnosers corresponding to the individual faults of the circuit are shown in Fig. 4.

The construction algorithm operates by tracing paths in the two given diagnosers. The algorithm is described as combining two diagnosers, but can be easily be modified to combine simultaneously k diagnosers. If the same event

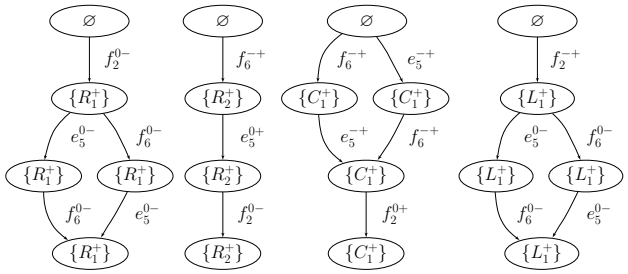


Fig. 4. Diagnosers for the individual faults of the circuit. The diagnosers for decreases in the parameter values are the same except for a reversal in the signs.

label is available in both current states, then we advance in both machines, i.e., $(q_1, q_2) \xrightarrow{\sigma} (\delta(q_1, \sigma), \delta(q_2, \sigma))$. Otherwise, we advance in only one, e.g. if σ can only be taken from q_1, then , $(q_1, q_2) \xrightarrow{\sigma} (\delta(q_1, \sigma), q_2)$. However, if the measurement associated with σ has already deviated along the current path (tracked using H), $\delta((q_1, q_2), \sigma)$ is set to $\varnothing$, because measurement deviations are only detected once per measurement. This also occurs if the computed diagnosis for the new state, d, is empty, because this means the current sequence of measurement deviations is inconsistent with the single fault assumption.

The diagnosis for the new state is formed by composing the current diagnosis with the hypothesis set. The hypothesis set, the set of faults consistent with the current event, is formed as the union of the active states' diagnoses (e.g. $\{f_i\} \cup \{f_j\}$), where the active states as the states of the diagnosers we are advancing to via σ. The new diagnosis for the composed diagnoser state is constructed as the intersection of the current diagnosis and the hypothesis set. For example, if $\{f_i, f_j\}$ is the current diagnosis and the hypothesis set is $\{f_i\}$ then the new diagnosis is $\{f_i\}$, which means that only f_i is consistent with the current sequence of measurement deviations.

The final composed diagnoser for the circuit is illustrated in Fig. 5. For example, consider the fault trace $f_6^{-+} e_5^{0+} f_2^{0-}$. For f_6^{-+} occurring as the first measurement deviation, only C_1^+ or R_2^+ could have occurred, given the known fault signatures and relative 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. The only possible measurement deviation from here is f_2^{0-} which must be consistent still with $\{R_2^+\}$.

Theorem 1: The diagnoser constructed by Algorithm 1 for fault sets F_1 and F_2 represents all valid single fault traces for the faults in F_1 and F_2 and associates correct diagnoses with the states.

Proof: By definition, the diagnoser for a single fault f is correct because it represents L_f, so represents all possible fault traces of f, and every state (except the initial state) of $\mathcal{L}_f$ is consistent with f occurring. Assume that diagnosers $\mathcal{D}_1$ and $\mathcal{D}_2$ are correct. Then they represent all possible fault traces for fault sets F_1 and F_2, respectively. At the initial state, if an event σ happens which can only happen in one of the diagnosers, $\mathcal{D}_i$, then the diagnosis is $Y_i(\delta(q_{oi}, \sigma))$, because it must be consistent with faults in F_i that are consistent with σ. If σ can occur in both diagnosers, then the diagnosis is $Y_1(\delta_1(q_{o1}, \sigma)) \cup Y_2(\delta_2(q_{o2}, \sigma))$ because either a fault in F_1 occurred or a fault in F_2 occurred, and the diagnosis must be consistent with any fault in $F_1 \cup F_2$ consistent with σ. Assume that for a given $(q_1, q_2) \neq q_o$ the diagnoses are correct for the event sequences leading up to (q_1, q_2). Then if an event σ happens which can only happen in one of the diagnosers, $\mathcal{D}_i$, then the diagnosis is $Y((q_1, q_2)) \cap Y_i(\delta_i(q_i, \sigma))$, because it must consistent with the previous diagnosis faults in F_i consistent with σ. If σ can occur in both diagnosers, then the diagnosis is $Y((q_1, q_2)) \cap (Y_1(\delta_1(q_1, \sigma)) \cup Y_2(\delta_2(q_2, \sigma)))$ because it must be consistent with the previous diagnosis and faults in $F_1 \cup F_2$ consistent with σ. Therefore, for any state q, $\delta(q, \sigma)$ has a correct diagnosis. So, for any two diagnosers, the resulting diagnoser is correct. ■

The diagnoser for the circuit example, shown in Fig. 5, illustrates certain properties of our approach. Since all the leaves have diagnoses with a unique fault, then the system is diagnosable. Any possible sequence of measurement deviations corresponding to a single fault occurring are captured in the diagnoser, and lead to unique diagnoses, therefore the system is diagnosable. We can also see that a unique diagnosis is obtained after only two of the three measurements deviate, therefore one measurement is redundant for single fault diagnosis of the selected faults.

B. Online Diagnoser Implementation

For a large number of faults and measurements, implementing the diagnoser as a LTS for use in online diagnosis is not space-efficient. Alternatively, we could create a single diagnoser for each fault, run them simultaneously, and combine the diagnoses. Individual diagnosers may be large, however, if there are few measurement orderings for the fault. Instead, we store only the single fault effects, i.e., for each fault we store its fault signatures and relative measurement orderings. As measurement deviations occur, we can check consistency using this stored information to generate our hypothesis sets and refine our diagnoses.

Given a current diagnosis of d_{i-1} and an event σ_i occurring, we can check which faults are consistent with σ_i occurring. The hypothesis set h_i consists of those faults. If $i = 1$, then the new diagnosis d_i is simply h_i. If $i > 1$, then 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.

Thus, we are only constructing the path of the diagnoser corresponding to the particular fault trace we are observing. This is more space-efficient than constructing the complete diagnoser offline and using it as an online diagnoser. The complete diagnoser has, in the worst case, $O(|M|!)$ possible fault traces, therefore the diagnoser has, in the worst case, $O(|M|!)$ states. Storing only fault signatures and relative measurement orderings for each

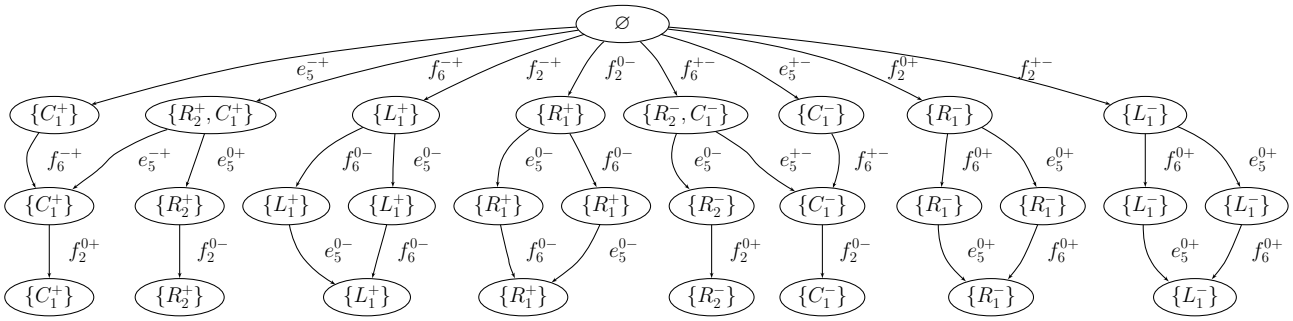


Fig. 5. Single fault diagnoser for the circuit.

fault, on the other hand, takes $O(|F||M|^2)$ space.

Time complexity, however, is in favor of the precomputed diagnoser. It needs only to wait for measurement deviations to occur, transition to the next state, and output the current diagnosis associated with the state. Using appropriate data structures, these operations can be achieved in constant time. For our online approach, we need to form the hypothesis set corresponding to the current measurement deviation. We need to look through the fault signatures and measurement orderings to do this, thus taking $O(|F||M|^2)$ time. We then need to 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 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, less and less faults are being considered.

V. Conclusions

In this paper, we presented discrete event system modeling and diagnosis methodologies and their application to continuous systems. The main issue in applying these 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 discrete event models of faulty system behavior and enables a diagnosis approach that is easily applicable to continuous systems.

Acknowledgments

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

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] 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-2002)*, Semmering, Austria, May 2002, pp. 179–188.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] ——, "Distributed diagnosis of coupled mobile robots," in *Proceedings 2006 IEEE International Conference on Robotics and Automation*, May 2006, pp. 3787–3794.

[12] 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.

[13] 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.

[14] 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.

[15] M.-O. Cordier, L. Travé-Massuyès, and X. Pucel, "Comparing diagnosability in continuous and discrete-event systems," in *Proceedings of the 17th International Workshop on Principles of Diagnosis (DX-06)*, 2006, pp. 55–60.

[16] 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.

[17] 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.

[18] D. C. Karnopp, D. L. Margolis, and R. C. Rosenberg, *Systems Dynamics: Modeling and Simulation of Mechatronic Systems*, 3rd ed. New York: John Wiley & Sons, Inc., 2000.

[19] 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.