Diagnosis approaches for detection and isolation of cyber attacks and faults on a two-tank system
Elodie Chantery, Audine Subias

To cite this version:
Elodie Chantery, Audine Subias. Diagnosis approaches for detection and isolation of cyber attacks and faults on a two-tank system. 30th International Workshop on Principles of Diagnosis DX’19, Nov 2019, Klagenfurt, Austria. hal-02439489

HAL Id: hal-02439489
https://laas.hal.science/hal-02439489
Submitted on 14 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Diagnosis approaches for detection and isolation of cyber attacks and faults on a two-tank system

Elodie Chantery and Audine Subias
LAAS-CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France
e-mail: [elodie.chanthery;audine.subias]@laas.fr

Abstract
Recently, a two-tank benchmark for detection and isolation of cyber-attacks has been proposed to the diagnosis research community to test different diagnosis methods. In this work, we use this benchmark, add some scenarios, and test several diagnosis techniques to evaluate their diagnosability power. We propose to use a well-known model-based diagnosis approach to identify the groups of faults and attacks that are not isolable or detectable. In a second part, we investigate improvements provided by data-based diagnosis techniques and show that they succeed to isolate faults and attack for this benchmark.

1 Introduction
Connected systems and intelligent systems are largely widespread in the majority of industrial domains (production, transport, networks, aeronautics, space,...). Critical systems as power plants, water plants, smart grids also rely heavily on these Information and Communication Technologies (ICT) to control, to monitor and to manage remotely several equipments. If ICT has contributed to improve the control and monitoring of installations it has also exposed critical systems to a new kind of threats by introducing a high degree of connectivity. Consequently, these systems which include both computational resources, communication capabilities and hardware also called cyber-physical systems, are exposed to malicious attacks, known as cyber-attacks. As in any computer system connected to internet the well known CIA (Confidentiality, Integrity, and Availability) triad of a cyber-physical system can be threatened by non-targeted attacks, but it can also fall prey of targeted attacks. In this case the hacker knows that he is targeting a control system and the cyber-attack is a malicious act aimed at degrading or interrupting the operation of an industrial installation or even destroying it [1]. These attacks exploit the vulnerabilities of the system and can have a very important impact not only from a process point of view but also from financial and environmental ones. Obviously, the influence the hacker can have on the system depends on his knowledge about the system. A prominent example is the attack on the water distribution system in Queensland, Australia [2]. The hacker used a laptop and took the control of 150 sewage pumping stations to release one million liters of untreated sewage into a storm-water drain connected to local waterways. Another example is the Stuxnet attack that taking the control of actuators and sensors struck the Iranian nuclear facility causing the failure of the centrifuges. More recently, we can cite the cyber-attack against Ukraine power grid that cut power to many customers for about six hours. Histories and descriptions of cyber-attacks can be found in [3][4][5][6].

In the context of a classical closed-loop control system architecture, the problem of attack diagnosis is getting closer to the problem of fault diagnosis. Nevertheless, the attack diagnosis problem addresses a larger spectrum of feared scenarios. Indeed, attacks as faults can concern sensors, actuators or the controller itself but the hacker is endowed with intelligence and has capabilities that multiply the possibilities of attacks. Whatever the attack type, its effect is at least one of the following: the sensor returns incorrect measurements, the controller sends out incorrect signals, the actuator ignores control input or executes incorrect commands. According to [7] two main types of attacks can be distinguished: those that concern data integrity issued from sensors or controllers (called by the authors deception attacks) and those of type Denial of Service (DoS) that threat the signals availability preventing the controller from receiving sensor measurements or the actuators from receiving control commands. In the first case, signals are corrupted by the hacker so that the measurement is incorrect, delayed, re-played etc.

The growing interest in this problem of attack detection and diagnosis is reflected in the Diagnosis community. In [8] an observer based method is developed to diagnose stealthy deception attack that can enable remote water pilfering from automated canal systems. The work of [9] define the notions of detectability and identifiability of an attack by its effect on output. [10] proposes an approach to design an input reconstruction filter to identify the effect of random deception attack on the system behavior in the context of limited access to sensor measurements. In [11] the authors propose a frequency based approach to detect replay attacks on cyber-physical systems. [12] propose an approach based on the introduction of a filter between the controller (Programmable Logic Controller) and the sensors/actuators to detect and prevent orders that could damage the system. Additional information issued from a notion of distance between states allow distinguishing cyber-attacks from classical failures. [13] consider the problem of intrusion detection and mitigation in supervisory control systems. Several types of attacks are considered (en-
Recently a two-tank system well known in the diagnosis community [14] has been extended and proposed to the scientific community to test different approaches of detection and isolation of cyber-attacks. To initiate the work the authors considered a classical model-based approach (Analytical Redundancy Relations) [13]. This paper is an answer to this invitation. The contribution of the paper is to show that they return incorrect measurements, on the actuators to enable/disabling of actuators events, erase/insertion of sensor events).

The attacks can then have some effects on the sensors so that the initial results obtained from Analytical Redundancy Relations can be improved by the use of data-based methods.

This paper is organized as follows. Section 2 presents the two-tank benchmark. Section 3 describes the different existing scenarios of faults and attacks, as well as some new scenarios proposed in this work. Model-based Diagnosis techniques are applied on the two-tank in Section 4 for detecting and isolating faults and attacks. The use of Data-based methods is developed in Section 5. Section 6 concludes the paper.

2 Benchmark Description

The two-tank benchmark is illustrated in Figure 1. The goal of the system is to provide water to customers with a continuous flow \( Q_o \). A valve \( V_o \) simulates the consumer. \( V_o \) is opened in nominal mode.

The system consists in two tanks \( T_1 \) and \( T_2 \) in series. \( h_1 \) and \( h_2 \) are supposed to be the water level in \( T_1 \) and \( T_2 \), respectively. \( T_1 \) is filled by a pump \( P_1 \) controlled by a PI level controller acting on the inlet flow \( Q_{in} \). It is assumed that \( Q_{in} \) is proportional to the PI controller output \( U_{P1}^m \), that is measured. Note that in the article, the letter "m" is added to a variable when it is measured. The water flow between the two tanks is \( Q_{in} \). It can be controlled by an ON/OFF valve \( V_o \). The water levels in the two tanks may be affected by real leaks but also by attacks simulating a leak. \( V_{leak1} \) (resp \( V_{leak2} \)) denotes a leakage in \( T_1 \) (resp. in \( T_2 \)) while \( Q_{leak1} \) (resp. \( Q_{leak2} \)) is an attack. Two sensors provide the measurements of the real water levels of the tanks: \( h_1^m \) (for water level in \( T_1 \)) and \( h_2^m \) (for water level in \( T_2 \)).

A detailed description of the dynamic model of the faultless two-tank system is given in [14]. The input \( u \) of the system is supposed to be known.

\[
\begin{bmatrix}
Q_p^m \\
U_{P1}^m \\
U_o^m \\
U_p^m
\end{bmatrix}
\]

(1)

where \( Q_p^m \) is the measured flow from the pump \( P_1 \), \( U_{P1}^m \) represents the position of valve \( V_{leak1}^m \) (0 if the valve is open, 1 if the valve is closed), \( U_o^m \) is the position of the valve \( V_o \), and \( U_p^m \) corresponds to the PI controller output.

3 Fault and attack scenarios

The next sections presents the faults and attack scenarios implemented in the two-tank system benchmark.

3.1 Fault Scenarios

A set of fault scenarios has been described in [14].

- \( f_0 \) faultless mode: the process runs without fault.
- \( f_1 \) Pump fault: from 40s up to 120s the pump is simulated off (like if it is broken).
- \( f_2 \) Level sensor \( mh_{1} \) stuck at zero fault: from 40s up to 120s the output of the sensor is stuck at zero.
- \( f_3 \) Level sensor \( mh_{2} \) fault: from 40s up to 120s the output of the sensor is stuck at zero.
- \( f_4 \) Leakage fault in Tank \( T_1 \) from 40s up to 120s. \( Q_{f1} = 10^{-4}m^3/s \).
- \( f_5 \) Leakage fault in Tank \( T_2 \) from 40s up to 120s. \( Q_{f2} = 10^{-4}m^3/s \).
- \( f_6 \) Sensor fault for the pump \( P_1 \) flow: the sensor is stuck at zero from 40s to up to 120s.
- \( f_7 \) Valve \( V_b \) stuck-closed fault: \( v_b \) is stuck closed from 40s to up to 150s.
- \( f_8 \) Valve fault: \( U_{P1}^m \) is stuck at zero from 40s to 120s.

Some additional fault scenarios have been defined and tested in this article.

- \( f_9 \) Pump sensor fault: \( Q_p^m \) is stuck at zero from 40s to up to 120s.
- \( f_{10} \) PI controller fault: \( U_{P1}^m \) is always equal to zero from 40s to up to 120s.
- \( f_{11} \) Valve \( V_b \) stuck-open fault: \( v_b \) is stuck open from 40s to up to 150s.
- \( f_{12} \) Level sensor \( mh_{1} \) stuck at non-zero fault: from 40s up to 120s the output of the sensor is stuck at a non-zero value, here \( h_{1}^m = 0.4 \).

3.2 Attack Scenarios

In this article we consider as in [15] that the final objective of an attacker is to steal water but we consider also the case where the aim of the attacker is the destruction of the installation or the degradation of the behaviour. We consider an active attacker who has the ability to altering the cyber-physical system by exploiting system security design holes.

The attacks can then have some effects on the sensors so that they return incorrect measurements, on the actuators to execute incorrect control actions but also on the controller to perform a physical action on the two-tank system.

We recall here the set of attack scenarios defined in [15].
Short-term water theft from $T_1$: this scenario is similar to the scenario with fault $f_3$. The difference is that it is cast maliciously, with the purpose of stealing water from $T_1$, $V_f = 10^{-4} m^3/s$ from 40s up to 80s.

Short-term water theft from $T_1$ with hiding signal added to the measurement from 40s up to 80s. The theft is hidden by adding a signal to the output of the level sensor in tank $T_1$. The PI controller works as if nothing had happened.

Long-term water theft from $T_1$ with hiding signal added to the measurement: the attack is the same as $a_2$ but its duration is extended so that Tank $T_1$ might become empty, affecting the tank $T_2$ and $Q_0$ due to interconnection.

Long-term water theft from $T_1$ with small signal added to the measurement: the attack is the same as $a_3$ but the added signal only compensates 50% of the stolen water, so that the pump will compensate half the theft and the attack will be harder to detect.

Short-term water theft from $T_2$ (same as $a_1$) $V_f = 10^{-4} m^3/s$ from 40s up to 80s.

Short-term water theft from $T_2$ with hiding signal added to the measurement from 40s up to 80s (see $a_2$).

Long-term water theft from $T_2$ with hiding signal added to the measurement (see $a_3$).

Short-term water theft from $T_2$ with small signal added to the measurement (see $a_4$).

Replay attack from 160s to up to 200s: the hacker records the measurements coming from the sensors without stealing water from the tanks. Then, when the system has reached its steady-state, the hacker steals water while replacing the sensor values by the recorded ones.

A set of five new attacks has been added to this initial set as follows:

Overflow with sensor hack from 40s up to 120s: the objective is not to steal water but to destroy the system. The value of $h_{3w}^m$ is modified to simulate a fake leakage, so that the pump compensates and causes a water overflow in $T_1$.

Overflow with controller hack from 40s up to 120s: the hacker disconnects the PI controller and sets a too high hydraulic flow coefficient of the valve $V_0$, $A_{i_{(i=1,2)}}$ is the cross-section of the cylindrical tank $T_i$, $C_{w}$ the hydraulic flow coefficient of the valve $V_0$, $K_P$ and $K_I$ the coefficients of the PI controller. The numerical values of these parameters are given in [15] and [14].

Two additional residuals have been included in the benchmark, suppose that the pressures in tank $T_1$ and $T_2$ may be measured by two new sensors.

We then defined:

\[ r_{5}(t) = P_{atm}^m(t) - \rho \cdot g \cdot h_{1w}^m(t) - P_{atm} \]

\[ r_{6}(t) = P_{atm}^m(t) - \rho \cdot g \cdot h_{2w}^m(t) - P_{atm} \]

### 4.1 Analytical Redundancy Relations

A set of four Analytical Redundancy Relations (ARR) has already been described in [14] and four residues have been included in the benchmark. We aim to use them for fault detection and isolation, but also for attack detection and isolation.

As a reminder, the ARR are calculated as:

\[ r_1(t) = -C_{ah}V_f^m(t) \sign(h_{1w}^m(t) - h_{1w}^m(0) - h_{2w}^m(t)) \frac{1}{(h_{2w}^m(t) - h_{2w}^m(0))} + Q_{p,\text{max}}^m(t) - A_1 \frac{dh_{2w}^m}{dt} \]

\[ r_2(t) = C_{ah}V_f^m(t) \sign(h_{1w}^m(t) - h_{1w}^m(0) - h_{2w}^m(t)) \frac{1}{(h_{2w}^m(t) - h_{2w}^m(0))} - C_{aw}V_f^m(t) \frac{dh_{2w}^m}{dt} \]

\[ r_3(t) = U_p^m(t) - K_P(h_{1w}^m(t) - h_{1w}^m(0) - K_I \int h_{1w}^m(t) - h_{1w}^m(0) dt) \]

\[ r_4(t) = Q_{p,\text{max}}^m - \begin{cases} \frac{0}{U_p^m(t) \leq 0} & \text{if } U_p^m(t) < Q_{p,\text{max}} \\ Q_{p,\text{max}} & \text{otherwise} \end{cases} \]

with: $C_{ah}$ the hydraulic flow coefficient of the valve $V_0$, $A_{i_{(i=1,2)}}$ is the cross-section of the cylindrical tank $T_i$, $C_{w}$ the hydraulic flow coefficient of the valve $V_0$, $K_P$ and $K_I$ the coefficients of the PI controller. The numerical values of these parameters are given in [15] and [14].

### 4.2 Definition of detectability and isolability for faults and attacks

Let $F = \{f_1, \ldots, f_n\}$ be the set of faults of the system, $A = \{a_1, \ldots, a_m\}$ be the set of attacks. Let $f_0$ correspond to the nominal mode without any fault or attack. We define in this section some fundamental definitions for fault diagnosis and attack detection and isolation.

We recall here the definition of a fault signature matrix and extend it for the attacks.

**Definition 1 (Fault Signature).** Given a set $ARR$ composed of $n$ faults and $F$ the set of considered $n$ faults for the system; consider the function $ARR \times F \longrightarrow \{0, 1\}$, then the signature of a fault $f \in F$ is the binary vector $FS(f) = [\tau_{f_1}, \ldots, \tau_{f_n}]^T$ where $\tau_k = 1$ if $f$ is a variable of the equation used to form $\alpha_k \in ARR$, otherwise $\tau_k = 0$.

The same definition holds for attacks.

**Definition 2 (Attack Signature).** Given a set $ARR$ composed of $n$ faults and $A$ the set of considered $n$ attacks for the system; consider the function $ARR \times A \longrightarrow \{0, 1\}$, then the signature of an attack $a \in A$ is the binary vector $FS(a) = [\tau_{a_1}, \ldots, \tau_{a_n}]^T$ where $\tau_k = 1$ if $a$ is a variable of the equation used to form $\alpha_k \in ARR$, otherwise $\tau_k = 0$.

### 4 Model-based diagnosis techniques on the benchmark

In this article we extend the diagnosis results obtained on the two-tank system based on an Analytical Redundancy Relations [16]. Nevertheless, it would be interesting to investigate other model-based approaches notably to take advantage of the dynamical information embedded in the fault and attack profiles. By considering the hybrid nature of the two-tank system, model based approaches developed in the field of switched systems could be investigated as observer based methods.
We can then define the fault and attack signature matrix as follows:

**Definition 3 (Fault and attack signature matrix FASM).**
The signatures of all the faults in $F$ and of all the attacks in $A$ together constitute the fault and attack signature matrix $FASM$ for the system, i.e $FASM = [FS(f_1), \ldots, FS(f_n), FS(a_1), \ldots, FS(a_m)]^T$.

This matrix obviously implies what is usually called the fault signature matrix and what could be called the attack signature matrix.

From the $FASM$ detectability and isolability properties can be defined for faults and attacks.

**Definition 4 (Detectable fault).** A fault $f \in F$ is detectable in the system if $FS(f) \neq FS(f_0)$.

The same definition can be applied for attacks.

**Definition 5 (Detectable attack).** An attack $a \in A$ is detectable in the system if $FS(a) \neq FS(f_0)$.

**Definition 6 (Isolability).** Two faults or attacks $fa \in F \cup A$ and $fa' \in F \cup A$ are isolable in the system if they are detectable and if $FS(fa) \neq FS(fa')$.

**Definition 7 (Diagnosability group).** Two faults or attacks $fa \in F \cup A \cup \{f_0\}$ and $fa' \in F \cup A \cup \{f_0\}$ are in the same diagnosability group if they are not isolable from each other.

In others words, a group of diagnosability is defined as a set of behaviours (nominal, faulty or attacks behaviours) that can not be isolated two by two as their signatures are identical.

### 4.3 Fault and attack signatures

Some diagnosis results have been provided in [14] but no link between the fault scenarios and the residuals have been clearly established. Here, we aimed at giving the fault and attack signatures for each defined scenario.

Using the four first residuals we succeeded in defining eight diagnosability groups as presented in Table [1].

From Table [1] we can deduce that all faults and attacks are detectable, except the replay attack. Only 3 faults ($f_1, f_6$ and $f_{10}$) are isolable from the others and from the attacks. Attacks are always included in diagnosability groups with at least one fault so it is impossible to distinguish them from faults with the $ARR$ approach.

We also tried to take benefit from the two additional residuals obtained from pressure sensors (see section [4.4]). From Table [2] we can deduce that all faults and attacks are detectable, except the replay attack, as with 4 residuals. The 2 new residuals increase isolability power: for example diagnosability group $G_2$ has been split into 3 other diagnosability groups $G_{21}, G_{22}$ and $G_{23}$, while $G_4$ is divided into 2 diagnosability groups. The only interesting point is that $G_{42}$ only refers to attack scenarios $a_9, a_7, a_8$ and $a_{12}$ so that it is possible to distinguish fault from attack in these cases, moreover $f_3$ is now isolable. However, much ambiguity remains between faults and attacks in 5 diagnosability groups ($G_{21}, G_3, G_4, G_6, G_8$).

### 4.4 Results

An HMI has been provided in the matlab benchmark to visualize online the residuals value and the possible diagnoses. For example, we simulated scenario $f_9$ and found the residuals $r_4$ is equal to 0, then rises to 1 at 41s.

**5 Data-based Diagnosis techniques on the benchmark**

Since model-based methods are not satisfactory, we decided to consider data-based methods. The final goal is the fusion of model-based and data-based methods, as proposed in [17][18]. At first, we would like to check the efficiency of data-based methods to better isolate faults from attacks for simple cases.

The fault diagnosis and attack isolation problem can be formulated as a classical classification problem. The faults and attacks are considered as different classes in the classification problem. Thanks to simulation, we created a large set of data for each scenario and tried to apply common methods of the machine learning literature, focused on supervised learning with a classification goal, each class corresponding to a fault or an attack. These methods are basically Discriminant Analysis [19], K-Nearest Neighbors [20], decision tree or bagged and boosted decision trees [21]. The goal was then to find a predictive model based on both input and output data.

#### 5.1 Features

The two-tank system has mainly 6 observable variables that have been taken as features and that correspond to the sensor data for the state variables (water level in each tank) and to the inputs. The features are then $h_{in}^1, h_{in}^2, Q_{in}, U_{in}^1, U_{in}^2$ and $T_{in}$. Note that the classification methods have been tested without taking into account the additional pressure sensors proposed for deriving residuals $r_4$ and $r_5$.

#### 5.2 Training Set

Each scenario has been run 10 times and each run varies because of noises included in the simulation. Each of them has a duration of 250s with a sampling period of 1s, so that each run corresponds to 250 input data. Each input data has been tagged by one of the 17 simplified tags, defined in Table [3].

These simplified tags have been used because some scenarios seemed to be exactly the same physically. Future work could investigate the search of differences between them.

As all the scenarios already include some "normal" data (for example, from $t=0s$ up to $t=40s$, all the data will be
5.3 Validation Scheme and Performances

The chosen validation scheme is the cross-validation method with 5 folds to partition the data set. The app partitions the data into 5 disjoint sets. For each fold, it trains a model using the out-of-fold observations, then assesses model performance using in-fold data.

Table 4 gives the performance results for the main methods included in the Matlab Classification Learner app.

The data-based methods have excellent results for classifying the different scenarios. Fine-kNN has an accuracy of 97.9% when only one neighbor is considered, while the accuracy is 97.4% when 5 neighbors are considered. The Bagged Trees method has an accuracy of 97.5%.

We show on Figure 3 and 4 the confusion matrices for the Fine-kNN and for the Bagged Trees methods. The confusion matrix gives the percentage of true class among the predicted class. As it is possible to see, the worst percentage of true detection is 78% for the Fine-kNN method and 82% for the Bagged Trees method.

From Figure 3 and 4 we can deduce that all the defined tagged cases can be detected by simple data-based methods. Faults and attacks can be isolated, except $a_9$, $a_1$, and $a_5$. These three attacks will be studied in future works. However, the isolation power of data-based methods is much better than model-based methods. For example, with model-based methods, $G_6$ includes $f_6$ and $a_{11}$ while data-based methods succeed in isolate both cases. The case $f_5 a_5$ remains the most difficult case to isolate from others.

5.4 New Data Prediction

We exported the models deduced from the Fine-kNN method and from the Bagged-Tree method to the workspace of Matlab and used these trained models to make predictions using new data.

A new scenario involving a failure of the pump has been designed, with different flows. From 0 to 40s, the system has no failure and no attack. From 40s to 120s, the pump is faulty, then after 120s the system again in nominal mode. New data have been tested with the predicted models. Figure 5 shows the prediction results for the Fine-kNN method: each new data has been well classified, except during the transition from normal mode to faulty mode where one data is tagged $f_6$ instead of $f_1$. The same problem occurs when using the model trained with Bagged Trees.

Another scenario has been tested, involving an attack corresponding of scenario $a_{13}$. Figure 6 shows the prediction...
<table>
<thead>
<tr>
<th>method</th>
<th>accuracy (%)</th>
<th>prediction speed (obs/sec)</th>
<th>training time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boosted Trees</td>
<td>85.8</td>
<td>21000</td>
<td>88.704</td>
</tr>
<tr>
<td>Bagged Trees</td>
<td>97.5</td>
<td>18000</td>
<td>64.451</td>
</tr>
<tr>
<td>Medium Tree</td>
<td>77.6</td>
<td>480000</td>
<td>2.826</td>
</tr>
<tr>
<td>Fine Tree</td>
<td>88.4</td>
<td>430000</td>
<td>4.4854</td>
</tr>
<tr>
<td>Coarse Tree</td>
<td>71.8</td>
<td>500000</td>
<td>2.45</td>
</tr>
<tr>
<td>Subspace Discriminant</td>
<td>66.7</td>
<td>12000</td>
<td>36.296</td>
</tr>
<tr>
<td>Linear Discriminant</td>
<td>64.6</td>
<td>240000</td>
<td>3.3491</td>
</tr>
<tr>
<td>Fine KNN</td>
<td>97.9</td>
<td>47000</td>
<td>8.5107</td>
</tr>
<tr>
<td>Medium KNN</td>
<td>96.6</td>
<td>36000</td>
<td>7.143</td>
</tr>
<tr>
<td>Coarse KNN</td>
<td>91.2</td>
<td>13000</td>
<td>16.26</td>
</tr>
</tbody>
</table>

Table 4: Performances Results for the main classification included in Matlab

Figure 3: Confusion matrix for the Fine-kNN method
Figure 4: Confusion matrix for Bagged Trees method

Figure 5: Predicted classes on a scenario of type $f_{1}$

Figure 6: Predicted classes on a scenario of type $a_{13}$

results for the Fine-kNN method: the attack is immediately detected and tagged $a_{13,14}$ with a Fine-KNN method, while the model-based methods kept the ambiguity with fault $f_{5}$. The same results are found with the Bagged Trees method.

6 Conclusion

This article presents the use of model-based diagnosis and data-based techniques for detection and isolation of cyber-attacks and faults on a two-tank system. A set of 12 fault scenarios and 14 attack scenarios was described, 9 of which are new compared to previous works. A first model-based technique was successfully implemented and tested on the benchmark. All faults and attacks are detectable, except replay attack. By introducing 2 new residuals implemented on the benchmark the isolability has been increased so that some attacks can be isolated from faults. However, most of the time the different scenarios have the same signature. In this article the authors propose a first step towards fault and attack detection and isolation. More efficient methods have to be tested to take the fault and attacks dynamic into account. Model-based approaches in the field of switched systems have been mentioned. Another idea could be to take advantage of structural analysis to analyze internal dynamics of the model and residuals.

Data-based diagnosis was performed using supervised machine learning methods. These methods obtained very good results in isolating faults and attacks and could be used in the future either alone, either using a fusion solution. Some aspects of the work however could be consolidated notably the characterization of the faults and attacks. Unsupervised learning methods were also tested. The goal is to find hidden pattern or groupings in data. This approach did not give satisfying results at the moment and should be investigated deeply. Methods for anomaly detection could be beneficial to consider cyberattacks not known in advance. Another research direction is data pre-processing to extract relevant features to fed machine learning methods. Here also dynamical aspects could be better considered.
Acknowledgments
We thank Julie Fouan, Maeva Kleinberg and Florian Valette, 3rd year students from INP-ENSEEIHT, Toulouse, who implemented the matlab simulator and the benchmark improvements presented in this paper.

References