



CBR-ODAF: A Case-Based Reasoning for the Online Diagnosis of All internal Faults in Automated Production Systems

Nourhene Ben Rabah¹, Ines Ben Tekaya¹ and Moncef Tagina²

Centre de recherche en informatique - University Paris 1 Pantheon Sorbonne, Paris, France.

COSMOS lab, ENSI, university of Manouba 2010, Manouba, Tunisia.

Nourhene.Ben-Rabah@univ-paris1.fr Ines.Ben-Tekaya@univ-paris1.fr

moncef.tagina@ensi-uma.tn

Abstract

In this paper, we focus on the online diagnosis of Automated Production Systems (APS) equipped with sensors and actuators emitting binary signals. These systems can be considered as Discrete Event Systems (DES). The paper presents a Case-Based Reasoning for the Online Diagnosis of All types of Faults in APS (CBR-ODAF). It is an improvement of our approach presented previously in order to remedy its limitations. Firstly, it proposes a new case representation format that describes all the faults to diagnose, adapts to the dynamic aspect of APS, is quite expressive and is easy to understand by human operators. Secondly, it allows to classify in real time each new observation as a 'normal case', 'faulty case' or 'unidentified case' based on a new dissimilarity index which is not intrinsic to the numerical type. It is an index that adapts to our proposed case representation format and describes the degree of difference between cases represented by data of different types (i.e. quantitative and qualitative).

1 Introduction

In recent decades, Automated Production Systems (APS) have evolved significantly. These systems have increased production, improved product quality, reduced labor costs and saved energy and raw material. In return, they have become increasingly complex and are quite often subject to faults. So in order to ensure their availability, guarantee their efficiency and ensure their safety, the scientific communities of automation engineers and computer scientists are coalescing to implement diagnosis approaches to detect, locate and identify potential faults in real time. In this context this paper focuses on the online diagnosis of APS. Our working hypotheses are as follows:

- **Hypothesis 1:** It should be noted that the diagnosis role depends on the dynamics of the APS functioning. It can be classified in three families: Continuous Systems, Discrete Event Systems (DES) [4] and Hybrid Systems. In this study, we are interested in the APS diagnosis with sensors and actuators delivering binary signals that fall within the DES scope.

- **Hypothesis 2:** Most of these systems are controlled by a programmable electronic device called Programmable Logic Controllers (PLC). After an initialization step, a PLC executes a cycle (T) comprising three successive operations: reading the inputs (E), executing the program (Pg) and updating the outputs (S). The PLC cycle can be run cyclically or periodically. In the first case, the PLC executes a new cycle as soon as the previous cycle is completed. Thus each cycle can have a different execution time. In the second case, the PLC executes a new cycle after receiving a synchronization by a clock (after some constant time). If another operation is running then it is canceled. Subsequently, our work revolves around APS controlled by PLC whose execution algorithm is periodic.
- **Hypothesis 3:** A fault is a malfunction in a system. The faults present in an APS can be classified according to several criteria such as the speed of appearance (sudden or gradual), the time of appearance (during the APS operation or when it stops), the importance degree (partial or complete), consequences (personal injury or property damage) and origin (internal factors or external factors). In our work, we will diagnose internal faults due to a sensor or actuator malfunction. They can be observable faults, such as the unexpected change from 0 to 1 or from 1 to 0 in a sensor or in an actuator. They can be also unobservable faults such as the stuck off a sensor or an actuator to 0 or to 1.
- **Hypothesis 4:** A digital twin is a virtual representation of a physical product, process, or service. It reflects the physical system state. It can be used when the physical system is offline or online. The capabilities, the functions, and the features of the digital twin will be defined by the physical system and represented by the twin. The digital twin must be accurate while representing physical, communication and operational aspects of the physical system. If the physical system can do something, the twin must do the same thing in the same way. The digital model is constantly updated. It changes its settings to better match the current operating mode of the physical system. In this paper, we are not interested in the design of a digital twin but we assume that we have one for the raw data acquisition.

Based on these assumptions, we propose an improvement and an extension of the previously proposed approach [2] in order to meet our main objective which is the online diagnosis of all internal faults in APS. This new approach presents two contributions: **(a)** a case representation format inspired by Causal Temporal Signatures (CTS) formalism which allows to describe all internal faults. In addition, this format adapts to the dynamic aspect of APS and **(b)** a new dissimilarity index which is not intrinsic to the numerical type. It is an index that adapts to our proposed case representation format and describes the degree of difference between cases represented by data of different types (quantitative and qualitative). The paper is organized as follows: section 2 presents the general concepts. Section 3 introduces our proposed method. Section 4 exposes our implementation details. In section 5, we present the results of applying our approach to a benchmark system and compare its performance to the old approach. A conclusion and overview of future work are presented in the last section.

2 General concepts

2.1 Case-Based Reasoning

Case-based reasoning (CBR) is a paradigm of artificial intelligence and cognitive science that models the reasoning process as primarily memory based. It use the specific knowledge of past

experiences, formalized in the form of concrete problem situations called cases. A new problem is solved by looking for a more similar past case and using it for the resolution of the new problematic situation. In 1994, Aamodt and Plaza defined a well received four step view on the process of case-based reasoning [1]. The four steps are: (a) **Retrieve** the most similar or the similar case(s) from the case base; (b) **Reuse** the solution of the most relevant case; (c) **Revise and adapt** the proposed solution to become compatible with the current problem and (d) **Retain** the confirmed solution in the case base for future use.

2.2 Causal Temporal Signatures

A Causal Temporal Signatures (CTS) is a subset of observable events partially ordered that characterizes a normal or a faulty behavior of a system. Events are linked by time constraints on their occurrence. A CTS is a rule that can be formally defined as follows: $X \rightarrow Y$ where X consists of two/several triplets and Y is a state of the system following this signature (normal or faulty behavior). Let (E, F, ct) be a triplet. 'E' and 'F' are two events with 'E' is the reference event and 'F' is the constrained event. 'ct' is a time constraint that can model a date, a period or a duration. If there are no time constraints between 'E' and 'F', 'ct' becomes 'nct' that means no time constraint, '*' is an operator which separates each triplet. It is used as an 'and' to connect triplets.

2.3 Why using CBR and CTS ?

We chose to use CBR for the following reasons: (i) it is a method that does not require a large amount of data to work properly. Indeed, an APS always executes the same work cycle for which it was programmed. This is why the amount of raw data describing its operation is limited. CBR can work with this amount of data unlike other machine learning methods that require a large amount of data to work properly and (ii) it allows a better understanding of the problem. Unlike 'black box' type algorithms such as deep learning, the CBR allows us to access to the reasoning. Its results can easily be justified by presenting similar cases as examples.

We chose to use CTS for the following reasons: (a) it is a formalism that is expressive enough since it is able to formally describe all the desired behaviors of the system to be diagnosed. This is done thanks to the temporal constraints and conjunction operators [8]; (b) CTS is understandable and readable by human supervisory operators to facilitate rules inspection, correlations discovery and relationships between them and (c) it has high efficiency in knowledge view of the symptoms of faults diagnosis. However, among the difficulties to overcome for the diagnosis using this formalism, we find the acquisition and the updating of a representative and rich knowledge base (that is to say a database of CTS) [8]. Indeed, a poor and unrepresentative base can lead to mediocre diagnosis system [3]. This database must contain sufficient knowledge of the system malfunction. Following this study, we concluded that the CTS represent a real trail that we can draw on to represent our cases.

3 Proposed Solution

Our online diagnosis approach (CBR-ODAF) is illustrated in figure 1. Its implementation requires an approach involving two phases: (a) *Phase 1: the initial case base construction phase* that transforms the various sensors and actuators signals of the virtual APS to empirical knowledge and (b) *Phase 2: the online reasoning and learning phase* that diagnoses online possible faults and updates the case base following the appearance of new unknown behaviours.

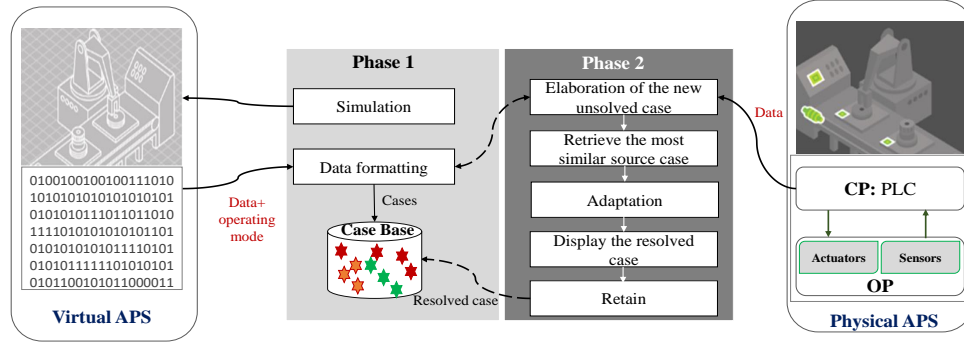


Figure 1: CBR-ODAF architecture.

3.1 The initial case base construction (learning base)

This phase combines a simulation module with a data formatting module to build the initial case base.

3.1.1 Simulation module

The simulation module provides different data describing the normal and faulty behaviors of the virtual APS. These data represent the signals of its sensors and actuators. We have chosen to use a digital twin instead of a real APS because it is possible to inject faults on sensors and actuators without any material damage. The principle of this module is as follows: **[Step 1]**: Simulate the virtual system in several normal operating scenarios; **[Step 2]**: For each possible scenario, collect data from sensors and actuators and transform them into empirical knowledge using the data formatting module. This module will be detailed in the subsection (3.1.2) and **[Step 3]**: For the P faults to be diagnosed in APS, simulate the virtual system by injecting each time a F_i fault and repeat step 2.

3.1.2 Data formatting module

In this module, the data obtained from the virtual system are processed to extract relevant information that is then modelled as cases. These cases describe normal and faulty behaviours of the system. To achieve such an objective, it is necessary to choose a case presentation format that is able to represent all the system faults. For this purpose, we propose in the following a case representation format to overcome the limitations of the case representation previously introduced [2].

A. Former case representation

A case is a vector formed by two parts: problem representation part and solution representation part. The problem representation represents the measurements of sensors and actuators (called descriptors) obtained during the current PLC cycle (noted T) and previous PLC cycles ($T-1$, $T-2$, $T-3$, ..., $T-z$) with z is the number of previous cycles whose inputs/outputs change. The solution part allows to qualify the type of case: normal or faulty. We have noticed that this representation has some limitations that are subject to extensions and improvements. These limits are as follows: (a) Incomplete representation: this format only describes observable

faults such as the unexpected change of a sensor or actuator from 1 to 0 or 0 to 1, but it cannot describe unobservable faults such as the stuck off a sensor or an actuator to 0 or to 1. **(b)** combinatorial explosion of descriptors (i.e. sensor and actuator measurements): this format can cause a combinatorial explosion of descriptors if the number of the system components is very large and **(c)** readability: this format is difficult for system users to understand (i.e. the binary format is not understandable for humans). To overcome these difficulties and improve the results, we present a new case representation.

B. New case representation

We are inspired by the CTS formalism to represent our cases in CBR-ODAF. We propose to adopt an object-based representation format. The latter can be seen as a format easily modelling a CTS. A case is then represented by an object described by an identifier (**ID**), a problem (**P**) and a solution (**S**). The problem **P** is represented by sets of Triplets. Each triplet is described by two observable events that are partially ordered and respect a temporal constraint. The first event is called **Referent Event (RE)**, the second event is a **Constrained Event (CE)** and the noted **Temporal Constraint (TC)** is a period described by a Lower Bound (*LB*) and a Upper Bound (*UB*). Solution **S** represents the behavior nature of the system to be diagnosed. This behavior may be '*normal*', '*faulty*' or '*unidentified*'. The *unidentified state* is a new state that is not mentioned in the former representation. It reduces the number of misclassified cases. In fact, storing a case as '*unidentified*' is better than storing it incorrectly.

According to this case representation, the problem part of a new case is a set of triplets. Each triplet is represented by (*RE, CE, TC*). In the following paragraph, we present an algorithm that transforms the sensor and actuator signals calculated during each PLC cycle to a 'Triplet' and builds the problem part of the new case from this triplet and the triplets associated with previous PLC cycles.

C. Generation of problems part

The proposed algorithm has as inputs the signals from APS sensors and actuators and the PLC cycle time. It has as output the problem part of a new case. It consists of six steps which are detailed by the following pseudo code. The notations used are the following: *Binarysignature*: groups the signals of sensors and actuators during a PLC cycle, *T*: Current PLC cycle, *T-1*: Previous PLC cycle, *CE*: Constrained Event, *RE*: Referent Event, *TC*: Temporal Constraint, *LB*: Lower Bound of the TC, *BS*: Upper Bound of the TC, *CurrentCycleTime*: time measurement associated with cycle T and *PreviousCycleTime*: time measurement associated with cycle T-1.

- **Step 0:** Group the signals from APS sensors and actuators during the *T* cycle to build a *Binarysignature*. If this *Binarysignature* is different from the *Binarysignature* of the *T-1* cycle then go to the step 1.
- **Step 1:** Formulation of the Referent Event (*RE*): If it is the first cycle executed then the *RE* is an '*IN*' event (i.e. it is an unconstrained event) otherwise the *CE* of the *T-1* cycle becomes the *RE* of the *T* cycle.
- **Step 2:** Formulation of the Constrained Event (*CE*): Each element of the binary signature is transformed to an event that can be either the **R**ising edge (designated by the prefix '**R**') or the **F**alling edge (designated by the prefix '**F**') of a sensor or actuator. This event is defined as a *CE*.

- **Step 3:** Formulation of the Temporal Constraint *TC*. If *RE* is equal to 'IN' then there is no temporal constraint (i. e. $LB=0$ and $UB=0$). If *RE* is different from 'IN' then the *TC* is calculated after an estimation of the random measurement errors due to hardware, physical parameters or operator. This leads to first expressing the estimated time: $EstimatedTime = CurrentCycleTime - PreviousCycleTime$ thereafter: $LB = EstimatedTime - \Delta t$ and $UB = EstimatedTime + \Delta t$ with Δt represents the uncertainty on the value of time, its value is > 0 and it is set in advance by an expert in the field.
- **Step 4:** Group the three previous results to formulate a new triplet.
- **Step 5:** Add the new triplet to the previous triplets to build the problem part of a new case.

D. Cases labeling

The operation of the APS in normal or faulty mode allows to automatically label each case without the need of an expert accompanying the process of data formatting. The normal operations are represented by a set of normal cases while the faulty operations are defined by a set of faulty cases. Both types of cases are stored in the initial case base.

3.2 The online reasoning and learning

This phase allows on the one hand the online recognition of this new problem and on the other hand the updating of the case base. To achieve these objectives, it consists of searching for the most similar source case, reusing the solution of the selected source case and saving the new solved case in the database.

3.2.1 Retrieve the most similar source case

As our case representation is made up of qualitative and quantitative data, we have proposed a Case Similarity Index Measure (CSIM). It calculates the degree of similarity between the new unsolved case and the source cases of the initial case base. This index allows us to avoid the dummy coding step¹ for qualitative data.

CSIM for the $Case_i$ and $Case_j$ consists in calculating the dissimilarity index for its P_i and P_j problems, which is noted *PDIS* index. The *CSIM* index is defined as follows:

$$CSIM(Case_i, Case_j) = 1 - PDIS(P_i, P_j) \quad (1)$$

where $P_i = \{Tr_{1,i}, Tr_{2,i}, Tr_{3,i}, \dots, Tr_{mt,i}\}$, $P_j = \{Tr_{1,j}, Tr_{2,j}, Tr_{3,j}, \dots, Tr_{mt,j}\}$ and mt is the triplet number. The *PDIS* index computes the dissimilarity index between the different triplets of the two problems. It is defined as follows:

$$PDIS(P_i, P_j) = \frac{\sum_{k=1}^{mt} DT(Tr_{k,i}, Tr_{k,j})}{3mt} \quad (2)$$

where $Tr_{K,i} = (RE_{K,i}, CE_{K,i}, TC_{K,i})$ is the k -th triplet of the i -th case, $Tr_{K,j} = (RE_{K,j}, CE_{K,j}, TC_{K,j})$ is the k -th triplet of the j -th case and *DT* is the dissimilarity index between the two triplets $Tr_{K,i}$ and $Tr_{K,j}$. It amounts to calculate two indices of dissimilarity: one for

¹Dummy coding provides one way of using categorical predictor variables in various kinds of estimation models.

the events (equation 4) and the other for the temporal constraints (equation 5). In fact, events are nominal qualitative data and temporal constraints are quantitative data of interval type. That's why DT is computed as follows:

$$DT(Tr_{k,i}, Tr_{k,j}) = DEV(RE_{k,i}, RE_{k,j}) + DEV(CE_{k,i}, CE_{k,j}) + \frac{IPOS(TC_{k,i}, TC_{k,j})}{Date_{max}} \quad (3)$$

where $Date_{max}$ corresponds to the maximum date of all time intervals constituting the case base. The dissimilarity index DEV between two events is computed as follows:

$$DEV(E_{k,i}, E_{k,j}) = \begin{cases} 0 & \text{if } E_{k,i} \text{ and } E_{k,j} \text{ are equal} \\ 1 & \text{otherwise.} \end{cases} \quad (4)$$

The positioning index $IPOS$ between two temporal constraints is defined as follows:

$$IPOS([x, y], [a, b]) = \begin{cases} 0 & \text{if } x \in [a, b] \cap [x, y] \\ MinPos([x, y], [a, b]) & \text{otherwise.} \end{cases} \quad (5)$$

where:

$$MinPos([x, y], [a, b]) = \min(Pos(x, [a, b]), Pos(y, [a, b])) \quad (6)$$

$[x, y]$ is the temporal constraint of the triplet $Tr_{k,i}$, $[a, b]$ is the temporal constraint of the triplet $Tr_{k,j}$ and Pos is the positioning index between a point x and an interval $[a, b]$.

$$Pos(x, [a, b]) = \begin{cases} 0 & \text{if } x \in [a, b] \\ Min(|x - a|, |x - b|) & \text{otherwise.} \end{cases} \quad (7)$$

3.2.2 Adaptation

The new case inherits the solution of the source case that has the maximum similarity and is greater than the '*decision threshold*'. Otherwise, if the similarity between the selected source case and the new case is less than or equal to the '*decision threshold*', then the new case is classified as an '*Unidentified case*'. The decision threshold must be set in advance by the system designer.

3.2.3 Retain

The new solved case is retained in the case base and is used immediately to solve future recognition problems.

4 Implementation details

4.1 Benchmark

To illustrate the performance of the proposed methodology, we used a benchmark [5] which is the sorting system. In Phase 1, we used a digital twin of the sorting system to build the initial case base, however, in phase 2 we operated a physical one for online diagnosis. The sorting

system is made up of eleven sensors and seven actuators. Each sensor and actuator have four types of faults: **(a)** The stuck off a sensor or an actuator to 0; **(b)** the stuck off a sensor or an actuator to 1; **(c)** the unexpected movement of a sensor or an actuator from 0 to 1 and **(d)** the unexpected movement of a sensor or an actuator from 1 to 0. Although the size of the system is not so large (18 I/O), it has 72 faults to be diagnosed, which is a relatively high number. For this reason, we restricted our study to only one sub-system. We chose the turntable, containing two sensors C_4 , C_5 and one actuator S_4 , since it was used as a benchmark sub-system in other diagnosis approaches [7, 5]. C_4 is the detector of the turntable loading position; C_5 is the detector of the turntable unloading position and S_4 is the turntable order.

In order to simulate different normal and faulty operating scenarios, we used the analysis of a human expert ([6]). The labels and descriptions of these faults are as follows: $F1$: C_4 stuck off at 0; $F2$: C_4 stuck off at 1; $F3$: C_5 stuck off at 0; $F4$: C_5 stuck off at 1; $F5$: S_4 stuck off at 0; $F6$: S_4 stuck off at 1; $F7$: Unexpected movement of C_4 from 0 to 1; $F8$: Unexpected movement of C_4 from 1 to 0; $F9$: Unexpected movement of C_5 from 0 to 1; $F10$: Unexpected movement of C_5 from 1 to 0; $F11$: Unexpected movement of S_4 from 0 to 1 and $F12$: Unexpected movement of S_4 from 1 to 0.

4.2 Initial case base

We assume that our initial case base is composed by 2 normal cases and 12 faulty. Faulty cases are labeled by the various faults presented previously.

5 Preliminary evaluation

The initial case base used for the evaluation contains 348 cases distributed over the different classes as follows: **(a)** 176 cases, labeled 'N', describe the turntable normal operations and **(b)** 172 cases, showing the different faults injected on the turntable, are distributed as follows: 11 cases of fault 'F2', 23 cases of fault 'F3', 30 cases of fault 'F4', 16 cases of fault 'F6', 27 cases of fault 'F7', 18 cases of fault 'F8', 13 cases of fault 'F9', 11 cases of fault 'F10', 15 cases of fault 'F11' and 4 cases of fault 'F12'.

Faults labeled 'F1' and 'F5' are not present in the experimental case base. In fact, at each simulation of these faults, the constructed cases remained the same and their values did not change.

5.1 Accuracy metrics

The proposed diagnosis system is considered as a multi-class classifier which tries to tidy up correctly every new case in one of the previously predefined classes. In such classifications, there are common metrics that are being used in the literature to evaluate the classifiers performance which can be computed from a confusion matrix (CM). This rows matrix shows the actual values of the target attribute for all instances of the test set (i.e. the target attribute is the case class). Its columns shows the predicted values. It can be represented through four types of responses: **(i)** TP_i (**True Positive** of the class C_i): Measures the number of correctly recognized cases for the class C_i ; **(ii)** TN_i (**True Negative** of the class C_i): Measures the number of correctly recognized cases that do not belong to the class C_i ; **(iii)** FP_i (**False Positive** of the class C_i): Measures the amount of cases that are incorrectly assigned to the class C_i and **(iv)** FN_i (**False Negative** of the class C_i): Measures the number of cases that are not recognized as belonging to the class C_i .

Although the confusion matrix gives us a lot of information about the quality of the classification system, we can calculate other more concise metrics like the average accuracy (AC). It presents the proportion of the correct predictions made by the classifier. It measures the average per-class effectiveness. AC is computed using equation (8).

$$AC = \frac{\sum_{i=1}^l \frac{TP_i + TN_i}{TP_i + FN_i + FP_i + TN_i}}{l} \quad (8)$$

To measure the performance of our approach and compare it with the old approach, we used a stratified cross-validation of 10 folds. This method ensures that each fold contains the same portions of instances of each class. The tests were performed on a 4-core processor with 32 GO RAM. They were repeated **10 000** times to measure the average accuracy.

5.2 Results

After applying the two approaches to the same experimental case base, it was obvious that the proposed approach performed the best prediction with an accuracy about 0.994 ± 0.002 while the old approach having an accuracy about 0.75 ± 0.003 . Figure 2 presents the multi-class confusion matrix obtained from a random execution of the approach proposed above. Figure 3 shows the CM of the CBR-ODAF approach. The number of cases in the diagonal gives an image of the right one classification:

- For the old approach: *261* cases are correctly classified, *176* cases are classified as normal cases (see green box) and *85* cases are classified as faulty cases (see red boxes). No false alarms have been generated (see the orange boxes) but *87* faulty cases are incorrectly classified (see the blue and the white boxes). They are classified as normal behaviours. These cases represent instances of faults *F2*, *F3*, *F4*, *F6* and *F8*. Indeed, the approach fails to recognize the faults of sensor or actuator blocking types at 0 or 1.
- For the CBR-ODAF: *345* cases are correctly classified, *176* cases are classified as normal cases and *169* cases are classified as faulty cases. No false alarms were generated (see the orange boxes in the matrix) and no faults were classified as normal behaviour (see the blue boxes). In overall, our proposed approach misses three cases [*2F8+1F4*]. The *F4* fault is classified as an unidentified case and two instances of the *F8* fault were misclassified.

The obtained results show that the CBR-ODAF can provide good prediction with the least error for the diagnosis of turntable sub-system. It makes it possible to recognize blocking faults at 0 and 1 and also to reduce the number of misclassified cases by classifying them as unidentified cases.

6 Conclusions and discussion

Within our CBR-ODAF, we offered an online diagnosis of APS based on a case representation format inspired by CTS formalism which allows to describe all types of faults to be diagnosed. We present a new dissimilarity index which describes the degree of difference between cases represented by data of different types. A revision step is therefore necessary to control the increase in the case base while retaining only the cases of good performance. In recent years the amount of data stored in case base is growing rapidly. In the future we propose for our research to further investigate the practical advantages that can be gain through big data approach.

| | N | F2 | F3 | F4 | F6 | F7 | F8 | F9 | F10 | F11 | F12 |
|-----|-----|----|----|----|----|----|----|----|-----|-----|-----|
| N | 176 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F2 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F3 | 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F4 | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F6 | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F7 | 0 | 0 | 0 | 0 | 0 | 26 | 0 | 0 | 0 | 0 | 0 |
| F8 | 1 | 0 | 0 | 0 | 0 | 1 | 16 | 0 | 0 | 0 | 0 |
| F9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 0 | 0 | 0 |
| F10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 0 | 0 |
| F11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 |
| F12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |

Figure 2: The multi-class confusion matrix of the former proposal.

| | N | F2 | F3 | F4 | F6 | F7 | F8 | F9 | F10 | F11 | F12 |
|-----|-----|----|----|----|----|----|----|----|-----|-----|-----|
| N | 176 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F2 | 0 | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F3 | 0 | 0 | 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F4 | 0 | 0 | 0 | 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| F6 | 0 | 0 | 0 | 0 | 16 | 0 | 0 | 0 | 0 | 0 | 0 |
| F7 | 0 | 0 | 0 | 0 | 0 | 26 | 0 | 0 | 0 | 0 | 0 |
| F8 | 1 | 0 | 0 | 0 | 0 | 1 | 16 | 0 | 0 | 0 | 0 |
| F9 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 0 | 0 | 0 |
| F10 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 0 | 0 |
| F11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 0 |
| F12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 |

Figure 3: The multi-class confusion matrix of the CBR-ODAF.

References

- [1] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *AI communications*, 7(1):39–59, 1994.
- [2] N Ben Rabah, R Saddem, F Ben Hmida, V Carre-Menetrier, and M Tagina. Intelligent case based decision support system for online diagnosis of automated production system. In *Journal of Physics: Conference Series*, volume 783, page 012009. IOP Publishing, 2017.
- [3] Nourhene Ben Ben Rabah, Ramla Saddem, Faten Ben Hmida, Véronique Carré-Ménétrier, and Moncef Tagina. Automatic acquisition and update of a causal temporal signatures base-for faults diagnosis in automated production systems. In *ICINCO (1)*, pages 262–269, 2017.
- [4] Christos G Cassandras and Stephane Lafortune. *Introduction to discrete event systems*. Springer Science & Business Media, 2009.
- [5] Julien Niguez, Saïd Amari, and Jean-Marc Faure. Active fault-tolerant control of timed automata with guards. *IFAC-PapersOnLine*, 50(1):13648–13653, 2017.
- [6] Alexandre Philippot, Pascale Marangé, Véronique Carré-Ménétrier, and Bernard Riera. Implementation of diagnosis approach for discrete event systems. In *International Symposium on Security and Safety of Complex Systems, 2SCS'12*, page CDROM, 2012.
- [7] Bernard Riera, Rachid Benlorhfar, D Annebicque, F Gellot, and B Vigario. Robust control filter for manufacturing systems: application to plc training. In *18th World Congress of the International Federation of Automatic Control*, 2011.
- [8] Ramla Saddem and Alexandre Philippot. Causal temporal signature from diagnoser model for online diagnosis of discrete event systems. In *Control, Decision and Information Technologies (CoDIT), 2014 International Conference on*, pages 551–556. IEEE, 2014.