Improving the Diagnostic Performance for Dynamic Systems through the use of Conflict-Driven Model Decomposition

Anibal Bregon[1], Alexander Feldman[2], Belarmino Pulido[3], Gregory Provan[4], and Carlos Alonso-González[5]

[1,3,5] *Depto. de Informática, Universidad de Valladolid, Spain*
{*anibal,belar,calonso*}*@infor.uva.es*

[2,4] *Dept. of Computer Science, University College Cork, Ireland*
alex@llama.gs;g.provan@cs.ucc.ie

Abstract

This work studies potential ways of integration of two techniques for fault detection, isolation, and identification in dynamic systems: Lydia-NG suite of diagnosis algorithms and Consistency-based Diagnosis with Possible Conflicts. By integrating both techniques, Lydia-NG will benefit from a more efficient fault detection and isolation task, and Possible Conflicts will benefit from the identification capabilities of Lydia-NG. In this paper, we define a common framework that integrates both techniques, and then we apply the proposed integrated approach to a three-tank system, and draw some conclusions about potential ways of integration.

1. Introduction

The need for safety and reliability in engineering systems provides the motivation for developing Integrated Systems Health Management (ISHM) methodologies that include efficient fault diagnosis mechanisms. In this work we focus on model-based approaches to on-line fault diagnosis of dynamic systems. Online methods for model-based diagnosis require the use of quick and robust fault detection methods to establish discrepancies between observed and expected system behavior. Discrepancies due to faults trigger fault isolation processes that are responsible for determining the root cause of the fault. However, accurate and timely online fault diagnosis of complex dynamic systems is difficult and can be computationally expensive (Pouliezos & Stavrakakis, 1994; Isermann, 2006; Gertler, 1998).

In this work we have used the Lydia-NG modeling language and the Lydia-NG suite of algorithms (Feldman, Provan, & Gemund, 2010). The main idea of Lydia-NG is to perform multiple simulations for various hypothesized health states of the plant. The output of these multiple simulations is then processed and combined into single diagnostic output. Lydia-NG implements several strategies for the generation of fault candidates and a number of algorithms for active testing. These algorithms are based on AI search and include best-first, and bottom-up greedy search. Lydia-NG has been successfully used for complex applications like space satellites (Feldman, Castro, Gemund, & Provan, 2013). However, when applied to online fault diagnosis of large dynamic systems, running all the hypothesized health states becomes a quite difficult and time consuming task.

Several approaches have been proposed in recent years to deal with the complexity issue. System decomposition methods, have been proposed to reduce the complexity in the fault diagnosis task (Bregon, Biswas, & Pulido, 2012) by generating smaller simulation submodels which can run in parallel and provide independent diagnosis decisions. The Possible Conflict, PC, approach (Pulido & Alonso-González, 2004), is an off-line dependency compilation technique from the DX community, which decomposes the global system model into minimal submodels, and performs on-line behavior estimation using simulation (Bregon, Alonso, Biswas, Pulido, & Moya, 2012), state observers (Daigle, Bregon, & Roychoudhury, 2012), dynamic Bayesian networks (Alonso-González, Moya, & Biswas, 2011), or state-based neural networks (Pulido, Zamarreño, Merino, & Bregon, 2012). If a discrepancy is found, a set of fault candidates is generated by a minimal hitting-set algorithm of the triggered PCs. However, additional techniques must be used to refine the set of

fault candidates.

The goal of this work consists of integrating PCs within the Lydia-NG diagnosis framework. First, PCs will decompose the global simulation model into a set of smaller simulation submodels. Then, PCs will be used for efficient online fault detection and fault localization, providing a subset of fault candidates from the minimal hitting-set of the deviated PC residuals. The subset of fault candidates is then used as input to Lydia-NG, where simulations are run only for each one of the fault candidates, and its result is processed and combined to provide the diagnosis output.

We have tested the proposed diagnosis framework by using a three-tank system. Theoretical study on the integration proposal shows that the complexity of LYDIA-NG is highly reduced by the inclusion of PCs in the framework. The experimental results present a particular diagnosis scenario where the proposed integration framework is used.

The rest of the paper is organized as follows. Section 2 presents the basic definitions and running example used in this work. Section 3.1 briefly introduces Lydia-NG, and section 3.2 describes basic ideas of system decomposition using PCs. Section 4 presents our proposal to integrate PCs within the Lydia-NG diagnosis framework. Section 5 describes the experimental results obtained for a three-tank system. Section 6 presents related work. And, finally, section 7 presents the discussion and conclusions.

2. Concepts and Definitions

In this section we present our basic definitions and a running example that we use to illustrate the significant concepts of this paper.

Since both LYDIA-NG and PCs are model-based diagnosis approaches, we provide a set of definitions about models and faults that will enable us to further explain both techniques using the same framework.

2.1. Definitions

For the purpose of this work we will focus our description on continuous systems diagnosis, with only one nominal state, and whose behavior can be described as a set Σ of Ordinary Differential Equations (ODEs). The model of our system will be the basic system description to perform diagnosis:

Definition 1 (Model)**.** The system model can be defined as $M(\Sigma, U, Y, X, \Theta)$, where: Σ is a set of ODEs, defined over a collection of known and unknown variables: U is a set of inputs, Y a set of outputs, X a set of state and intermediate i.e., unknown) variables, and Θ is the set of Model Parameters.

Definition 2 (System Description)**.** SD is made up of (M, H, σ, Π), where,

- H is the health-vector defined by means of $(h_1, \ldots, h_k)$, health variables, that allow us to characterize the set of states in the system, i.e. each $h_i \in H$ is a potential mode for the system. either nominal or faulty.
- σ is a mapping function: $\sigma(M, H_c) \rightarrow M_{Hc}(\Sigma_{Hc}, U_{Hc}, Y_{Hc}, X_{Hc}, \Theta_{Hc})$, that given the system description, M, and the current health status, Hc, provides the model for behavior estimation for the current mode (or current system description M_{Hc}): $\Sigma_{Hc} \subseteq \Sigma$, $U_{Hc} \subseteq U$, $Y_{Hc} \subseteq Y$, $X_{Hc} \subseteq X$, and $\Theta_{Hc} \subseteq \Theta$).
- Π is a mapping function $\Pi(\theta_{cc}) \rightarrow \{Hc \mid Hc \subseteq H\}$ that, given a set of parameters, provides the set of health statuses that relate to the set of model parameters: $\theta_{cc} \subseteq \Theta_{cc}$.

For consistency-based diagnosis using PCs, we only use $\sigma(M, Hn)$ with Hn corresponding to a nominal mode. Since we are dealing with a continuous system working in one nominal mode, we can compute offline the set of PCs for $M_{Hn}(\Sigma_{Hn}, U_{Hn}, Y_{Hn}, X_{Hn}, \Theta_{Hn})$, as will be described later.

An implicit assumption in our modeling approach is that we can use the same set of equations for both the nominal behavior estimation and the faulty behavior estimation, just changing the value of θ_i in equation $c_i \in \Theta$.

Consistency Based Diagnosis (CBD) performs fault detection and fault isolation using only models of correct behavior in a *two stage process*. First, we identify if there exists a discrepancy between the observed behavior (obtained via sensor outputs y_i) and the expected behavior (estimated by the Possible Conflict, $\hat{y}_{i_{pc}}$). We define a discrepancy in terms of a residual.

Definition 3 (Residual)**.** A residual is a real-valued measure $R(y_i, \hat{y}_{i_{pc}})$ of the difference between real and simulated system output at time t.

Corresponding to each residual is a conflict (Reiter, 1987). Hence, fault detection consists of computing every conflict.

The second step is fault isolation. *Fault isolation* consists of computing the minimal hitting sets of the conflicts. Conflicts are important since the *set of minimal diagnoses* of a system is given by the minimal hitting set of the set of minimal conflicts (Reiter, 1987). Intuitively, a conflict is a set of components that cannot be-

have properly simultaneously, given the system description and current observations of abnormal behavior. In other words, given a system description and some current observations, they entail that at least one component of the conflict must be faulty, in the sense that it departs from the behavior described by its model of correct behavior.

There is no general framework for CBD of dynamic systems. Nevertheless, most existing approaches rely upon an iterative process of behavior estimation, conflict detection, and fault candidate generation (i.e. fault isolation). In this work we use the Possible Conflicts (PCs) approach to avoid the on-line computation of conflicts and speed up overall fault isolation. PCs are designed to compute off-line those subsystems capable of becoming conflicts online.

The output of the consistency-based diagnosis using PCs is a set of fault candidates C defined in the lattice provided by Θ^*.

2.2. Running Example

In this paper, we use the three-tank system shown in figure 1 as the running example. The three tanks are denoted as T_1, T_2, and T_3. They all have the same area $A_1 = A_2 = A_3 = 3$ [m^2]. The experiments are performed assuming the gravity $g = 10$ and the liquid with density $\rho = 1$.

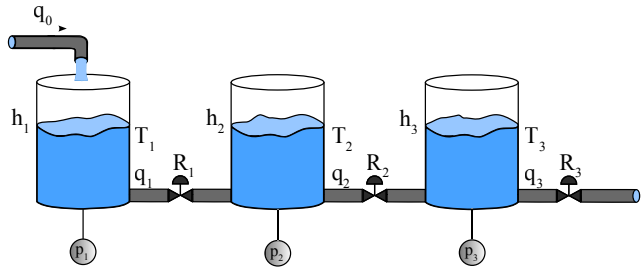

Figure 1. Diagram of the three-tank system.

Tank T_1 gets filled from a pipe q_0 with a constant flow of 1.5 [m^3/s]. It drains into T_2 via a pipe q_1. The liquid level is denoted as h_1. There is a pressure sensor p_1 connected to T_1 that measures the pressure in Pascals [Pa]. Starting from the Newton's (and Bernoulli's) equations and manipulating them (the actual derivation is trivial and irrelevant in this paper) we derive the following Ordinary Differential Equation (ODE) that gives the level of the liquid in T_1:

$$\frac{dh_1}{dt} = \frac{q_0 - k_1\sqrt{h_1 - h_2}}{A_1} \tag{1}$$

In eq. 1, the coefficient k_1 is used to model the area of the drainage hole and its friction factor. We emphasize the use of k_1 because, later, we will be "diagnosing" our system in term of changes in k_1. Consider a physical valve R_1 between T_1 and T_2 that constraints the flow between the two tanks. We can say that the valve changes proportionally the cross-sectional drainage area of q_1 and hence k_1. The diagnostic task will be to compute the true value of k_1, given p_1, and from k_1 we can compute the actual position of the valve R_1. The water levels of T_2 and T_3, denoted as h_2 and h_3 respectively, are given by:

$$\frac{dh_2}{dt} = \frac{k_1\sqrt{h_1 - h_2} - k_2\sqrt{h_2 - h_3}}{A_2}, \tag{2}$$

$$\frac{dh_3}{dt} = \frac{k_2\sqrt{h_2 - h_3} - k_3\sqrt{h_3}}{A_3}. \tag{3}$$

Values k_1, k_2, and k_3, are constant values with no physical meaning, and we have set them with a value of 0.75. Finally, we turn the water level into pressure:

$$p_i = \frac{g\, h_i\, A_i}{A_i} = g\, h_i \tag{4}$$

where i is the tank index ($i \in \{1, 2, 3\}$). Hence, there are three equations relating the pressure in the tank with the level in the tank. To observe the behavior of the system we have an observational model, that allows us to know or read each value p_i. We use p_i^* to distinguish the measured variable from the model output p_i:

$$p_i^* = p_i \tag{5}$$

It is assumed that the initial water level in the three tanks is zero. Additionally, we make explicit the relation between the state variables, h_i in our example, and their derivatives, dh_i:

$$h_i = \int dh_i \cdot dt \tag{6}$$

These equations allow us to select an integral or differential approach for behavior simulation, depending on the selected causality. These equations make no influence in the diagnosis results, because they will have no θ_i, and consequently no health status.

3. Algorithms

This section presents the fundamental ideas of the Lydia-NG diagnosis framework and the structural model decomposition approach with PCs.

3.1. Lydia-NG

We next show an algorithmic framework for computing diagnoses. The basic idea of the LYDIA-NG diagnostic library (shown in Fig. 2) is to perform multiple simulations for various hypothesized health states of the plant. The output of these multiple simulations is then processed and combined into single diagnostic output.

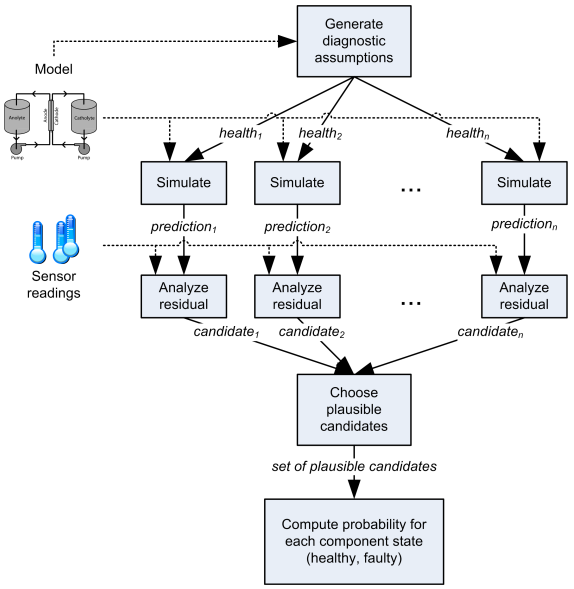

Figure 2. Overview of the LYDIA-NG diagnostic method

The LYDIA-NG diagnostic library consists of the following building blocks:

Generator of Diagnostic Assumptions: A diagnostic assumption is a set of hypothetical assignments for the health or fault state of each component in the system. The "all nominal" diagnostic assumption assigns healthy status to each component. LYDIA-NG allows one nominal and one or more faulty states per component.

Simulation Engine: Given a diagnostic assumption, LYDIA-NG can construct a simulation model of the system. This simulation model consists of equations. By solving this system of equations LYDIA-NG computes values for one or more *observable* variables. The values of these observable variables is also referred to as a *prediction*.

Residual Analysis Engine: A prediction is compared to the sensor data by a residual analysis engine. This engine combines the individual discrepancies in each sensor data/predicted variable pair to produce a single real value that indicates how close is the prediction of the simulation engine to the sensor data obtained from the plant. A simulation that results in all predicted values coincide with the measured ones will result in the residual being close to zero. The data structure containing predictions, their corresponding sensor data and the computed residual is called a *diagnostic candidate* or simply *candidate*.

Candidate Selection Algorithm: Not all candidates generated by the residual analysis engine are used for computing the final system health. The candidate selection algorithm discards each candidate whose residual is larger than the residual of the "all nominal" candidate.

System State Estimation Algorithm: LYDIA-NG uses the set of candidates that is computed by the candidate selection algorithm to compute an estimate for the health of each component. This is done by the system state estimation algorithm.

Algorithm 1 shows the top-level diagnostic process. The inputs to Alg. 1 are a model and a scenario, and the result is a diagnosis.

Algorithm 1 supports a large variety of simulation methods that may or may not use time as an independent variable. The only requirement toward the simulation engine is to predict a number of variables whose types can be mapped to LYDIA-NG and to be relatively fast.

Algorithm 1 Diagnosis framework

```
1: function DIAGNOSE(SCN) returns a diagnosis
       inputs: SCN, diagnostic scenario
       local variables: h, FDI vector, health assignment
                        p, real vector, prediction
                        Ω, a set of diagnostic candidates
                        DIAG, diagnosis, result
2:     while h ← NEXTHEALTHASSIGNMENT() do
3:         p ← SIMULATE(M, γ, h)
4:         r ← COMPUTERESIDUAL(p, α)
5:         Ω ← Ω ∪ ⟨h, r⟩
6:     end while
7:     DIAG ← COMBINECANDIDATES(Ω)
8:     return DIAG
9: end function
```

The basic idea of Alg. 1 is to simulate for various health assignments and to compare the predictions with the observed sensor data (i.e., telemetry). There are several important aspects of this algorithms that ultimately affect the diagnostic accuracy as measured by various performance metrics.

The first algorithmic property that determines many of the diagnostic performances is the order in which health-assignments are generated. In Alg. 1 this is implemented in the NEXTHEALTHASSIGNMENT function. The latter subroutine also determines when to stop the search and

should be properly parametrized depending on the model and the user requirements. In the standard LYDIA-NG diagnostic library we provide the following diagnostic search policies:

Breadth-First Search (BFS): This policy first generates the nominal health assignment, then single-faults, double-faults, etc.

Depth-First Search (DFS): This search policy starts with the nominal health assignment, then adds a single-fault, continues with a double fault including the first, and so on, until all components are failed. After the all-faulty assignment is generated, the algorithm backtracks one step and generates a sibling assignment and continues traversing down and backtracking in the same manner until no more backtracking is possible.

Backwards Greedy Stochastic Search (BGSS): In this mode, the search start from the all-faulty assignment. A random health variable is then flipped and the flip is retained iff the flip leads to a decrease in the residual. The order of health variables is arbitrary. As the whole search process is stochastic, it needs to be run multiple iterations in order to achieve the desired completeness. A formal description of this method for Boolean circuit models can be found in (Feldman et al., 2010).

Each simulation produces what we call a *candidate*: a set of predicted values for a given health-assignment. The second important property of Alg. 1 is the comparison and ordering of the diagnostic candidates. This is done by mapping the predicted and observed variables into a single real-number, called a *residual*.

Residual generation functions in LYDIA-NG bear resemblance to loss functions in decision theory. For example, residuals may be squared or absolute residuals (Feldman et al., 2013). A disadvantage of the squared residuals function R_{sq} is that it adds a lot weight to outliers. In decision theory, the absolute loss function that corresponds to the R_{abs} function is discontinuous. The latter, however, is not a problem for the algorithms described in this paper and we prefer R_{abs} over R_{sq}.

3.2. Consistency-based diagnosis with PCs

3.2.1. Model decomposition with PCs

The Possible Conflicts (PCs) approach (Pulido & Alonso-González, 2004) is a model decomposition method that finds (off-line) every subset of equations capable of generating conflicts. PCs provide the structural and causal model of a subsystem with minimal redundancy. The set of equations in a PC can be used to simulate the correct behavior of the subsystem. Hence, PCs can be used in CBD of dynamic systems (PULI01, 2001). For the sake of self-containment, we summarize here the proposal for PCs computation given in (Pulido & Alonso-González, 2004).

To compute PCs, we need the structural model of the system under study, which can be obtained from the set of equations in the system description, once we select a given working mode, tailored for our new problem formulation, instead of the original process which was suitable for system descriptions provided as hypergraphs (Pulido & Alonso-González, 2004).

We will illustrate the process using the three-tank system in Fig. 1, and the set of equations in its model as described in Section 2.2.

We need an abstraction of our model description $SD = (M, H, \sigma, \Pi)$. Let's assume we compute the set of PC for a given nominal mode characterized by H_n. Using $\sigma(M, H_n)$, we obtain $M_{H_n} = (\Sigma_{H_n}, U_{H_n}, Y_{H_n}, X_{H_n}, \Theta_{H_n})$. For the structural model, we only need the information about the measured and unknown variables in each model equation. Thus each equation in $\Sigma_{H_{ni}}$ will provide one structural constraint $(c_i,\ S_i,\ X_i)$, where S_i accounts for the measured variables from U_{H_n}, Y_{H_n} in c_i, and X_i accounts for the unknown (state or intermediate variables in c_i).

For the three tank system the structural model is made up of the following constraints:

Constraint	*Sensors*	*Unknowns*
c_1	$\{q_0\}$	$\{d_{h1}, h_1, h_2\}$
c_2	$\{\}$	$\{d_{h2}, h_1, h_2, h_3\}$
c_3	$\{\}$	$\{d_{h3}, h_2, h_3\}$
c_4	$\{\}$	$\{p_1, h_1\}$
c_5	$\{\}$	$\{p_2, h_2\}$
c_6	$\{\}$	$\{p_3, h_3\}$
c_7	$\{p_1^*\}$	$\{p_1\}$
c_8	$\{p_2^*\}$	$\{p_2\}$
c_9	$\{p_3^*\}$	$\{p_3\}$
c_{10}	$\{\}$	$\{h_1, d_{h1}\}$
c_{11}	$\{\}$	$\{h_2, d_{h2}\}$
c_{12}	$\{\}$	$\{h_3, d_{h3}\}$

where constraints c_1 to c_3 are related to equations (1) to (3); constraints c_4 to c_6 are related to the equation (4) for each one of the tanks; constraints c_7 to c_9 make explicit the diagnosis observational model, relating the output variable p_i and its associated sensor p_i^*; and constraints c_{10} to c_{12} make explicit the dynamic in the system: relation between the derivative of the state variables and the state variable itself.

The first step in PC computation is to look for the complete set of minimally redundant subsets of equations. The redundancy is related to the set of unknown

variables in the equations.

Definition 4 (Minimal Evaluation Chain (MEC))**.** A subset of equations that represents a potential source of discrepancy if the set of equations could be actually solved[1].

A MEC represents a strictly overdetermined[2] set of equations that can potentially be solved using local propagation (elimination method): each MEC will have n constraints and $n-1$ unknowns. MECs are computed to guarantee that there is a complete matching in its associated bipartite graph (made up of unknowns as nodes and equations as edges). This is a necessary condition to obtain a causal assignment and potentially provide a computational model.

A summary of the algorithms used to compute MECs in a system can be found in Appendix A. The set of MECs in the system in Fig. 1 is:

- $mec_1 = \{c_7, c_4, c_{10}, c_1, c_5, c_8\}$
- $mec_2 = \{c_8, c_5, c_{11}, c_2, c_4, c_6, c_7, c_9\}$
- $mec_3 = \{c_9, c_6, c_{12}, c_3, c_5, c_8\}$

We need to know the different ways an equation can be solved, because we can deal with non-linear models. These ways are usually called the set of possible causal assignments for the variables in an equation. Using this set of causal assignments, we can define the set of possible causal assignments for every unknown variable in the bipartite graph. We assume that the set of possible causal assignments is known for the system model, and we build the complete set of valid causal assignments for the set of MECs, using exhaustive search (Pulido & Alonso-González, 2004). We call each valid causal assignment Minimal Evaluation Model (MEM), because each MEM represent the precise order to solve or to evaluate the overdetermined set of equations in a MEC. Given the equations and the evaluation order provided by a MEM, they can be used to build simulation models[3].

Definition 5 (Minimal Evaluation Model (MEM))**.** A MEM is a MEC with a valid global causal assignment for every unknown in the MEC.

For the three tank system, we assume that the causality is given by the expression in equations (1) to (6), except for the observational model (in this case we allow solving constraints ec_7 to ec_9 in both directions because we need to convert some system measurements Y in MEM inputs). The set of MEMs for the three-tank system and their discrepancy nodes are shown in Table 1.

[1] A MEC is equivalent to the structure of an Analytical Redundancy Relation, ARR, or a Minimally Structurally Overdetermined set, MSO, in works by Staroswiecki and co-workers and Nyberg and co-workers, respectively.

[2] A redundant set of equations would be an Evaluation Chain. Since we are interested only on minimal conflicts, we just focus on the set of MECs that are by definition minimally overdetermined.

[3] A MEM is equivalent to the analytical expression of an ARR, an R-Conflict in the work by Cordier and the HIMALAYA group, and also is the set of formulas used to compute a conflict in GDE.

3.2.2. Fault detection and isolation using PCs

In the MEM there is a special node called *discrepancy* node (representing the only variable that is estimated by two different ways). Therefore, that node is the potential source of a discrepancy using only the values of measured variables as inputs, and the past value of state-variables.

In CBD (Reiter, 1987; Kleer & Williams, 1987) a conflict arises given a discrepancy between observed and predicted values for a variable. Under fault conditions, conflicts are observed when the model described by a MEM is evaluated with available observations and produce a discrepancy, because the model equations and the input/measured values are inconsistent (Reiter, 1987; Kleer & Williams, 1987). This notion of possible discrepancy generation leads to the definition of *Possible* Conflict:

Definition 6 (Possible Conflict)**.** The set of constraints in a MEC that give rise to at least one MEM.

Every MEC in the three-tank system has one MEM. Then, there are three PCs in the system, one for each MEC.

Each MEM is the computational model for a PC, and each equation in a MEM contains zero or more parameters that can be the source of potential faults (θ_{cc} in our model description). The set of parameters related to each PC is also shown in the fourth column in Table 1. Given a non-zero residual, we then isolate the fault parameters involved in the *pc* structural model: Θ_{pc}. This information is the basis for the integration of Consistency-based diagnosis of dynamic systems with Possible Conflicts and Lydia-NG.

4. On-line Fault diagnosis with Lydia-NG and PCs

In CBD, diagnosis must discriminate among 2^N behavioral mode assignments when just correct, $ok(\cdot)$, and incorrect modes, $\neg ok(\cdot)$, are present for N components. When B behavioral models are allowed, diagnosis must discriminate among B^N mode assignments. This is the problem faced by any model-based diagnosis proposal which attempts fault identification (DRES96, 1996), and it is also present in Lydia-NG. In this section, we present an integration proposal, where the system model is partitioned using PCs. As explained in Section 2, the output of the consistency-based diagnosis using PCs is a set of

Table 1. MEMs for the three-tank system and their discrepancy nodes.

MEM	Associated MEC	Discrepancy node	Fault Parameters
$\{c_7, c_4, c_{10}, c_1, c_5, c_8\}$	mec_1	p_1^*	k_1, A_1
$\{c_8, c_5, c_{11}, c_2, c_4, c_6, c_7, c_9\}$	mec_2	p_2^*	k_1, k_2, A_2
$\{c_9, c_6, c_{12}, c_3, c_5, c_8\}$	mec_3	p_3^*	k_2, k_3, A_3

fault candidates C defined in the lattice provided by Θ^*. Then, this set of diagnosis candidates is used as input to LYDIA-NG, thus reducing the number of health state simulations that needs to be considered by LYDIA-NG.

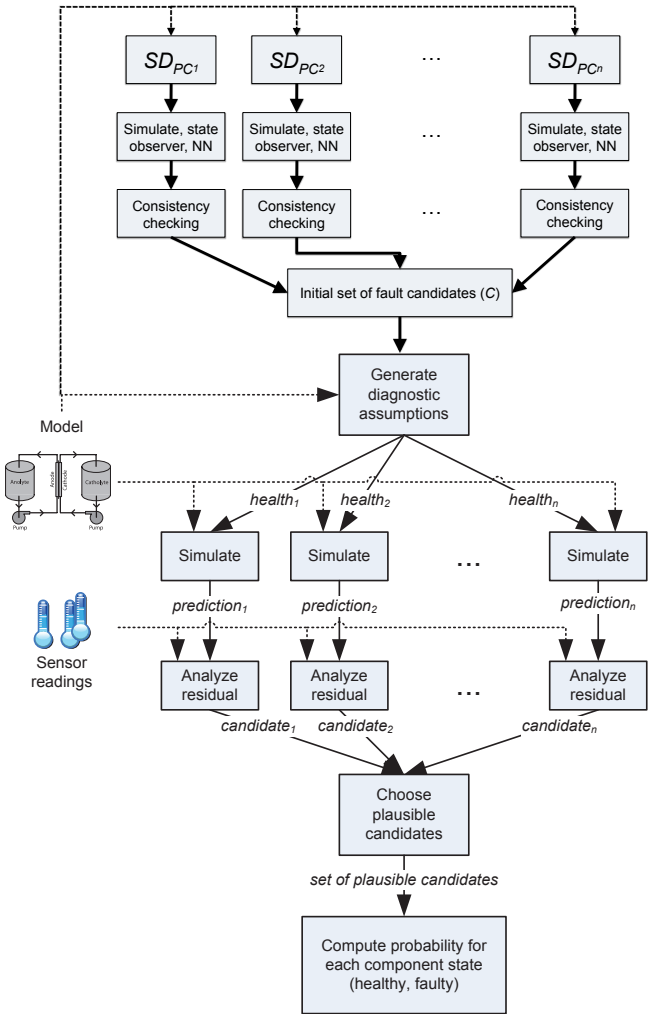

Figure 3. LYDIA-NG and PCs integration framework.

Fig. 3 shows the basic idea of our integration proposal. The simulation model for each PC uses some of the system measurements as input, and provides an estimation for exactly one variable (the potential discrepancy). Then, an executable model, SD_{pc_i} for each pc_i, is built. This executable model can be a simulation model, a state observer, or even a neural network (Bregon, Biswas, & Pulido, 2012; Pulido et al., 2012). Summarizing, the integration of Lydia-NG and CBD with PCs is possible given the set of candidates, C: each candidate C_i is a subset of Θ_{pc}. Then invoking $\Pi(C_i)$, Lydia-NG can obtain the set of health statuses, Hc related to C_i, and use them as input for its search. Given the current implementation of Lydia-NG, we can obtain the system description (system model) imposed by $Hc : \sigma(M, Hc)$, which is enough to characterize the current model and perform simulation of the Hc health status.

Algorithm 2 shows the algorithm for our integrated diagnosis framework. Y_{pc_i} denotes the set of input observations available for the executable model of a PC, SD_{pc_i}; and $\hat{Y}_{pc_i}$ represents the set of predictions obtained from SD_{pc_i}. The function OBTAINOBSERVATIONS obtains from the diagnostic scenario the observations which have to be used as input for each PC. Function ESTIMATEBEHAVIOR provides an estimation of a measured variable by using the executable model of each PC (either a simulation model, a state observer model, or a neural network).

For the detection part, to determine significant deviations from the PC residuals (PC residuals are computed by using an absolute residual function). We use the Z-test for robust fault detection using a set of sliding windows as detailed in (Daigle et al., 2010). A small window, N_2, is used to estimate the current mean of the residual signal, μ_r. The variance of the nominal residual signal is computed using a large window N_1 preceding N_2, by a buffer N_{delay}, which ensures that N_1 does not contain any samples after fault occurrence. The variance and the confidence level determined by the user are then used to dynamically compute the detection thresholds ϵ_r^- and ϵ_r^+. Other approaches can be used to determine significant deviations from the residuals, such as the Dynamic Time Warping distance, DTW (Keogh & Ratanamahatana, 2005).

Once the initial set of fault candidates has been isolated, the LYDIA-NG part of the algorithm is run (as shown in algorithm 2). The algorithm takes the set of isolated fault candidates as input, and the NEXTHEALTHASSIGNMENT function only considers the health assignments related to the fault candidates. In this version of the integrated framework, the global system model is used as the simulation model, instead of the PC submodels, thus providing a more direct way to integrate both approaches. In future versions of the

Algorithm 2 Integrated PCs and LYDIA-NG diagnosis approach.

function PCs-LYDIA-DIAGNOSIS(SCN) **returns** a diagnosis
  inputs: SCN, diagnostic scenario
  local variables: Y_{pc_i}, set of input observations
    $\hat{Y}_{pc_i}$, estimation from the PC
    Θ_{pc_i}, fault parameters involved in the PC
    h, FDI vector, health assignment
    p, real vector, prediction
    Ω, a set of diagnostic candidates
    DIAG, diagnosis, result
  repeat
    $Y_{pc_i} \leftarrow$ OBTAINOBSERVATIONS(SCN)
    $\hat{Y}_{pc_i} \leftarrow$ ESTIMATEBEHAVIOR(SD_{pc_i}, Y_{pc_i})
    $r_{pc_i} \leftarrow$ COMPUTERESIDUAL($\hat{Y}_{pc_i}, Y_{pc_i}$)
    if $r_{pc_i} < \epsilon_r^-$ or $r_{pc_i} > \epsilon_r^+$ **then**
      Θ_{pc_i} = confirm pc_i as a real conflict
      $C \leftarrow$ MHS(C, Θ_{pc_i})
    end if
  until Every pc_i is activated or time elapsed or a unique fault candidate has been isolated
  while h $\leftarrow$ NEXTHEALTHASSIGNMENT(Π, C) **do**
    p $\leftarrow$ SIMULATE(M, γ, **h**)
    $r \leftarrow$ COMPUTERESIDUAL(**p**, α)
    $\Omega \leftarrow \Omega \cup \langle$**h**$, r\rangle$
  end while
  DIAG $\leftarrow$ COMBINECANDIDATES(Ω)
  return DIAG
end function

framework, the PC submodels will also be used as the simulation model in LYDIA-NG, thus providing faster simulation results.

5. Results

In this section we show some diagnosis results for our integrated framework. We present an on-line fault diagnosis scenario for a particular fault in the three-tank system and discuss the results obtained with and without our integrated framework.

5.1. Fault Diagnosis Scenarios

This section describes the scenario of nominal conditions for the system, and the fault scenario of partial blockage of valve R_1 at time 100 s.

Nominal Scenario: Figure 4 shows a simulation experiment for the three-tank model. It is intuitive from the tanks equations that the pressure in Tank 1, p_1, is larger than the pressure in Tank 2, p_2, which is larger than the pressure in Tank 3, p_3. This is confirmed by the plot in fig. 4. For this nominal scenario, none of the three PCs found for the system is triggered. The advantage of including PCs within the LYDIA-NG framework is evident for this case. Since none of the PCs is triggered, LYDIA-NG is not run, thus avoiding the time-consuming simulations for the different health states when no actual fault has occurred in the system. **Fault Scenario:** Now

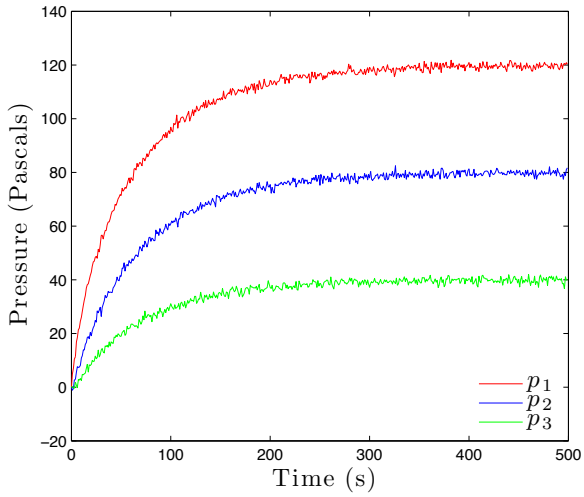

Figure 4. Simulation results for the three-tank model.

consider we introduce a 40% blockage fault in valve R_1 occurring at time 100 s. Figure 5 shows the plots of the three-tank system simulation for this fault, and Figure 6 shows the plots for the PCs for such a fault. The left column in Figure 6 shows the measured and estimated pressure for each one of the PCs, while the right column shows the residual signal computed for each PC.

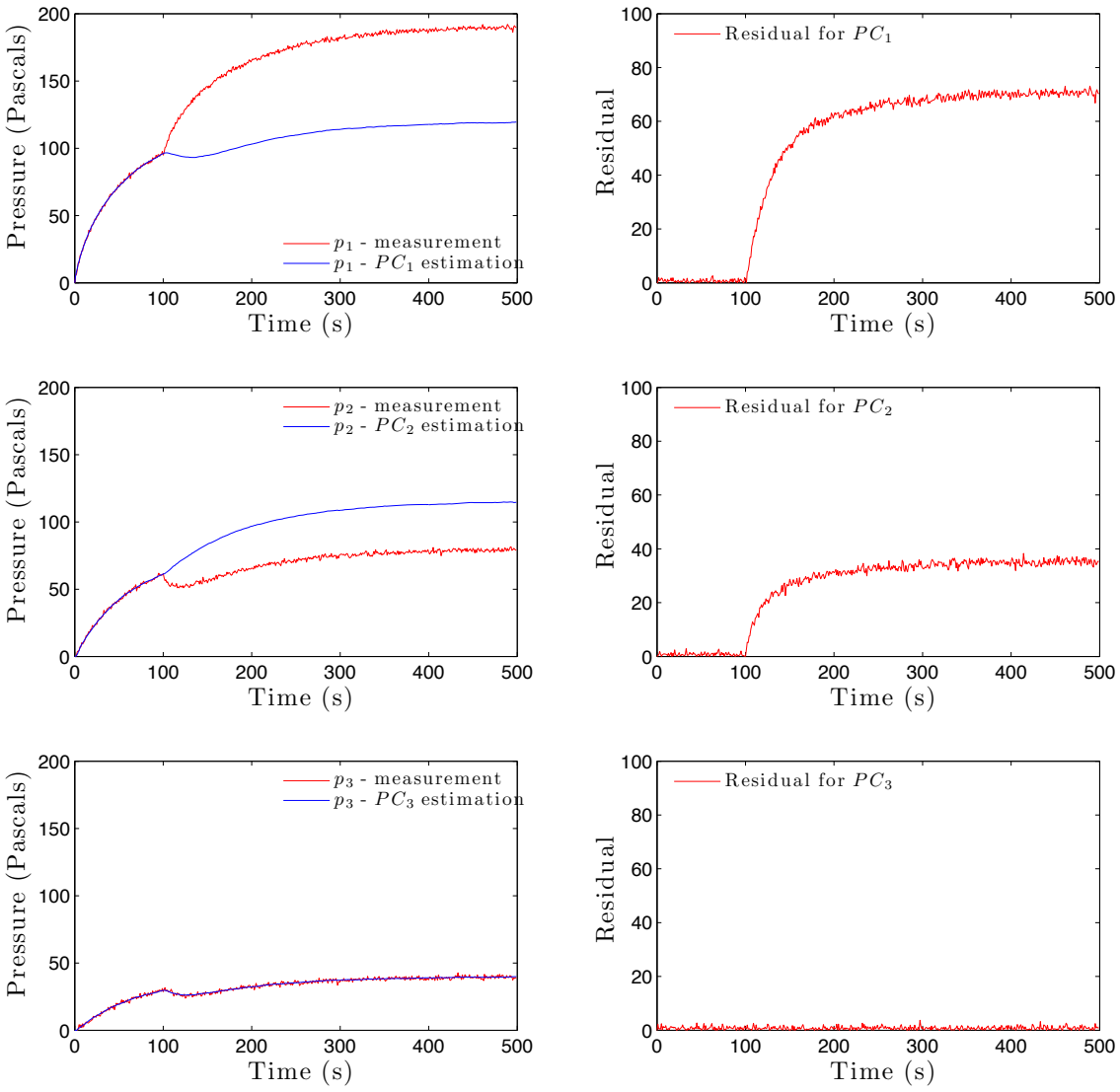

Figure 6. Simulation results for a 40% blockage fault at time 100 s in valve R_1.

5.2. On-line fault diagnosis

This section briefly describes how Lydia-NG runs with and without the use of PCs. The first phase is residual analysis, where Lydia-NG runs a set of simulations such that a residual is computed for each simulation: see Figure 2. Because Lydia-NG uses real-value health variables, the space of potential diagnostic assumptions, and the corresponding set of simulations, is enormous, and infinite in the worst case. The heuristics used for generation of diagnostic assumptions are critical to the success and efficiency of Lydia-NG.

Lydia-NG ranks the residual outputs, discarding those candidates whose residual value is larger than the residual of the "all nominal" candidate. The remaining candidates are assigned probabilities of occurrence, using a method described in (Feldman et al., 2013). The fault isolation process assigns probabilities of failure to system components, and these are reported as ranked diagnoses.

In the following we compare the results for running Lydia-NG with and without PCs. Without PCs, Lydia-NG uses the global system model described earlier; with PCs (i.e., using Algorithm 2), the generation of diagnostic assumptions is governed by the PC-based algorithm.

For the diagnosis scenario with a 40% blockage fault in valve R_1 occurring at time 100 s, our results are as follows.

Non-PC-based Approach: Lydia-NG computes residuals based on the difference between the pressures

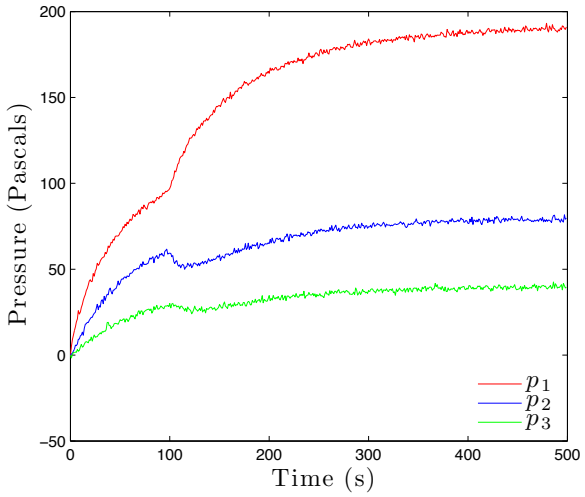

Figure 5. Simulation results for a 40% blockage fault at time 100 s in valve R_1.

shown in Figures 4 and 5. The non-zero residual at time 104 s creates a set of simulations in which LYDIA-NG analyzes several valve %-blockage cases for R_1, R_2 and R_3. LYDIA-NG estimates the valve positions by "guessing" the *true* valve positions and computes the health probability by subtracting the commanded valve position from the estimated one. LYDIA-NG is able to isolate the most-likely fault as (R_1, 40%).

PC-based Approach: When computing diagnoses for this fault, at time 104 s, an increase in the residual of PC_2 is detected, and consequently k_1, k_2, and A_2 are selected as the initial set of fault candidates. At the next time step, at time 105 s, PC_1 is triggered, thus selecting k_1 and A_1 as possible fault candidates. A minimal hitting set algorithm is run, determining that the only single fault candidate in the system is k_1. At this point, the fault identification for k_1 is triggered by using LYDIA-NG.

Running this diagnosis scenario with a (trivial) input of R_1 (as derived from the candidate k_1), as opposed to R_1, R_2 and R_3, results in an 80× speedup of LYDIA-NG as compared to the non-PC approach. This is a result of reducing the diagnosis assumption space.

6. Related work

LYDIA-NG belongs to a class of MBD methods that use continuous-valued models and sensor data, and use entropy based methods for test selection to disambiguate diagnoses. It is a generalization of LYDIA, which used discrete-value models.

In terms of diagnostics solvers, LYDIA-NG is related to the HyDE (Hybrid Diagnosis Engine) solver (Narasimhan & Brownston, 2007). The HyDE-S variant accepts as input interval-valued hybrid models and continuous-valued sensor data. Another solver, FACT (Daigle et al., 2010), can also use continuous-valued models and sensor data, but requires that the model be represented as a hybrid bond graph. Given an anomaly, FACT first uses an observer-based approach (adopted from the FDI community) with statistical techniques for robust fault detection. Fault isolation is performed using qualitative inference, i.e., by matching qualitative deviations caused by fault transients to those predicted by the model.

Recent works have demonstrated the similarities between model-based diagnosis approaches from the DX and the FDI communities (Cordier et al., 2004). In such framework, it has been demonstrated the equivalence of several structural model decomposition techniques such as PCs, minimal ARRs and Minimally Structurally Overdetermined sets (Armengol et al., 2009). As a consequence, the proposal in this work can be easily extended to other structural methods.

Using CBD we need to generate the set of candidates C and wait for every *pc* to be confirmed. An FDI approach would use exoneration using the structural information in the set of PCs. In CBD we wait for additional observations in order to reject modes that are not consistent with available information. Combining our results with LYDIA-NG provides an additional boost for candidate discrimination by including fault models through health statuses.

The approach can be further refined using qualitative information (for instance residual qualitative signatures), or add a quantitative parameter estimation (Bregon, Biswas, & Pulido, 2012). Additionally, different methodologies can be coupled to estimate correct and/or faulty behavior (Alonso-González et al., 2011).

This work is clearly a first step to combine both techniques for hybrid non-linear systems. PCs has been extended to work with hybrid systems (Bregon, Alonso, et al., 2012), and can greatly benefit from LYDIA-NG state estimation capabilities. The approach then would be similar to coupling different techniques for continuous/discrete state estimation as in (Hofbaur & Williams, 2004; Bayoudh, Travé-Massuyès, & Olivé, 2008).

7. Conclusions

This work has presented an integrated framework for on line fault detection, isolation and identification of dynamic systems.

Two different approaches have been integrated: The LYDIA-NG suite of diagnosis algorithms and the Pos-

sible Conflicts framework for on-line consistency based diagnosis. Lydia-NG is a simulation based diagnosis system that filters out diagnosis candidates discarding those of them that generates residuals larger than the *all-nominal* assumption, i.e., fault free and nominal system configuration. Although the system incorporates important facilities, such as diagnostic test generation based on entropy measure, its main drawback is the lack of focus for the initial set of candidates, which may be large, and the cost of simulating the complete system for every considered candidate. On the contrary, the set of Possible Conflicts identifies minimal computational subsystems that decompose the complete system and that can be simulated independently. PCs are based on Reiter's theory of diagnosis from first principles and are able to generate fault isolation candidates from model of correct behavior without hypothesizing an initial set of candidates. Hence, using consistency-based diagnosis with PCs candidate generation is rather efficient, although additional techniques are required to further refine fault candidates for fault isolation and identification. They also lack some of the facilities incorporated in Lydia-NG like generation of diagnostic tests.

In this paper we have combined both approaches, complementing each other, looking to preserve the best of each approximation. This integration can be tackled in different ways. We have opted for a simple integration approach that still is able to improve any of them. PCs are used to generate the initial set of isolation candidates *a la Reiter*. These candidates are later refined by Lydia-NG, which simulates the complete system in the modes –potential health statuses– defined by the candidates. In this way, we exploit PCs ability to generate isolation candidates with Lydia-NG ability to reject fault candidates that do not comply with current observations and diagnosis assumptions.

Our three tank system running example shows the potential of this approach. First, when the system is fault free, no PC becomes a real conflict and no candidate is generated. This avoids running Lydia-NG for fault detection, which is performed by the PCs approach, thus potentially providing a significant saving on computing time, depending on the size of the complete system and on the number and overlapping degree of the PCs. Second, when a fault is detected, PCs may generate a low number of fault candidates, depending on the number of PCs and its overlapping degree but also on the real faulty parameter, thus providing an automatic focus for Lydia-NG fault candidate search.

Acknowledgment

A. Bregon, B. Pulido, and C. Alonso's funding for this work was provided by the Spanish MCI TIN2009-11326 grant.

References

Alonso-González, C., Moya, N., & Biswas, G. (2011). Dynamic Bayesian network factors from possible conflicts for continuous system diagnosis. In *Proc. of the 14th Int. Conf. on Advances in AI* (pp. 223–232). Berlin: Springer-Verlag.

Armengol, J., Bregon, A., Escobet, T., Gelso, E., Krysander, M., Nyberg, M., et al. (2009). Minimal Structurally Overdetermined sets for residual generation: A comparison of alternative approaches. In *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS09* (p. 1480-1485). Barcelona, Spain.

Bayoudh, M., Travé-Massuyès, L., & Olivé, X. (2008). Coupling Continuous and Discrete Event System Techniques for Hybrid System Diagnosability Analysis. In *Proc. of the 18th European Conf. on Artificial Intelligence, ECAI08* (pp. 219–223). Amsterdam, The Netherlands.

Bregon, A., Alonso, C., Biswas, G., Pulido, B., & Moya, N. (2012). Fault Diagnosis in Hybrid Systems using Possible Conficts. In *Proc. of Safeprocess'12.* Mexico City, Mexico.

Bregon, A., Biswas, G., & Pulido, B. (2012). A Decomposition Method for Nonlinear Parameter Estimation in TRANSCEND. *IEEE Trans. Syst. Man. Cy. Part A*, *42*(3), 751-763.

Cordier, M., Dague, P., Lévy, F., Montmain, J., Staroswiecki, M., & Travé-Massuyès, L. (2004). Conflicts versus Analytical Redundancy Relations: a comparative analysis of the Model-based Diagnosis approach from the Artificial Intelligence and Automatic Control perspectives. *IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics*, *34*(5), 2163-2177.

Daigle, M., Bregon, A., & Roychoudhury, I. (2012, aug). Qualitative Event-based Diagnosis with Possible Conflicts Applied to Spacecraft Power Distribution Systems. In *Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes* (p. 265-270).

Daigle, M., Roychoudhury, I., Biswas, G., Koutsoukos, X., Patterson-Hine, A., , et al. (2010, September). A Comprehensive Diagnosis Methodology for Complex Hybrid Systems: A Case Study on Spacecraft Power Distribution Systems. *IEEE Transac-*

tions of Systems, Man, and Cybernetics, Part A, *4*(5), 917–931.

Daigle, M., Roychoudhury, I., Biswas, G., Koutsoukos, X., Patterson-Hine, A., & Poll, S. (2010). A comprehensive diagnosis methodology for complex hybrid systems: A case study on spacecraft power distribution systems. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, *40*(5), 917–931.

Dressler, O. (1996). On-line diagnosis and monitoring of dynamic systems based on qualitative models and dependency-recording diagnosis engines. In *Proceedings of the Twelfth European Conference on Artificial Intelligence, ECAI-96* (p. 461-465).

Feldman, A., Castro, H. V. de, Gemund, A. van, & Provan, G. (2013). Model-Based Diagnostic decision-support system for satellites. In *Aerospace Conference, 2013 IEEE* (pp. 1–14).

Feldman, A., Provan, G., & Gemund, A. van. (2010). Approximate Model-Based Diagnosis Using Greedy Stochastic Search. *Journal of Artificial Intelligence Research*, *38*, 371–413.

Gertler, J. (1998). *Fault Detection and Diagnosis in Engineering Systems*. Marcel Dekker, Inc.

Hofbaur, M., & Williams, B. (2004, Oct.). Hybrid estimation of complex systems. *IEEE T. Syst. Man. Cy. Part B*, *34*(5), 2178 -2191.

Isermann, R. (2006). *Fault-Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance*. Springer.

Keogh, E., & Ratanamahatana, C. (2005). Exact indexing of dynamic time warping. *Knowledge and Information Systems*, *7*(3), 358-386.

Kleer, J. de, & Williams, B. C. (1987). Diagnosing multiple faults. *Artificial Intelligente*, *32*, 97-130.

Narasimhan, S., & Brownston, L. (2007). HyDE– A General Framework for Stochastic and Hybrid Model-based Diagnosis. In *Proc. 18th International Workshop on Principles of Diagnosis (DXâĂŹ07), Nashville, USA* (pp. 162–169).

Pouliezos, A., & Stavrakakis, G. (1994). *Real Time Fault Monitoring of Industrial Processes*. Kluwer Academic Publishers.

Pulido, B., & Alonso, C. (2001). Dealing with cyclical configurations in MORDRED. In *IX Conferencia Nacional de la Asociacion Española de Inteligencia Artificial, (CAEPIA-01)* (p. 983-992). Gijon, Spain.

Pulido, B., Alonso, C., & Acebes, F. (2001). Lessons learned from diagnosing dynamic systems using possible conflicts and quantitative models. In *Engineering of Intelligent Systems. XIV Conf. IEA/AIE-2001* (Vol. 2070, p. 135-144). Budapest, Hungary.

Pulido, B., & Alonso-González, C. (2004, Octubre). Possible Conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. on Systems, Man, and Cybernetics. Part B: Cybernetics*, *34*(5), 2192-2206.

Pulido, B., Bregon, A., & Alonso-González, C. (2010). Analyzing the influence of differential constraints in Possible Conflicts and ARR Computation. In P. Meseguer, L. Mandow, & R. Gasca (Eds.), *Current Topics in Artificial Intelligence* (Vol. 5988, p. 11-21). Springer Berlin.

Pulido, B., Zamarreño, J., Merino, A., & Bregon, A. (2012, July). Using structural decomposition methods to design gray-box models for fault diagnosis of complex systems: a beet sugar factory case study. In A. Bregon & A. Saxena (Eds.), *Procs. of the First European Conference of the Prognostics and Health Management Society* (p. 225-238). Dresden, Germany.

Reiter, R. (1987). A Theory of Diagnosis from First Principles. *Artificial Intelligence*, *32*, 57-95.

A. Possible Conflicts Computation algorithms

Each PC is a set of overdetermined set of equations, derived from the model, that can be solved. The algorithm to compute PCs finds recursively every combination of model equations that are strictly overdetermined using depth first search (i.e. we find any possible overdetermined system of equations for every equation). The algorithm proceeds finding one new equation potentially solving one of the remaining unknowns.

Algorithm 3 Step 1: Find every strictly overdetermined sub-system, PC, in M_H for each equation c in Σ_H

```
1: function FIND_EVERY_PC(M_H) returns SPCs
2:     for all equation c in Σ_H do
3:         find_pc (M_H \ {c}, {c}, c_unknowns, SPCs)
4:     end for
5: end function
```

A similar algorithm should analyze the set of equations and unknown variables in each *pc*, and find out if its associated set of equations have a globally consistent causal assignment, i.e. we can obtain a solution for the model and generate a simulation model, for instance. This is the basic algorithm that pays no attention to potential cyclical configurations – algebraic loops or loops containing differential equations–. These checks can be done later, and depend on the kind of equations in the model, the simulation language used, and the presence of appropriate equation solvers for loops (Pulido & Alonso, 2001; Pulido, Bregon, & Alonso-González, 2010).

Algorithm 4 Step 1.2: Find possible conflict, pc, from available equations in the model, RM: find a new equation for each remaining unknown variable, $unknowns$, in the pc

```
1: function FIND_PC(RM, pc, unknowns, SPCs)
2:     if unknowns == {} then
3:         if pc is minimal w.r.t. SPCs then
4:             remove every superset of pc in SPCs
5:             insert pc in SPCs
6:         end if
7:     else
8:         for all equation c' ∈ RM do
9:             for all y ∈ (c'_unknowns ∩ unknowns) do
10:                find_pc(RM \ {c'}, pc ∪ {c'},
   unknowns ∪ {c'_unknowns} \ {y}, SPCs)
11:            end for
12:        end for
13:    end if
14: end function
```

Biographies

Anibal Bregon received his B.Sc., M.Sc., and Ph.D. degrees in Computer Science from the University of Valladolid, Spain, in 2005, 2007, and 2010, respectively. From September 2005 to June 2010, he was Graduate Research Assistant with the Intelligent Systems Group at the University of Valladolid, Spain. He has been visiting researcher at the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN, USA; the Dept. of Electrical Engineering, Linkoping University, Linkoping, Sweden; and the Diagnostics and Prognostics Group, NASA Ames Research Center, Mountain View, CA, USA. Since September 2010, he has been Assistant Professor and Research Scientist at the Department of Computer Science from the University of Valladolid. Dr. Bregon is a member of the Prognostics and Health Management Society and the IEEE. His current research interests include model-based reasoning for diagnosis, prognostics, health-management, and distributed diagnosis and prognostics of complex physical systems.

Alexander Feldman is a postdoc at University College Cork and a visiting researcher at Ecole Polytechnique Federale de Lausanne (EPFL), Delft University of Technology and PARC (former Xerox PARC). He has obtained his Ph.D. (cum laude) in computer science/artificial intelligence and M.Sc. (cum laude) in parallel and distributed systems from the Delft University of Technology. He has published in leading conference proceedings and international journals covering topics in artificial intelligence, model-based diagnosis, and engineering. In cooperation with NASA Ames Research Center and PARC, Alexander Feldman has co-organized the International Diagnostic Competitions (DXC). Alexander Feldman's interest cover wide spectrum, including topics such as model-based diagnosis, automated problem solving, software and hardware design, design of diagnostic space applications, digital signal processing, and localization.

Belarmino Pulido received his degree, MsC degree, and PhD degree in Computer Science from the University of Valladolid, Valladolid, Spain, in 1992, 1995, and 2001 respectively. In 1994 he joined the Departamento de Informática in the University of Valladolid, where he is Associate Professor since 2002. His main research interests are Model-based reasoning and Knowledge-based reasoning, and their application to Supervision and Diagnosis. He has worked in different national and European funded projects related to Supervision and Diagnosis of complex industrial systems. He is member of different professional associations (IEEE since 2000, ACM since 2003, CAEPIA -Spanish section of ECCAI- since 1997). Moreover, he is the coordinator of the Spanish Network on Supervision and Diagnosis of Complex Systems since 2005.

Gregory Provan is a Professor at the Computer Science Department at University College Cork (UCC), in Cork, Ireland. He received his BSE from Princeton University, USA, his MSc from Stanford University, USA, and his DPhil from University of Oxford, UK. He is currently Director of the Complex Systems Lab at UCC. Prior to joining UCC he was on faculty at the University of Pennsylvania, USA, and spent over 10 years in industrial research. His interests are in complex systems (design, analysis and control), diagnostics, algorithm design, and bioinformatics. His current research focuses on modelling and analysis of sustainable energy systems, model-based diagnostics of a variety of complex systems, and design of methods for efficiently embedding code in such systems. Prof. Provan is the coauthor of over 100 refereed articles, and has been Principal Investigator in a variety of grants and research contracts.

Carlos Alonso-González got his degree in Sciences in 1985, and his PhD in Physics in 1990, from the University of Valladolid, Valladolid, Spain. Currently he is Associate Professor with the Department of Computer Science from the University of Valladolid. He has being working in different national funded projects related to Supervision and Diagnosis of continuous industrial environments. His main research interests are Knowledge-based systems for Supervision and Diagnosis of dynamic systems, model-based diagnosis, Knowledge Engineering and Machine Learning.