Improving Fault Isolation and Identification for Hybrid Systems with Hybrid Possible Conflicts

Anibal Bregon and **Carlos J. Alonso-González** and **Belarmino Pulido**
Departamento de Informática, Universidad de Valladolid, Valladolid, 47011, Spain
e-mail: {anibal,calonso,belar}@infor.uva.es

Abstract

Model-based fault isolation and identification in hybrid systems is computationally expensive or even unfeasible for complex systems due to the presence of uncertainty concerning the actual state, and also due to the presence of both discrete and parametric faults coupled with changing modes in the system. In this work we improve fault isolation and identification performance for hybrid systems diagnosis using Hybrid Possible Conflicts. The Hybrid Bond Graph modeling approach makes feasible to track system behavior without enumerating the complete set of system modes. Hybrid Possible Conflicts focus the analysis on potential mode changes on those subsystems whose behavior deviates from expected. Moreover, using information derived from the Hybrid Bond Graph model, we can cope with both discrete and parametric faults in a unique framework.

Fault detection with Hybrid Possible Conflicts relied upon an statistical test to decide when a significant deviation in the residual occurs. Fault detection time was later used to start the fault isolation and identification stages. In this work we propose to analyze the evolution of the residual signal using CUSUM to find a more accurate estimation of the time of fault occurrence, which allows to improve both the potential new modes tracking and the parametric fault identification. Moreover, we extend our previous proposal for fault identification in continuous systems to cope with fault identification along a set of mode changes while performing parameter identification. We have tested these ideas in a four-tank hybrid system with satisfactory results.

1 Introduction

Complex hybrid systems are present in a broad range of engineering applications, such as mechanical systems, electrical circuits, or embedded computation systems. The behavior of these systems is made up of continuous and discrete event dynamics.The main sources of hybrid behavior are discrete actuators, like discrete valves or switches in fluid or electrical systems, respectively. These changes in the continuous behavior increase the difficulties for accurate and timely online fault diagnosis. Our focus in this paper is on developing efficient model-based methodologies for online fault isolation and identification in complex hybrid systems.

Both the DX and the FDI communities have approached hybrid systems modeling and diagnosis during the last 20 years. They have used different modeling proposals [1; 2; 3], and have approached diagnosis either as hybrid state estimation [2] or as online state tracking [4; 5; 6], or a combination of both methods [7]. The main difficulties in any approach is to estimate the current state or set of states, and to diagnose that set of feasible states. Both tasks are computationally expensive or even unfeasible for complex systems. Several approaches have been proposed in the DX field to tackle these problems [4; 6].

In this work we have selected the hybrid system modeling based on Hybrid Bond Graphs (HBGs) [1; 6], together with consistency-based diagnosis using Possible Conflicts (PCs) [8]. HBGs are an extension of Bond Graphs (BG) [9], which models the discrete changes as ideal switching junctions that can be set to ON or OFF according to an automaton. In [10] we presented Hybrid Possible Conflicts (HPCs) as an extension of Possible Conflicts using HBGs to track hybrid systems behavior. Later, the HPCs approach was extended to integrate fault diagnosis of both parametric and discrete faults using HPCs [11] in a unique framework.

In order to achieve efficient fault identification, it is very important to determine the time of fault occurrence as accurately and quickly as possible. But there is a required trade-off between fast and reliable fault detection. In our approach we relied upon an statistical test to decide when a residual deviates from the current mode, and used this time to start the fault isolation and identification stages, however, the fault detection instant can be delayed from the fault occurrence time and this has some problems (e.g., that the fault identification process is delayed, or that we have to assume that we know the value of the state variables at the beginning of the identification process). In this work we propose to analyze the evolution of the residual signal using the CUSUM algorithm [12; 13] to find a more accurate estimation of the time of fault occurrence, both for potential new modes tracking and for parametric fault identification. Moreover, we extend our previous proposals for fault identification [14; 15] to cope with fault identification along a set of mode changes while performing the parameter identification.

The rest of the paper is organized as follows. Section 2 presents the case study used along the paper and introduces the Hybrid Bond Graph (HBG) modeling technique. Section 3 summarizes the Hybrid Possible Conflicts (HPCs) background, while section 4 explains the unified framework for both discrete and parametric faults. Section 5 introduces some concepts related to the CUSUM algorithm required in our approach. Section 6 explains our approach for fault identification. Section 7 introduces some results obtained applying our proposal on our case study. Finally, Section 8 draws some conclusions.

2 Case Study

The hybrid four-tank system in Figure 1 will be used to show some concepts and to present some results in this work. The system has an input flow which can be sent to tank 1, to tank 3 or to both tanks. Next to tank 1 there is tank 2, once the liquid in tank 1 reaches a level of h it starts to fill also tank 2. The lower part of the system has the same configuration, tank 4 is next to tank 3 connected by a pipe at a distance h above the base of the tanks.

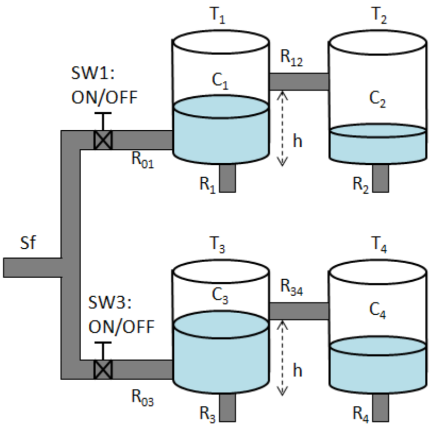

Figure 1: Schematics of the four-tank system

The methodology chosen to model the system in this work is Hybrid Bond Graph (HBG), which is an extension of Bond Graphs (BGs). BGs are defined as a domain-independent energy-based topological modeling language for physical systems [9]. Several types of primitive elements are used to build BGs: storage elements (capacitances, C, and inductances, I), dissipative elements (resistors, R) and elements to transform energy (transformers, TF, and gyrators, GY). There are also effort and flow sources (Se and Sf), which are used 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 associated two variables (effort and flow). The power is defined as *effort* $\times$ *flow* for each bond. The SCAP algorithm [16] is used to assign causality automatically to the BG.

To model hybrid systems using BGs we need to use some kind of connections which allow changes in their state. Hybrid Bond Graphs (HBGs) [1] extend BGs by including those connections. They are idealized switching junctions that allow mode changes in the system. 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 [17]. Figure 2 shows the HBG model of the four-tank system in Figure 1.

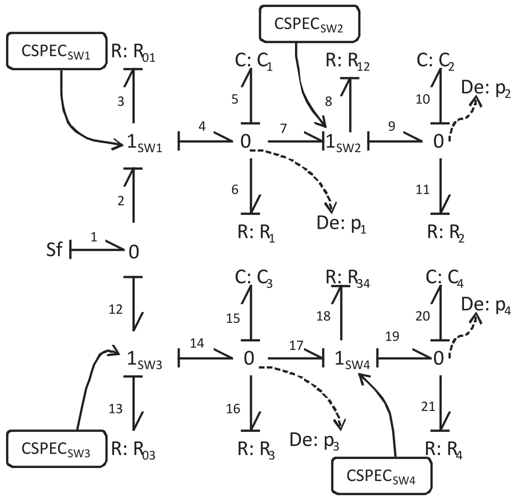

Figure 2: Bond graph model of the plant.

The system 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. Both kinds of transitions are represented using a finite state machine. Figure 3 shows: a) the automaton associated with switching junction 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 are equivalent to the ones shown in Figure 3.

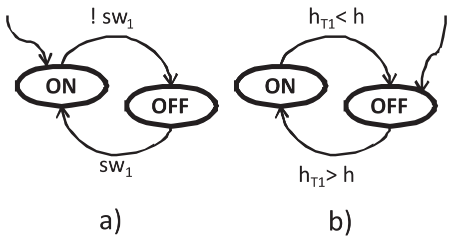

Figure 3: a) Automaton associated with the *ON/OFF* switching junction SW_1; b) Automaton representing the autonomous transition in SW_2.

3 Hybrid Possible Conflicts background

Consistency-based diagnosis of continuous systems using Possible Conflicts (PCs) [8] is based upon a dependency-compilation technique from the DX community. PCs are computed offline, finding minimal structurally overdetermined subsets of equations with sufficient analytical redundancy to generate fault hypotheses from observed measurement deviations. Only structural and causal information about the system description is required. This information can be obtained from a set of algebraic and/or differential equations, or can be automatically derived from bond graph models [18; 19]. Once the set of PCs is found, they can be implemented as simulation, state-observers or gray-box models for tracking online actual system behavior [20], or for online fault identification [14].

The PCs approach has been recently extended to cope with hybrid system dynamics, and the set of PCs for hybrid systems were called Hybrid Possible Conflicts (HPCs) [10]. HPCs rely upon the Hybrid Bond-Graph modeling formalism [1], whose main advantage is that the set of possible modes in the system do not need to be enumerated. Moreover, HBGs are capable to track online hybrid system behavior, performing online causality reassignment in the system model by means of the HSCAP algorithm [17]. Using HPCs we make even more efficient the HSCAP algorithm, because causality needs only to be revised within the subsystem defined for each HPC, and these changes are local to the switching junction affected by the mode change.

For the four-tank system 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 BG fragments of these four HPCs. In this example, the four HPCs were computed assuming that all switching junctions are set to *ON*.

As mentioned before, when any of these junctions is switched to *OFF*, causality in the system needs to be reassigned, but the HPCs generation process does not need to be restarted again [10]. The decomposition of a hybrid system model obtained from HPCs is unique, and after a mode change some portions of some HPCs can disappear (or even the entire HPC), but no additional HPC appears. It is proved in [10] that once PCs of the system have been generated considering all switching junctions set to *ON* mode, turning a switch from *ON* to *OFF* or viceversa, no genuine new HPCs will ever appear.

Regarding fault profiles, our current proposal works with single fault, and abrupt fault assumptions. Abrupt faults are modeled as an instantaneous change in a parameter, whose magnitude does not change afterwards (can be modeled as a step function).

Regarding parametric faults, fault isolation is performed by means of the Reduced Qualitative Fault Signature Matrix (RQFSM). Table 1 shows the RQFSM for the mode where each switch is set to *ON*. For a given mode, the RQFSM can be computed online from the TCG associated to an HPC [1]. In this table there is a row for each fault considered. And there is a column for each HPC. The entry in the table represent the Qualitative Fault Signature of the fault in the HPC residual, as computed in TRANSCEND [1]. The "reduced" tag means that the Qualitative Fault Signature is computed within the subsystem delimited by a HPC, and not for the whole set of measurements [18]. Once fault detection is performed, we can use this information to reject those faults whose residual evolution does not match the qualitative signatures in this table.

We also consider discrete faults, i.e. faults in discrete actuators, as commanded mode switches which do not perform the correct action. In our case study, there are four faulty situations to be considered, where SW_i denotes the switching junction i of the system.

1. SW_i = 11: SW_i stuck ON (1).
2. SW_i = 00: SW_i stuck OFF (0).
3. SW_i = 01: Autonomous switch ON (SW_i is OFF (0) and it switches to ON itself (1)).
4. SW_i = 10: Autonomous switch OFF (SW_i is ON (1) and it switches to OFF itself (0)).

Table 1: Reduced Qualitative Fault Signature Matrix.

	$HPC1$	$HPC2$	$HPC3$	$HPC4$
C_1^+	$-+$			
C_2^+		$-+$		
C_3^+			$-+$	
C_4^+				$-+$
R_{01}^+	$0-$		$0+$	
R_{03}^+	$0+$		$0-$	
R_1^+	$0+$			
R_2^+		$0+$		
R_3^+			$0+$	
R_4^+				$0+$
R_{12}^+	$0-$	$0-$		
R_{34}^+			$0+$	$0-$

The relation between the HPCs and their related switching junctions can be seen in Table 2, which is called Hybrid Fault Signature Matrix (HFSM). This information can be used in the unified framework for discrete and parametric fault isolation and identification [11].

Table 2: Hybrid Fault Signature Matrix (HFSM) showing the relations between switching junctions and each HPC.

	$HPC1$	$HPC2$	$HPC3$	$HPC4$
1_{SW_1}	1		1	
1_{SW_2}	1	1		
1_{SW_3}	1		1	
1_{SW_4}			1	1

Discrete faults usually introduce high non-linearities in the system outputs, that should be easily detected if magnitudes related to the failing switch were measured, generating almost instantaneous detection for discrete faults. In this case, exoneration could be applied. But even if those measurements are not available we can still use the qualitative signature of the effects of the discrete faults in the HPC residuals. With this information we can build the so-called Hybrid Qualitative Fault Signature Matrix (HQFSM) that can also be used for exoneration purposes in the fault isolation stage. In our system we can build the following HQFSM for $HPC1$ and $HPC3$, which are linked to commanded switches SW_1 and SW_3, which are the potential source of discrete faults in our system. We do not show SW_2 and SW_4 in the table since they introduce hybrid dynamics in the system, but they can not be the source of a discrete fault.

Table 3: Hybrid Qualitative Fault Signature Matrix.

	$HPC1$	$HPC3$
$1_{SW_1}(11)$	$+$	$-$
$1_{SW_1}(00)$	$-$	$+$
$1_{SW_1}(01)$	$+$	$-$
$1_{SW_1}(10)$	$-$	$+$
$1_{SW_3}(11)$	$-$	$+$
$1_{SW_3}(00)$	$+$	$-$
$1_{SW_3}(01)$	$-$	$+$
$1_{SW_3}(10)$	$+$	$-$

Next section presents our diagnosis framework for hybrid systems using HPCs.

4 Hybrid Systems Diagnosis using HPCs

As we mentioned before, tracking of hybrid systems can be performed using Hybrid PCs [10]. Initially, the set of HPCs is built assuming all switching junctions are set to *ON*.

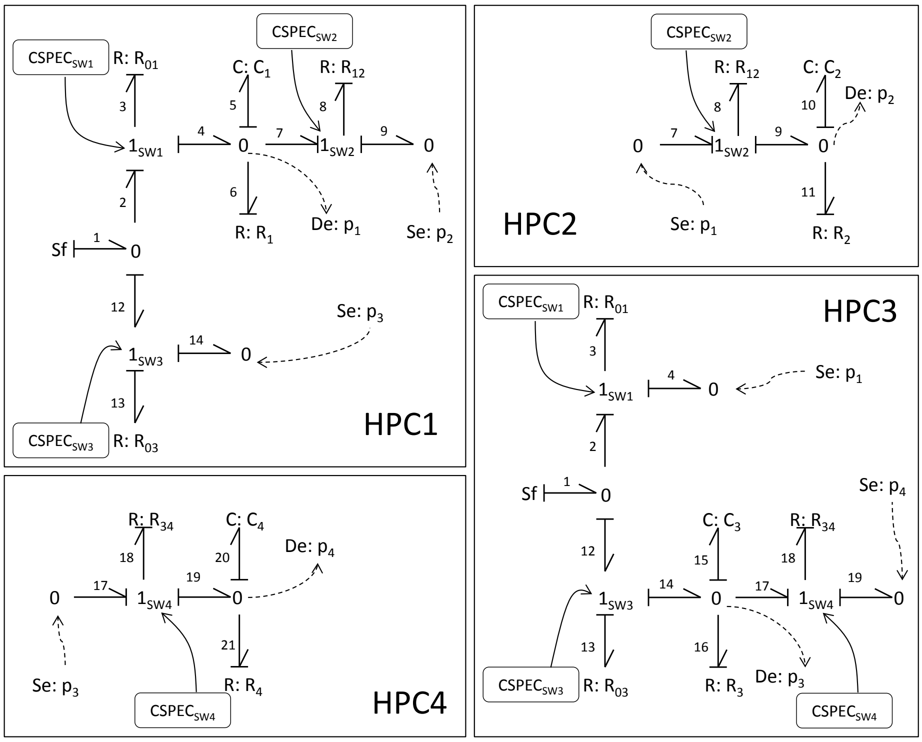

Figure 4: Bond graphs of the four PCs found for the four-tank system.

Afterwards, the set of models for the HPCs for the actual mode are efficiently built, and they start tracking the system. Whenever a mode change, commanded or autonomous, is detected, a new set of models for the HPCs is computed online.

In case a fault occurs, one or more HPC residuals will trigger. Significant deviations in the residuals are found using the statistical Z-test. Based on the activated residuals for the set of HPCs in the current mode, the structural information in the HQFSM (Table 3), and the RQFSM (Table 1), we build the current set of fault candidates. This set can contain both discrete and parametric faults. Since discrete faults generally have a bigger and potentially more dangerous influence in the system behavior, in our framework we consider discrete faults as preferred candidates before considering the parametric ones. If there is no discrete fault as candidate, then we directly go to the fault identification as described in Section 6.

At this point we run the CUSUM algorithm (described in Section 5) to approximately determine the time of fault occurrence. Once this is done, we create a new simulation model using the HPCs, and starting at the fault time determined by the CUSUM, we begin tracking the system behavior in each one of the hypothesized mode changes (the HQFSM and the qualitative value of the HPC residuals are used to reject those modes that are inconsistent with expected deviations in the HQFSM). If the hypothesized mode is the correct one, the residual for that mode will go to zero after a relatively small period of time (this is possible, as we will show later, thanks to the accurate estimation of the fault time provided by the CUSUM). If the hypothesized mode is not correct, the residual will keep deviating from zero, and after an empirically determined time window without converging, the discrete fault candidate will be discarded. If only one mode has the residual close to zero, this is the new system mode.

If the residual for each hypothesized new mode does not converge to zero, discrete faults (as mode changes) are discarded and we focus on parametric faults, starting the identification stage. As mentioned before, qualitative fault signatures in the RQFSM can be used to reject those parametric faults non consistent with current observations thus focusing even further the fault identification stage.

Finally, once the set of parametric fault candidates is refined through the RQFSM, we perform fault identification for the set of remaining parametric fault candidates. Fault identification is done with hybrid parameter estimators, which are presented in Section 6.

5 Time of Fault Estimation using CUSUM

In the previous section we have presented our fault isolation approach of discrete faults by hypothesizing the faults compatible with the Hybrid Qualitative Fault Signature Matrix and filtering out those faults whose models do not converge. Divergence of non-current models is usually easy to check when we are dealing with discrete faults. However, the convergence of the current model may be slow if initial values of the state variables of the model are not known or our initial guess is far from the actual value. We are assuming that we are able to track the system dynamic before the occurrence of a fault. In other words, we are assuming that we know -or we are able to estimate- the state variables before the time of fault occurrence. Hence, in order to speed up the convergence of the current model, it is important to have a good estimation of that time.

The cumulative sum algorithm, CUSUM, introduced by [12] and discussed in detail in [13] and elsewhere, is an optimal fault detection algorithm that can also provide a estimation of the time of fault occurrence t_0, as we will detail later. Nevertheless, it makes the strong assumption that the signal we are tracking changes its mean value from a constant initial mean μ_0 to a final constant mean μ_1.

On the other hand, the Z-test [21] is a sub-optimal fault detection algorithm compared to CUSUM, but it makes no assumptions concerning the properties of the new mean value. Particularly, it does not require this to be constant.

In order to have a robust fault detection mechanism and a good approximation of the fault time, we have opted for combining both tests. We use Z-test to perform fault detection and, afterwards, we estimate the fault time using CUSUM.

CUSUM was designed to detect abrupt changes in the mean of stochastic signals. In the simple case of a Gaussian residual, $res(i)$, of constant variance σ^2, constant and known initial mean μ_0 and constant and known final mean μ_1, the decision signal, S_k, is $S_k = \sum_{i=1}^{k} s_i = \sum_{i=1}^{k} \frac{\mu_1-\mu_0}{\sigma^2}(res(i) - \frac{\mu_0+\mu_1}{2})$. Hence, for a window of N samples with a change in mean at $1 \leq t_0 \leq N$, S_k decreases at the constant rate $\mu = \frac{\mu_1-\mu_0}{2}$ for $k < t_0$ and increases by μ for $t_0 \leq k$. It can be shown [13] that the change time t_0 can be estimated as $\hat{t}_0 = \arg\min_k S_k$.

When μ_1 is unknown, it can be set to the residual corresponding to the smallest fault to be detected, typically some units of the residual noise deviation, σ. This can be done without increasing the fault positive alarm rate because we use Z-test to perform fault detection, and we only use this CUSUM variant to estimate the time of fault occurrence, t_0. We have also tried estimating μ as the empirical mean of the residual, with similar results. In all the cases we have tested, the estimated time of fault occurrence, $\hat{t}_0$, computed by CUSUM, is smaller than the detection time provided by Z-test.

6 Fault Identification with HPCs

Once all the discrete fault candidates have been discarded, we have to do fault identification for the set of isolated parametric faults. In previous work [14] we proposed to use minimal parameter estimators computed from PCs to generate parameterized estimators. However, that approach is not applicable for hybrid systems fault identification since we can have mode changes during the identification process. As a solution, we propose a extension of our minimal parameterized estimators which are computed directly from HPCs, thus being able to handle mode changes during the identification process.

The fault identification process is done by the following steps: (i) model decomposition by offline computation of the set of HPCs from the hybrid bond graph model; (ii) offline computation and selection of the better hybrid estimator for each fault candidate; (iii) after the fault isolation process, online quantitative parameter estimation procedure over the hybrid estimators related with the set of isolated fault candidates; and (iv) decision procedure to select the faulty candidate.

Using HPCs we can derive the structure of a hybrid parameterized estimator, e_{hpc_k}, for a hybrid system. The parameterized estimator e_{hpc_k} can be used as a hybrid estimator as stated in the following proposition:

Proposition 1. *A HPC, HPC_k, along with its set of input variables, u_{hpc_k}, the commanded signals of the switching junctions, sw_{hpc_k}, and initial value of the parameter to identify, θ_f, can be used as a parameter estimator using $\hat{y}_{hpc_k} = e_{hpc_k}(u_{hpc_k}, \theta_f, sw_{hpc_k}(t))$, where the measured variable estimated by the HPC, $\hat{y}_{hpc_i}$, is solved in terms of the remaining measured variables.*

Each estimator is uniquely related to one HPC, hence it contains minimal redundancy required for parameter estimation. In this case, each HPC has an executable model that can be used for simulation purposes. For the four-tank system we have obtained four hybrid parameter estimators shown in table 4, one for each HPC.

Estimator	*Related PC*	*Parameters*	*Inputs*	*Output*
e_1	HPC_1	$R_{01}, R_{03}, R_{12}, R_1, C_1$	S_f, p_2, p_3	p_1
e_2	HPC_2	R_{12}, R_2, C_2	p_1	p_2
e_3	HPC_3	$R_{01}, R_{03}, R_{34}, R_3, C_3$	S_f, p_1, p_4	p_3
e_4	HPC_4	R_{34}, R_4, C_4	p_3	p_4

Table 4: Hybrid parameter estimators found for the four-tank system, and their related HPCs.

The basic idea is to use the estimator e_{hpc_k} to compute estimations for $\hat{y}_{hpc_k}$ with different values of the parameter θ_f, so that we can find a value of the parameter that minimizes the least squares (LS) error between the estimation $\hat{y}_{hpc_k}$ and the measured value y_{hpc_k}.

Fig. 5 shows the parameter estimation process using the hybrid estimators. A parametrized estimator, e_{hpc_k}, uses the inputs of the system, u_{hpc_k}, and a parameter value, θ_f, to generate an estimation of the output, $\hat{y}_{hpc_k}$. This estimated output is compared against the observed output, y_{hpc_k}, by the quadratic error calculator block. This block computes the quadratic error between $\hat{y}_{hpc_k}$ and y_{hpc_k} for the fault candidate f, E_f^2. Then, the iteration engine block, that contains a nonlinear optimization algorithm, finds the minimum of the error surface $E_f^2(\theta_f)$, by iteratively invoking the estimator with different parameter values. The value of the parameter and its minimum LS error will be the output of the parameter estimation block (and the input for the decision procedure block).

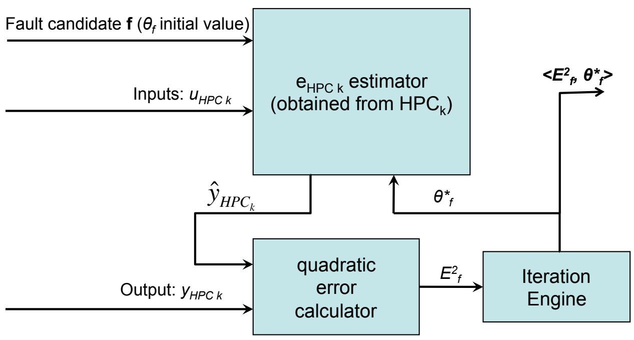

Figure 5: Parameter estimation using the hybrid estimators from HPCs.

7 Results

To test the validity of the approach, we implemented the four hybrid HPCs for the four-tank system, with its corresponding estimators, and run different simulation experiments.

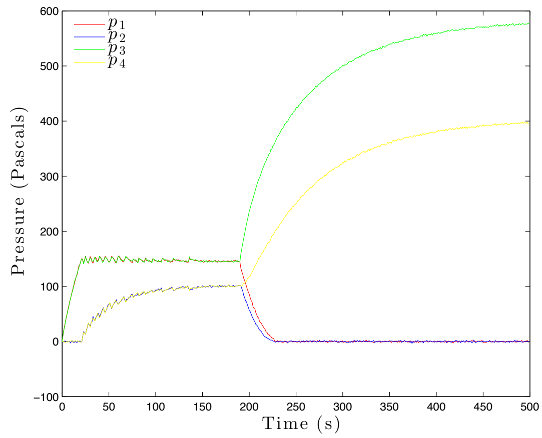

Figure 6: Measured pressures in the four tanks when a fault in SW_1 is introduced at $t = 190$ s.

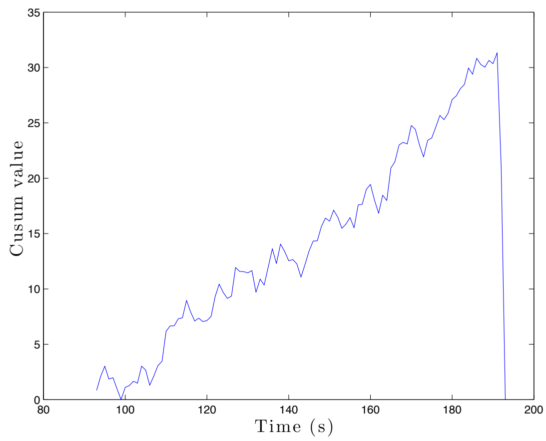

Figure 7: CUSUM output for a fault in SW_1.

In the first experiment, we assume that the water tanks are initially 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 time 20 s level in both tanks reach the height of the connecting pipes and tanks 2 and 4 start to fill in. At time 190 s, a fault occurs in the controlled junction SW_1, which switches off (see Fig. 6 for the measured pressures in the four tanks for this experiment).

Four seconds after the fault is introduced, at $t = 194$ s, both HPC_1 and HPC_3 trigger, and consequently both SW_1 or SW_3 are initially considered as discrete fault candidates. At this point, the CUSUM algorithm is run, determining that the fault has occurred at $t = 191$ s. In this case study we use a CUSUM window of size 100. Figure 7 shows the output of the CUSUM algorithm where the absolute maximum represents the approximate time (due to noise in the system) of fault occurrence.

Once the point of fault occurrence has been determined at $t = 191$ s, the diagnosis framework takes the values of the simulation at such time instant and launches two parallel diagnosis experiments, one for each hypothesized fault candidate, i.e., $SW_1(10)$ and $SW_3(10)$. Figs. 8 and 9 show the

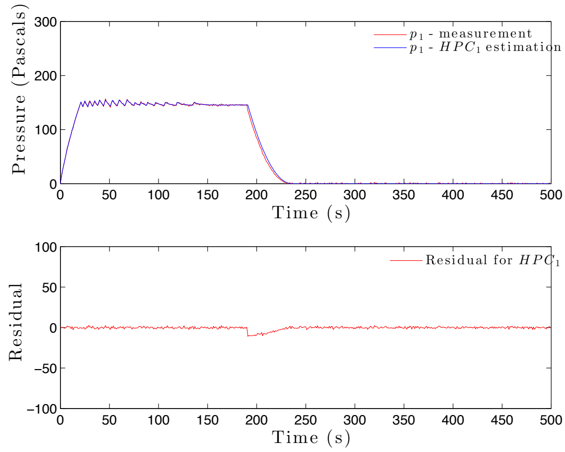

Figure 8: Estimation and residual for HPC_1 (using CUSUM) when a fault in SW_1 occurs and the hypothesized fault is SW_1.

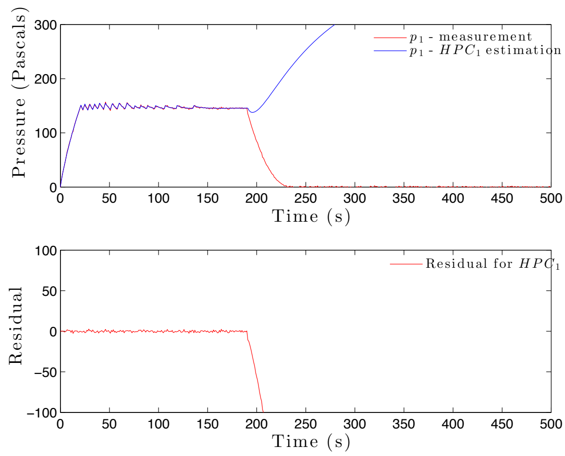

Figure 9: Estimation and residual for HPC_1 (using CUSUM) when a fault in SW_1 occurs and the hypothesized fault is SW_3.

estimation and the residual for HPC_1 when the hypothesized faults are $SW_1(10)$ and $SW_3(10)$, respectively (we do not show the result for HPC_3 since are similar to the results obtained for HPC_1). Looking at the results, it is obvious that the residual converges to zero when a fault in $SW_1(10)$ is hypothesized, while the residual when $SW_3(10)$ is hypothesized does not converge. Hence, $SW_1(10)$ is confirmed as the fault. This confirmation is done by continuously analyzing residual signals with the Z-test. Please note that, since the CUSUM algorithm gives a good approximation of the point of failure, the residual is able to converge very quickly when the true fault is hypothesized. For comparison purposes, Fig. 10 shows the estimation and residual for HPC_1 when CUSUM is not used to re-initialize the simulation (for the hypothesized fault SW_1). By comparing this figure with Fig. 8 it is clear that using CUSUM allows the HPC to converge faster.

As a second diagnosis experiment, we start off from the same situation of the previous experiment, but in this case, we introduce a small parametric fault and after a short while,

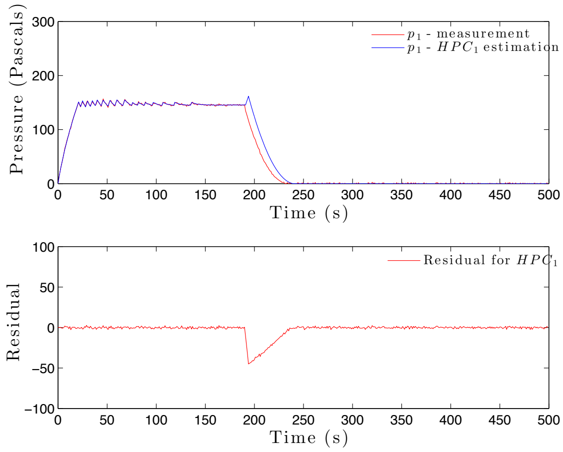

Figure 10: Estimation and residual for HPC_1 (without using CUSUM to re-initialize the simulation) when a fault in SW_1 occurs and the hypothesized fault is SW_1.

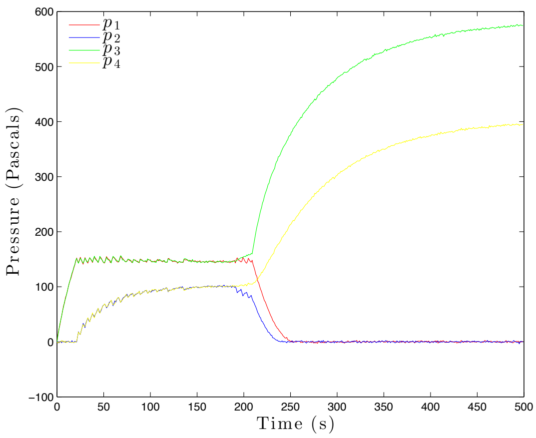

Figure 11: Measured pressures in the four tanks when a fault in R_{01} is introduced at $t = 190$ s and the switching junction SW_1 is turned off at $t = 210$ s.

a discrete change. Specifically, a 20% blockage in the input pipe of tank 1, R_{01}, is introduced at $t = 190$ s, and then SW_1 is commanded to switch OFF at $t = 210$ s (Fig. 11 shows the measured pressures in the four tanks for this experiment).

For this experiment, both HPC_1 and HPC_3 trigger at $t = 198$ s (as an example, see Fig. 12 with the estimation and residual for HPC_1), and consequently both $SW_1(10)$ and $SW_3(10)$ are initially considered as discrete fault candidates. However, in this scenario, after running the CUSUM (see Fig. 13 for the CUSUM output), which estimated the fault time at $t = 191$s, and the diagnosis experiments for both fault candidates, none of the residuals was able to converge within a reasonable, empirically determined, amount of time, thus concluding that a parametric fault has occurred. At this point, the fault identification process is triggered for R_{01}, which is the only parametric fault candidates (R_{03} is discarded due to the qualitative sign in the residuals). The estimated value for parameter R_{01} was 0.1937, i.e., a 19.37% blockage in the pipe. Please note that the estimator

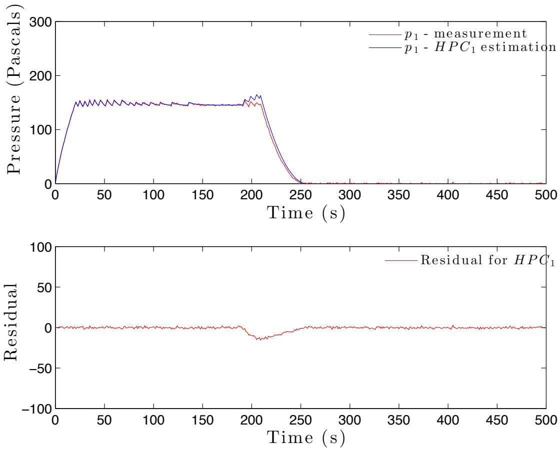

Figure 12: Estimation and residual for HPC_1 when a fault in R_{01} occurs and then SW_1 is set to OFF mode.

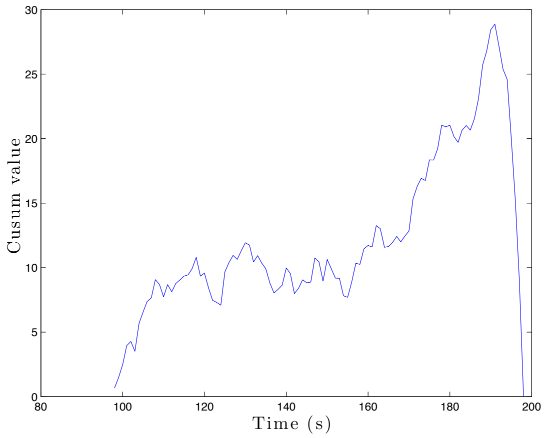

Figure 13: CUSUM output for a fault in R_{01}.

used a total of 60 seconds of data starting from $t = 191$ s, hence, the estimator was capable of correctly estimating the value of the faulty parameter even if the system transitions from one mode to another during the estimation process.

We run several experiments with different mode configurations and different faults, varying the size, time of fault occurrence (in some of them by introducing faults immediately after the mode change). Results for all these situations were equivalent to the examples shown in this section.

8 Conclusions

In this work we have presented an approach for hybrid systems fault identification using Hybrid Possible Conflicts. Using HBGs we can generate minimal estimators that can be used for fault identification just considering the possible mode changes within the estimators. Additionally, we have proposed the integration of the CUSUM algorithm to accurately determine the time of fault occurrence. A more accurate estimation of the fault instant allows to quickly isolate discrete faults, and to obtain a better approximation of the values of the state variables, which are needed as initial values for the fault identification.

Diagnosis results using a four-tank system showed that the proposed approach can be successfully used for fault identification of hybrid systems.

In future work, we will test the approach in more complex systems with real data, and will propose a distributed approach for hybrid systems fault diagnosis.

Acknowledgments

This work has been funded by the Spanish MINECO DPI2013-45414-R grant.

References

[1] P. Mosterman and G. Biswas. Diagnosis of continuous valued systems in transient operating regions. *IEEE Trans. on Sys., Man, and Cyber. Part A*, 29(6):554–565, 1999.

[2] M.W. Hofbaur and B.C. Williams. Hybrid estimation of complex systems. *IEEE Trans. on Sys., Man, and Cyber. Part B*, 34(5):2178 –2191, oct. 2004.

[3] E. Benazera and L. Travé-Massuyès. Set-theoretic estimation of hybrid system configurations. *IEEE Trans. on Sys. Man Cyber. Part B*, 39:1277–1291, October 2009.

[4] S. Narasimhan and L. Brownston. Hyde - a general framework for stochastic and hybrid model-based diagnosis. In *Proc. of the 18th Int. WS on Pples. of Diagnosis, DX07*, pages 186–193, Nashville, TN, USA, May 29-31 2007.

[5] Mehdi Bayoudh, Louise Travé-Massuyès, and Xavier Olive. Fault detection and diagnosis; hybrid systems modeling and control; discrete event systems modeling and control. In *Proc. of the Int. Conference on Control, Automation and Systems, ICCAS08*, pages 7265–7270, Seoul, Korea, October 2008.

[6] S. Narasimhan and G. Biswas. Model-Based Diagnosis of Hybrid Systems. *IEEE Trans. on Sys., Man and Cyber., Part A*, 37(3):348–361, May 2007.

[7] Th. Rienmüller, M. Bayoudh, M.W. Hofbaur, and L. Travé-Massuyès. Hybrid Estimation through Synergic Mode-Set Focusing. In *Proc. of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS09*, pages 1480–1485, Barcelona, Spain, 2009.

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

[9] D.C. Karnopp, D.L. Margolis, and R.C. Rosenberg. *System Dynamics: Modeling and Simulation of Mechatronic Systems*. John Wiley & Sons, Inc., New York, NY, USA, 2006.

[10] A. Bregon, C. Alonso, G. Biswas, B. Pulido, and N. Moya. Fault diagnosis in hybrid systems using possible conficts. In *Proc. of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS12*, Mexico City, Mexico, 2012.

[11] N. Moya, A. Bregon, CJ Alonso-González, and B. Pulido. A Common Framework for Fault Diagnosis of Parametric and Discrete Faults Using Possible Conflicts. In *Advances in Artificial Intelligence, CAEPIA 2013*, volume 8109 of *Lecture Notes in Computer Science*, pages 239–249. Springer-Verlag Berlin, 2013.

[12] E.S. Page. Continuous inspection schemes. *Biometrika*, 41:100–115, 1954.

[13] M. Basseville and I.V. Nikiforov. *Detection of Abrupt Changes: Theory and Applications*. Prentice Hall, 1993.

[14] A. Bregon, G. Biswas, and B. Pulido. A Decomposition Method for Nonlinear Parameter Estimation in TRANSCEND. *IEEE Trans. Syst. Man, Cyber. Part A*, 42(3):751–763, 2012.

[15] N. Moya, B. Pulido, CJ Alonso-González, A. Bregon, and D. Rubio. Automatic Generation of Dynamic Bayesian Networks for Hybrid Systems Fault Diagnosis. In *Proc. of the 23rd Int. WS on Pples. of Diagnosis, DX12*, Great Malvern, UK, Jul-Aug 2012.

[16] D.C. Karnopp, R.C. Rosenberg, and D.L. Margolis. *System Dynamics, A Unified Approach*. 3nd ed., John Wiley & Sons, 2000.

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

[18] A. Bregon, B. Pulido, G. Biswas, and X. Koutsoukos. Generating Possible Conflicts from Bond Graphs Using Temporal Causal Graphs. In *Proc. of the 23rd European Conference on Modelling and Simulation, ECMS09*, pages 675–682, Madrid, Spain, 2009.

[19] A. Bregon, G. Biswas, B. Pulido, C. Alonso-González, and H. Khorasgani. A Common Framework for Compilation Techniques Applied to Diagnosis of Linear Dynamic Systems. *IEEE Trans. on Syst., Man, and Cyb.: Systems*, 44(7):863–873, 2013.

[20] A. Bregon, C. Alonso-González, and B. Pulido. Integration of simulation and state observers for on line fault detection of nonlinear continuous systems. *IEEE Trans. on Syst., Man, and Cyb.: Systems*, 44(12):1553–1568, 2014.

[21] G. Biswas, G. Simon, N. Mahadevan, S. Nararsimhan, J. Ramirez, and G. Karsai. A robust method for hybrid diagnosis of complex systems. In *Proceeding of the 5th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, SAFEPROCESS 2003*, pages 1125–1130, Washington D.C., USA, June 2003.