Efficient Generation of Minimal Dynamic Bayesian Networks for Hybrid Systems Fault Diagnosis using Hybrid Possible Conflicts

Belarmino Pulido, Noemi Moya, Carlos J. Alonso-González, Anibal Bregon

Departamento de Informática, Universidad de Valladolid, Valladolid, 47011, Spain
{belar,noemi,calonso,anibal}@infor.uva.es

Abstract

Hybrid systems diagnosis requires different sets of equations for each operation mode in order to estimate the continuous system behaviour. In this work we rely upon Hybrid Possible Conflicts (HPCs), which are an extension of Possible Conflicts (PCs) for hybrid systems, that introduce the information about potential system modes as control specifications that activate/deactivate different sets of equations. We also introduce the concept of Hybrid Minimal Evaluation Models (H-MEMs) to represent the set of globally consistent causal assignments in an HPC for any potential mode.

H-MEMs can be explored for a specific operation mode, and its computational model automatically generated. In this work, the selected computational models are minimal Dynamic Bayesian Networks (DBNs). Since DBNs can be directly generated from PCs, and can be used for fault detection and isolation, we propose to efficiently generate Minimal DBNs models on-line using the H-MEM structure. By introducing fault parameters in the DBN model, we can also perform fault identification, providing an unifying framework for fault diagnosis, under single fault assumption. We test the approach in a simulation four-tank system.

1. Introduction

Dynamic systems with hybrid behaviour are present in almost every field in our society. Fault diagnosis for these systems is of capital importance to prevent malfunctions or breakdowns, and to increase the security and the quality of the final products. However, it is difficult to provide accurate and timely online fault diagnosis. because their behaviour is made up of continuous behaviour commanded by discrete events.

For the last 15 years two research communities: the Control Theory, known as the FDI[1] approach, and Artificial Intelligence, known as the DX approach, have worked on hybrid systems modelling and diagnosis (Cocquempot, El Mezyani, & Staroswiecki, 2004; Hofbaur & Williams, 2004; Narasimhan & Biswas, 2007; Narasimhan & Brownston, 2007). Current research activities are focused on two main issues: full or approximate estimation of the set of possible states, and tracking of nominal and faulty system behaviour (Rienmüller, Bayoudh, Hofbaur, & Travé-Massuyès, 2009). To tackle the first issue, different kinds of automata have been used to model the complete set of modes, and transitions between them, which introduces the need to enumerate all the set of modes (states) and transitions, and to track the entire set of consistent modes. Both issues do not scale well for complex systems. To avoid the pre-enumeration of modes, we have followed the proposal by Narasimhan et al. (Narasimhan & Biswas, 2007), which uses Hybrid Bond-Graphs (HBGs) to model the whole system, and depending on the value of the switching junctions, used to model the hybrid behaviour, it is able to generate on-line new models for tracking the new system mode. That work has been recently extended to efficiently generate simulation models using model block diagrams based on HBGs properties (Roychoudhury, Daigle, Biswas, & Koutsoukos, 2011), and to efficiently generate state observers (Podgursky, B., & Koutsoukos, 2010).

Bregon et al. (Bregon, Alonso, Biswas, Pulido, & Moya, 2012) introduced Hybrid Possible Conflicts (HPCs) as an extension of Possible Conflicts (Pulido & Alonso-González, 2004) using HBGs (Narasimhan & Biswas, 2007), and Block Diagrams (Roychoudhury et al., 2011). HPCs can track hybrid systems behaviour, efficiently changing on-line for each mode the PC simulation model, and performing diagnosis without pre-enumerating the set of modes in the system. But HPCs do not provided a unified diagnosis framework using one technique.

In this work we propose to use minimal Dynamic Bayesian Networks, DBNs, derived from HPCs as a unique modelling framework for hybrid systems fault detection, isolation, and identification, together with new algorithms to automatically generate on-line the DBN computational model, and then using that model to track the system. To automatically generate

Belarmino Pulido et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] Acronym for Fault Detection and Isolation.

on-line the computational model we introduce the concept of H-MEM, that provides the potential set of equations that can appear in any HPC. The approach has been tested on a four-tank laboratory plant with satisfactory results.

The rest of the paper is organized as follows. Section 2 presents the case study used along the paper and introduces HBG modelling technique. Section 3 summarizes the HPCs background. Section 4 describes our new diagnosis framework with DBNs computed from HBGs. Section 5 presents some results obtained applying our proposal to the case study. Section 6 describes some related work and Section 7 draws some conclusions and future work.

2. Case Study

The hybrid four-tank system in Figure 1 will be used along the paper. It has one input flow to tanks 1 and 3, that can be sent to both tanks or only to one of them. Tanks 2 and 4 are connected to tanks 1 and 3, respectively, through a pipe placed at a distance *h* above their bases. Sources of discrete behaviour are: commanded valves at the input of tanks 1 and 3, and the pipes connecting tanks 1 and 2, and between tanks 3 and 4, at a given height $h_i > 0$. There are four measurements in the system: pressure sensors at the bottom of every tank.

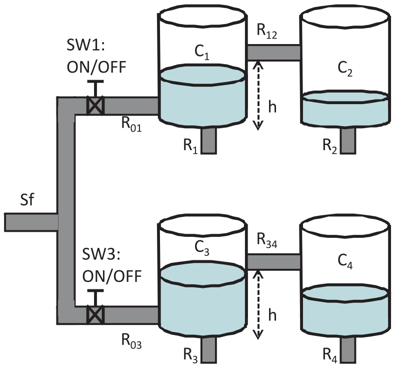

Figure 1. Schematics of the four-tank system

We used HBGs, an extension of Bond-Graphs (BGs) (Karnopp, Margolis, & Rosenberg, 2006), to model our system. BGs is a domain-independent energy-based topological modelling language for physical systems. BGs rely upon primitive elements: storage (capacitances, C, and inductances, I), dissipative (resistors, R) and energy transformation (transformers, TF, and gyrators, GY) elements. There are also effort and flow sources (Se and Sf) to define interactions between the system and the environment. Elements in a BG are connected by 0 or 1 junctions (representing ideal parallel or series connections between components). Each bond has two associated variables (effort and flow). The rate of energy is defined as $effort \times flow$ for each bond. The SCAP algorithm (Karnopp et al., 2006) is used to assign causality automatically to the BG.

Related to the primitive elements, sources and junctions there is a set of well-established equations relating flow and effort variables. The exact expression of each equation depends on the assigned causality. For instance, for a resistance, R, element with effort and flow variables, e_1, f_1, the corresponding equation would be $e_1 = R \times f_1$ or $f_1 = R \times e_1$. Energy storage elements, such as a capacitor C, provide the following equation $e_2 = \frac{1}{C} \int f_2 dt$, for variables effort and flow, e_2, f_2, in integral causality. Finally, 0-junctions and 1-junctions model ideal common effort or common flow connections, where efforts (equivalently flows) are all equal ($e_1 = e_2 = e_3$), while sum of flows (correspondingly efforts) must equal zero ($f_1 - f_2 + f_3 = 0$). Additionally, there are effort and flow detectors, De and D_f respectively, that provide measurements of system magnitudes.

To model hybrid systems we need to use some kind of connections which allow changes in their state. HBGs (Narasimhan & Biswas, 2007) extend BGs by including those connections. If a switching junction is set to ON, it behaves as a regular junction. When it changes to OFF, all bonds incident on the junction are deactivated forcing 0 flow (or effort) for 1 (or 0) junctions. A finite state machine control specification (CSPEC) implements those junctions. Transitions between the CSPEC states can be triggered by endogenous or exogenous variables, called guards. CSPECs capture controlled and autonomous changes as described in (Roychoudhury et al., 2011).

Figure 2 shows the HBG model of the four tanks system, where there are four measurements for diagnosis: the level of the four tanks by means of pressure sensors, p_i related to capacitances C_i, that are represented as effort detectors $De : p_i,\ i = 1..4$ in the BG.

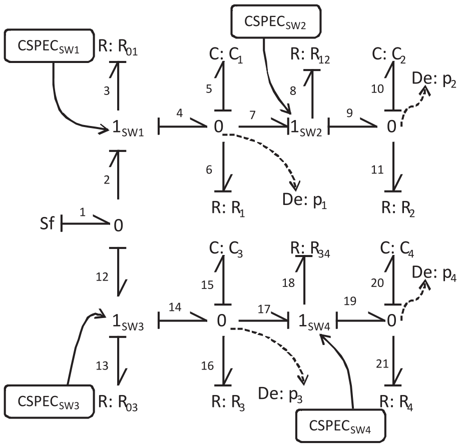

Figure 2. Bond graph model of the plant.

Regarding the hybrid behaviour, it has four switching junctions: SW_1, SW_2, SW_3 and SW_4. SW_1 and SW_3 are controlled *ON/OFF* transitions, while SW_2 and SW_4 are autonomous transitions related to water level of tanks 1 and 3 surpassing the height h and overflowing to tanks 2, and 4 respectively. Both kind of transitions are represented using a finite state machine. Figure 3 shows: a) the automaton associated with switching SW_1 and b) the automaton representing the autonomous transition in SW_2 (since the system is symmetric, automata for SW_3 and SW_4 will be equivalent).

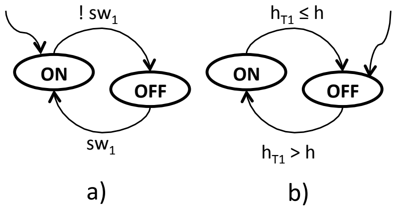

Figure 3. a) Automaton associated with the *ON/OFF* switching junction SW_1; b) Automaton representing the autonomous transition in SW_2.

3. Hybrid PCs and minimal DBNs background

3.1. Hybrid Possible Conflicts (HPCs)

The Possible Conflict, PC, approach is a dependency-compilation technique from the DX community (Pulido & Alonso-González, 2004), that have been successfully used for system model decomposition in consistency-based diagnosis of continuous systems. PCs define minimal structurally overdetermined subsets of equations with sufficient analytical redundancy to generate fault hypotheses from observed measurement deviations. In the original approach, only structural and causal information from the system model is used. PCs are computed using a hypergraph abstracting the structural model of the system. Recently, we have proposed an extension that allows to compute PCs directly from bond graph models (Bregon, Biswas, & Pulido, 2012).

The PC approach has been recently extended to cope with hybrid system dynamics, using Hybrid Bond-Graphs (Roychoudhury et al., 2011; Narasimhan & Biswas, 2007) as the modelling approach. The extension is called Hybrid Possible Conflicts (Bregon, Alonso, et al., 2012). Main advantage of HBG modelling technique is that pre-enumeration of the modes in the system is not necessary. However, its main concern when applied to fault diagnosis of hybrid systems (Narasimhan & Biswas, 2007) is related to the task of causality reassignment for the entire bond graph model, because during this causality reassignment process, the diagnosis system needs to stop tracking the behaviour of the system, making it sensitive to miss faults that occur during (or immediately after) such reassignment process. However, recent proposals for fast causality reassignment in HBGs can be used to speed up this process for efficient on-line simulation (Roychoudhury et al., 2011). Typically, changes in causality do not propagate within the model, or only a small part of the model causality needs to be reassigned. Moreover, when causality needs to be reassigned, changes will be typically local to the hybrid junction. HPCs incorporate the proposal by Roychoudhury et al. (Roychoudhury et al., 2011) to generate new causality assignments for HPCs once a mode change is observed. Currently, our main assumption is that we are able to track the current system mode.

For the case study we have found four HPCs. Each one of them estimates one of the measured variables (p_1, p_2, p_3, or p_4). Figure 4 shows the HBG fragments of these four HPCs.

In this example, we first computed HPCs assuming that all switching junctions are set to *ON*, but when any of these junctions is switched to *OFF*, causality in the system needs to be reassigned. Even though causality may change, the HPC generation process does not need to be restarted again (Bregon, Alonso, et al., 2012).

There are two basic possibilities for the existing HPCs depending on whether the change in the switching junction induces a change in causality or not. First, the change in the switching junction induces a change in causality which affects the HPC. A new causality will be assigned to the HPC and it will be updated. If there is not a valid causal assignment, the HPC will disappear. Second, as a result of the change in the switching junction there is no change in causality. In this case, either a PC can remain the same, or a part of the PC can be affected by the switch and disappear or the whole PC can disappear (the discrepancy node disappears).

HBGs main advantage is that the complete set of modes do not need to be known or enumerated in advance. However, many times there is no such HBG model available. In this work we propose to compute HPCs for a generic set of ODEs, given that some of them are only valid under given system configurations, thus it is needed to extend original algorithms to compute PCS by introducing the information about discrete dynamics. To show our approach we have used as system model the set of equations that can be derived from an HBG model, as explained above in order to ease the comparison of this approach with the previous one. But in general, any set of ODEs can be our system model because for computing our HPC models we work mainly at the structural and causal level.

Figure 5 represent one possible MEM for PC_2 in Figure 4. The MEM provides a computational model that can be implemented as a simulation or state-observer model (Pulido, Bregon, & Alonso-González, 2010). In this work we propose to implement our PCs as a set of Dynamic Bayesian Networks, providing a framework capable to perform not only fault detection and isolation but fault identification using the same computational model.

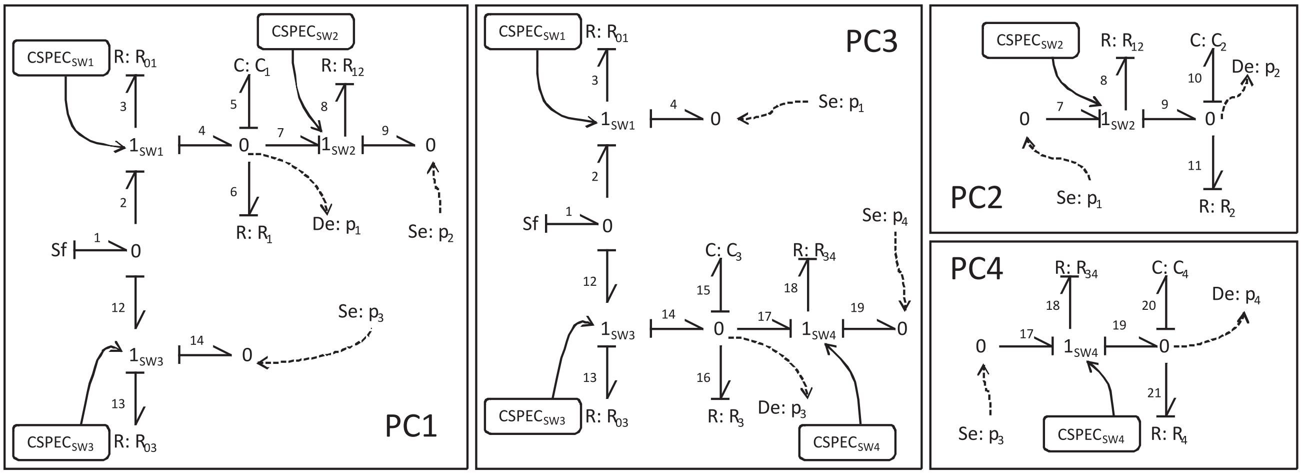

Figure 4. Hybrid Bond graphs of the four HPCs found for the four-tank system.

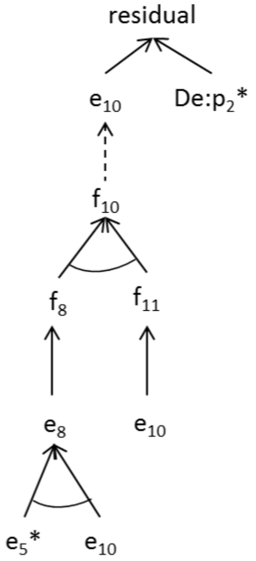

Figure 5. MEM for PC_2 subsystem in Figure 4. The effort and flow variables in the graph correspond to pressures and flows in PC_2 in Figure 4.

3.2. Minimal Dynamic Bayesian Networks(DBNs)

Dynamic Bayesian Networks (DBNs) are a probabilistic temporal model representation of a dynamic system. Basically, a DBN is a two slices Bayes Network (BN). Assuming that the system is time invariant and a First Order Markov process, two static and identical BN connected by inter slice arcs are enough to model the system (Murphy, 2002). Inter slice arcs model system dynamics. Intra slice arcs model instantaneous (algebraic) relations.

The system variables (X, Z, U, Y) represented in a DBN are the inputs (U), the state variables (X), the observed or measured variables (Y) and, in some cases, other hidden variables (Z) . Once we have the nodes, we need to define the arcs and the parameters in the model, the state transition model (graphically represented by the inter slice arcs) and the observational model (represented by intra slice arcs).

Figure 6 represent the DBN for MEM2 in Figure 5. Blue arrows represent the inter slice arcs modelling system dynamics. Orange arrows represent the intra slice or instantaneous relations among system variables. Alonso-Gonzalez et al (Alonso-Gonzalez, Moya, & Biswas, 2011) provided the method to automatically transform a MEM from a PC to a DBN model. Following the method we obtain the DBN model in Figure 6.

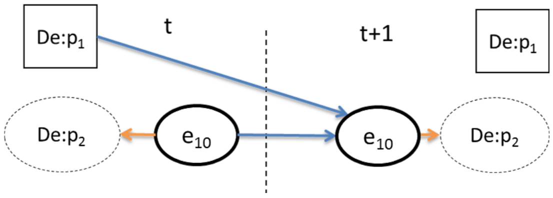

Figure 6. DBN for the PC_2 subsystem in Figure 5. Input node is measured pressure $e5$ = *De:p1*, state variable is the pressure in tank 2, $e10$, and the output node is the measured pressure in tank 2 *De:p2*.

Exact inference in DBNs is not computationally tractable. Hence, Monte Carlo simulation methods are used for approximate inference, particularly Particle Filter algorithm (Koller & Lerner, 2001). The unknown continuous stochastic distribution of the state is approximated by a discrete distribution obtained by weighted samples. After propagation of the state, the weights are updated with current observations. In this work, we assume a Gaussian distribution.

DBNs can be used along all the stages in the diagnosis process. They provide a unified framework for fault diagnosis. DBNs can be generated from a PC derived from a BG model (Alonso-Gonzalez et al., 2011) and have been successfully applied for fault diagnosis of continuous systems (Roychoudhury, Biswas, & Koutsoukos, 2008; Alonso-Gonzalez et al., 2011) .

In this work, we propose to integrate DBNs to monitor the continuous behaviour of the system, and to use HPCs to generate different DBNs for each new mode. We propose to build a different DBN for each mode, instead of using a hybrid DBN able to track the complete set of modes related with the HPC.

4. Efficiently computing Hybrid PCs

In (Pulido & Alonso-González, 2004) PCs were computed for a unique mode. The computation was made in two steps: first, we obtained the set of minimally overdetermined sets of equations, which are called Minimal Evaluation Chains, MECs –equivalent to minimal ARRs (Analytical Redundancy Relations) or MSO (Minimal Structurally Overdetermined) sets of equations (Armengol et al., 2009)–. Second, introducing causal information in the model[2], for each MEC we obtained the complete set of globally consistent causal assignments, each one called Minimal Evaluation Model or MEM. Each MEM provides the computational model required to build a simulation or a state-observer model (Pulido et al., 2010).

In previous works we have demonstrated that the structural and causal models can be automatically obtained from Bond-Graph models, deriving a Temporal Causal Graph, TCG, that represent a consistent causal assignment for the system in one mode. And we can compute the set of Possible Conflicts from the TCG (Bregon, Pulido, Biswas, & Koutsoukos, 2009). But in this section we propose to extend the approach to any causal model (we always can start from a system model made up of a set of ODEs, and then to abstract the structural and causal information in them to generate a causal model where only the presence of measured and unknown variables in an equation is relevant). In (Bregon, Alonso, et al., 2012) we proposed how to obtain HPCs from HBGs, using HSCAP (Roychoudhury et al., 2011) to avoid computing a new causal assignment in the HBG whenever there is a mode change.

In this work we propose to compute the HPCs directly from a set of labelled equations, obtained as an abstraction of the set of ODEs which is our model. In order to efficiently generate computational models as minimal DBN factors, we need to extend the algorithms computing HPCs in two ways. First, we need to compute the set of HPCs, i.e. PCs with labelled equations related to discrete dynamics. Second, we need to automatically and efficiently build the DBN behavioural models from the computational model provided by a MEM.

4.1. Inclusion of constraints to represent discrete dynamics

To fulfill the first requirement we first introduce information about discrete dynamics in the model, and later on we modify the original algorithms to compute PCs. In this work we assume that each equation in the system model is valid in a set of configurations, and these configurations can be characterized as constraints: one constraint is a well-formed formula, WFF, in propositional logic. The propositions in the WFF will represent the control specifications related to the switching junction automata (as shown in Figure 3) because they will have only boolean values related to the ON/OFF state of the switching junction.

These modifications can be summarized as follows:

- first, we add the information about constraints in the equations as WFF. Each switching junction introduces an atomic proposition, whose values *true* or *false* will be function of the switching junction control specifications being *ON* or *OFF*.
- Second, the automaton representing the switching junction is explicitly modelled as a set of constraints in adjacent equations, forcing different causal assignments.

For 1-junctions (and 0-junctions have a corresponding dual version):

- when the switch is *ON*: flows must be equal $f_i = f_j = f_k$, and effort variables must sum up to 0: $\sum e_l = 0$
- when the switch is *OFF*: there is no effort related to the junction, and each flow is transformed into a zero flow source: $Sf : f_i = 0$, $Sf : f_j = 0$, and $Sf : f_k = 0$.

As an example, the behaviour of switching junction SW_1 in Figure 2 will provide the set of equations in Table **??**, and their corresponding evaluation forms[3].

Table 1. Equations with no causality, their associated evaluation form (with some possible causality assignments), and constraints for the equations in the four tank system model.

Equation	Evaluation form	Constraint
$(ec_3 : e_2, e_3, e_4)$	$ec_{3_1} : e_3 := e_2 - e_4$	sw_1
$(ec_2 : f_2, f_3, f_4)$	$ec_{2_1} : f_2 := f_3; f_2 := f_4; f_3 := f_4$	sw_1
$(ec_4 : f_3, e_3)$	$ec_{4_1} : f_3 := e_3/R01$	sw_1
$(ec_4 : f_3, e_3)$	$ec_{4_2} : f_3 := 0$	$\neg sw_1$

It must be noticed that there is no valid evaluation form for $(ec_3 : \; e_2 \; e_3 \; e_4)$ when $\neg sw_1$ is $true$. Then, even if ec_3

[2] We made difference between static and dynamic relations, called differential equations. Different causal assignments for differential equations provide integral or derivative approaches for behaviour estimation.

[3] HBGs provide a systematic way to derive the ODEs. But we can start the process from any kind of modelling language.

and ec_4 share the unknown variable e_3, they can only be used together when sw_1 is evaluated to $true$. Those equations with no contraints represent equations that are valid in any working mode.

Using this new system model, we only need to introduce slight modifications in the algorithms developed to compute PCs as described in (Pulido & Alonso-González, 2004); we just need to include the information about the constraints modelling discrete behaviour while building the set of MEMs for a given HPC.

The algorithm for computing MECs is the same. Only those MECs with a valid switching configuration will provide a valid MEM, i.e., those MECs will be PCs. We consider a switching configuration as valid if its associated WWF is satisfiable, that is, there is at least one configuration where it is consistent.

Algorithms for computing PCs can be easily extended to introduce constraints related to mode changes: when building each MEM, in each step we try to justify or remove one unknown variable using known values. Now, we impose the additional requirement that the set of constraints in the MEM and the constraint in the new equation, if any, provide a satisfiable formula (i.e., it contains no contradiction). For any expression or equation where there is more than one constraint, we must explore in parallel every potential solution. Hence, for a given MEC we can obtain a collection of MEMs, and each MEM will be valid in a limited set of operation modes, determined by satisfiable WWFs.

For instance, we can use ec_{3_1} to estimate the value of e_3, then use e_3 and equation ec_{4_1} to estimate the value of f_3 under the constraint sw_1, i.e. when switching junction SW_1 is set to *ON*. However, when switching junction SW_1 is set to *OFF*, $\neg sw_1$ is $true$ and we can not use both equations ec_{3_1} and ec_{4_1}. We can only use ec_{4_2} that fixes the value of f_3 to zero.

Hence, each extended MEM represent now a global consistent causal assignment for the equations in a HPC, together with a WFF in propositional logic made up of the conjunction of every constraint. For instance, both $sw_1 \wedge sw_2 \wedge sw_3$ and $sw_1 \wedge \neg sw_2 \wedge sw_3$ are satisfiable formula, but $sw_1 \wedge \neg sw_1$ is not. We term *label* of a MEM to its WFF.

The complete set of MEMs plus their associated constraints represent all the consistent causal assignments for the equations in a HPC, i.e. they represent the evaluation form for all the possible behavioural models in a HPC. Each MEM in a HPC will have one discrepancy node[4], but will have different sets of equations depending of the current mode on the system. For any given MEM, the hyperarcs represent the equations used to compute the head of the hyperarc using the variables in the tail of the hyperarc. Leaf nodes in the hypergraph are either measured variables or previously estimated unknown variables, i.e. potential cyclical configurations. In the hypergraph differential constraints are represented as dashed hyperarcs, and they do not introduce loops.

We call that complete collection of extended MEMs for any HPC, Hybrid MEM (H-MEM). In Figure 7 we show the H-MEM for HPC_1, in our case study. Labels in the right hand side of the hyperarcs as $\{sw_j\}$ or $\{\neg sw_j\}$ represent the constraints related to the original switching junction CSPECs. Remainder labels represent either the name of the equation, ec_k, or the faulty parameter related with the equation: either R_x, or C_x for faults in resistance or capacitance elements in the original BG model. This H-MEM represents the most complex configuration in our case study, since the models for HPC_1 in different modes require different causality assignments. For a given mode, we will use only those paths from the leaf nodes to the discrepancy node (to compute a residual) whose constraints are consistent.

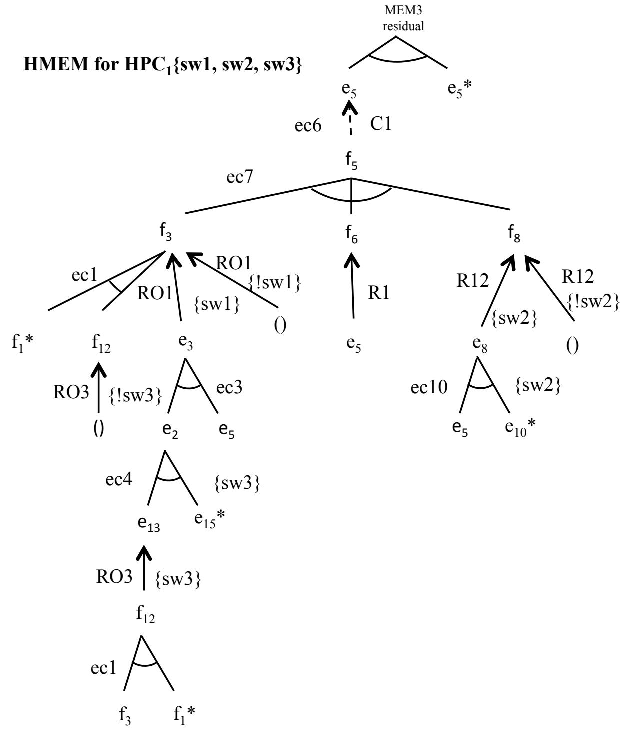

Figure 7. Hybrid Minimal Evaluation Model for HPC_1. We use dashed arcs for differential equations, and solid arcs for instantaneous equations. Observed variables are marked with an asterisk. Different arcs entering in a node represent different paths.

We do not compute the complete H-MEM. We only need to know the set of constraints contained in the HPC, because each mode is defined by a WFF involving just the atomic propositions for switching-junctions involved in the HPC. Once we know the For instance, the H-MEM for HPC_1 in

[4] The only variable estimated and measured, which can be the origin of a real conflict. The discrepancy node is equivalent to a residual in FDI terminology.

Figure 7[5] has consistent MEMs in the following configurations, among others:

$$sw_1 \wedge sw_2 \wedge sw_3$$
$$\neg sw_1 \wedge sw_2 \wedge sw_3$$
$$sw_1 \wedge \neg sw_2 \wedge sw_3$$

It must be noticed that in engineering systems, not every possible configuration is feasible for security reasons. For instance, in our system is not possible to have both switches, sw_1 and sw_3 both *off* at the same time. Hence, those models will never be generated.

In fact, we do not need to know all these configurations. Once a new mode is suspected, if one H-MEM contains constraints related to that mode, we build its actual MEM, just using the expressions of the equations that are valid under the current mode, depending on the actual values of the switches. We depth-first traverse, dft, the hypergraph from the discrepancy node to the leaf nodes:

```
FUNCTION  mem = create-MEM(hmen: H-MEM, m: mode)
begin
  dft(hmem.discrepancy_node, hmem, m, mem);
 return mem;
end FUNCTION

FUNCTION dft (c: node, hmen: H-MEM, m: mode, mem: MEM)
begin
insert(c) in mem; // c is current node
for y = each node in hmem | (c,y) is an edge in hmem
 AND (c,y).label is compatible with m
do
  if y was not visited yet then
    insert(c, y) in mem;
    insert(c,y).label in mem.label;
    dft(y, hmem, m, mem);
  end
end
end FUNCTION
```

The model described by that MEM can be implemented in many different ways. In this work we opted for DBNs implemented as Particle Filters.

4.2. From H-MEMs to minimal DBNs

In (Alonso-Gonzalez et al., 2011) it was described how to derive the transition and the observational model of a DBN-PC factor from a MEM. The transition model estimates the next state(s) value(s) in the DBN from the inputs, and the current state(s) value(s); the observational model computes the value of the system output (only one output in a MEM because it is minimal) given the state(s). In that work the model was built manually. In this section we explain how to efficiently and automatically derive the model for the DBN-PC factor from the MEM; that model will be later implemented as a particle filter in Matlab(©MathWorks).

While building a MEM it is trivial to identify the discrepancy node, input and output variables (measurements), and state variables: first, we are using integral causality to build the DBN-PC factor, then it is straightforward to determine the state variables that will be the nodes computed by the set of differential constraints, i.e. those modeling dynamics. In our case study, we have only one state variable for each HPC, and they correspond to the capacitor elements in the original Hybrid Bond-Graph: e_5, e_{10}, e_{15}, and e_{20}. Second, in the HBG framework, only measured outputs can be the origin of a discrepancy. Hence, there will be only one output variable in each H-MEM, and that will be the discrepancy node.

Once we identify those elements, we implement the proposal by Alonso-González et al. (Alonso-Gonzalez et al., 2011). We assume that the analytical expression for each equation in the MEM is known. Then, we just need to find the transition model and the observational model for the DBN factor.

Obtaining the observational model is simple: we depth-first search in the MEM from the discrepancy node to state variables and inputs, just using instantaneous constraints. That sequence of equations in reverse order is the analytical expression of the observational model.

```
FUNCTION get_observational_model (mem: MEM,
om: observational_model)
begin
   return dfs(MEM.discrepancy_node, mem, om);
end FUNCTION

FUNCTION om = dfs (n: node, mem: MEM,
om: observational_model)
begin
for y = each node in mem | (n,y) is an edge in mem
do
  add (n,y) at the beginning of om;
  if (n,y) is a differential edge then
    mark n as mem.observed-state-variable;
  else
    dfs (y, mem, om);
  end
end
end FUNCTION
```

The transition model can be obtained searching depth first the MEM from the state variables to state variables and inputs, following the requirements in (Alonso-Gonzalez et al., 2011). The analytical model is obtained from the transcription of that sequence of equations in reverse order.

```
FUNCTION get_transition_model (mem: MEM,
  tm: transition_model, om: observational_model)
begin
  for st = each state-variable
   in mem.observed-state-variable
  do
    dfs2 (st, mem, tm, om);
  end
end FUNCTION

function dfs2 (st: node, mem: MEM,
tm: transition_model, om: observational_model)
begin
 for y= each node | (n, y) is an edge in mem
  AND (n,y) is not an edge in om
```

[5] To make the H-MEM more readable, we have collapsed every effort or flow variable of the same junction with a unique index. For instance e4=e5=e6=e7 or f2=f3=f4

```
  do
    if (st,y) has not yet been visited then
      mark (st,y) as visited;
      insert (st,y) at the beginning of tm;
      tm= dfs2 (y, mem, tm, om);
    end
  end
end FUNCTION
```

Different proposals have been made to improve the algorithms computing the set of PCs. One of them is able to find any cyclical configuration (Pulido & Alonso, 2001). Those configurations can introduce algebraic loops in the model. Those loops containing differential constraints are no longer loops; in fact, they represent the integration step in the simulation model. Those loops containing just algebraic loops can be solved if there is a direct path from observed variables to an unknown variable in the loop. Otherwise, we need to create a subset of equations that need to be solved using a numeric solver (Pulido et al., 2010). In this case, there could be efficiency problems for the DBN-PC factor. All these analysis must be done before we build either the observational or the transition model.

Regarding the efficiency of this proposal, there will be always a trade-off in terms of space and computation time. Depending on the system under study, different heuristics can be applied to customize the algorithms performance. For small systems with a limited number of modes, most of the computation can be done off-line and cached to speed up the on-line code generation. In the H-MEM there will be a number of equations that will always appear in the system because they are causality independent. These could be also pre-compiled, since they will always provide the same analytical expression in any MEM in any mode.

5. Results

The four-tank hybrid system in Figure 1 has been used to show the applicability of our proposal.

Simulated data has been generated with 5% level of noise, during 1000 s with a sample period of 0.1. We run several experiments with different mode configurations and different faults, varying the size and time of fault occurrence. Results for all these situations were equivalent to the example presented next.

5.1. Tracking and fault isolation results

For the four tank system we computed the set of four HPCs and their corresponding H-MEMs, using the new algorithms. Since our models were provided by the HBG of the whole system when every switching-junction was set to ON, we obtained the same results.

Figure 8 shows the results obtained for one of the experiments run. First row (Figure 8) compares the three measurements and its estimation by the DBN-PC, while second row shows the residual obtained for each DBN-PC. DBN from HPC4 has not been included in the figure as this PC is always deactivated during the experiment. The results of the experiments have 10000 time stamps. The graphs built with those signals were difficult to read due to its size. Some time intervals during stationary state have been ommited to avoid that problem. Because of that, the time stamps that are mentioned below will not match with the time stamps in Figure 8, but the comments about the real time stamps are correct.

Initially, water tanks are empty, and start to fill in at constant rate. Hence, the initial configuration of the system is SW_1 and SW_3 set to *ON*, and SW_2 and SW_4 set to *OFF*. Tanks 1 and 3 start to fill in, and approximately at instant 500 sampling periods both tanks reach stationary state. At this time, the level in tank 1, h_{T1}, and the level in tank 3, h_{T3}, are lower than the height of the connecting pipes, h, and consequently, there is no flow through the connecting pipes.

At instant 2000 sampling steps, controlled junction SW_3 is set to *OFF*, so the system mode changes. Simultaneously, $HPC1$ and $HPC3$, which contain constraints related to SW_3, must change their models to accommodate the new operation mode. It is necessary to reassign causality in our H-MEMs. Once the new computational expression for the HPCs have been generated, the corresponding DBNs are built. As shown in Figure 8, DBN-PC1 and DBN-PC3 are able to correctly estimate the level of tank 1 and 3, respectivley, immediately after the mode change. Regarding $HPC2$ and $HPC4$, since both HPCs do not contain constraints related to the switching junction SW_3, none of them is affected by the mode change so their DBNs do not need to be generated.

SW_3 has been set to *OFF*, so the level of tank 3 decreases until it becomes zero, while the level of tank 1 increases. At instant 2100 sampling periods, the level of tank 1 reaches the height of the connecting pipe between tanks 1 and 2. At this point, the equations related to autonomous transition in SW_2 are set to *ON* and water begins to fill in tank 2. HPC_1 and HPC_2 are affected by this mode change. In both cases, the models of their H-MEMs are updated and the DBNs are generated quickly. Both of them are able to correctly estimate the measurements for the new mode.

At instant 7000 sampling periods a 20$\%$ leak in tank 1 occurs. As a consequence, the level of tank 1 decreases, while the estimation of $HPC1$ does not. Hence, residual of $HPC1$, which is the only one containing C_1 as a fault candidate, activates, triggering the fault isolation procedure. Regarding $HPC2$, since the level of tank 1 decreases due to the fault, at instant 7050 sampling periods the autonomous junction SW_2 transitions again to *OFF* mode, and $HPC2$ changes mode again. The H-MEM for $HPC2$ is updated immediately and DBN-PC2 is built; it is able to correctly estimate the level of tank 2 for the new mode.

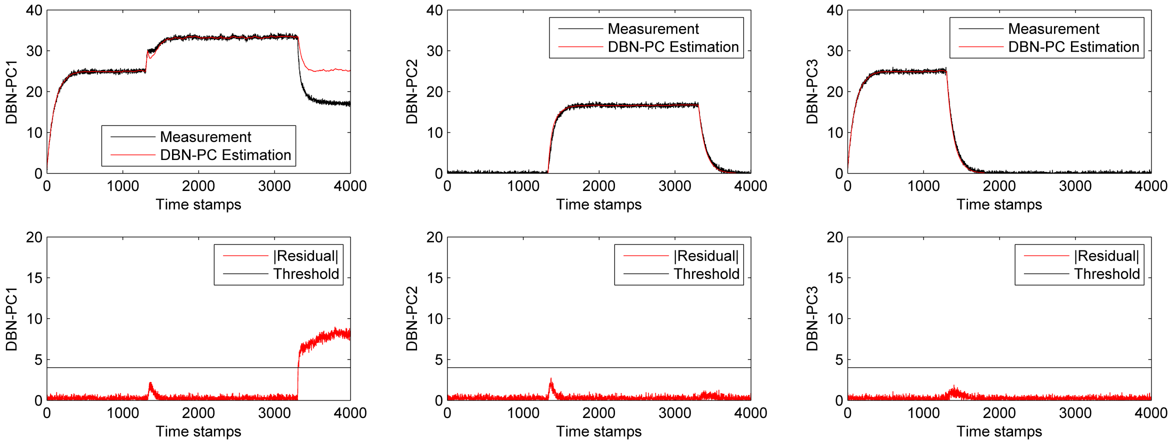

Figure 8. Experiment for the four-tank system.

For the case study, the average time for updating the H-MEM to generate the current MEM, and to generate the analytical model for the DBN is less than 1 ms. The algorithms were built in Java and they were run in a Intel Core i3 processor with RAM of 4GB.

5.2. Fault identification results

A 20% leak in tank 1, which is related to parameter C_1 in the models, was introduced at instant 7000 sampling periods and 9 sampling periods later (0.9 seconds) DBN for HPC1, DBN-PC1, detects a fault. According to the Fault Signature Matrix, FSM, in Table 2, the set of fault candidates is $\{C_1, R_1, R_01, R_{03}, R_{12}\}$. The FSM describes the relation between the set of faulty parameters in the model and the set of HPCs.

Table 2. Fault Signature Matrix for the four tank system. The parameters in row are directly obtained from the BG model, and their corresponding constituent equations.

	$HPC1$	$HPC2$	$HPC3$	$HPC4$
C_1	1			
C_2		1		
C_3			1	
C_4				1
R_{01}	1		1	
R_{03}	1		1	
R_1	1			
R_2		1		
R_3			1	
R_4				1
R_{12}	1	1		
R_{34}			1	1

DBN-PC1 can be extended with a node for the faulty parameter which needs to be identified as explained in (Alonso-Gonzalez et al., 2011). In this scenario, five DBNs were built, one for each fault candidate. Figure 9 shows the results obtained using the DBN-PC1 to estimate C_1. The DBN is able to track the system behavior and to obtain an estimation for the parameter quickly converging to a 19.3% fault in C_1. DBN-PCs to estimate the remaining faults were not able to converge. Hence, the candidates were discarded.

6. Related Work

In our approach we do not need to enumerate the complete set of modes, as required by other works using parameterized ARRs (Cocquempot et al., 2004; Bayoudh, Travé-Massuyès, & Olivé, 2009), or using pure discrete models. We just need to provide the constraints for the equations. Later on, for each HPC, its model will be generated for a specific working mode. However, in the general case, we would need to compute the set of HPCs at least for the configuration where every switch is on. But we don't need to instantiate them. We just need to check what HPC has a valid causal assignment for the current operation mode. In that set we build a superset of parameterized ARRs. Our main assumption to do so is that every structural model is a subset of the structural model where every switch is ON.

Using DBNs derived from HPCs for fault detection, isolation and identification avoids using several techniques as in (Narasimhan & Biswas, 2007; Rienmüller et al., 2009). But the mode must be observed, thus requiring an hybrid state estimation (Hofbaur & Williams, 2004; Koutsoukos, Kurien, & Zhao, 2003).

Using minimal DBN-PCs as a unified model, we do not provide general solutions for hybrid systems diagnosis such as HyDe (Narasimhan & Brownston, 2007). But we combine continuous estimation with discrete changes, and we do not restrict our solution to pure discrete systems as in HyDe or probabilistic approaches. Moreover, interleaving different DBN continuous models we avoid the usage of hybrid DBNs.

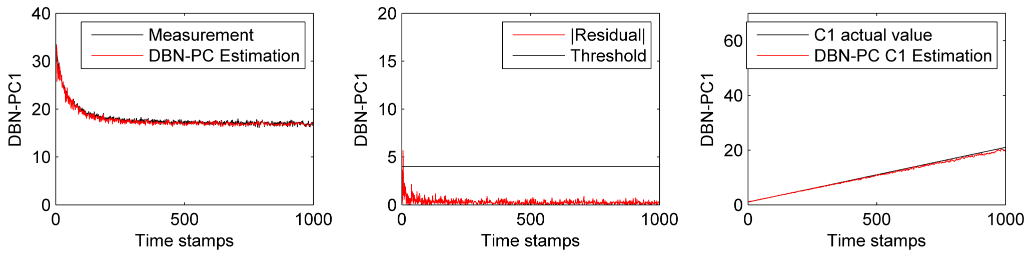

Figure 9. Identifiying the fault in parameter C1.

Our work is closely related to efficient generation of simulation (Roychoudhury et al., 2011) and state observer models (Podgursky et al., 2010) in TRANSCEND (Narasimhan & Biswas, 2007), because we assume that changes in model causality will be mostly local, but we do not rely upon the Hybrid SCAP algorithm to generate a valid HBG model for the entire system. Instead, we need to know every feasible causal assignment in the system description, and perform on-line local search in the H-MEMs. Finally, (Bregon, Alonso, et al., 2012) proposed to obtain the set of HPCs from block diagrams derived from HBGs (Roychoudhury et al., 2011). We improve that proposal by directly generating DBN computational models instead of simulation models, thus improving fault detection capabilities, and the process can be performed for any structural and causal model conform with our definitions in section 4.

7. Conclusions

This work proposes an efficient and unified solution for hybrid systems fault detection, isolation and identification, assuming that it is possible to identify the current system state.

Efficiency is obtained by avoiding the explicit consideration of every possible mode configuration. HPCs, avoid computing PCs from scratch for every new configuration. Finally, a new algorithm is proposed for efficient on-line computation of minimal DBN-PCs.

Implementing HPCs as minimal DBNs provides a unified solution, because DBNs naturally allow fault detection, fault isolation and fault identification of continuous systems. Using HPCs we transform a hybrid diagnosis problem in a sequence of continuous diagnosis problems, avoiding the use of hybrid DBNs. An additional effect of using HPCs to generate DBNs is that we must not simulate the complete system DBN model, thus improving on-line computational efficiency.

As further work, we plan to integrate this proposal in a common framework including both discrete and parametric faults (Moya, Bregon, Alonso-González, Pulido, & Biswas, 2012; Moya, Bregon, Alonso-González, & Pulido, 2013). Besides, we want to incrementally generate the MEM in new modes from the MEM in the previous one; and to test the approach in a more demanding scenario with larger set of modes, and faster dynamics. Finally, we will couple this framework with a reliable hybrid state estimator.

Acknowledgments

This work has been partially supported by the Spanish MCI TIN2009-11326 grant.

References

Alonso-Gonzalez, C., Moya, N., & Biswas, G. (2011). Dynamic Bayesian network factors from possible conflicts for continuous system diagnosis. In *Proc. of the 14th Intl. Conf. on Advances in AI* (pp. 223–232). Berlin: Springer-Verlag. Available from `http://dl.acm.org/citation.cfm?-id=2075561.2075588`

Armengol, J., Bregón, 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* (pp. 1480–1485).

Bayoudh, M., Travé-Massuyès, L., & Olivé, X. (2009). Diagnosis of a Class of Non Linear Hybrid Systems by On-line Instantiation of Parameterized Analytical Redundancy Relations. In *Proc. of the XX Intl. Workshop on Principles of Diagnosis, DX'09* (p. 283-289). Stockholm, Sweden.

Bregon, A., Alonso, C., Biswas, G., Pulido, B., & Moya, N. (2012). Fault Diagnosis in Hybrid Systems using Possible Conficts. In *Proc. of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, 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.

Bregon, A., Pulido, B., Biswas, G., & Koutsoukos, X. (2009). Generating Possible Conflicts from Bond Graphs Using Temporal Causal Graphs. In *Proceedings of the 23rd European Conference on Modelling and Simulation, ECMS'09* (p. 675-682). Madrid, Spain.

Cocquempot, V., El Mezyani, T., & Staroswiecki, M. (2004, Jul.). Fault detection and isolation for hybrid systems using structured parity residuals. In *Proceedings of the 5^{th} Asian Control Conference.*

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

Karnopp, D., Margolis, D., & Rosenberg, R. (2006). *System Dynamics: Modeling and Simulation of Mechatronic Systems*. New York, NY, USA: John Wiley & Sons, Inc.

Koller, D., & Lerner, U. (2001). Sequential Monte Carlo Methods in Practice. In N. d. F. A. Doucet & N. Gordon (Eds.), (chap. Sampling in factored dynamic systems). Springer.

Koutsoukos, X., Kurien, J., & Zhao, F. (2003). Estimation of Distributed Hybrid Systems Using Particle Filtering Methods. In *Proc. of the Intl. Workshop on Hybrid Systems: Computation and Control, HSCC'03* (p. 298-313). Springer.

Moya, N., Bregon, A., Alonso-González, C., & Pulido, B. (2013). A Common Framework for Fault Diagnosis of Parametric and Discrete Faults using Possible Conflicts. In *Proc. of the 15th Intl. Conf. on Advances in AI* (pp. 239–249). Madrid, Spain: Springer-Verlag, Berlin.

Moya, N., Bregon, A., Alonso-González, C., Pulido, B., & Biswas, G. (2012, Jul-Aug). Automatic Generation of Dynamic Bayesian Networks for Hybrid Systems Fault Diagnosis. In *Proc. of the XXIII Intl. Workshop on Principles of Diagnosis, DX'12.* Great Malvern, UK.

Murphy, K. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. Unpublished doctoral dissertation, University of California, Berkeley.

Narasimhan, S., & Biswas, G. (2007, May). Model-Based Diagnosis of Hybrid Systems. *IEEE T. Syst. Man Cy. Part A*, *37*(3), 348-361.

Narasimhan, S., & Brownston, L. (2007). HyDE: A General Framework for Stochastic and Hybrid Model-based Diagnosis. In *Proc. the XVIII Intl. Workshop on Principles of Diagnosis, DX'07* (p. 186-193). Nashville, USA.

Podgursky, B., G. B., & Koutsoukos, X. (2010, Oct). Efficient Tracking of Behavior in Complex Hybrid Systems via Hybrid Bond Graphs. In *Proc. of the Prognostics and Health Management Conf., PHM'10.* Portland, OR, USA.

Pulido, B., & Alonso, C. (2001). Dealing with cyclical configurations in (MORDRED). In *Proc. of the 9th Intl. Conf. on Advances in AI, CAPIA'01* (p. 983-992). Gijon, Spain.

Pulido, B., & Alonso-González, C. (2004, Oct.). Possible Conflicts: a compilation technique for consistency-based diagnosis. *IEEE Trans. Sys. Man Cy. Part B*, *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.

Rienmüller, T., Bayoudh, M., Hofbaur, M., & Travé-Massuyès, L. (2009). Hybrid Estimation through Synergic Mode-Set Focusing. In *Proc. of 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Safeprocess'09* (p. 1480-1485). Barcelona, Spain.

Roychoudhury, I., Biswas, G., & Koutsoukos, X. (2008, Sept.). Comprehensive Diagnosis of Continuous systems Using Dynamic Bayes Nets. In *Proc. of XIX Intl. Workshop on Principles of Diagnosis, DX'08.* Blue Mountains, Australia.

Roychoudhury, I., Daigle, M., Biswas, G., & Koutsoukos, X. (2011, June). Efficient simulation of hybrid systems: A hybrid bond graph approach. *SIMULATION: Transactions of the Society for Modeling and Simulation International*(6), 467-498.

BIOGRAPHIES

Belarmino Pulido 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 Informatica in the University of Valladolid, where he is Associate Professor since 2002. He has been working on 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. He is a member of the IEEE (M'2000), ACM (M'2003), and CAEPIA (1997, part of EC-CAI) professional associations. He is also the coordinator of the Spanish Network on Supervision and Diagnosis of Complex Systems since 2005. Currently he is working on model-based diagnosis and prognosis of continuous and hybrid systems, using both centralized and distributed approaches.

Noemi Moya receives her Degree, her MsC Degree, and PhD Degree in Computing Sciences from the Universidad de Valladolid, Valladolid, Spain, in 2006, 2008, and 2013, respectively. From 2008 to 2013 she has been a research and teaching assistant at the Departamento de Informatica from the Universidad de Valladolid. She has been working in Model-Based Diagnosis of continuous and hybrid systems. She has been part of a national project related to diagnosis and she has been a member of the Spanish National Network on Supervision and Diagnosis of Complex Systems.

Carlos J. 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.

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.