Improving Distributed Diagnosis Through Structural Model Decomposition

Anibal Bregon 1, Matthew Daigle 2, Indranil Roychoudhury 3
Gautam Biswas 4, Xenofon Koutsoukos 4, and Belarmino Pulido 1

1 Department of Computer Science, University of Valladolid, Valladolid, 47011, Spain
{anibal,belar}@infor.uva.es
2 University of California, Santa Cruz, NASA Ames Research Center, Moffett Field, CA, 94035, USA
matthew.j.daigle@nasa.gov
3 SGT Inc., NASA Ames Research Center, Moffett Field, CA, 94035, USA
indranil.roychoudhury@nasa.gov
4 Institute for Software Integrated Systems, Dept. of EECS, Vanderbilt University, Nashville, TN, 37235, USA
{gautam.biswas,xenofon.koutsoukos}@vanderbilt.edu

ABSTRACT

Complex engineering systems require efficient fault diagnosis methodologies, but centralized approaches do not scale well, and this motivates the development of distributed solutions. This work presents an event-based approach for distributed diagnosis of abrupt parametric faults in continuous systems, by using the structural model decomposition capabilities provided by Possible Conflicts. We develop a distributed diagnosis algorithm that uses residuals computed by extending Possible Conflicts to build local event-based diagnosers based on global diagnosability analysis. The proposed approach is applied to a multi-tank system, and results demonstrate an improvement in the design of local diagnosers. Since local diagnosers use only a subset of the residuals, and use subsystem models to compute residuals (instead of the global system model), the local diagnosers are more efficient than previously developed distributed approaches.

1 INTRODUCTION

The need for increased performance, safety, and reliability of complex engineering systems motivates the development of efficient fault diagnosis methodologies. Accurate and timely centralized fault diagnosis of complex systems is difficult and can be computationally expensive. Typically, centralized solutions have been proposed to approach the fault diagnosis problem, but these solutions do not scale well as the size of the system increases, and serve as single points of failure. These shortcomings, together with the widespread use of distributed, networked components, encourages the development of distributed diagnosis frameworks.

In previous work, we have developed a distributed design approach based on global diagnosability analysis (Roychoudhury et al., 2009), where the local diagnosers are designed to provide globally correct diagnosis results, without a centralized coordinator, and by communicating a minimal number of measurements among themselves. Later on, this work was integrated to the formal event-based framework developed in (Daigle et al., 2009) to include measurement orderings within the local diagnosers. Inclusion of measurement orderings improves diagnosability, allowing the local diagnosers to be more efficient (Daigle et al., 2010). However, the approach proposed in (Daigle et al., 2010) still uses residual generators based on a global model of the system.

On the other hand, system decomposition methods, such as Possible Conflicts (PCs) (Pulido and Alonso-González, 2004), have been proposed to decompose a system model into minimal over-determined subsystems that suffice for fault diagnosis. PCs capture a subset of constraints or relations among the system variables that produce inconsistencies when faults occur. More formally, PCs are minimal subsets of equations containing sufficient analytical redundancy to generate fault hypotheses from observed measurement deviations. However, PCs were developed within the classical Consistency-based Diagnosis paradigm (Reiter, 1987), require the use of a central coordinator to compute the set of minimal diagnosis candidates based on activated or confirmed PCs.

In this work, we build on ideas from system decomposition with Possible Conflicts and event-based distributed diagnoser design as in (Daigle et al., 2010) to improve the design of independent local event-based diagnosers. This work contributes by incorporating PCs into the event-based distributed diagnosis framework, leading to more robust local diagnosers (if one local diagnoser fails, it does not affect the others), better design (obtaining smaller local event-based diagnosers, that are also independent on every level, even residual generation), and a generalization of PCs to multi-output residual generators. Results, using a multi-tank system as a case study, demonstrate the improved design of the proposed approach.

The paper is organized as follows. Section 2 describes the system modeling methodology and introduces the case study. Section 3 presents the theoretical concepts of our residual design approach. Section 4 describes the theoretical background for qualitative fault isolation and event-based diagnosis used.
in this paper. Section 5 discusses the local diagnoser design approach. Section 6 demonstrates the approach with different scenarios of the case study. Finally, Section 7 concludes the paper.

2 SYSTEM MODELING

We consider the problem of single fault diagnosis in continuous systems. We assume the system, \( S \), is described by

\[
\begin{align*}
\dot{x}(t) &= f(x(t), \theta(t), u(t)) + v(t), \\
y(t) &= h(x(t), \theta(t), u(t)) + w(t),
\end{align*}
\]

where \( x(t) \in \mathbb{R}^{n_x} \) is the state vector, \( \theta(t) \in \mathbb{R}^{n_{\theta}} \) is the parameter vector, \( u(t) \in \mathbb{R}^{n_u} \) is the input vector, \( v(t) \in \mathbb{R}^{n_v} \) is the process noise vector, assumed to be zero-mean Gaussian, \( f \) represents the set of state equations, \( y(t) \in \mathbb{R}^{n_y} \) is the output vector, \( w(t) \in \mathbb{R}^{n_w} \) is the measurement noise vector, assumed to be zero-mean Gaussian, and \( h \) represents the set of output equations. The dimension of a vector \( a \) is denoted by \( n_a \).

We denote a measurement as \( m \), which is a time-varying signal of \( y(t) \) obtained from an associated sensor. The measurement set is denoted as \( M \).

We consider single, abrupt, parametric faults, where faults are modeled as unexpected step changes in system parameter values. We name faults by the associated parameter and the direction of change, i.e., \( \theta^+ \) denotes a fault defined as an increase in the value of parameter \( \theta \), and \( \theta^- \) denotes a fault defined as a decrease in the parameter value. We denote a fault as \( f \) and a set of faults as \( F \).

Throughout the paper, we will use a multi-tank system as a running example. The tanks are connected serially as shown in Fig. 1, and we will consider a variable number of tanks. For tank \( i \), \( u_i \) denotes the input flow, \( C_i \) denotes the capacitance, and \( R_i \) denotes the resistance of the connected drain pipe. For tanks \( i \) and \( j \), \( R_{ij} \) denotes the resistance of the connecting pipe. For an \( n \)-tank system, the pressure of tank \( i \) is described by

\[
\dot{p}_i = \frac{1}{C_i} \left( u_i + q_{i-1,i} - q_i - q_{i,i+1} \right),
\]

with \( q_{0,1} = 0 \) and \( q_{n,n+1} = 0 \) for tanks \( i = 1 \) and \( i = n \), respectively. The output flow is defined as \( \{ q_i : i = 1, \ldots, n \} \), where \( q_i \) describes the output flow of tank \( i \), i.e.,

\[
q_i = \frac{1}{R_i} (p_i).
\]

The flows between tanks are defined as \( \{ q_{i,i+1} : i = 1, \ldots, n - 1 \} \), where \( q_{i,i+1} \) describes the flow from tank \( i \) to tank \( i+1 \), i.e.,

\[
q_{i,i+1} = \frac{1}{R_{i,i+1}} (p_i - p_{i+1}).
\]

The complete fault set \( F \) consists of \( \{ C_i^-, C_i^+, R_i^-, R_i^+ : i = 1, \ldots, n \} \cup \{ R_{i,i+1}^{-}, R_{i,i+1}^{+} : i = 1, \ldots, n - 1 \} \). The complete measurement set \( M \) is defined as \( \{ p_i, q_i : i = 1, \ldots, n \} \cup \{ q_{i,i+1} : i = 1, \ldots, n - 1 \} \).

3 RESIDUAL DESIGN

In previous works, we have developed a diagnosis framework, called TRANSCEND, where an observer, based on the global model of the system, is used to estimate the behavior of the system based on the set of measurements (Mosterman and Biswas, 1999). This estimation is then used to compute a residual for the measurement. We denote a residual \( r \), as a signal (typically generated by using the inputs and measurements of the system) that is zero when the system is fault-free, and non-zero when a fault appears in the system. The residual set is denoted as \( \hat{R} \). In TRANSCEND, a residual \( r \) is computed as the difference between an observation, \( y \), and the predicted nominal behavior of the output, \( \hat{y} \). Recently, system decomposition methods, like PCs, have been proposed to decompose a system model into minimal over-determined subsystems that suffice for fault diagnosis (Pulido and Alonso-González, 2004). These approaches decompose the global model into several independent minimal submodels, each with a single output. Each one of these minimal submodels estimates one measured variable, \( \bar{y} \), that it is compared with the observation, \( y \), to build the residual \( r \). Observers based on PCs are independent of each other, unlike a distributed observer scheme that uses the global model.

In both approaches, we define residuals with respect to a particular measurement. The main difference is the observer which produces the estimation \( \bar{y} \). With TRANSCEND, it is computed using the global model, whereas with PCs, it is computed using a minimal observer which estimates only a single variable using other measurements as additional input.

These two approaches represent two endpoints in the space of residual design. In this section, we first describe the fundamentals of the PC approach, then we generalize PCs to submodels with multiple outputs, and show how the TRANSCEND approach to residual design and the PC approach are special cases of a more general one. We will show in Section 5 how this generalization is necessary for efficient diagnoser design.

3.1 Possible Conflicts

PCs are minimal subsets of equations with sufficient analytical redundancy to generate fault hypotheses from observed measurement deviations. However, the PCs approach requires the use of a central coordinator to reason over the residual deviations among the different PCs to provide diagnosis results. PCs can be computed using hypergraphs (Pulido and Alonso-González, 2004) or Temporal Causal Graphs.
(TCGs) (Bregen et al., 2009b) as input. Here, we use the TCG-based approach as described in (Bregen et al., 2009b), since it allows to automatically include the temporal information in the PCs.

In this work, the global model of the system is denoted as $M$, and minimal submodels obtained from PCs are denoted as $M_i = (X_i, U_i, Y_i)$, where $X_i, U_i,$ and $Y_i$ are the state, input, and output variables of the submodel with measured variable $i$ as output, respectively. Using the PC approach with a three-tank system with $M = \{p_1, p_2, p_3\}$ we find a set of three minimal submodels: $M_{p_1} = \{\{p_1\}, \{u_1, p_2\}, \{p_1\}\}$, $M_{p_2} = \{\{p_2\}, \{u_2, p_1, p_3\}, \{p_2\}\}$, and $M_{p_3} = \{\{p_3\}, \{u_3, p_2\}, \{p_3\}\}$. For example, since the pressure in tank 1, $p_1$, is measured, a PC that estimates the pressure in tank 1 (that corresponds to minimal submodel $M_{p_1}$) is defined as follows:

$$p_1 = \frac{1}{C_1} \left( u_1 - \frac{1}{R_1} (p_1) - \frac{1}{R_{12}} (p_1 - p_2) \right),$$

where $p_1$ is the state variable, $u_1$ is the input to the tank, $p_2$ is the measured pressure of tank 2 that is it used as input for the PC, and $\{C_1, R_1, R_{12}\}$ is the subset of faults that affects the estimation of this PC. Note that this PCs are independent from $p_3$.

### 3.2 Generalizing Possible Conflicts

With PCs, each submodel is minimal, in the sense that it contains the minimum number of state variables to compute only a single output. Therefore, one PC, i.e., one minimal submodel, is derived for each system measurement. However, it is also possible to derive minimal multi-output submodels, i.e., submodels with multiple outputs. These may be constructed by merging the minimal submodels in various combinations. Additional residuals may then be defined for measurements within these minimal multi-output submodels. By merging all minimal submodels, we regain the global model, and the residuals defined using this model are the same as those defined in the TRANSCEND approach.

Formally, the merge operation $\oplus$ between two submodels is defined as follows.

**Definition 1** (Submodel Merging). Given two submodels $M_i = (X_i, U_i, Y_i)$ and $M_j = (X_j, U_j, Y_j)$, the merged submodel $M_{i,j} = M_i \oplus M_j$ is defined as $M_{i,j} = (X_{i,j}, U_{i,j}, Y_{i,j})$, where $X_{i,j} = X_i \cup X_j$, $U_{i,j} = (U_i \cup U_j) - (X_i \cup X_j)$, and $Y_{i,j} = Y_i \cup Y_j$.

The merged submodel must have all the states and outputs of its constituent submodels, and must have all the inputs, minus those that have become states in the merged submodel. We denote merged submodels by the outputs of their constituent submodels, e.g., the submodel formed by merging minimal submodels $M_{p_1}$ and $M_{p_2}$ is denoted as $M_{p_1, p_2}$. For the global model, we drop the subscripts and denote it as $M$.

For the three-tank, where the pressures are measured, the complete set of submodels is the following:

$$M_{p_1} = \{\{p_1\}, \{u_1, p_2\}, \{p_1\}\}$$
$$M_{p_2} = \{\{p_2\}, \{u_2, p_1, p_3\}, \{p_2\}\}$$
$$M_{p_3} = \{\{p_3\}, \{u_3, p_2\}, \{p_3\}\}$$
$$M_{p_1, p_2} = \{\{p_1, p_2\}, \{u_1, u_2, p_3\}, \{p_1, p_2\}\}$$
$$M_{p_1, p_3} = \{\{p_1, p_3\}, \{u_1, u_3, p_2\}, \{p_1, p_3\}\}$$
$$M_{p_2, p_3} = \{\{p_2, p_3\}, \{u_2, u_3, p_1\}, \{p_2, p_3\}\}$$
$$M = \{\{p_1, p_2, p_3\}, \{u_1, u_2, u_3\}, \{p_1, p_2, p_3\}\}$$

A residual may be defined for each measurement in each submodel. We denote a residual as $r_{m(M_i)}$, where $m$ is the measured variable estimated by the residual, and $(M_i)$ refers to the submodel with measurements $M_i$ as outputs used to compute the residual. For example, $r_{p_1(p_1, p_2)}$ denotes the residual that estimates the measured variable $p_1$ from submodel $M_{p_1, p_2}$. When the global system model is used, we drop the submodel subscript, e.g., $r_{p_1}$ denotes the residual that estimates the measured variable $p_1$ and uses the global system model $M$. For a system with $n_p$ measurements, a total of $2^{n_p} - 1$ submodels may be constructed. This results in at most $n_p2^{n_p-1}$ possible unique residuals.

### 4 QUALITATIVE EVENT-BASED DIAGNOSIS

Residuals, as described in the previous section, are triggered when faults occur in the system. Faults manifest as persistent abrupt changes in the values of the system parameters. The effects of the faults cause deviations in the observed measured variables from the nominal values. This section recapitulates the basic theoretical concepts needed to describe our diagnosis approach. We first review the theoretical framework for qualitative fault isolation and then the framework for event-based fault modeling.

#### 4.1 Qualitative Fault Isolation

Residual deviations caused by faults are abstracted using qualitative $+$, $-$, and $0$ values to form fault signatures (Mosterman and Biswas, 1999). Fault signatures represent these deviations as the immediate change in magnitude and the first nonzero derivative change.

**Definition 2** (Fault Signature). A fault signature for a fault $f$ and residual $r$ is the qualitative magnitude and slope change in $r$ caused by the occurrence of $f$, and is denoted by $\sigma_{f,r} \in \Sigma_{f,r}$.

In general, ambiguities may exist in the fault signatures, so $\sigma_{f,r}$ may not be unique. A fault signature is written as $\sigma_1 \sigma_2$, where $\sigma_1$ is the qualitative magnitude change and $\sigma_2$ is the qualitative slope change, e.g., $+-$.

We also capture the temporal order of residual deviations for a given submodel, termed relative measurement orderings (Daigle, 2008). Relative measurement orderings are based on the intuition that fault effects will manifest in some parts of the system before others. As described in Section 3, for a given submodel there is a residual defined for each measurement. Within this submodel, the relative ordering of the residual deviations for these measurements can be computed based on analysis of the transfer functions from faults to residuals defined for measurements.
between residuals of two different submodels because set of all measurement orderings for and of fault signatures and measurement orderings for and is shown in Table 2. In this case, the PCs are able to decouple the system, and so no effect appears in residuals and , but no effect appears in residuals and since the minimal submodels have an output measurement each, there are no orderings to be computed.

### 4.2 Event-based Fault Modeling

Fault signatures combined with relative measurement orderings provide event-based information for diagnosis. For a given fault, the combination of all fault signatures and measurement orderings yields all the possible ways a fault can manifest in the residuals. We denote each of these possibilities as a fault trace.

**Definition 4 (Fault Trace).** A fault trace for a fault over residuals , denoted by , is a string of length \(|R|\) that includes, for every \(r \in R\) that will deviate due to , a fault signature , such that the sequence of fault signatures satisfies \(\Omega_{f,R}\).

Note that the definition implies that fault traces are of maximal length, i.e., a fault trace includes deviations for all residuals 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 of a fault.

**Definition 5 (Fault Language).** The fault language of a fault \(f \in F\) with residual set \(R\), denoted by \(\mathcal{L}_{f,R}\), is the set of all fault traces for \(f\) over the residuals in \(R\).

**Definition 6 (Fault Model).** The fault model for a fault \(f \in F\) with residual set \(R\), denoted by \(\mathcal{L}_{f,R}\), is the finite automaton that accepts exactly the language \(\mathcal{L}_{f,R}\), and is given by \(\mathcal{L}_{f,R} = (S, s_0, \Sigma, \delta, \lambda)\) where \(S\) is a set of states, \(s_0 \in S\) is an initial state, \(\Sigma\) is a set of events, \(\delta : S \times \Sigma \rightarrow S\) is a transition function, and \(A \subseteq S\) is a set of accepting states.

The finite automata representation allows for the composition of the fault signatures and measurement orderings into fault models. The possible fault signatures and measurement orderings can be composed automatically to form the fault models based on the synchronization operation (Daigle et al., 2009).

Selected fault models for a three-tank system are shown in Fig. 2. For example, as seen in \(\mathcal{L}_{C_2,R}\), the fault \(C_2\) may manifest as the fault traces \(r_{p_2}^r p_1^r p_3^r p_1^r\) and \(r_{p_2}^r p_3^r p_1^r\), as implied by the fault signatures and measurement orderings.

<table>
<thead>
<tr>
<th>Fault</th>
<th>(r_{p_1})</th>
<th>(r_{p_2})</th>
<th>(r_{p_3})</th>
<th>Measurement Orderings</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C_1)</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>- 0+ 0+ + 0+ 0+</td>
</tr>
<tr>
<td>(R_1)</td>
<td>+</td>
<td>- +</td>
<td>- + +</td>
<td>0+ 0+ 0+ + - 0+</td>
</tr>
<tr>
<td>(R_{12})</td>
<td>-</td>
<td>+</td>
<td>- +</td>
<td>0+ 0+ 0+ - + - 0+</td>
</tr>
<tr>
<td>(C_2)</td>
<td>+</td>
<td>-</td>
<td>- +</td>
<td>0+ 0+ 0+ - + + 0+</td>
</tr>
<tr>
<td>(R_2)</td>
<td>+</td>
<td>-</td>
<td>- +</td>
<td>0+ 0+ 0+ - + + 0+</td>
</tr>
<tr>
<td>(R_{23})</td>
<td>-</td>
<td>+</td>
<td>- +</td>
<td>0+ 0+ 0+ - + + 0+</td>
</tr>
<tr>
<td>(C_3)</td>
<td>-</td>
<td>-</td>
<td>- +</td>
<td>0+ 0+ 0+ - + + 0+</td>
</tr>
<tr>
<td>(R_3)</td>
<td>+</td>
<td>-</td>
<td>- +</td>
<td>0+ 0+ 0+ - + + 0+</td>
</tr>
</tbody>
</table>

**Table 2: Fault Signatures and Relative Measurement Orderings for the set of minimal submodels of the Three-tank System.**

<table>
<thead>
<tr>
<th>Fault</th>
<th>(r_{p_1}(p_1))</th>
<th>(r_{p_2}(p_2))</th>
<th>(r_{p_3}(p_3))</th>
<th>Measurement Orderings</th>
</tr>
</thead>
<tbody>
<tr>
<td>(C_1)</td>
<td>+</td>
<td>0</td>
<td>0</td>
<td>- 0+ 0+</td>
</tr>
<tr>
<td>(R_1)</td>
<td>+</td>
<td>0</td>
<td>0</td>
<td>+ 0+ 0+</td>
</tr>
<tr>
<td>(R_{12})</td>
<td>+</td>
<td>0</td>
<td>0</td>
<td>- 0+ 0+</td>
</tr>
<tr>
<td>(C_2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+ 0+ 0+</td>
</tr>
<tr>
<td>(R_2)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+ 0+ 0+</td>
</tr>
<tr>
<td>(R_{23})</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+ 0+ 0+</td>
</tr>
<tr>
<td>(C_3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+ 0+ 0+</td>
</tr>
<tr>
<td>(R_3)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+ 0+ 0+</td>
</tr>
</tbody>
</table>

Figure 2: Fault models for some faults of the three-tank system, where \(R = \{r_{p_1}, r_{p_2}, r_{p_3}\}\).
5 DISTRIBUTED DIAGNOSER DESIGN

Diagnoser design is based on the diagnosability of the system. In this work we use the notions of global and local diagnosability as the conditions for the local diagnoser to achieve globally correct results, as in (Roychoudhury et al., 2009; Daigle et al., 2010). So we first define notions of diagnosability in our framework, then, we describe the diagnoser design algorithm, and finally, we summarize how we build the local event-based diagnosers.

5.1 Diagnosability

Given a model of a system, and the set of faults \( F \) and residuals \( R \), we may now establish the notions of distinguishability and diagnosability. Using these definitions, we can then formally define the distributed diagnoser design problem. Distinguishability between faults is characterized as follows.

Definition 7 (Distinguishability). With residuals \( R \), a fault \( f_i \) is distinguishable from a fault \( f_j \), denoted by \( f_i \not\cong_R f_j \), if \( f_i \) always eventually produces effects on the residuals 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 (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.

As we previously described, the set of possible effects on residuals due to a fault is called a fault language. Using this definition we define a system within our framework as follows.

Definition 8 (System). A system \( S \) is a tuple \((F, M, R, L_F, R)\), where \( F = \{f_1, f_2, \ldots , f_n\} \) is a set of faults, \( M = \{m_1, m_2, \ldots , m_n\} \) is a set of measurements, \( R \) is a set of residuals, and \( L_F = \{L_{f_1, R}, L_{f_2, R}, \ldots , L_{f_n, R}\} \) is the set of fault languages.

If a system is diagnosable, then we can make guarantees about the unique isolation of every fault in the system.

Definition 9 (Diagnosability). A system \( S = (F, M, R, L_F, R) \) is diagnosable if \((\forall f_i, f_j \in F)f_i \neq f_j \implies f_i \not\cong_R f_j \).

If \( S \) is diagnosable, then every pair of faults is distinguishable using the residual set \( R \). Hence, we can uniquely isolate all faults of interest. If \( S \) is not diagnosable, then ambiguities will remain after fault isolation, i.e., after all possible fault effects on the residuals have been observed. For example, consider the \( M \)-based residual set given in Table 1. The system defined with these residuals is diagnosable when both signatures and orderings are used (without orderings, faults \( R^+_1 \), \( R^-_1 \), and \( R^+_2 \) cannot be distinguished because they all produce the same signature). However, given the PC-based residuals (derived from the minimal submodels), the system is not diagnosable since fault \( R^+_1 \) cannot be distinguished from fault \( R^-_1 \).

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

2 If the system \( S \) is not diagnosable, we can define aggregate faults, where an aggregate fault is a set of faults that are indistinguishable from each other. The diagnosis methodology can be applied to the modified fault set that includes the aggregate faults (Roychoudhury et al., 2009).
in $R_i^+$, and (iii) $S_i^* = (F_i, M_i^*, R_i^+, L_{F_i}^+, R_i^-)$ is globally diagnosable.

5.2 Diagnoser Design Algorithm

The diagnoser design problem is, in general, a measurement selection problem, which is an instance of the set covering problem, known to be NP-complete (Narasimhan et al., 1998). The complexity of the design problem increases with the number of residuals, and, as shown in Section 4, the complete residual set grows exponentially with the number of measurements. Therefore, we need to use heuristics to guide the search.

The advantage of PCs is that they decouple the effects of all faults, whose effect on the output measurement of the PC residual only happens through one of the measurements that are considered as input to the PC, i.e., there is no direct path in the TCG from a fault to the measurement residual (without going through other measurements that are considered inputs to this residual). This results in an improvement in diagnosability in a local sense. The intuition, then, is that including PC-based residuals will lead to improved diagnoser designs because of this improvement of diagnosability. So, one may simply apply the algorithm presented in (Daigle et al., 2010) on the PC-based residual set. However, there are two problems. First, the system may not be diagnosable with only the PC-based residuals (see the example in the previous subsection), even if it is diagnosable with the residuals based on the global model $M$, and second, measurement orderings cannot be derived for the PC-based residuals, so diagnosable performance may be decreased relative to a centralized diagnoser that uses measurement orderings.

Assume that the system is split into three subsystems, $S_1$, $S_2$, and $S_3$, where for $S_1$, $F_1 = \{C_1^-, R_1^+, R_{12}^+\}$, $M_1 = \{p_1\}$, for $S_2$, $F_2 = \{C_2^-, R_2^+, R_{23}^+\}$, $M_2 = \{p_2\}$, and for $S_3$, $F_3 = \{C_3^-, R_3^+, R_{34}^+\}$, $M_3 = \{p_3\}$. Say that we use the $M$-based residuals, so $R_1 = \{r_{p_1}\}$, $R_2 = \{r_{p_2}\}$, and $R_3 = \{r_{p_3}\}$. Analyzing global diagnosability, we see that none of the subsystems are globally diagnosable, i.e., we will have to add new residuals to each subsystem in order to satisfy our design constraints. Now assume that we use the PC-based residuals, $R_1 = \{r_{p_1(p_1)}\}$, $R_2 = \{r_{p_2(p_2)}\}$, and $R_3 = \{r_{p_3(p_3)}\}$. We see that now $S_3$ is globally diagnosable because only one nonlocal fault, $R_{23}^+$, produces an effect on $r_{p_3(p_3)}$, and it is a different effect from those produced by the local faults. So if $S_3$ uses the PC-based residual instead of the global model-based residual, it can have an improved diagnoser design. But, the other subsystems are not globally diagnosable, and cannot be made so by including any other PC-based residual, because those subsystems contain the faults that make the system as a whole nondiagnosable using only the PC-based residuals. This suggests that we require a more general approach that combines both PC-based residuals and the $M$-based residuals. In general, we need to consider residuals from the complete set considering all possible submodels.

But, as previously pointed out, the complexity of the design problem is dependent on the number of possible residuals, and the complete set is too large. Further, there is much overlap of information between the different residuals, for instance, compare Tables 1 and 2. Instead, we perform a search over the measurement space, which is much smaller, and define residuals in a particular way for a given set of measurements. Specifically, given a set of measurements $M_i$, we use the residuals for the submodel that includes exactly the measurements in $M_i$ as outputs, denoted using $r_{m(M_i)}$ for measurement $m$. We then incrementally expand the submodel of each subsystem to include additional measurements (and, hence, a larger set of residuals) in order to satisfy the global diagnosability criterion.

The diagnoser design algorithm is shown in Algorithm 1. For each subsystem, we first construct the set of residuals for its current measurement set. We then determine a subset of measurements over $M' \subseteq M - M_i$ over which we will consider adding to the subsystem using the computeMSubset function. In our particular implementation, we simply set $M'$ equal to $M - M_i$, but, in general, this may include heuristics such as the subsystem distance heuristic developed in (Roychoudhury et al., 2009). We then identify the best (with respect to global diagnosability) subset of measurements $M_i'$ within $M'$ to add to $M_i$, using the findBestM function. We then update $M_i$, reconstruct the residual set for the new measurement set, and continue in this fashion until $S_i$ is globally diagnosable.

In our particular implementation, we used the findBestM function shown as Function 2. Here, we select only the single best measurement, rather than a subset of measurements. For each possible measurement to add, we construct the new set of residuals, then determine the set of faults $F_i^*$ that are not globally distinguishable for the subsystem and this residual set. The measurement that results in the smallest $F_i^*$ is selected as the best measurement and becomes the output $M_i'$. Adding measurements incrementally, and especially one at a time, is, in general, nonoptimal, but here we tradeoff optimality for computational efficiency. More complex versions of this function are also possible.

We apply this algorithm to the three-tank system, where for tank $i$, for $i = 1, \ldots, n - 1$, $S_i$ is defined by $F_i = \{C_i^-, R_i^+, R_{i+1}^+\}$ and $M_i = \{p_i\}$, and for $i = n$, $S_i$ is defined by $F_i = \{C_n^-, R_n^+\}$ and $M_i = \{p_i\}$. As a result, we have to add one
residual only to the subsystems $S_1$ and $S_2$, and none have to be added to subsystem $S_3$, because, as shown previously, the subsystem is already globally diagnosable with only $r_{p_3(p_3)}$.

Subsystem $S_1$ gets residuals $r_{p_3(p_1,p_2)}$ and $r_{p_3(p_1,p_2)}$, and subsystem $S_2$ gets residuals $r_{p_2(p_2,p_3)}$ and $r_{p_3(p_2,p_3)}$. This improves the algorithm presented in (Daigle et al., 2010), because in that case, subsystem $S_2$ needs three residuals, and subsystem $S_3$ needs two residuals, so the size of the event-based diagnosers is improved.

There is also a second way in which the design is improved over the approach of (Daigle et al., 2010). In that approach, each subsystem used the global model for residual generation. In the approach developed in this paper, however, each subsystem needs only a submodel for residual generation. So, residual generation will be more efficient. In fact, this will always be the case, because the only time a subsystem will end up using the global model is if it adds all the measurements to the subsystem. This is a worst-case design, and, on average, each subsystem will only use a subset of the measurements, and, therefore, a subset of the global model for residual generation.

5.3 Diagnoser Implementation

Once we have designed the distributed diagnosis system, event-based diagnosers may be constructed. An event-based diagnoser, $D_{F,R}$, for fault set $F$ and residual set $R$, is a finite automaton extended by a set of diagnoses and a diagnosis map and is similar in concept to DES diagnosers such as (Sampath et al., 1996). It takes events as inputs, which, as with fault models, correspond to residual deviations. From the current state, a residual 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. The diagnoser is constructed to capture the fault languages and link fault traces to diagnoses. Details of this procedure can be found in (Daigle et al., 2009). The design of local diagnosers follows the same procedure as the global diagnoser, i.e., given $F_i$ and $R_i$ for subsystem $S_i$, we construct $D_{F_i,R_i}$. The local diagnosers for the distributed diagnoser design example for the three-tank system are given in Fig. 3. Accepting states correspond to globally correct diagnosis.

6 RESULTS

This section shows the applicability of the proposed design approach. First, we show different design scenarios and compare the design obtained with the new approach against the design obtained using the previous approach in (Daigle et al., 2010). Then we show an example to demonstrate online diagnosis in this new framework.

6.1 Distributed Design Experiments

As a first design scenario, consider the three-tank system with $F = \{C^-_3, C^-_1, C^-_3, R^+_{12}, R^+_{23}\}$ and $M = \{p_1, p_2, p_3\}$. Now, assume that the system is split into three subsystems, $S_1$, $S_2$, and $S_3$, where for $S_1$, $F_1=\{C^-_1, R^+_{12}\}$, $M_1=\{p_1\}$, for $S_2$, $F_2=\{C^-_2, R^+_{23}\}$, $M_2=\{p_2\}$, and for $S_3$, $F_3=\{C^-_3\}$, $M_3=\{p_3\}$. If we use the PC-based residuals, $R_1=\{r_{p_1(p_1)}\}$, $R_2=\{r_{p_2(p_2)}\}$, and $R_3=\{r_{p_3(p_3)}\}$ we see that all three subsystems, $S_1$, $S_2$, and $S_3$, are globally diagnosable. This is clear from the set of fault signatures obtained using these residuals, shown in Table 3. The PCs decouple the subsystems to the extent that only the $R^+_{12}$ faults affect multiple subsystems, and the effects they produce are unique. Hence, no design is needed in this case, and we will be able to use the minimal PC-based residuals instead of the global model-based residuals. This improves over the previous, because in that case, subsystem $S_1$ needs two residuals, and subsystem $S_2$ also needs two residuals, so the size of the event-based di-
Table 3: Fault Signatures and Relative Measurement Orderings for the Set of Minimal Submodels of the Three-tank System with \( F=\{C_1^-, C_2^+, C_3^+, R_{12}, R_{23}\} \) and \( M=\{r_{p1}, p_{p2}, p_{p3}\} \).

<table>
<thead>
<tr>
<th>Fault</th>
<th>( T_{p1}(p_{p1}) )</th>
<th>( T_{p2}(p_{p2}) )</th>
<th>( T_{p3}(p_{p3}) )</th>
<th>Measurement Orderings</th>
</tr>
</thead>
<tbody>
<tr>
<td>( C_1^- )</td>
<td>+</td>
<td>0 0</td>
<td>0</td>
<td>∅</td>
</tr>
<tr>
<td>( R_{12}^+ )</td>
<td>+ 0</td>
<td>0</td>
<td>0</td>
<td>∅</td>
</tr>
<tr>
<td>( C_2^+ )</td>
<td>0</td>
<td>+ 0</td>
<td>0</td>
<td>∅</td>
</tr>
<tr>
<td>( R_{23}^+ )</td>
<td>0 0</td>
<td>0</td>
<td>+</td>
<td>∅</td>
</tr>
<tr>
<td>( C_3^- )</td>
<td>0</td>
<td>0</td>
<td>+</td>
<td>∅</td>
</tr>
</tbody>
</table>

The distributed diagnosis framework relates to distributed discrete-event system (DES) diagnosis methods like (Debeuk et al., 2000). The local diagnosers are designed to provide globally correct diagnosis results, contrasting with other DES approaches such as (Pencolé and Cordier, 2005), where a merge operation of diagnosis results is necessary to obtain the global diagnosis. The abstraction of the continuous dynamics into an event-based representation is also similar to (Meseguer et al., 2010; Bayoudh et al., 2006).

In future work, we will integrate the proposed approach within a diagnosis framework that goes from ...
fault detection to fault identification, where we will exploit additional properties of the minimal submodels (like the computation of minimal parameter estimators for fault identification (Bregon et al., 2009a)). We also plan to extend the approach to multiple faults, based on results presented in (Daigle, 2008).

REFERENCES


