

# DIAGNOSIS OF CONTINUOUS DYNAMIC SYSTEMS: INTEGRATING CONSISTENCY BASED DIAGNOSIS WITH MACHINE-LEARNING TECHNIQUES

Belarmino Pulido <sup>\*,1</sup> Juan José Rodríguez Díez <sup>\*\*</sup>  
Carlos Alonso González <sup>\*</sup> Oscar J. Prieto <sup>\*</sup>  
Esteban R. Gelso <sup>\*</sup>

<sup>\*</sup> *Grupo de Sistemas Inteligentes, Dpto. de Informática,  
Universidad de Valladolid, Spain. e-mail:  
{belar,calonso,oscapri,egelso}@infor.uva.es*

<sup>\*\*</sup> *Lenguajes y Sistemas Informáticos, Universidad de Burgos,  
Spain. e-mail: jjrodriguez@ubu.es*

**Abstract:** This paper describes an integrated approach to diagnosis of complex dynamic systems, combining model based diagnosis with machine learning techniques, proposing a simple framework to make them cooperate, hence improving the diagnosis capabilities of each individual method.

First step in the diagnosis process resorts to consistency-based diagnosis, via possible conflicts, which allows fault detection and localization without prior knowledge of the device fault modes. In the second step, a classification system, obtained via machine learning techniques, is used to propose a ranked sequence of fault modes, coherent with the previous localization step. This cycle iterates in time, generating more focused and precise diagnosis as new data are available.

A laboratory plant has been built to test this proposal. Simulation results are shown for a total number of 14 different faults. *Copyright ©2005 IFAC*

**Keywords:** Fault Diagnosis, Model-based diagnosis, Machine-learning

## 1. INTRODUCTION

### 1.1 *The context*

Diagnosis of complex dynamic systems is still an open research problem. It has been approached using a wide variety of techniques, (Balakrishnan and Honavar, 1998), being the four main approaches: Knowledge Based – including expert systems—, Case Based Reasoning, Machine Learning and Model Based Systems. Currently, it seems clear that no single technique is capable to claim its success in every field. Therefore, an increasing number of diagnosis systems has opted

for hybrid solutions. This is our case. Our approach relies primarily upon model-based diagnosis, but it has been enhanced via machine-learning techniques to overcome some drawbacks.

Within the model-based approach we can make difference of two major fields: one coming from the Engineering community –known as FDI approach–, and another one coming from the Computer Science-Artificial Intelligence community –known as DX approach–. Focusing our discussion on the DX community the main research effort in order to tackle real world difficult problems has been directed toward modelling issues, recognizing that modelling is a key question in model based diagnosis. This is particularly severe in the consistency based approach to diagnosis (de Kleer *et al.*, 1992), where fault identifica-

---

<sup>1</sup> This work has been supported by the Spanish MCyT project DPI2002-01809 and the “Junta de Castilla y León” project VA101/01.

tion requires explicit modelling of faulty behaviors. Consistency-based diagnosis, CBD, provides an elegant theoretical framework for fault detection and isolation. CBD can be summarized as an iterative cycle of behaviour prediction, discrepancy or conflict detection, fault isolation or candidate generation, and candidate refinement by means of new measurements. In this cycle, diagnosis candidates can be automatically obtained from conflicts using a minimal hitting-set algorithm.

Recently, researchers from DX and FDI communities have worked to bring together a common framework for model-based diagnosis, which is known as BRIDGE. In such framework, Cordier et al. (Cordier et al., 2004) have pointed out that CBD using conflicts and FDI using ARR based on structural analysis are equivalent under given assumptions. The possible conflict concept (Pulido and Alonso, 2000), which will be thoroughly used in this work must be also understood within this BRIDGE framework. Possible conflicts are those sub-systems capable to become conflicts in CBD, i.e. minimal subsets of equations containing analytical redundancy. The set of possible conflicts can be obtained through off-line analysis of the set of equations in the original model. Hence, this technique is rather similar to the ARR approach for FDI (Staroswiecki and Declerk, 1989). We have recently demonstrated that possible conflicts and ARRs are also equivalent under given assumptions (Pulido and Alonso González, 2004).

## 1.2 The problem

The theory underlying CBD was originally developed for static systems, and has been successfully applied to diagnose complex dynamic systems. Although there is no general extension for dynamic systems, authors usually rely upon qualitative or semi-qualitative models to overcome uncertainty in the models, and noisy measurements. However, this kind of modelling is inherently ambiguous concerning behaviour estimation, and does not allow early detection of non-abrupt faults. Since CBD is able to perform both fault detection and isolation with just models for correct behaviour, our proposal uses quantitative models for correct behaviour of dynamic systems within such framework. Also, to overcome problems related with uncertainty and noise, we perform the consistency-check for a data series using a dissimilarity value. The whole process is described in (Pulido et al., 2001).

Within the BRIDGE framework Cordier et al. (Cordier et al., 2004) have stated that main difference between DX and FDI approaches to model-based diagnosis comes from the way the fault signature matrix, FSM, is analyzed: while matching the real and the theoretical FSM, both communities used different diagnosis assumptions. FDI methods usually rely upon *single-fault* and *no-compensation* assumptions. This is not

the case in DX. This fact provides another drawback for precise fault isolation in DX: fault isolation and identification in CBD can not be done unless any other fault mode has been rejected. Hence, the FSM is revised just to reject those fault signatures which are not consistent with current conflicts, i.e. residuals. Therefore, for continuous dynamic processes with limited observability, usually there is no specific candidate localization, but a collection of components or fault modes which are consistent with current observations (i.e. they have not been rejected yet). This problem can be even worse when fault symptoms exhibit different dynamics. Conflicts can be found at different time steps, and fault identification requires additional time to be sound from a logical point of view.

Our proposal consists on enhancing the precision in the fault localization step within CBD based on possible conflicts using machine-learning techniques, while keeping diagnosis soundness.

The organization of this paper is as follows. First, we summarize the machine-learning concepts used in the approach. Second, we provide a description of the enhanced CBD protocol. Finally, we show some results on a case study, and draw some conclusions.

## 2. MACHINE LEARNING TECHNIQUES FOR FAULT IDENTIFICATION

Machine learning techniques has been successfully used to automate fault diagnosis, inducing trees or rules from examples, (Quinlan, 1993), or training artificial neural networks, (Venkatusugramanian and Chan, 1995). These techniques try to identify behavioral patterns associated to the different faults, and allow to perform fault identification. However, the majority of the machine learning techniques do not take into account the dynamic aspects of a problem and, consequently, fail to exploit the temporal information that so meaningful seems to be to human troubleshooters.

Since main patterns for identifying faults in dynamic environments consist on the evolution over time of variables related with the current fault, we have approached the diagnosis problem as classification of multivariate time series.

### 2.1 Time Series Classifiers

The considered classification system is based on the family of learning methods named “boosting”, using very simple base classifiers: only one literal. At present, an active research topic is the use of *ensembles* of classifiers. One of the most popular methods for creating ensembles is boosting, (Schapire, 1999). It works assigning a weight to each example. Initially, all the examples have the same weight. In each iteration a *base* (also named *weak*) classifier is constructed,

using any other classification method, according to the distribution of weights. Afterwards, the weight of each example is readjusted, based on the correctness of the class assigned to the example by the base classifier. The final result is obtained by weighted votes of the base classifiers.

The base classifier used here are interval predicates. There are two kinds: relative and region based. Relative predicates consider the differences between the values in the interval. Region based predicates are based on the presence of the values of a variable in a region during an interval. This section only introduces the predicates; (Rodríguez *et al.*, 2001) gives a more detailed description, including how to select them efficiently. These predicates are:

- increases (Example, Variable, Beginning, End, Value). It is true, for the Example, if the difference between the values of the Variable at End and Beginning is greater or equal than Value.
- decreases (Example, Variable, Beginning, End, Value). Idem when the difference is less or equal.
- stays (Example, Variable, Beginning, End, Value). It is true, for the Example, if the range of values of the Variable in the interval is less or equal than Value.
- always (Example, Variable, Region, Beginning, End). It is true, for the Example, if the Variable is always in this Region in the interval between Beginning and End.
- sometime (Example, Variable, Region, Beginning, End). Similarly, when the variable stays for sometime.
- true\_percentage (Example, Variable, Region, Beginning, End, Percentage). It is true, for the Example, if the percentage of the time between Beginning and End where the variable is in Region is greater or equal to Percentage.

## 2.2 Using the Classifiers for Fault Identification

Normally, when a classifier is used, the only expected result is the selected class. For fault identification, it would be desirable to obtain a ranking of the different fault modes. This ordering contains information that may be useful for the human responsible on the command of the process. On the other hand, some faults will be discarded in the model-based stage, so the desired output is an ordering of the remaining candidates. The ADABOOST (Schapire, 1999) algorithm assigns for each class a determined value: the weighted vote of the individual classifiers for that class. These value can be used for considering the result of the classifier not as a unique class but as an ordering of the set of classes. If some classes have been discarded in a previous step, then it is only necessary to compute the weighted vote for the remaining classes.

In order to measure the adequacy of a classification with these features we have considered the average number of classes that appear before the correct class. In order to obtain a measurement with a range independent of the number of classes, the value of the average number of classes that are before the correct one is divided by the number of classes minus one. In this way, that value will be between 0 and 1. This measure is called *position error*. The classical error,

here will be called *classification error*, with the objective of avoiding confusions. A position error of 0% indicates that the correct class is always the first, a 100% indicates that the correct one is always the last. If the order of classes were assigned randomly, the average value of this error would be 50%.

The classifiers have been obtained using full series as training examples. They are trained with series that start and end in a steady state, with faults happening somewhere in between. Nevertheless, it is not an option to wait for a full series in order to use the classifiers, we want to apply the classifiers as soon as a fault is detected. Hence, the classifiers must deal with partial time series, and they must produce a classification, as good as possible, considering the available information. We call this feature *early classification*. From all the literals in the classifier, some of them will have a defined result for the partial example, because their intervals refer to areas that are already available in the example. Nevertheless, for other literals their results will be unknown because their intervals still are not available for the example. The learning method produces as a result a linear combination of literals. The literals that still have an unknown result, will be simply omitted from the classifier. The classification given to a partial example will be the linear combination of those literals that have known results.

## 3. INTEGRATION OF CONSISTENCY-BASED DIAGNOSIS USING POSSIBLE CONFLICTS AND TIME SERIES CLASSIFIERS

The CBD approach based on possible conflicts for continuous dynamics systems has been previously introduced in (Pulido *et al.*, 2001). The integration of model based diagnosis and machine learning techniques may be accomplished in several ways. It basically depends on which properties we desire the end system to exhibit. Since we wanted to preserve the logical soundness of consistency based diagnosis, we have opted for giving higher priority to its output, constraining the machine learning methods to refine previously localized faults.

To achieve this behaviour, the induced time series classifiers are slightly modified. Let us denote by  $CLASSIFIER(t)$  an invocation of the induced time series classifier with a fragment of series from time  $t$  to  $\min(\text{current time}, t + \text{maximum series length})$ . Each call to  $CLASSIFIER(t)$  will return a list of fault modes ranked by their voted weight. Let us denote by  $CLASSIFIER(t, c)$  an invocation to the modified classifier, being  $c$  a set of consistency based diagnosis candidates. A call to  $CLASSIFIER(t, c)$ , will compute the list obtained firstly invoking  $CLASSIFIER(t)$  and secondly removing those fault modes not associated to components of  $c$ . To further simplify the problem, singled fault hypothesis is assumed; otherwise, the

induction of the time series classifiers becomes a combinatorial problem.

Due to the capability of the induced classifiers to consider only a fragment of a time series –early classification– and to discard the fault modes not associated with the current candidates –as the previous paragraph shown– the integration of both techniques is particularly simple if consistency based diagnosis relies on the concept of *possible conflict*: the integration only requires to invoke the time series classifiers from the iterative and incremental cycle of diagnosis with *possible conflicts*. Additional details about the integration architecture and results for typical benchmark in machine learning can be found in (Alonso González *et al.*, 2004).

In a dynamic environment, diagnosis with possible conflicts is performed in an iterative and incremental way, assuming the hypothesis of non intermittent faults. To improve fault localization and facilitate fault identification, we only have to add a new step, 2d, to the basic cycle<sup>2</sup>.

- (1) OFF-LINE:
  - (a) analyze the model  $SD$  looking for possible conflicts,  $pc_i$
  - (b) build appropriate executable models for each possible conflict,  $SD_{pc_i}$
- (2) ON-LINE:
 

*repeat*

  - (a) simulate  $SD_{pc_i}$  using  $OBS_{pc_i}$  and producing  $PRED_{pc_i}$ ,
  - (b) if  $|PRED_{pc_i} - OBS'_{pc_i}| > \delta_{pc_i}$  then confirm  $pc_i$ ,
  - (c) if a new  $pc_i$  is confirmed, then compute the new set of candidates
  - (d) update fault modes ranking with  $CLASSIFIER(t_0, \text{set of candidates})$

*until* there is no  $pc_i$  to be simulated.

The proposed diagnosis process will incrementally generate the set of candidates obtained from available observations. Simultaneously, it will order the available fault modes according to their confidence, in a process with an error rate that decreases as bigger fragments of the variables evolution is available.

## 4. A CASE STUDY

### 4.1 The plant to be diagnosed

The designed laboratory plant, which can be seen in figure 1, tries to resemble common industrial continuous processes. It is made up of 4 tanks – $\{T_1, T_2, T_3, T_4\}$ –, 5 pumps – $\{P_1, P_2, P_3, P_4, P_5\}$ –, and 2 PID controllers – $\{PI_1, PI_2\}$ – acting on pumps  $\{P_1, P_5\}$  to keep the level of  $\{T_1, T_4\}$  close to the

<sup>2</sup> Where  $OBS_{pc_i}$  denotes the set of input observations available in  $SD_{pc_i}$ ,  $PRED_{pc_i}$  represents the set of predictions obtained from  $SD_{pc_i}$ ,  $OBS'_{pc_i}$  denotes the set of output observations in  $SD_{pc_i}$ , and  $\delta_{pc_i}$  is the maximum value allowed as the dissimilarity value between  $OBS'_{pc_i}$  and  $PRED_{pc_i}$ .  $t_0$  is the starting time of the series, prior to the first conflict confirmation.

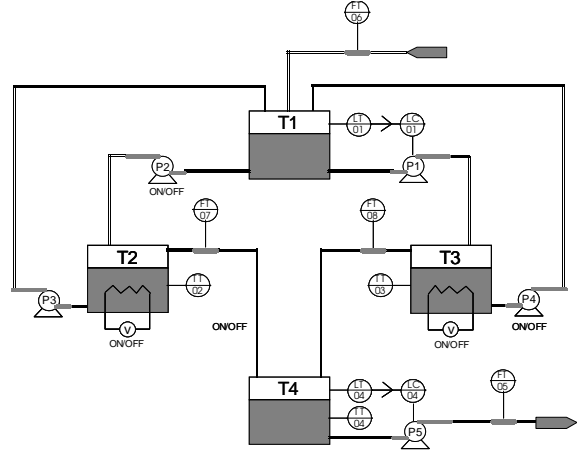


Fig. 1. Reconfigurable laboratory plant made up of 4 tanks, 5 pumps, 2 controllers, and 2 resistors.

specified set point. To control temperature on tanks  $\{T_2, T_3\}$  we use two resistors – $\{R_1, R_2\}$ .

In this plant we have the following set of measurements: levels of tanks  $T_1$  and  $T_4$  – $\{LT_1, LT_2\}$ –, the value of the PID controllers on pumps  $\{P_1, P_5\}$  – $\{LC_1, LC_2\}$ –, in-flow on tank  $T_1$  – $\{FT_1\}$ –, outflow on tanks  $\{T_2, T_3, T_4\}$  – $\{FT_2, FT_3, FT_4\}$ –, and temperatures on tanks  $\{T_2, T_3, T_4\}$  – $\{TT_1, TT_2, TT_3\}$ –. Action on pumps  $\{P_2, P_3, P_4\}$  and on resistors – $\{R_1, R_2\}$ – are also known.

We have used common physics equations to model the behavior of each component:

|          |  |
|----------|--|
| $t_{dm}$ | mass balance in tank $t$                         |
| $t_{cm}$ | evolution of mass from mass balance in $t$       |
| $t_{ct}$ | computing temperature in tank $t$                |
| $t_{dE}$ | energy balance in tank $t$                       |
| $t_{cE}$ | computing temperature from energy balance in $t$ |
| $t_{fb}$ | flow from tank $t$ to pump                       |
| $t_f$    | flow from tank $t$ through a pipe                |
| $p_1$    | relation between flow and pressure within a pump |
| $p_2$    | relation between inflow and outflow in a pump    |

This plant can work on different situations because our diagnosis approach is just a task within a global supervision problem (Acosta Lazo *et al.*, 2002). We have defined three working situations which are commanded through four different operation protocols. In the operation protocol used in this paper resistor  $R_1$  is switch off, while resistor  $R_2$  is on. Also, pumps  $\{P_3, P_4\}$  are switch off; hence, just flow  $FT_1$  is an input to tank  $T_1$ .

### 4.2 Consistency-based diagnosis using possible conflicts

In this system we have found 10 different possible conflicts, all of them minimal w.r.t. the equations used in the models. These possible conflicts can be seen in Table 1.

In this plant we have considered the following classes of faults for the current protocol: related to leakages in tanks – $\{f_1, f_2, f_5, f_7, f_9\}$ –, pipes blockages – $\{f_3, f_4, f_6, f_8, f_{10}\}$ –, pumps failures – $\{f_{11}, f_{12}, f_{13}\}$ –, and resistor failure – $\{f_{14}\}$ –. Table 2 show what

|           |   |
|-----------|---|
| $PC_1$    | $\{t1_{dm}, t1_{fb1}, t1_{fb2}, p1_1, p1_2, p2_1, p2_2\}$                                     |
| $PC_2$    | $\{t1_{ct}, t1_{dE}, t1_{fb1}, t1_{fb2}, p1_1, p1_2, p2_1, p2_2, t2_{dm}, t2_{ct}, t2_{dE}\}$ |
| $PC_3$    | $\{t1_{ct}, t1_{dE}, t1_{fb1}, t1_{fb2}, p1_1, p1_2, p2_1, p2_2, t3_{dm}, t3_{ct}, t3_{dE}\}$ |
| $PC_4$    | $\{t1_{fb1}, p1_1, p1_2, t3_{dm}, t3_{fp}\}$  |
| $PC_5$    | $\{t1_{fb2}, p2_1, p2_2, t2_{dm}, t2_{fp}\}$  |
| $PC_6$    | $\{t4_{dm}, t4_{fb5}, p5_1\}$   |
| $PC_7$    | $\{t4_{dm}, p5_2\}$   |
| $PC_8$    | $\{t4_{ct}, t4_{dE}, t4_{fb5}, p5_1\}$  |
| $PC_9$    | $\{t4_{ct}, t4_{dE}, p5_2\}$  |
| $PC_{10}$ | $\{t4_{fb5}, p5_1, p5_2\}$  |

Table 1. Possible conflicts found for the laboratory plant.

|           | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ | $f_9$ | $f_{10}$ | $f_{11}$ | $f_{12}$ | $f_{13}$ | $f_{14}$ |
|-----------|-------|-------|-------|-------|-------|-------|-------|-------|-------|----------|----------|----------|----------|----------|
| $PC_1$    | 1     | 1     | 1     | 1     |       |       |       |       |       |          | 1        | 1        |          |          |
| $PC_2$    |       |       | 1     | 1     | 1     |       |       |       |       |          | 1        | 1        |          |          |
| $PC_3$    |       |       | 1     | 1     |       |       | 1     |       |       |          | 1        | 1        |          | 1        |
| $PC_4$    |       |       |       | 1     |       |       | 1     | 1     |       |          | 1        |          |          |          |
| $PC_5$    |       |       | 1     |       | 1     | 1     |       |       |       |          |          | 1        |          |          |
| $PC_6$    |       |       |       |       |       |       |       |       | 1     | 1        |          |          | 1        |          |
| $PC_7$    |       |       |       |       |       |       |       |       | 1     |          |          |          | 1        |          |
| $PC_8$    |       |       |       |       |       |       |       |       |       | 1        |          |          | 1        |          |
| $PC_9$    |       |       |       |       |       |       |       |       |       |          |          |          | 1        |          |
| $PC_{10}$ |       |       |       |       |       |       |       |       |       | 1        |          |          | 1        |          |

Table 2. PCs and their related fault modes

would be the relation between possible conflicts and described faults<sup>3</sup>.

### 4.3 Induced Classifiers

Since fault detection is obtained via possible conflicts, it was only necessary to consider 14 classes: one for each fault mode. For each faulty behaviour 20 examples were generated. These simulations differ because we have randomly selected the free parameters for the operation protocol and noise was added to the input flow. Also, the values for parameters modelling the fault were also tossed. Every simulation always has at least 3 minutes of correct behaviour. Faults randomly occur, between 3 and 5 minutes from the start. The series are simulated for 15 minutes (time required for variables with slow dynamic to reach a new steady state).

Figure 2 shows the evolution of some variables for  $f_1$  –a small leakage in  $T_1$ –, and  $f_2$  –medium or big leakage in  $T_1$ –, respectively. Keep in mind that this is a ten variable problem.

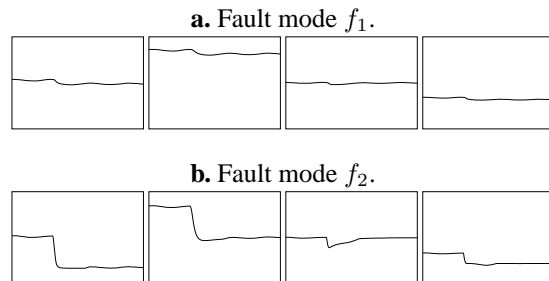


Fig. 2. Meaningful magnitudes ( $FT_3$ ,  $FT_4$ ,  $LT_1$ ,  $LC_1$ ) to distinguish fault modes  $f_1$  and  $f_2$ .

The classifiers were built from the relative and region based predicates (section 2) and the classifiers were

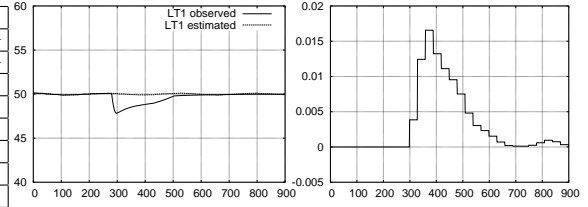


Fig. 3. Fault detection using using  $PC_1$ .

formed by 100 literals. Experimental results were obtained using 10-fold cross-validation. The classification error is 2.86% when using complete series, 7.14 when the 50% of the series are available and 23.93% when only the 35% is available. The position errors are 0.49%, 1.43% and 4.94% for, respectively, full series, half the series and 35% of the series. Since this is a 14 class problem, these errors mean that the average position of the correct class in the ranking are, respectively, 1.06, 1.19 and 1.64.

### 4.4 A diagnosis episode

To illustrate the behaviour of the system we include a complete diagnosis episode for the second fault mode,  $f_2$ : medium to large leak in the first tank. The free variables of the operation protocol and the beginning of the fault were randomly selected. Figure 2.b shows the evolution of the four most significant variables obtained from simulation.

The episode starts with a detection due to the confirmation of the first possible conflict,  $PC_1$ . Figure 3, left, shows the observed and predicted values of  $LT_1$  via  $PC_1$ . Right hand side shows the output of the detection phase for this possible conflict: using a sliding window approach, every 30 seconds a set of 60 values are compared using dynamic time wrapping, DTW. DTW is used for detection due to its robustness to compare time series. If the returned dissimilarity value exceeds the threshold, the possible conflict is confirmed as a real conflict. With a detection threshold of 0.01, the possible conflict is confirmed at  $t = 330$  seconds. It is important to note that this is the only established conflict in the whole episode.

Considering the delay introduced by the monitoring process and the methodology applied to simulate the training example we situate the origin of the time series at  $t_0 = 60$ . Thus, at detection time, we start to classify fragments of 270 seconds —30% of the whole series—.

A call to  $CLASSIFIER(60)$  at  $t = 330$  returns  $f_1$ ,  $f_{10}$  and  $f_2$  as the three most plausible faults. However  $f_{10}$  may be filtered out, because  $f_{10}$  is not a fault mode associated to the relations of  $PC_1$ . Hence, in this example, from detection time we have a satisfying ranking:  $f_1$ ,  $f_2$  which stand for small, medium to big leak in  $T_1$ .

A call to  $CLASSIFIER(60)$  at  $t = 420$  —that is, 1.5 minutes after detection time and 40% of the

<sup>3</sup> Fault Signature Matrix in FDI terminology.

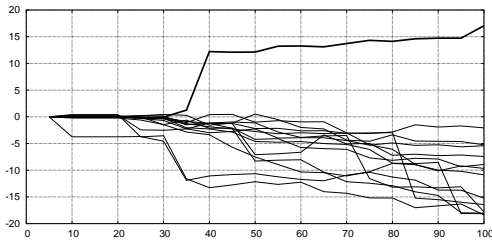


Fig. 4. Evolution of the confidence assigned to each class by the classifier. The thicker curve corresponds to the correct class.

series— returns  $f_2$ ,  $f_7$  and  $f_8$  as the three most plausible faults. Now,  $f_7$  and  $f_8$  may be discarded. The remaining fault,  $f_2$ , is the correct one.

From now on, the weight of  $f_2$  keeps increasing, maintaining a significant gap with the class in the second place. This situation is illustrated in figure 4, which shows the evolution of the confidence assigned by the classifier to the different classes as a function of the series percentage for this particular example.

## 5. DISCUSSION

An integrated approach to diagnosis of dynamic systems, combining consistency-based diagnosis and machine-learning techniques, has been introduced. It pretends to be effective for complex scenarios. The proposal has been illustrated with a non trivial example, which shows the viability of the approach. Regarding to classification error, the induced classifiers behaves quite well. Moreover, the cooperation of both methods may improve the diagnostician performance as the selected example pointed out.

Special effort has been done to keep the best properties of consistency based diagnosis: diagnosis results are sound and complete, using just models of correct behaviour. It should be noticed that the proposed way to integrate both techniques does not lose the completeness of the system, guaranteed by the consistency based phase. If a non considered faulty model arises, the system is still able to do fault localization. At the same time, we try to alleviate its major drawback: diagnosis tends to be unfocused due to the absence of fault information. This fault information is introduced resorting to machine learning techniques. A major advantage of the proposed method is that fault models for training do not require to know the precise value of the parameters modelling the fault. Moreover, some degree of variations on these parameters may facilitate the induction process.

The induced models, time series classifiers, describe in a natural way some temporal properties of the faulty behaviours: they are designed to work with time series and their symbolic nature allows to adapt them to accept series of different length. This property provides several opportunities for a natural integration on the iterative cycle of consistency based diagnosis.

This work is part of an ongoing research activity, and further experimental effort has still to be done. Particularly, we would like to test the approach on even more demanding scenarios.

## REFERENCES

- Acosta Lazo, G., C. J. Alonso-González and B. Pulido Junquera (2002). Basic tasks for knowledge based supervisin in process control. *ENG APPL ARTIF INTEL* **14**, 441–455.
- Alonso González, C., J.J. Rodriguez and B. Pulido (2004). *Current topics in Artificial Intelligence: X Conf. CAEPIA'2003, and V Conf. TTIA'2003*. Chap. Enhancing consistency-based diagnosis with machine-learning techniques, pp. 312–321. Vol. 3040 of *LNAI*. Springer.
- Balakrishnan, K. and V. Honavar (1998). Intelligent diagnosis systems. *INT J INTELL SYST*.
- Cordier, M.O., P. Dague, M. Dumas, F. Levy, J. Montmain, M. Staroswiecki and L. Travé-Massuyés (2004). Conflicts versus analytical redundancy relations - a comparative analysis of the model based diagnosis approach from the artificial intelligence and automatic control parspectives. *IEEE T SYST MAN CY B. Special Issue on Diagnosis of Complex Systems: bridging the methodologies of the FDI and DX communities* **34**(5), 2163–2177.
- de Kleer, J., A. K. Mackworth and R. Reiter (1992). Characterising diagnosis and systems. In: *Readings in Model-based Diagnosis* (W. Hamscher, L. Console and J. de Kleer, Eds.). pp. 54–65. Morgan-Kaufmann.
- Pulido, B. and C. Alonso (2000). An alternative approach to dependency-recording engines in consistency-based diagnosis. In: *AIMSA-00*. Vol. 1904 of *LNAI*. pp. 111–120. Springer Verlag.
- Pulido, B. and C. Alonso González (2004). Possible conflicts: a compilation technique for consistency-based diagnosis. *IEEE T SYST MAN CY B. Special Issue on Diagnosis of Complex Systems: bridging the methodologies of the FDI and DX communities* **34**(5), 2192–2206.
- Pulido, B., C. Alonso and F. Acebes (2001). Lessons learned from diagnosing dynamic systems using possible conflicts and quantitative models. In: *IEA/AIE-2001*. Vol. 2070 of *LNAI*. pp. 135–144.
- Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann.
- Rodríguez, J. J., C. J. Alonso and H. Boström (2001). Boosting interval based literals. *Intelligent Data Analysis* **5**(3), 245–262.
- Schapiro, Robert E. (1999). A brief introduction to boosting. In: *16<sup>th</sup> IJCAI*.
- Staroswiecki, M. and P. Declerk (1989). Analytical redundancy in non linear interconnected systems by means of structural analysis. In: *Proc. of the IFAC AIPAC-89*. Nancy, France. pp. 51–55.
- Venkatusugramanian, V. and K. Chan (1995). A neural network methodology for process fault diagnosis. *AICHE J.* **35**, 1993–2001.