Fault Detection and Isolation in Critical Infrastructure Systems

Vicenç Puig, Teresa Escobet, Ramon Sarrate and Joseba Quevedo

Advanced Control Systems (SAC), Universitat Politècnica de Catalunya (UPC),
Campus de Terrassa, Rambla Sant Nebridi, 10
08222 Terrassa, Barcelona, Spain
{vicenc.puig,teresa.escobet,
ramon.sarrate,joseba.quevedo}@upc.edu

Abstract. Critical infrastructure systems (CIS) are complex large-scale systems which in turn require highly sophisticated supervisory control systems to ensure that high performance can be achieved and maintained under adverse conditions. The global CIS Real-Time Control (RTC) need of operating in adverse conditions involves, with a high probability, sensor and actuator malfunctions (faults). This problem calls for the use of an on-line Fault Detection and Isolation (FDI) system able to detect such faults. This paper proposes a FDI mechanism that extends the classical Boolean fault signature matrix concept taking into account several fault signal properties to isolate faults in CIS. To exemplify the proposed FDI scheme in CIS, the Barcelona drinking water network is used as a case study.

1 Introduction

Critical infrastructure systems (CIS), such as water, gas or electrical networks, are complex large-scale systems which in turn require highly sophisticated supervisory control systems. CIS are geographically distributed and decentralized with a hierarchical structure. Each subsystem is composed of a large number of elements with time-varying behavior, exhibiting numerous operating modes and subject to changes due to external conditions (e.g., weather) and operational constraints. But, in order to take profit of these expensive infrastructures, it is also necessary to have a highly sophisticated real-time control (RTC) scheme which ensures that high performance can be achieved and maintained under adverse conditions (Schütze et al., 2004; Ocampo et al., 2008). The advantage of RTC applied to CIS has been demonstrated by an important number of researchers during the last decades. Comprehensive reviews that include a discussion of some existing implementations are given by Schilling et al. (1996), Schütze et al. (2004) and Ocampo et al. (2013), and cited references therein, while practical issues are discussed by Schütze et al. (2002), among other. The RTC scheme in CIS might be local or global. When local control is applied, regulation devices use only measurements taken at their specific locations. While this control structure is applicable in many simple cases, in large systems with a strongly interconnected and complex infrastructure of sensors and actuators, it may not be the most efficient alternative. Conversely, a global control strategy is suitable for large scale systems with slow and coupled multivariable dynamic response such as water networks, which computes control actions taking into account real-time measurements all through the network, is likely the best way to use the infrastructure capacity and all the available sensor information.

The global RTC need of operating in adverse conditions involves, with a high probability, sensor and actuator malfunctions (faults) since due to the large scale nature of the systems, an important number of components are involved. This problem calls for the use of an on-line fault detection and isolation (FDI) system able to detect locally such faults, and correct them (if possible) by activating fault tolerant control (FTC) mechanisms. FTC techniques prevent the global RTC system from stopping every time a fault occurs by using techniques such as virtual sensors/actuators or retuning of the controller,

The FDI process aims at carefully identifying which fault (including hardware or software faults, and malicious attacks) can be hypothesized to be the cause of some monitored events. In general, when addressing the FDI problem, two approaches can be found in the literature: hardware redundancy based on the use of redundancies (adding extra sensors and actuators), and software (or analytical) redundancy based on the use of software/intelligent sensors (or model) combining information provided by sensor measurements or using other actuators to compensate a faulty actuator. In CIS, hardware redundancy is preferred. However, for large-scale systems, the use of hardware redundancy is very expensive and increases the number of maintenance and calibration operations. This is the reason why, in CIS applications, systems that allow combining both hardware and analytical redundancy (Carrozza, 2008) must be developed.

This paper proposes a FDI mechanism that extends the classical Boolean fault signature matrix (FSM) concept taking into account several fault signal properties to isolate faults in CIS. To exemplify the proposed FDI scheme in CIS, the Barcelona drinking water network is used as a case study.

2 Proposed Methodology

2.1 Foundations

The proposed FDI procedure is based on checking the consistency between the observed and the normal system behavior using a set of analytical redundancy relations, which relate the values for measured variables according to a normal operation (fault-free) model of the monitored system. When some inconsistency is detected, the fault isolation mechanism is activated in order to identify the possible fault.

The design of a model-based FDI system is based on utilizing the CIS mathematical model (that is obtained from the constitutive elements and their basic relationships) to build a set of consistency tests that only involve observed variables, known as Analytical Redundancy Relations (ARRs). A convenient description of the mathematical model of a CIS regarding FDI is by means of the following discrete-time model:

$$\begin{aligned} x_{k+1} &= g\left(x_k, u_k, \theta_k\right) + w_k \\ 0 &= f\left(x_k, u_k, \theta_k\right) + \eta_k \\ y_k &= h\left(x_k, u_k, \theta_k\right) + v_k \end{aligned} \tag{1}$$

where: $x \in \mathbb{R}^{n_x}$ is the vector of system states, $u \in \mathbb{R}^{n_u}$ is the vector of control actions and $y \in \mathbb{R}^{n_y}$ is the vector of system outputs; $\theta_k \in \mathbb{R}^{n_\theta}$ is a vector of uncertain parameters; $w_k \in \mathbb{R}^{n_w}$ and $\eta_k \in \mathbb{R}^{n_\eta}$ are unmodelled dynamics and disturbances and ;$v_k \in \mathbb{R}^{n_v}$ are measurement noises; $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ and $h : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ are the state-space and measurement nonlinear functions, respectively; and f is the nonlinear static relation function.

To obtain ARRs for state space representation such as (1), it is necessary to manipulate the model to eliminate unobserved variables (i.e., the state *x*).

As it has been defined in Cordier et al. (2004), an ARR is a constraint derived from the system model which contains only observed variables, and which can therefore be evaluated from any observation obtained from measurements provided by the installed sensors. The evaluation of an ARR is denoted as *r* and is called the residual of the ARR. In ideal conditions (no uncertainty and no noise), r=0 in a non-faulty situation, while r≠0 otherwise. Thus, residual *r* is the basis for fault detection.

Given the model defined in (1) with observed variables y_k and u_k, consistency tests can be derived from an ARR by generating a computational residual in the following way:

$$r_i = \Psi_i\left(y_k, u_k\right) = 0 \tag{2}$$

where Ψ_i is called the residual ARR expression. The set of ARR can be represented as

$$\mathcal{R} = \left\{ r_i = \Psi_i\left(y_k, u_k\right) = 0, i = 1, \cdots, n_r \right\} \tag{3}$$

where n_r is the number of obtained ARRs.

In CIS, these ARRs can be efficiently derived applying structural analysis techniques. The analysis of the model structure has been widely used in the area of model-based diagnosis (Blanke et al., 2006). A structural model of a system is an abstraction of the analytical model where only the relation between variables and equations is taken into account, neglecting the mathematical expression of this relation. The diagnosis analysis based on structural models is performed by means of graph-based methods which have no numerical problems and are more efficient, in general, than analytical methods. In (Sarrate et al., 2014), a structural model of a water distribution network is obtained for FDI system design. See (Rosich et al., 2012) and (Travé-Massuyés et al, 2006) for a comprehensive description of ARR design methodologies based on structural analysis.

2.2 Fault Detection

In the literature, there are different approaches to solve this problem. For example, statistical decision methods (Basseville and Nikiforov, 1993) can be used when unknown dynamics and measurement noise are stochastically modeled. In many practical situations, this assumption is not realistic, being more natural to assume that disturbances/model errors and measurement noise are bounded and their effect is propagated to the residuals using, for example, interval methods (Puig et al., 2008). Taking into account bounded uncertainties, the residual of the ARR (2) is monitored by evaluating an interval:

$$\left[r_i\right]=\left\{r_i \middle| r_i=\Psi_i\left(y_k,u_k,\delta_k\right),\delta_k\in D\right\} \tag{5}$$

where D is the interval box $D=\left\{\delta\in\mathbb{R}^{n_\delta}\middle|\underline{\delta}\le\delta\le\overline{\delta}\right\}$, that includes all the bounded uncertainties. Fault detection is formulated as ARR consistency checking using a set-membership approach (Tornil-Sin et al., 2012).

Given a system described by (3) and a sequence of measured inputs u_k and outputs y_k of the real system at time *k*, an ARR is consistent with those measurements and the known bounds of uncertain parameters and noise if there exists a set of sequences $\delta_k \in \mathrm{D}$ which satisfies the ARR.

Given a sequence of observed inputs u_k and outputs y_k of the real system, a fault is said to be detected at time *k* if there does not exist a set of sequences $\delta_k \in \mathrm{D}$ to which the set of ARRs is consistent.

Based on interval reasoning, a fault is detected when $0\notin\left[r_i\right]$ where $\left[r_i\right]$ is defined in (5) . The information provided by the consistency checking is stored as fault signal $\phi_i\left(k\right)$:

$$\phi_i(k) = \begin{cases} 0 & if\ 0 \in [r_i] \\ 1 & if\ 0 \notin [r_i] \end{cases} \tag{6}$$

From computation point of view (6) are generated as $r(k) = y(k) - \hat{y}(k,\delta)$, where $\hat{y}(k,\delta)$ is the estimated value of the output obtained from (1), using for example parity equations or observers.

2.3 Fault isolation

While a single residual is sufficient to detect faults, a set (or a vector) of residuals is required for fault isolation (Gertler, 1998). Once the j^{th} residual has been generated, it is evaluated in order to detect normal or abnormal behaviors. In general, a fault *f* affects a subset of ARRs, $R_f \subseteq \mathcal{R}$.

In model based FDI, the fault effects on the residual can be expressed in terms of the residual fault sensitivity that leads to the residual internal form (Gertler, 1998). For example, in the case of residual r_1 is affected by faults f_1 and f_2, the internal form can be expressed as follows

$$r_1(k) = S_{f_1}(q^{-1})\, f_1(k) + S_{f_2}(q^{-1})\, f_2(k) \tag{7}$$

where, $S_{f_1}(q^{-1})$ and $S_{f_2}(q^{-1})$ are the residual fault sensitivity transfer functions that characterize the fault effect on the residual and q^{-1} is the delay operator of discrete time models.

The fault isolation module proposed in this paper is a generalization to a CIS of the one used in Puig et al. (2005) (see Figure 1). The first component is a memory that stores information on the fault signal occurrence history and it is cyclically updated by the fault detection module. The pattern comparison component compares the memory contents with the stored fault patterns. The classical Boolean fault signature matrix concept (Gertler, 1998) is generalized by extending the binary interface to take into account more fault signal properties. The last component represents the decision logic part of the method whose aim is to propose the most probable fault candidate.

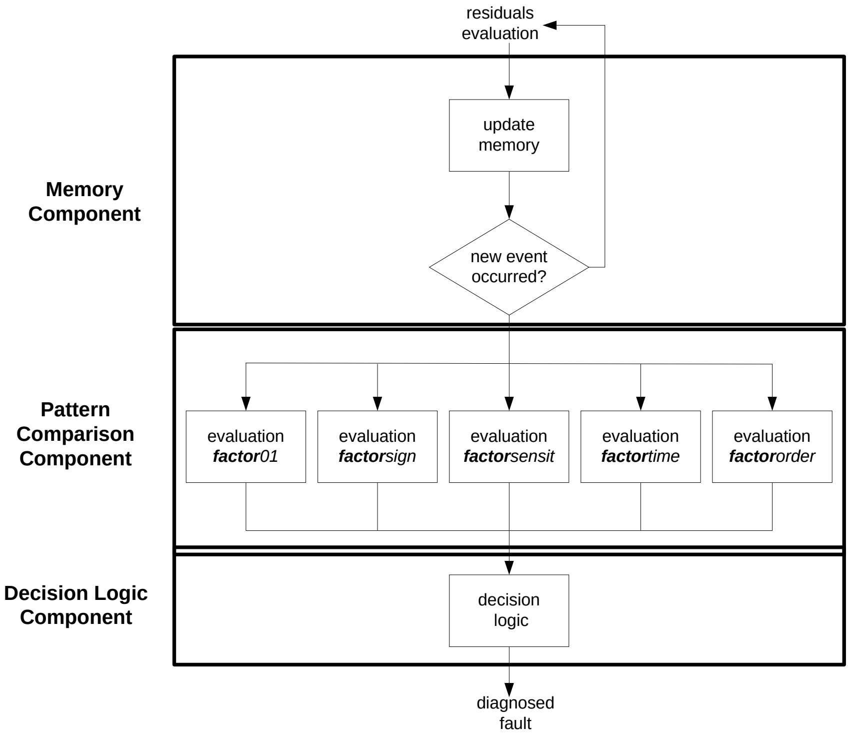


Figure 1. Fault detection and isolation logic scheme.

2.3.1Memory component

The memory component consists of a table in which events in the residual history are stored. When $\phi_i = 1$, the occurrence time, identified by k_o, is stored in the first column; the maximum nominal residual $r_{i,\max}$ is stored in the second column and computes as follow:

$$r_{i,\max} = \max_{k\in[k_o,k_o+T_w]}\left(\left|r_i^o(k)\right|\right) \tag{9}$$

where r_i^o is computed according to (5) considering the center of the uncertainty interval δ_o ; and, the *sign* of the residual is stored in the last column. If the fault detection component detects a new fault signal, the memory is updated by filling out all those fields. The problem of different time instant appearances of the fault signal $\phi_i(k)$ is solved by disabling the isolation decision until a prefixed waiting time T_w has elapsed from the first fault signal appearance. This T_w is calculated from the larger transient time response from a non-faulty situation to any faulty situation. After this time has elapsed, a diagnosis is proposed and the memory component is reset in order to be ready to start the diagnosis of a new fault. Following the approach of Combastel et al. (2003), inside this diagnosis time window, the maximum activation value of the memory-table $r_{i,\max}$ at time k_0 and for one residual i changes only if the current nominal residual is superior to the previous ones. Due to the max-operator activation values can only rise. Using this strategy the effect of noise and non-persistence fault indicators are filtered because just the activation peaks are stored. The memory table makes the residual history accessible for later computation by explicitly storing that data. In this way, temporal aspects of fault isolation can be handled in a very easy and straightforward way.

2.3.2 Pattern comparison component

The pattern comparison component compares the memory contents with the stored fault patterns. Fault patterns are organized according to a theoretical ***FSM***. This interpretation assumes that the occurrence of f_j is observable in r_i , hypothesis known as *fault exoneration* or no *compensation*, and that f_j is the only fault affecting the monitored system. Five different fault signature matrices are considered in the evaluation task: Boolean fault signal activation (***FSM***01), fault signal signs (***FSM****sign*), fault residual sensitivity (***FSM****sensit*), and, finally, fault signal occurrence order (***FSM****order*) and time after the first residual is activated (***FSM****time*). Theses matrices can be obtained from the analysis of residual fault sensitivity (8). Details on the general rules to obtain those matrices from (8) can be found in Meseguer et al. (2010).

2.3.3 Decision logic component

The decision logic algorithm starts when the first residual is activated (that is, $\phi_i = 1$) and lasts T_w time instants or till all fault hypotheses except one are rejected because they do not fulfill the observed residual activation order/time or because an unexpected activation signal has been observed according to those fault hypotheses. Rejection is based on using the results of ***factor01****$_j$*, ***factor****sign$_j$* and ***factor****order$_j$*. If any of these factors is ’zero’ for a given fault hypothesis, it will be rejected. Every factor, with a range of [0,1], represents some kind of a filter, suggesting a set of possible fault hypotheses. At the end of the time window T_w, for each non-rejected fault hypothesis, a fault isolation indicator is calculated using ***factor****sensit$_j$* and ***factor****time$_j$* factors. Thus, the biggest fault isolation indicator will determine the diagnosed fault. The fault isolation indicator associated to the fault hypothesis f_j is determined as it follows:

$$d_j = \max(\left|\boldsymbol{factor}\textit{sensit}_j\right|, \boldsymbol{factor}\textit{time}_j) \qquad (10)$$

So, the final diagnosis result can be expressed as a set of fault candidates with their associated fault isolation indicator.

3 Application to the Barcelona Water Transport Network

3.1 Description of network

The Barcelona water network supplies water to approximately 3 million consumers, distributed in 23 municipalities in a 424 km^2 area. Water can be taken from both surface and underground sources. The most important ones in terms of capacity and use are Ter, which is a surface source, and Llobregat, where water can be taken from one surface source and one underground source. Water is supplied from these sources to 218 demand sectors through around 4645 km of pipes. The complete transport network has been modeled using: 63 storage tanks, 3 surface sources and 7 underground sources, 79 pumps, 50 valves, 18 nodes and 88 demands. The network is controlled through a SCADA system (Figure 2) with sampling periods of 1 hour. For the predictive control scheme a prediction horizon of 24 h is chosen. This record is updated at each time interval.

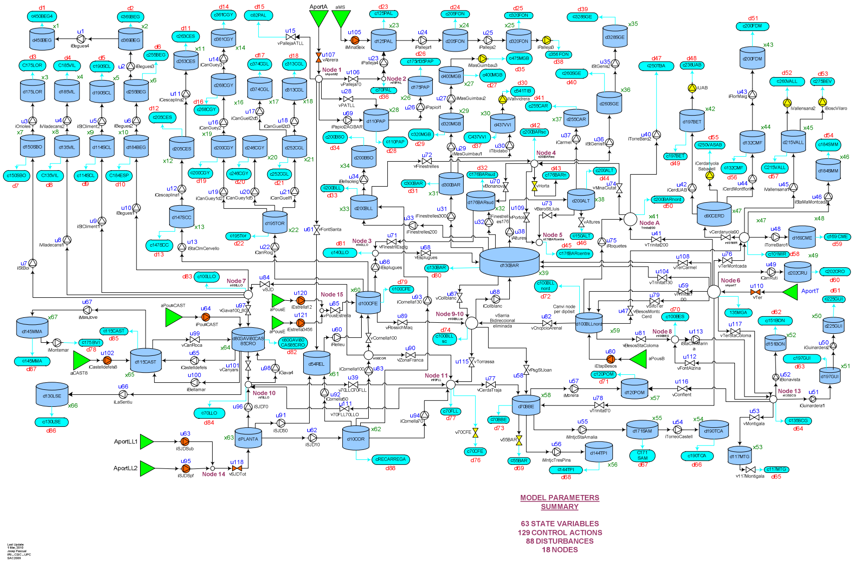


Figure 2. Barcelona water transport network description

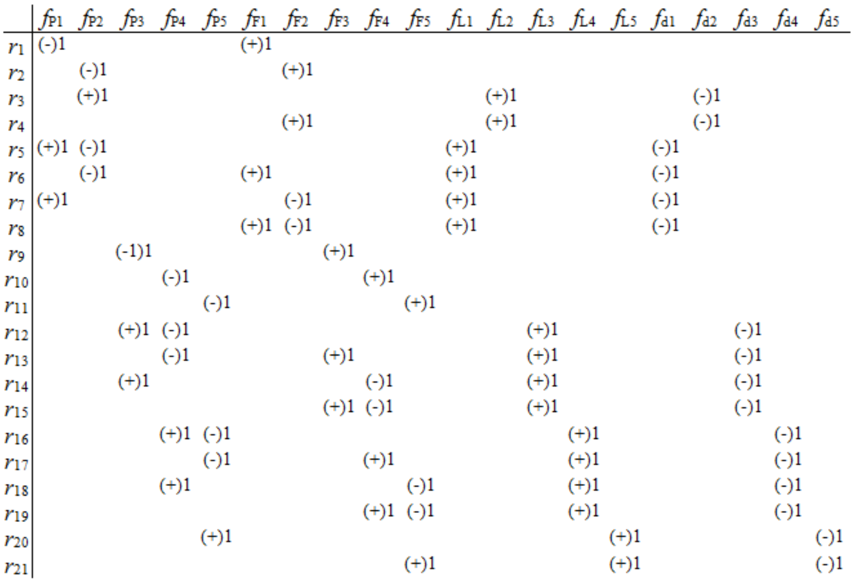
3.2 FDI in the Barcelona water network

The case study used to illustrate the FDI methodology proposed in this paper is based on part of this network. It includes two subsystems, known as Orioles and Cervello. This part of the network includes the following elements:

- Tanks: d150SBO, d175LOR, d147SCC, d205CES, d263CES
- Actuators with sensor flows: iStBoi, iOrioles, iStaClmCervello, iCesalpina1
- Demands with sensor flows: c157SBO, c175LOR, c147SCC, c205CES, c263CES
- Sensor levels: d150SBO, xd175LOR, xd147SCC, xd205CES, xd263CES

This case study can be modeled by the system described by (1), with a 5-dimentional state space vector where each x_i is the i^{th} tank level, $q_{in,i}$ and $q_{out,i}$ are the input and output tank flows, and d_i is the demand. The set of known variables is $\mathcal{O} = \{u_i, y_j\}$ for i=1,..,5 and j=1,..,15, where u_i are the actuator command variables and y_j concerns all measured variables , including the sensors described above.

Applying the algorithm proposed by (Travé-Massuyés et al, 2006), 21 ARRs have been obtained. From these ARRs, the same number of residuals can be generated. Considering faults in the actuators, f_{Pi}, flow transducers, f_{Fi}, level transducers, f_{Li}, and demand transducers f_{di}, for i=1,…,5, the fault signature matrix shown in Figure 3 is obtained. This fault signature matrix includes binary and sign information.



	f_{P1}	f_{P2}	f_{P3}	f_{P4}	f_{P5}	f_{F1}	f_{F2}	f_{F3}	f_{F4}	f_{F5}	f_{L1}	f_{L2}	f_{L3}	f_{L4}	f_{L5}	f_{d1}	f_{d2}	f_{d3}	f_{d4}	f_{d5}
r_1	(-)1					(+)1														
r_2		(-)1					(+)1													
r_3		(+)1										(+)1					(-)1			
r_4							(+)1					(+)1					(-)1			
r_5	(+)1	(-)1									(+)1					(-)1				
r_6		(-)1				(+)1					(+)1					(-)1				
r_7	(+)1						(-)1				(+)1					(-)1				
r_8						(+)1	(-)1				(+)1					(-)1				
r_9			(-1)1					(+)1												
r_{10}				(-)1					(+)1											
r_{11}					(-)1					(+)1										
r_{12}			(+)1	(-)1									(+)1					(-)1		
r_{13}				(-)1				(+)1					(+)1					(-)1		
r_{14}			(+)1						(-)1				(+)1					(-)1		
r_{15}								(+)1	(-)1				(+)1					(-)1		
r_{16}				(+)1	(-)1									(+)1					(-)1	
r_{17}					(-)1				(+)1					(+)1					(-)1	
r_{18}				(+)1						(-)1				(+)1					(-)1	
r_{19}									(+)1	(-)1				(+)1					(-)1	
r_{20}					(+)1										(+)1					(-)1
r_{21}										(+)1					(+)1					(-)1

Figure 3. Theoretical fault signature matrix FSM using binary and sign information

If just binary information is considered, all faults are detectable, but only f_{Pi} and f_{Fi} are isolable. For instance, faults {f_{Li}, f_{di}} can not be isolated because both can not observed independently. But if *sign* information is taken into account, both can be distinguished. Moreover, notice that the information provided by both sensors, {f_{Li}, f_{di}} is essential for computing residuals because there is not enough redundancy, Thus, they can be considered as critical sensors. A fault in one of these sensors modifies the

ARR sets, resulting to an undetectable fault. The fault detection and isolation procedure described in Section 2 has been applied in a simulation case. Figure 4 shows the first 8 ARR residuals and fault signal evolution when a drift in sensor iOrioles flow, f_{F2}, is introduced at hour 362. Notice that residuals r_2, r_4, r_7 and_r_8 are non-consistent, indicating as potential fault $\{f_{P2}, f_{F1}, f_{F2}, f_{L1}, f_{L2}, f_{d1}, f_{d2}\}$.

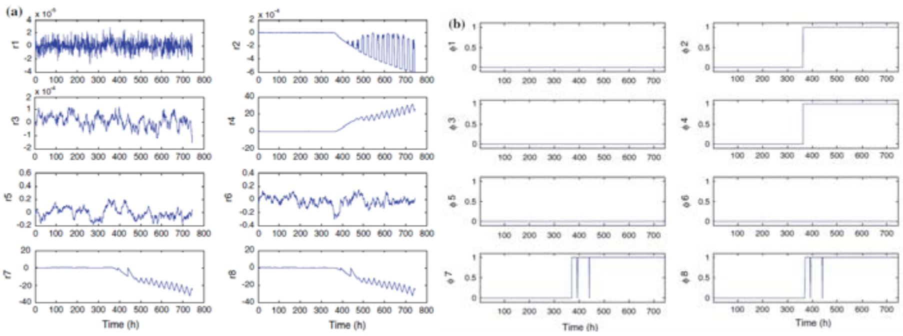


Figure 4. (a) Residuals and (b) fault signal evolution with a drift fault in sensor iOrioles flow.

The time evolution of *factor*01 and *factor*sign are plotted at every time instant in Figure 5. It can be seen that both factors indicate as a maximum fault hypothesis f_{P2}, with $d_{P2} = 1$ (10), There are also others activated factors but with a smaller indication magnitude. In this example, the time needed for detection and isolation is of two sampling times.

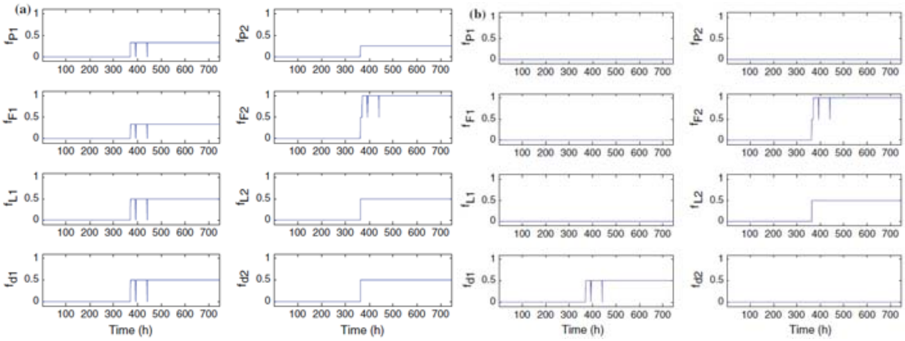


Figure 5. Fault signal analysis based on (a) *factor*01 and (b) *factor*sign.

4 Conclusions

CIS are complex large-scale systems which in turn require highly sophisticated supervisory-control systems to ensure that high performance can be achieved and maintained under adverse conditions. The global RTC need of operating in adverse conditions involve, with a high probability, sensor and actuator malfunctions (faults). This problem calls for the use of an on-line FDI system able to detect such faults and correct them (if possible) by activating fault tolerant mechanisms. The proposed FDI mechanism extends the classical Boolean fault signature matrix concept taking into account several fault signal properties to isolate the faults in CIS. To exemplify the FDI methodologies in CIS, the Barcelona drinking water network is used as the case study.

References

Basseville, M., Nikiforov, I. *Detection of Abrupt Changes: Theory and Applications*. Prentice Hall. 2003.

Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M.(2006). *Diagnosis and Fault-Tolerant Control*. Springer, 2nd edition.

Carrozza, G., Cotroneo, D. and Russo, S. (2008). Software Faults Diagnosis in Complex OTS Based Safety Critical Systems. *Proceedings of Seventh European Dependable Computing Conference*, 25-34.

Combastel, C., S. Gentil, and J. P. Rognon (2003). Toward a better integration of residual generation and diagnostic decision. *Proceedings of IFAC Safeprocess'03*, Washington, USA.

Cordier, M.-O., Dague, P., Levy, F., Montmain, J., Staroswiecki, M., Trave-Massuyes, L. (2004). Conflicts versus analytical redundancy relations: a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control perspectives, *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* , 34(5), 2163-2177.

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

Meseguer, J., Puig, V., Escobet, T. (2010) "Fault Diagnosis Using a Timed Discrete-Event Approach Based on Interval Observers: Application to Sewer Networks". *IEEE Transactions on Systems, Man and Cybernetics: Part A*, Volume 40(5), pp. 900-916

Ocampo-Martínez, C. and Puig, V Fault-tolerant model predictive control within the hybrid systems framework: Application to sewer networks. *International Journal of Adaptive Control and Signal Processing*, 23(8): 1099-1115, 2008.

Ocampo-Martínez, C., Puig, V., Cembrano, G. and J. Quevedo, Application of predictive control strategies to the management of complex networks in the urban water cycle, *IEEE Control Systems Magazine*, vol. 33, no. 1, pp. 15–41, 2013.

Puig, V. J. Quevedo, T. Escobet and B. Pulido (2005). A New Fault Diagnosis Algorithm that Improves the Integration of Fault Detection and *Isolation. Proceedings of ECC-CDC'05,* Sevilla, Spain.

Puig, V., Quevedo, J., Escobet, T., De las Heras, S. (2008) "Passive Robust Fault Detection of Dynamic Processes Using Interval Models". *IEEE Transactions on Control Systems Technology*, Volume 16, Issue 5, pp. 1083 – 1089.

Rosich, A., Frisk, E., Åslund, J., Sarrate, R., and Nejjari, F. (2012). Fault diagnosis based on causal computations, *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 42(2), 371-381.

Sarrate, R., Blesa, J., and Nejjari, F. (2014). Sensor placement for leak detection and location in water distribution networks, *Water Science & Technology: Water Supply* , (in press), doi:10.2166/ws.2014.037.

Schilling, W., Anderson, B., Nyberg, U., Aspegren, H., Rauch, W., and Harremoës, P. (1996). Real-time control of wasterwater systems. *Journal of Hydraulic Resourses*, 34(6), 785–797.

Schütze, M., Butler, D., and Beck, B. (2002). *Modelling, Simulation and Control of Urban Wastewater Systems*. Springer.

Schütze, M., Campisanob A., Colas, H., W.Schilling, and Vanrolleghem, P. (2004). Real time control of urban wastewater systems: Where do we stand today? *Journal of Hydrology* **299**, 335–348.

Tornil-Sin, S.; Ocampo-Martinez, C.; Puig, V.; Escobet, T. (2014), "Robust Fault Diagnosis of Nonlinear Systems Using Interval Constraint Satisfaction and Analytical Redundancy Relations," *IEEE Transactions on Systems, Man, and Cybernetics: Systems-Part B,* , vol.44, no.1, pp.18,29,

Travé-Massuyès, L., Escobet, T. and Olive, X. (2006). Diagnosability analysis based on component supported analytical redundancy relations. *IEEE Transactions on Systems, Man, and Cybernetics-Part A,* 36(6):1146–1160.