Using Supervised Learning Techniques for Diagnosis of Dynamic Systems

Pedro J. Abad[1], **Antonio J. Suárez**[1], **Rafael M. Gasca**[2], **Juan A. Ortega**[2]

Abstract. This paper describes an approach based on supervised learning techniques for the diagnosis of dynamic systems. The methodology can start with real system data or with a model of the dynamic system. In the second case, a set of simulations of the system is required to obtain the necessary data. In both cases, obtained data will be labelled according to the running conditions of the system at the gathering data time. Label indicates the running state of system: correct working or abnormal functioning of any system component. After being labelled, data will be treated to add additional information about the running of system. The final goal is to obtain a set of decision rules by applying a classification tool to the set of labelled and treated data. This way, any observation on the system will be classified according to those decision rules, having a return label indicating the currently running state of system. Returned label will be the diagnostic. This entire learning task is carried out off-line, before the diagnosing.

1 INTRODUCTION

Diagnosis determines why a system, correctly designed, doesn't work like it was expected. Explanation, for this erroneous behaviour, represents a discrepancy with the system design. One diagnosis task is to determine the system elements that could cause the erroneous behaviour according to the system observations. Monitoring process is fundamental to avoid non-real faults by small alterations in variables values. [1] Proposes a knowledge model for dynamic systems monitoring.

Fault detection consists on determining, starting from the system observations, when an incorrect operation of the observed system exists. When failure is detected then diagnosis will take the control to find the reasons of that incorrect behaviour.

Fault detection and diagnostic of faulty components are very important from the strategic point of view of the companies, due to the economic demands and environment conservation required to remain in competitive markets. This is one of the reasons causing that this is a very active investigation field. Components faults and process faults can cause systems damages and undesirable halt of the system. This causes the increase of costs and decrease of production. Therefore developing mechanisms to detect and to diagnose systems faults are needed to maintain the systems in levels of security, production and reliability.

Inside the Artificial Intelligent community the dynamic systems diagnosis task has been approached, in most of the cases, adapting the techniques coming from the static systems diagnosis to the dynamic behaviour of the systems. This way [2] or [3] try to add temporary information to GDE [4]

On the other hand, qualitative models have also been commonly used for this purpose [5] [6].

In [7] the fundaments of the based-models diagnosis, applied to the dynamic systems, are presented, and more recently [8] proposes a consistency-based approach with qualitative models.

Other techniques, coming from the AI, have also entered in the diagnosis field. Following this line, learning techniques tries to identify the system behaviour basing on a previous training.

Lately, some works using learning-based techniques have been presented, like stochastic methods [9], neural network based learning [10] and classification systems [11]. Neural network techniques have recently been applied in diverse fields, as medicine [12] or power supply [13].

Machine Learning techniques, inside the supervised learning field, are automated procedures based on logical operations that learn a task starting from a suite of examples. In the classification field the attention has been centred, concretely, in approaches with decision trees [14], where classification is the result of a series of logical steps. These approaches are able to represent the most complex problems if they have enough data. Applied to the diagnosis, we can find these methods used for the classification of temporary patterns [15] or in previous works to the current one [16] [17].

The present work is centred in quantitative models. It uses supervised learning techniques to obtain a rules-based model to diagnose dynamic systems by recognizing the correct behaviour models and faulty behaviour models. An approach to offer several fault causes, when there isn't an only clear cause, is presented.

Rest of the document has been organized in the following way: in the next section the used methodology will be exposed and the form to carry out the diagnosis. Next a problem application example is described for the developed approach. To illustrate the operation of these techniques a wide set of tests is presented. Lastly some improvements that are in development process are discussed.

2 PROPOSED METHODOLOGY

To carry out diagnosis of dynamic systems a set of decision rules should be generated. It can be done starting from the known

[1] Dpto de Ingeniería Electrónica, Sistemas Informáticos y Automática. Universidad de Huelva. E-Mail: {abadhe,asuarez@uhu.es}
[2] Dpto de Lenguaje y Sistemas Informáticos. Universidad de Sevilla. E- Mail: {gasca,ortega@lsi.us.es}

trajectories of the system or the simulations generated from a model.

Before starting with the methodology some concepts need to be defined.

2.1 Definitions and notation.

Definition 1: Behaviours Family. It is a finite group of trajectories having a similar behaviour from the point of view of the diagnosis.
Definition 2: Correct behaviour. It is the finite group of trajectories belonging to evolutions of the system without any fault type.
Definition 3: Perfect behaviour. It is the trajectory describing the system when all parameters take the central values of the ranges defined as correct.
Definition 4: Observation. It is a real trajectory of the dynamic system containing values of the observational variables in the system.
Definition 5: Diagnosis. It is the identification of the observed behaviour of the system as belonging to a certain behaviour family (diagnosis label) and according to decision rules.

Proposed approach can be generated from two different ways:

- Rules are generated starting from a group of different behaviour models.

  $\vee$ *Model (behaviour)* $\Rightarrow$ *labelled trajectories*
- Rules are generated starting from a group of experimental trajectories of dynamic system for the correct behaviour and possible fault behaviour.

  $\vee$ *Trajectories (behaviour)* $\Rightarrow$ *labelled trajectories.*

Leaving of one of these situations the process can continue like that:

1. Similar trajectories belonging to different behaviours family are identified. These trajectories are labelled again as belonging to both behaviours family.

   $\vee$ *Similar Trajectories (different behaviour family)* $\Rightarrow$ *relabelled trajectories.*
2. Decision rules are generated using a supervised learning tool.

   $\vee$ *Relabelled trajectories* $\Rightarrow$ *Decision rules*
3. Diagnosis consists in associating an observation as corresponding to behaviours family by using decision rules.

   Classification (observation, rules) $\Rightarrow$ *Diagnostic label*

2.2 Methodology

Proposed methodology to diagnose is an amplification of other one developed in [16]. This basic methodology may present some problems when the same system behaviours can be associated to different fault reasons. In order to don't diagnose incorrectly these cases, in this new approach, those behaviours will be associated with all the possible behaviours family that can cause this concrete behaviour. In this way several fault causes will be offered for observations that can correspond to different behaviours family.

Basic idea consists in obtaining a set of classification rules from a suite of system data in different behaviours modes: the correct behaviour and the faulty behaviours. After, those obtained classification rules can be used to associate an observation with model behaviour. Thus diagnosis of the observation is obtained.

Process can start with real system data or with a model of the dynamic system. In the second case, a set of simulations of the system is required to obtain the necessary data. In both cases, obtained data will be labelled according to the running conditions of the system at the gathering data time. Label indicates the running system state: correct working or abnormal function of any system component. Final result consists in a database containing all labelled trajectories.

Obtained database contains very similar trajectories corresponding to different behaviour family and therefore with different labels. To solve this problem the set of all similar trajectories will be relabelled with new labels. This new labels will be composed as a mix of the older labels. Thus, relabelled trajectories will be associated with anyone of the original behaviours family. The problem is to define when two or more trajectories are similar. Decision taken is that several trajectories

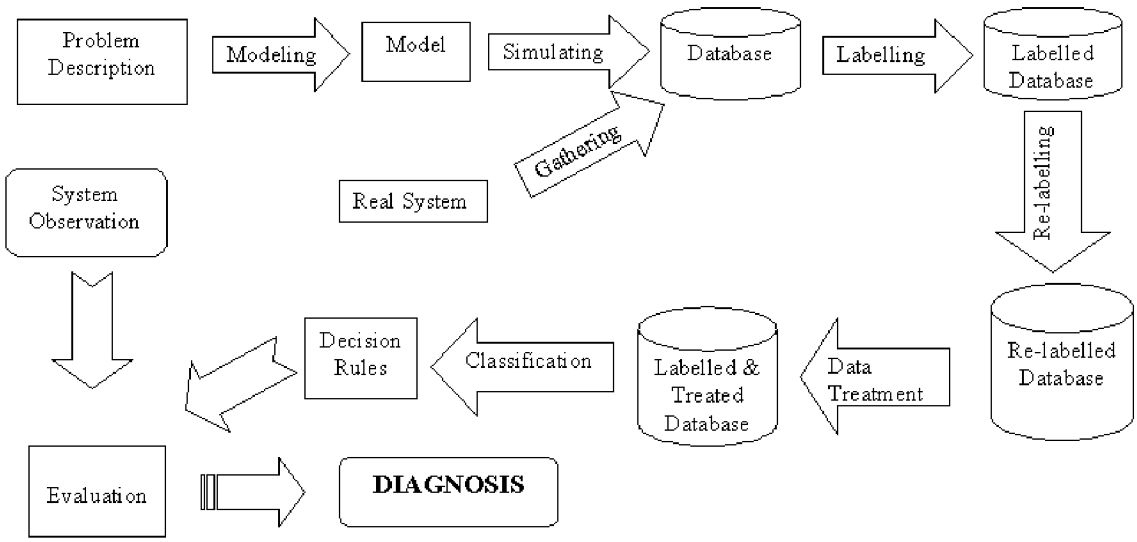

Figure 1. Proposed Methodology

are similar when distance between them is lower than a magnitude. That magnitude should be specified for each treated system. Used distance is Euclidean distance.

After being labelled and relabelled, trajectories data will be treated to add additional information about running of the system. This additional information will be very useful when classification tool tries to find decision rules, because available information will be greater. This additional information should characterize the system further than gathering data and it is specified for each treated systems.

A new database, which contains original trajectories plus new attributes and the corresponding label, is obtained.

Final step, to obtain decision rules, is to use a classification tool with the labelled and treated database.

An aspect to highlight is that all process, until this moment, have been development off-line, and time needed for this process is not important for the diagnosis process.

Diagnosis process consists on evaluating an observation with the obtained decision rules. Time spending to diagnose is only the time of evaluating obtained decision rules. Decision rules returns the label associated to the behaviour by correspondence between training data and observed data. This returned label is offered as diagnosis.

Next a case study will be presented to develop this methodology.

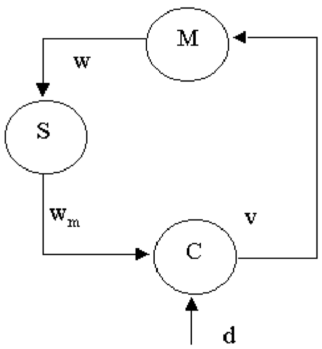

Figure 2. The example system

3 CASE STUDY

As it has been commented previously, methodology can be used with real system data or with obtained data of a model simulation. In our case, the methodology will be applied to a model, which is an idealized situation, but it offers us a clear idea of the way to act. In case of application on a real system, many difficult aspects, not mentioned here (as monitoring or small phase shift), need to be taken in account, but with the model we are only trying to present the approach.

As example of dynamic system to diagnose we consider the controller electric motor in [18] and [19]. Figure 2 represents treated system. The motor 'M', whose rotational speed is '*w*', is driven through a voltage '*v*' by the controller 'C' which acts based on the desired speed '*d*' and the speed 'w_m' measured by the revolution counter 'S'. Controller 'C' is considered as an I-controller.

System can be modelled by the following equations, which include a constant for each component that is used to model also the faulty behaviour of the component:

$$Motor: T * \frac{dw}{dt} = c_m * v - w \quad (1)$$

$$I - Controller: \frac{dv}{dt} = c_c * (d - w_m) \quad (2)$$

$$Sensor: w_m = c_s * w \quad (3)$$

Where T is the inertia of the motor, c_m is the constant of the motor; c_c is the constant of the controller and c_s is the constant of the revolution counter.

Component anomalous operation is caused, mainly, by the deviation of the component constant nominal value. These constants stray of the considered correct values range

Some faults represent that constants take values above the correct ones and others faults represent that constants take values below the correct ones. Diagnosis result should indicate, in addition to the faulty component, if taken values for the component constant are below correct values or above them.

Possible fault reasons that we want to identify are therefore: 'CmHigh' when values of Cm are above the correct ones; 'CmLow' when values of Cm are below the correct ones; 'CsHigh' when values of Cs are above the correct ones; 'CsLow' when values of Cs are below the correct ones; 'CcHigh' when values of Cc are above the correct ones and 'CcLow' when values of Cc are below the correct ones.

To describe the system correct behaviour, it is considered that values of all constants don' t have only one correct value, but rather they can take values inside an interval that will be considered as correct.

This way, operation flexibility is allowed and system real behaviour is better simulated, where there is not a correct value but rather correction margins are flexible. This produces that system doesn' t have an only correct behaviour, but rather a correct behaviours family. It represents all possible combinations of the constants values that are inside of the defined tolerance limit.

A correct behaviours family does the diagnosis more difficult, because it is necessary to recognize different behaviours as correct, but on the contrary it provides a more realistic vision of the system.

In our model the constant values considered as correct are:

Table 1. Values for OK behaviours

Cm	[0.98-1.02]
Cc	[0.98-1.02]
Cs	[0.98-1.02]

Other considered characteristics in our system are:

1. Fault is present from the beginning and it doesn' t evolve in the time.
2. Behaviour change occurs instantly and starting from here it doesn' t change again.
3. Once the wanted angular speed has been indicated, it doesn' t change until this angular speed is reached.

This way, diagnosis will be carried out when the desired angular speed (d) is changed. The way to diagnose is by checking the evolution to reach the final speed. It is necessary to keep in mind that in spite of existence of a failure in some component, I-controller is able to act on the motor to reach the required final speed. Of course evolution of the system to reach the desired final speed will be different. This difference in the behaviour will allow the diagnosis.

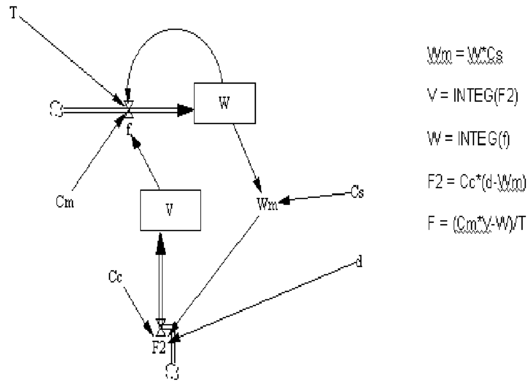

Figure 3. Forrester diagram

First step, therefore, is performing system simulations in different behaviours modes. In our case, system has been modelled as a Forrester diagram [20], to be able to simulate using the simulation tool VEMSIM®. Forrester diagram generated for the system is presented in figure 3.

Simulated behaviours will be those that we want to diagnose. They will be: OK for correct behaviour and CmHigh, CmLow, CsHigh, CsLow, CcHigh, CcLow for each component fault above mentioned.

A behaviour family will represent each one of these behaviours.

Simulations values are shown in table 2.

Table 2. System values for simulation

T	3
D	10
W	5
Time Step	0.1

For the correct behaviour the constant values are into [0.98-1.02]. Values to simulate behaviours above the correct one are into [1.02-5]. Values to simulate behaviours bellow the correct one are into [0-0.98].

Constants values for simulated behaviours have been elected by random with the Monte Carlo method following a uniform distribution. Number of simulations per behaviour will be 100.

Label corresponding to behaviour is placed to each one of the trajectories. This way, a database containing 700 labelled trajectories is obtained.

Trajectories are composed with values of the variable 'w_m' in each time step. Reason to select variable 'w_m' and not 'w' is that 'w_m' is the only observable variable in the real system.

In figures 4, 5 and 6 different system behaviours are shown.

Obtained database has similar trajectories belong to different behaviours. This way several very similar trajectories have different labels. This is a problem, because our final goal is to use a classification tool to obtain a set of decision rules, and if we have similar trajectories with different labels then classifier can't correctly work; that is to say, those similar trajectories will be incorrectly classified. Figure 7 shows an example of this.

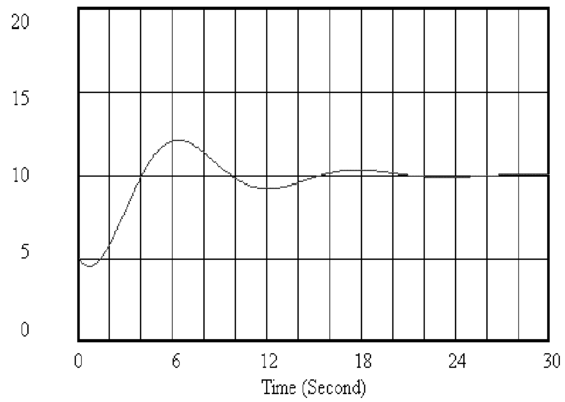

Figure 4. OK Behaviour

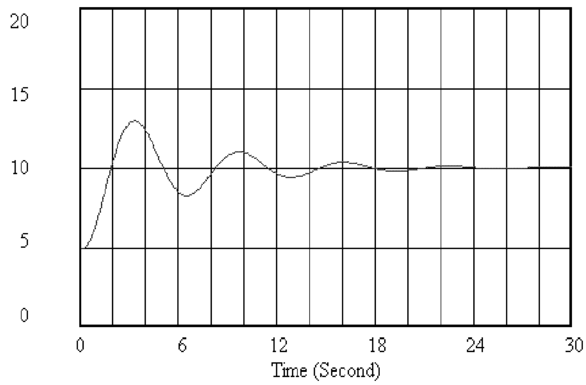

Figure 5. CmHigh Behaviour

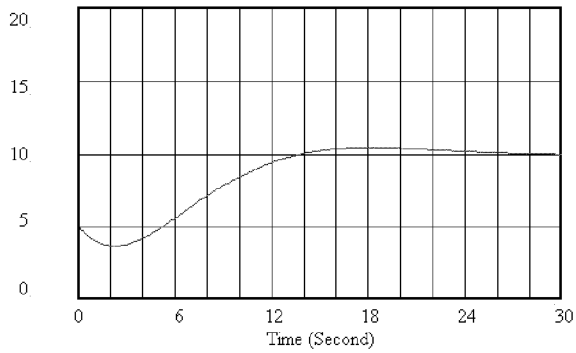

Figure 6. CcLow Behaviour

To solve this problem a new label will be assigned to very similar trajectories. A mixture of labels of all similar trajectories will compose the new label. This way, next step is to find all similar trajectories into the database and assigning a new label.

It is necessary to define when two or more trajectories are similar. Two trajectories are considered similar when distance between them is smaller than a magnitude. Distance between trajectories is measured as Euclidean Distance and magnitude chosen is 10% of the Euclidean distance between the two further away trajectories for the correct behaviour. This magnitude in our example is 0.45.

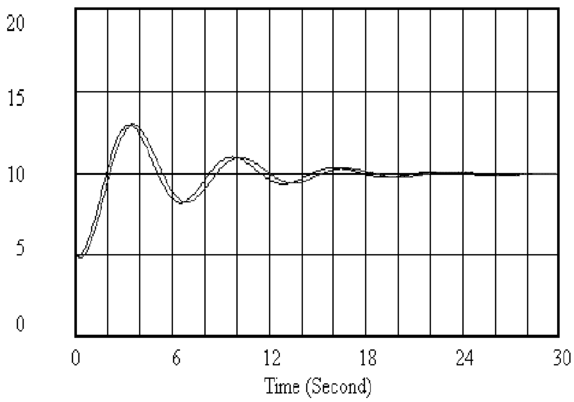

Figure 7. Behaviour CcHigh vs CmHigh

After this process we obtain a new database with all similar trajectories re-labelled as corresponding with all behaviours of the similar trajectories.

Next step is to calculate new attributes of each trajectory with the goal that classifier has more information to generate decision rules. These new attributes must be representative for each trajectory.

For each trajectory point next attributes have been calculated:

- Distance to perfect behaviour. It indicates how far away is current trajectory from perfect behaviour (above defined). It is calculated as:

$$DP(i) = Wm[i] - Wmpf[i] \tag{4}$$

Where *Wm[i]* is the treated point in the current trajectory and *Wmpf[i]* is the correspondent point in the perfect behaviour.

- Integral. It is the magnitude returned by numerical integration between current point and the precedent one. It represents the closed area between them. It is calculated by approximating as follow:

$$I(i) = Ts \times \frac{p[i] - p[i-1]}{2} \tag{5}$$

Where *Ts* is the time step in the simulation, *p[i]* is the current treated point and *p[i-1]* the precedent one.

In addition next attributes will be calculated for each trajectory:

- Rise Time (RT). It is the moment in which desired revolution speed is reached for first time.
- Steady state (SS). It is the moment in which desired revolution speed is reached definitively.
- Max speed (MS). It is the value of the highest revolution reached speed.
- Max speed time (MST). It is the moment in which the highest revolution speed is reached.

This way a new database containing trajectories plus new attributes is generated.

Data in new database have the following form:

RT, SS, MS, MST, Wm[1], DP[1], I[1], ..., Wm[n], DP[n], I[n], LABEL

Final step is performing supervised learning with the obtained database. Classification tool selected to perform the supervised learning is C4.5 [21]. What is gotten with this tool is to characterize each one of the behaviour families according to the values of the attributes that have been provided. Result is a decision tree and an equivalent set of decision rules. These rules will be the way to do the diagnosis. In our example classifier obtains 27 rules with an error rate of 1.2%. This mean that 1.2% of trajectories are not correctly classified with those rules.

3.1 Diagnosis

The way to do the diagnosis is evaluate the observed data with the obtained rules.

Because in rules appear attributes that have been calculated and not appear in observed data, same attributes should be calculated for observed data in order to be able to classify with those rules.

This way in the moment that one observed data is gathered all possible attributes should be calculated. After that, decision rules are evaluated with two possible results: a label is returned or information is insufficient to evaluate all rules. In the first case the returned label is the result of the diagnosis. In the second one we need to wait more information in further moments.

If we want to diagnose the system with another running conditions, we should have prepared the decision rules set for those specific conditions. I. e. if we want to diagnose this system when current rotational speed is 12 rad/sec and desired rotational speed is 7 rad/sec, we should have generated a set of decision rules for those conditions and we will use them in the diagnosis moment.

4 RESULTS ON THE EXAMPLE SYSTEM

To evaluate the proposed methodology a set of tests have been done.

Observational data have been obtained by simulating the system with specific conditions for the test. This way a test trajectory is obtained and the diagnosis correct result is known, because it must be the corresponding to the simulated conditions.

Conditions of the test are the same above mentioned. We remember them in table 3:

Table 3. Tests conditions

T	3
D	10
W initial	5
Time Step	0.1
Values for OK	[0.98 - 1.02]
Values for HIGH	[1.02 - 5]
Values for LOW	[0 - 0.98]

In table 4 we can see results for the tests:

Table 4. Tests results

VALUE OF THE CONSTANT			CORRECT DIAGNOSIS	DIAGNOSIS WITH SIMPLE LABELLED	DIAGNOSIS WITH RE-LABELLED
Cm	Cc	Cs			
1	1	1.03	CS HIGH	CS HIGH	CS HIGH
1	1	1.07	CS HIGH	CS HIGH	CS HIGH
1	1	1.1	CS HIGH	CS HIGH	CS HIGH
1	1	1.5	CS HIGH	CS HIGH	CS HIGH
1	1	2	CS HIGH	CS HIGH	CS HIGH
1	1	3	CS HIGH	CS HIGH	CS HIGH
1	1.03	1	CC HIGH	OK	OK
1	1.07	1	CC HIGH	CM HIGH	CC HIGH | CM HIGH
1	1.1	1	CC HIGH	CM HIGH	CC HIGH | CM HIGH
1	1.5	1	CC HIGH	CC HIGH	CC HIGH
1	2	1	CC HIGH	CC HIGH	CC HIGH
1	3	1	CC HIGH	CC HIGH	CC HIGH
1.03	1	1	CM HIGH	OK	OK | CS LOW
1.07	1	1	CM HIGH	CM HIGH	CC HIGH | CM HIGH
1.1	1	1	CM HIGH	CM HIGH	CC HIGH | CM HIGH
1.5	1	1	CM HIGH	CM HIGH	CM HIGH
2	1	1	CM HIGH	CM HIGH	CM HIGH
3	1	1	CM HIGH	CM HIGH	CM HIGH
1	1	0.97	CS LOW	OK	CS LOW | OK
1	1	0.93	CS LOW	CS LOW	CS LOW
1	1	0.89	CS LOW	CS LOW	CS LOW
1	1	0.85	CS LOW	CS LOW	CS LOW
1	1	0.5	CS LOW	CS LOW	CS LOW
1	1	0.1	CS LOW	CS LOW	CS LOW
1	0.97	1	CC LOW	OK	OK
1	0.93	1	CC LOW	CC LOW	CC LOW | CM LOW
1	0.89	1	CC LOW	CC LOW	CC LOW | CM LOW
1	0.85	1	CC LOW	CC LOW	CC LOW | CM LOW
1	0.5	1	CC LOW	CC LOW	CC LOW
1	0.1	1	CC LOW	CC LOW	CC LOW
0.97	1	1	CM LOW	OK	OK
0.93	1	1	CM LOW	CC LOW	CC LOW | CM LOW
0.89	1	1	CM LOW	CM LOW	CC LOW | CM LOW
0.85	1	1	CM LOW	CM LOW	CC LOW | CM LOW
0.5	1	1	CM LOW	CM LOW	CM LOW
0.1	1	1	CM LOW	CM LOW	CM LOW
0.99	0.98	1.02	OK	OK	OK
1	1.02	1.02	OK	OK	OK
0.98	1	0.98	OK	OK	OK
0.98	1.02	1.02	OK	OK	OK
0.99	1.01	1.01	OK	OK	OK
1.01	1	0.99	OK	OK	OK

We can see that diagnosis methodology with simple labelled doesn't offer a correct diagnostic in tests that are very near of the correct behaviour. In those cases the fault is not detected. Other times, methodology returns an incorrect diagnosis, but in general offered results are acceptable.

This occurs because there are very similar trajectories belonging to different behaviours, and classifier cannot correctly select the rules to difference them.

To solve this problem the new methodology proposes the re-labelled of all similar trajectories as have been above mentioned. Obtained results show that the new methodology offers a multiple diagnosis when the previous one can't find the correct fault. Among the multiple offered diagnoses, near to all tests return the correct one.

It is important to highlight that, in tests where behaviour is far of the correct one, offered diagnosis is the correct one.

In the set of presented tests the diagnosis is correct in 58.33 % of the cases. Correct diagnosis is offered, among others, in 30.55 % of the cases. An incorrect diagnosis is offered in 2.7 % of the cases. The fault is not detected in 8.33 % of the cases. Otherwise, never detect failure when failure doesn't exist.

5 CONCLUSIONS AND FURTHER WORKS

Presented methodology is able to perform diagnosis of dynamic systems and it is independent of the system type. In fact, one of further works is to apply this methodology to a non-linear dynamic system.

This capacity is due to the fact that the methodology is only centred in the evolution characteristics of the system for the correct behaviour or faulty behaviours.

Another characteristic of the methodology is that the diagnosis can be performed in a very simple way, and a very little computational time is required.

Certain systems, as the presented in the example, can produce similar behaviours for different fault reasons. This is due to relationship among variables that govern the system behaviour. This relationship, among system variables, can produce that an alteration of a variable would be compensated by the alteration of another variable in contrary sense. To solve this problem, methodology assigns multiple fault reasons to system behaviours that could be produced by different fault reasons. This way a multiple diagnosis is offered in those situations.

Another further work is to be able to diagnose dynamic system when multiple fault occurs at the same time, is to say, identifying system behaviours when more than one component is faulty.

ACKNOWLEDGMENTS

This work has been partiality financed by the Comisión Interministerial de Ciencia y Tecnología (DPI2000-0666-C02-02) and the Modelización Matemática Redes y Multimedia investigation group of the University of Huelva.

REFERENCES

[1] C. J. Alonso, J. A. Maestro, J. B. Pulido y C. Llamas. *Monitorización de Sistemas Dinámicos: hacia una Caracterización en el Nivel de Conocimiento*. In proceedings of the I Jornadas de Trabajo sobre Diagnosis. Valladolid 2001.

[2] W. Hamscher. *Diagnosis devices with hierarchic structure and known component failure models.* In proceedings of the 6th Conference on AI Applications.. 1990

[3] Dague, P y otros. *When Oscillators stop oscillating*. In proceedings of IJCAI-91

[4] J. De Kleer y B. Williams. *Diagnosing multiple faults.* Artificial Intelligence 32, 97-130, 1987

[5] K. Bousson, y L. Trave-Massuyes A computational causal model for process supervision. Technical Report 92147, LAAS-CNRS, Toulouse, France. 1992

[6] P. Mosterman *Hybrid dynamic systems: a hybrid bond graph modeling paradigm and its applications in diagnosis*. Tesis Doctoral Vanderbilt University, Nashville, Tennessee, USA. 1997.

[7] P. Struss. Fundamentals of model-based diagnosis of dynamic systems. Proc. IJCAI'97. 1997.

[8] B. Pulido, C. Alonso y F. Acebes. *Consistency-based Diagnosis of Dynamics Systems using quantitative models and off-line dependency-recording*. In proceedings of DX-01.2001

[9] A.D. Pouliezos y G.S. Stavrakakis. Real time fault monitoring of industrial process. Microprocessor-based systems engineering. Kluwer Academic Publishers, Dordrecht, 1994.

[10] V. Venkatusubramanian and K. Chan. *A neural network methodology for process fault diagnosis*. Journal of Artificial Intelligence in Chemical Engineering, 35:1993-2001. 1995

[11] S. Leonhart y M. Ayoubi. *Methods of fault diagnosis*. Control Engineering Practice, 5. 1997.

[12] A. Simón, L. Alonso y A. Antón. *Sistema híbrido borroso para ayuda del diagnóstico del glaucoma*. In proceedings of the I Jornadas de Trabajo sobre Diagnosis. Valladolid 2001.

[13] S. Saludes, A. Vargas y J. R. Perán. *Aplicación de la red neuronal SOM para la detección de fallos desconocidos en un grupo hidroeléctrico.* In proceedings of the I Jornadas de Trabajo sobre Diagnosis. Valladolid 2001.

[14] J. Ross Quinlan. *Induction of decision trees.* Machine learning, 1986

[15] Juan J. Rodríguez, Carlos J. Alonso y Q. Isaac Moro. *Clasificación de patrones temporales en sistemas dinámicos mediante Boosting y Alineamiento dinámico temporal.* In proceedings of the I Jornadas de Trabajo sobre Diagnosis. Valladolid 2001.

[16] Pedro J. Abad y Antonio J. Suárez. Diagnosis de Sistemas Dinámicos Basada en Aprendizaje Supervisado *Off-line.* In proceedings of the I Jornadas de Trabajo sobre Diagnosis. Valladolid 2001.

[17] Antonio J. Suárez y Pedro J. Abad. *Aplicación de Técnicas de Aprendizaje a la diagnosis de Sistemas Dinámicos con Etiquetado Múltiple*. In proceedings of the IX CAEPIA –TTIA. 2001.

[18] A. Panati and D. T. Drupé *Stated based vs simulation-based diagnosis of dynamic system.* ECAI2000. !4Th European Conference on Artificial Intelligent. 2000.

[19] A. Malik and P. Struss. *Diagnosis of Dynamic Systems does not necessarily require simulation.* Workshop Notes of the Seventh International Workshop on Principles of Diagnosis DX-96 Montreal. 1996.

[20] J. W. Forrester *Principles of systems.* Wright-Allen Press 1968.

[21] J. Ross Quinlan. *C45:Program for Machine Learning.* Morgan Kaufman, 1993.