Fault Detection and Isolation by Using Bayesian Networks: A Case Study

Giuseppe Nunnari, F. Cannavò
Dipartimento di Ingegneria Elettrica, Elettronica e dei Sistemi
Università di Catania
Viale A. Doria, 6, 95125 Catania
Italy
gnunnari@diees.unict.it, fcannano@diees.unict.it

Radu Vrânceanu
Department of Signal Circuits and Systems
The "Gh. Asachi" Technical University of Iasi
Bvd. Carol 11, Iasi 700050
Romania
radu@altastiinta.ro

Abstract – **This paper presents a Fault Detection and Isolation (FDI) approach based on the use of Bayesian networks (BN). The peculiarity of the proposed approach is that an analytical dynamic model of the process to be monitored is not required. Instead it is hypothesized that input/output measures performed on the considered process during different working conditions, including faults, are available. In the paper the proposed FDI approach is described and the performances are evaluated considering a standard benchmark consisting of an hydraulic actuators available in literature. The goodness of the proposed approach is assessed by using appropriate performance indices. An intercomparison among the BN approach and two others approaches, namely a Multilayer Perceptron (MLP) neural network and a Fuzzy network (NF) is given. Results show that the BN approach outperforms in same case the MLP and the NF approach but it requires a high design and computational effort.**

I. INTRODUCTION

The problem of Fault Detection and Isolation (FDI) is a complex task of paramount importance for industrial applications. Several authors have proposed a large variety of techniques, based upon both physical and analytical redundancy [1], which can be roughly classified as model based and non-model based. Despite model based approaches are widely studied they cannot be applied when an accurate analytical model of the plant is not available. For this reason other approaches have been studied, mainly investigating in the Artificial Intelligence field. It is to be stressed here that several fault-symptom relations are not deterministic but rather probabilistic. Indeed, given a fault, the symptom may be detected with a certain probability, and, moreover, probability may change depending on particular working conditions. It is straightforward to conclude that probability theory provides an interesting approach to solve FDI problems. Among probability theory based approaches Bayesian networks (BN) represent a powerful tool to dealing at the same time with uncertainty, probability and graph theory [2]. In particular Bayesian Networks has been usefully considered as a very attractive alternative technology for diagnostic decision tools by [3] for complex nuclear power systems and by [4] for chemical plants. Others applications are described in [5] and [6]. To further investigate the peculiarities of the BNs based approach, in this paper we report some results concerning the application of Dynamic Bayesian Networks (DBN) to a benchmark concerning hydraulic actuators installed on particular industrial process. The performances of the proposed BN approach are evaluated according to a predefined set of indices are compared with that obtained by using a Multi-layer Perceptron based approach (MLP).

II. THE PROPOSED FDI APPROACH

The purpose of this section is to clarify the basic terms and to describe how to formulate a FDI problem in terms of Bayesian networks.

A Bayesian network, also referred as a belief network or a causal network in literature, is a graphical representation of relationships within a problem domain [5]. The peculiarity of a Bayesian network is to compute joint probability distributions over a set of random variables X using an approach consisting of the following steps:

- Find a directed a-cyclic graph (DAG), denoted by G, with nodes corresponding to random variables in X.
- Find a set of Conditional Probability Distributions (CPD), one for each node in G, represented graphically by directed arcs between nodes.

The conditional independences shown in DAG simplifies the computation of joint probability distribution (1) limiting the products to the parents of each random variable (*pa(xi)*).

$$P(x_1 \ldots x_n) = \prod_i P(x_i \mid pa(x_i)) \qquad (1)$$

Diagnostic inferences are also possible by Bayesian Networks getting conditional probability when some variable values are known (priors).

During this calculation, if some state of nodes is not available, the propagation of priors finds the best hypothesis consistent with the actual data.

In a BN for FDI, priors are typically represented by measurements, symptoms and/or other evidences about the physical system. The physical system, in this approach, is described by stochastic time series of I/O measures. Its behavior is described in terms of a system state which evolves stochastically at discrete time steps. The system state is modeled by a set of random variables X. Physical systems commonly comprise both continuous and discrete quantities. To introduce the time variable in the framework of probabilistic models, Dynamic Hybrid Bayesian Networks (HDBN) are needed, whose nodes are random variables in two consecutive time steps.

Each measured variable and other additional knowledge (e.g. due to human experts, other FDI models etc) are represented by random nodes. If a given variable is continuous the node is also continuous and the probability distribution is supposed to be Gaussian. If

the variable is discrete the node is also a discrete ones and the probability is supposed uniformly distributed, unless other information is available. Variables that represent failure nodes of the system are typically Boolean, the two states representing the normal and fault behavior respectively. To build a dynamic net it is necessary to insert a certain number of regression nodes representing the values of given variables at previous time instants. The number of regressions for each variable depends on the particular dynamic system considered. A crucial step is the FDI process is to find a DAG. The influence arcs are inserted on the base of a-priori information, heuristic knowledge, temporal causal graph of the system, or evidences obtained from the structure learning algorithms. In a simplistic network the fault node influences all measures.

The hybrid dynamic Bayesian network (HDBN) represents the complete fault state of the system at each time step [7]. A fault variable in belief state takes into consideration all the evidences available up to the present time in order to determine a probability distribution of a residual for the considered fault. In the FDI approach considered in this paper, a BN for each class of fault was implemented. This implies that a unique fault node is present in a given BN, as represented in the example reported in the next section. Since an individual BN does not represent the whole system behavior, appropriate fault examples must be contained in the fault data base.

III. THE CONSIDERED CASE STUDY

The proposed algorithm has been tested on a public domain benchmark appropriately studied to provide the possibility of evaluating and ranking research approaches. The benchmark have been considered in the framework of the DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems) http://diag.mchtr.pw.edu.pl/damadics/benchmark.html) and concern a particular kind of hydraulic actuators installed on a sugar plant. The main goal of the benchmarks is the generation of well defined, repeatable single actuator faults. The structure of such an actuator is presented in Fig. 1.

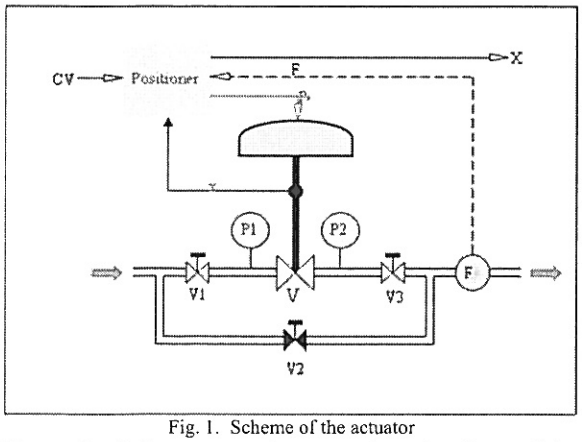

Fig. 1. Scheme of the actuator

The task of the actuator is to regulate the flow of the liquid circulating through a plant for juice production, taking into account some inputs values and the external conditions. Three mainly parts can be identified inside the actuator: the *positioner,* the *pneumatic servo-motor* and the *control valve.* The control valve is the element used to prevent, allow and/or limit the flow of fluids through the control systems. Changing the state of the control valve is accomplished by a servomotor. The pneumatic servomotor is a device which provides a linear motion of the servomotor stem. The *positioner* is a device applied to eliminate the control-valve-stem miss-positions produced by the external or internal sources such as friction, pressure unbalance, hydrodynamic forces etc. The original benchmark is based on the use of both Simulink-Matlab model of the actuators and on real process data files with artificial generated faults. However in this paper only patterns generated by using the Simulink model are considered. The analytical model of actuator is not available as is typical when considering industrial implementation The original benchmark is characterised by the following limitations:

- The considered number of different classes of faults is equal to 19,
- two fault simulation scenarios are assumed: abrupt and incipient.
- only single fault scenarios are considered and simulated.

In addition to these limitations in this paper we have considered the following restriction: results refer to abrupt faults only. Moreover, for the lack of pages we reports results concerning to six of the ninentheen possible faults. According with the symbols considered in the original benchmark, the six considered faults will be referred as:

f1 – valve clogging
f7- medium cavity or critical flow
f15 – positioner supply pressure drop
f17- pressure difference drop on valve
f18 – fully or partly opened bypass valve
f19 – flow rate sensor fault

IV. THE STRUCTURE OF THE HDBN

The BN designed to address the considered FDI problem is shown in Figure 2.

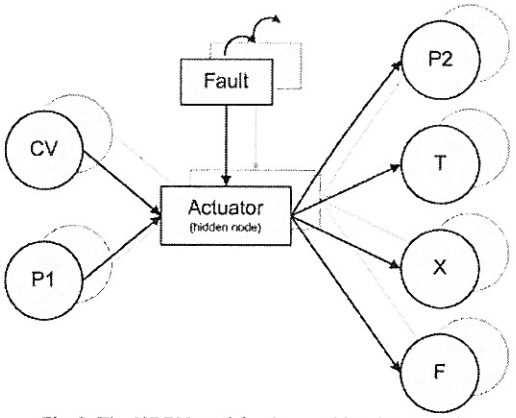

Fig. 2. The HDBN used for the considered case study

The meaning of the nodes represented in this figure is the following:

- P1 – represents the pressure from the inlet valve (input, continuous node);
- P2 –represents the pressure sensor from the outlet valve (output, continuous node);
- CV – represents the control signal from external controller (input, continuous node);
- T – represents the temperature of the liquid (output, continuous node);
- X –represents the piston rod displacement (output, continuous node)
- F – represents the process media flow meter (output, continuous node);
- Fault –represents an individual class of fault (discrete node with 2 states: False and True).
- Actuator – a hidden node that models the system and through which the relations between all the other nodes are learned (discrete node with 8 states, the number of states were determined by a trial and error approach).

The shaded circles, boxes and arrows shown in Fig. 2 are a way to represent the dynamic behaviour of the network. In more detail, since a HDBN consists of an initial DAG and of a transition DAG containing the nodes and links among variables at two consecutive time slice, we have represented the initial DAG by using continuous line and the transition DAG by shadow lines. A network for each fault was preferred instead of one large network with all the faults implemented, due to the exponential increase, with the number of faults, of the size of the Actuator node, the time to compute the inference and memory needed for this task. The structure of the HDBN shown in Fig. 2 was obtained after a trial-and-error approach. The total set of variables (CV, P1, P2, T, X and F) was heuristically divided in two sets referred here as the input set (CV, P1), and output set (P2, T, X, F) respectively. The output set was obtained considering those variables which exhibit significant changes when a fault occurs, while the input set includes those variables that are independent from faults.

The BN used to address this FDI problem was developed in Matlab using the BNT Toolbox written by Kevin Murphy (http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html). This toolbox was chosen due to its very good implementation of the inference algorithms, the possibility of implementing real dynamic Bayesian Networks, the possibility to work with discrete nodes that have continuous parents (thanks to the softmax CPDs), its computational power and the fact that it is free. Also, being written in Matlab it was much easier to manipulate the data, for training and testing, generated with the Simulink model of the actuator. On the other hand the toolbox is a little bit slower than its equivalent written in C. The data used for training the BNs networks was thought such that it provides training patterns for a variety of cases. For testing, the DGEN block (provided in the Simulink model) was used to generate data. A number of 1500 patterns were used for each fault case and the simulation time was more or less 15 min. for each fault on a 2.4 GHz computer with 384 MB RAM.

V. PERFORMANCE INDICES

In order to evaluate the performance of the considered FDI approaches, an appropriate set of indices were considered whose definition is reported below according with the symbols in Fig. 3

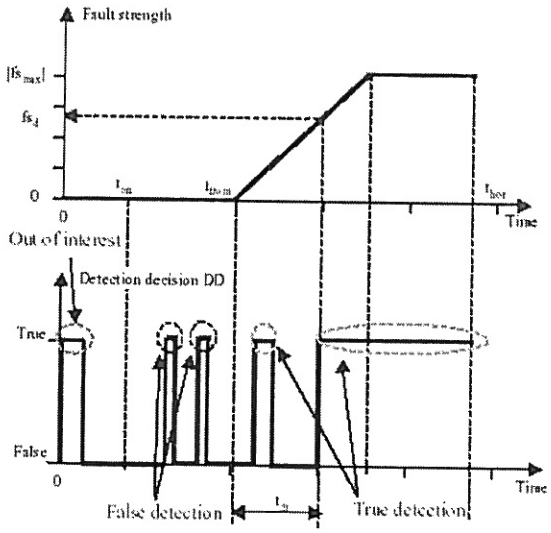

Fig. 3. Explanation of parameters used in performance indices

Detection time (t_{dt}): period of time from the begin of fault start-up to the moment of the last leading edge of the detection decision (DD) signal.

False detection rate (r_{fd}):

$$r_{fd} = \frac{\sum_i t_{fd}^{i,DD}}{t_{from} - t_{on}} \quad (2)$$

where $t_{fd}^{\,i,\,DD}$ is a i-th period of high DD signal value between t_{on} to t_{from}.

True detection rate (r_{td})

$$r_{td} = \frac{\sum_i t_{td}^{i,DD}}{t_{hor} - t_{from}} \quad (3)$$

where $t_{td}^{\,i,\,DD}$ is a i-th period of high DD signal value between t_{from} to t_{hor}.

Isolation time (t_{it}): Period of time from the begin of fault start-up to the moment of the last leading edge of the Primary Isolation Decision (PID) signal.

False isolation rate (r_{fi}):

$$r_{fi} = \frac{\sum_i t_{fi}^{i,PID}}{t_{from} - t_{on}} \quad (4)$$

where $t_{fi}^{\,i,\,DD}$ is a i-th period of high PID signal value between t_{on} to t_{from}.

True isolation rate (r_{ti}):

$$r_{ti} = \frac{\sum_{i} t_{ti}^{i,PID}}{t_{hor} - t_{from}} \quad (5)$$

It is straightforward to understand that the best situations in terms of t_{dt} (or t_{it}) are obtained when its value approach zero. The same consideration can be made for r_{fd} (or r_{fi}), while the best value for r_{td} (or r_{ti}) is 1.

VI. INTERCOMPARED TECHNIQUES

In order to appreciate the goodness of the proposed approach, an inter-comparison among BN based approach and two different techniques has been performed. The first technique is based on the use of Multilayer Perceptron (MLP) neural networks while the second considered approach is based on a Neuro-Fuzzy (NF) approach.

The MLP approach was implemented by the authors while results reported for the NF approach are due to Jegorow and Wasiewicz [8] who have separately considered the same benchmark. The interested reader's can read the paper [8] for detail about the implementation of the NF approach. Some detail concerning the MLP approach is reported below.

The MLP neural networks were trained as classifiers. In particular six classifiers were implemented, each devoted to detect and isolate one of the six different fault classes. MLP with one hidden layer were considered and the best number of neurons in the hidden layer was determined by a trial-and-error approach. In this work this number was chosen to be 6 and adopted for all the considered classifiers. The number of input neurons was set to 18 since the number of input variable is 6 (i.e. CV, P1, P2, X, F, T1) and for each of them 3 regression were considered. The regression time slice was chosen to be 20 seconds in order to avoid drawbacks due to the transient behaviours. Bearing in mind the above considerations it is easy to conclude that the structure of each classifier was set to 18-6-1 (i.e. 18 neurons in the input layer, 6 in the hidden layer and 1 in the output layer). In order to train a MLP classifier to detect and isolate a given fault class, the training patterns were obtained setting the output values to 0 if the pattern represents the fault-free class or if it belongs to a different fault class, otherwise it was set to 1. In this way, each MLP can detect and, at the same time, isolate only one fault class.

Data used for generating both training and testing patterns has been generated in order to have as many operating points as possible. To satisfy this requirement the frequencies of P1, P2 and CV have been chosen as prime numbers. Moreover CV was varied from 0.3 to 0.7 through sinusoidal function with frequency of 0.01 Hz. The same function was also used to vary P1 from 0.83 to 0.92 with frequency of 0.51 Hz and P2 from 0.64 to 0.66 with frequency of 0.57 Hz. All variable values were normalized in the range [0, 1].

The block used to generate data was the Act block of the DAMADICS Simulink Library. For each fault, we have generated learning data for 1500 samples with period of 1 second. The faults were simulated from second 900. For testing we used the DGEN block, also provided in the benchmark library, to generate data that simulates 2000 seconds of activity.

VII. SIMULATION RESULTS

Results obtained by using the Bayesian Network shown in Fig. 2 (one for each class of fault) are summarised in terms of performance indices in Table I, while Table II and III summarise results for the MLP and the NF approach respectively

TABLE I

BEST PERFORMANCE INDEXES OBTAINED BY BAYESIAN NETWORK APPROACH.

Fault	t_{dt}[sec]	r_{fd}	r_{td}	t_{it}[sec]	r_{fi}	r_{ti}
F1	0	0.057	1	0	0.057	1
F7	0	0	1	0	0	1
F15	4	0.004	0.996	4	0.004	0.996
F17	0	0	1	0	0	1
F18	0	0.078	1	0	0.078	1
F19	ND	0	0	ND	0	0

TABLE II

BEST PERFORMANCE INDEXES OBTAINED BY NEURAL CLASSIFIER APPROACH.

Fault	t_{dt}[sec]	r_{fd}	r_{td}	t_{it}[sec]	r_{fi}	r_{ti}
F1	23	0	0.979	23	0	0.979
F7	75	0	0.981	75	0	0.981
F15	14	0	0.97	14	0	0.97
F17	0	0.002	1	0	0.002	1
F18	11	0	0.859	11	0	0.859
F19	ND	0	0	ND	0	0

TABLE III

BEST PERFORMANCE INDEXES OBTAINED BY MIXED NEURO-FUZZY APPROACH [8].

Fault	t_{dt}[sec]	r_{fd}	r_{td}	t_{it}[sec]	r_{fi}	r_{ti}
F1	5	0	0.98	5	0	0.98
F7	5	0	0.98	5	0	0.98
F15	231	0	0.92	240	0	0.66
F17	6	0	0.97	35	0	0.90
F18	5	0	0.98	5	0	0.98
F19	7	0	0.97	7	0	0.97

As a general comment it is possible to say that all the three inter-compared techniques perform quite well. However a ranking of these techniques in terms of t_{dt}, r_{td}, t_{it} and r_{ti} points out that the BN approach outperform both the MLP and NF while these latter techniques work better than BN in terms of r_{td} and r_{fi}.

Details about the capability of the proposed BN approach during the detection and isolation of samples of the six considered fault classes are shown in Fig. 4a) trough 4f). For the lack of pages the corresponding figures for the MLP approach are given for faults f1 (Fig. 5a) and f15 (Fig. 5b) only. Comparing Fig. 4a and

Fig. 5a, both referring to fault f1, we can observe that MLP exhibits a longer detection delay with respect to the BN approach but the transient behavior seems *cleaner*. On the contrary, comparing Fig. 4c and Fig. 5b, both referring to fault f15, the transient behavior of the BN is better than that shown by the MLP. However, in general it seems that the BN works as well as the MLP approach or better. It is to be stressed that only fault f19 was undetectable by using BN but this fault was undetectable also by using MLP, while, as claimed in [8] it was detectable by using the NF approach.
A ranking of the inter-compared approaches in terms of design and computational effort says that BN requires in general higher efforts with respect to both MLP and NF. Indeed for medium skilled designers it is usually more difficult to obtain an acceptable architecture of an HDBN with respect to the effort needed to find the structure of a MLP or NF neural networks. Furthermore the algorithms available for training a BN require a higher computational effort with respect to the training of a MLP or NF network that is usually accomplished by using the traditional Backpropagation algorithms. The use of approximate algorithms for making inferences with BN, as done in this work, could help to reduce the computational effort that however remains still higher that that required by MLP or NF.

VIII. CONCLUSIONS

In this paper a Bayesian Network FDI based approach has been proposed and applied to a benchmark considered in literature consisting of an hydraulic actuators which is a subsystem of several industrial plants. The results shows that the implemented systems performs quite well in terms of the most common performance indices usually considered for such a kind of application. The comparison with other techniques traditionally considered for FDI applications shows that there could be some advantage in terms of performances but there are also some drawbacks in terms of design and computational effort. However it is to be stressed that the results reported in this paper have been obtained working on data generated by a SIMULINK model of the real system. This means the further confirmation about the reliability of the proposed approach on real process data. Work is in progress to this end.

IX. ACKNOWLEDGES

This paper has been supported by the research project "Rivelazione e diagnosi di malfunzionamenti, riconfigurazione del controllo e monitoraggio delle prestazioni nei processi industriali, PRIN (Progetti di Rilevante Interesse Nazionale) 2002-2003, Ministero dell'Istruzione, dell'Università e della Ricerca.

X. REFERENCES

[1] Simani, S., Fantuzzi C., Patton R.J., Model-base Fault Diagnosis in Dynamic Systems Using Identification Techniques, Springer, 2002.

[2] Karny, M., Hangos K., Tesar L.; On probabilistic modeling in fault detection, European Control Conference. ECC '99, Karlsruhe, 1999.

[3] Kang C.W. ; Golay M.W., A Bayesian belief network-based advisory system for operational availability focused diagnosis of complex nuclear power systems, Expert Systems with Applications , volume 17, pages 21--32, 1999.

[4] Lerner U., Parr R., Koller D., Biswas B., Bayesian fault detection and diagnosis in dynamic, in AAAI/IAAI, pages 531--537, 2000.

[5] Przytula, K.W.; Thompson, D., Construction of Bayesian networks for diagnostics, Aerospace Conference Proceedings, 2000 IEEE , volume 5, pages 193--200, March 2000.

[6] Karny, M., Hangos K., Tesar L.; On probabilistic modeling in fault detection, European Control Conference. ECC '99, Karlsruhe, 1999.

[7] R. E. Neapolitan, *Learning Bayesian Neyworks,* Prentice-Hall, 2004, pp. 205-265

[8] Sergej Jegorov, Piotr Wasiewicz "Actuator Benchmark Results: Step I and Step II", 5-th DAMADICS Workshop, Łagów, Poland, April 5th-7th, 2004.

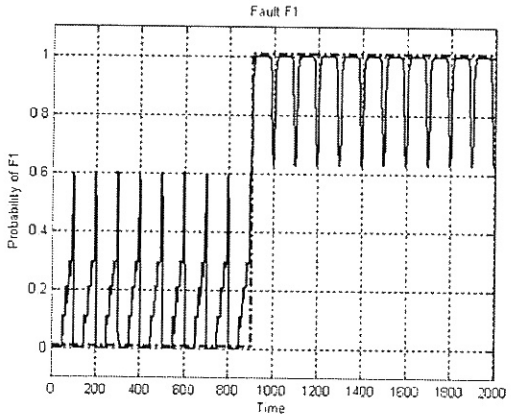

Fig. 4a. Output of BN used to detect F1, in terms of probability of fault occurrence (continuous line). The fault occurs at second 900 (dotted line).

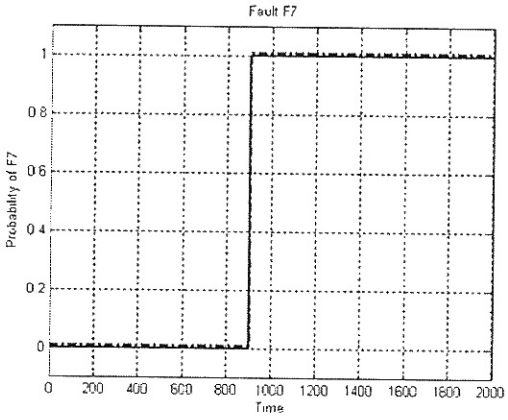

Fig. 4b. Output of BN used to detect F7, in terms of probability of fault occurrence (continuous line).

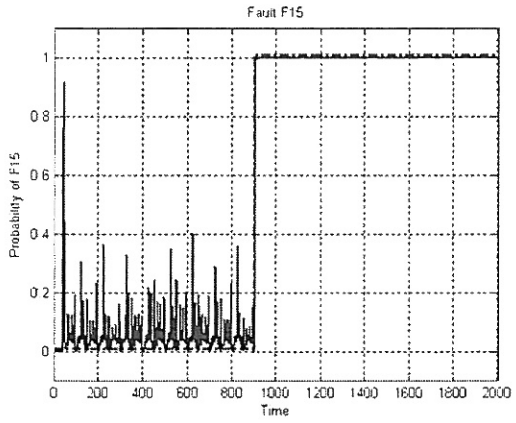

Fig. 4c. Output of BN used to detect F15, in terms of probability of fault occurrence (continuous line). The fault occurs at second 900 (dotted line).

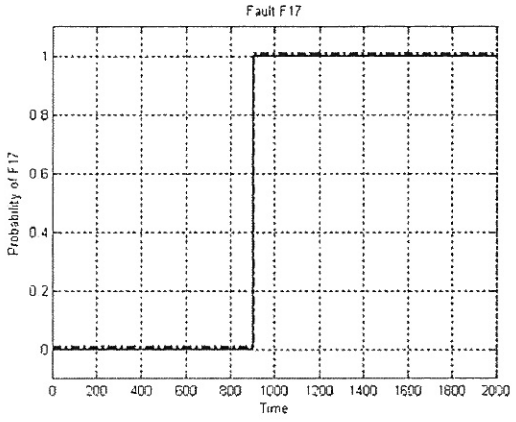

Fig. 4d. Output of BN used to detect F17, in terms of probability of fault occurrence (continuous line). The fault occurs at second 900 (dotted line).

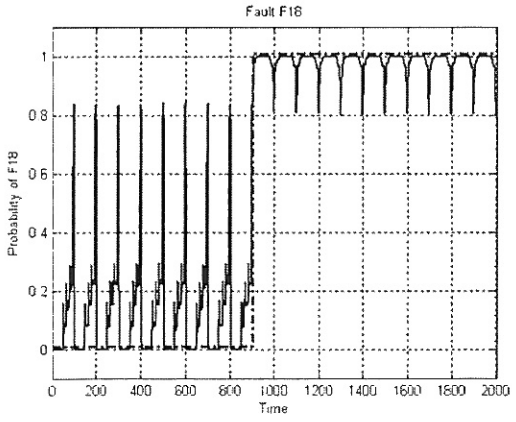

Fig. 4e. Output of BN used to detect F18, in terms of probability of fault occurrence (continuous line). The fault occurs at second 900 (dotted line).

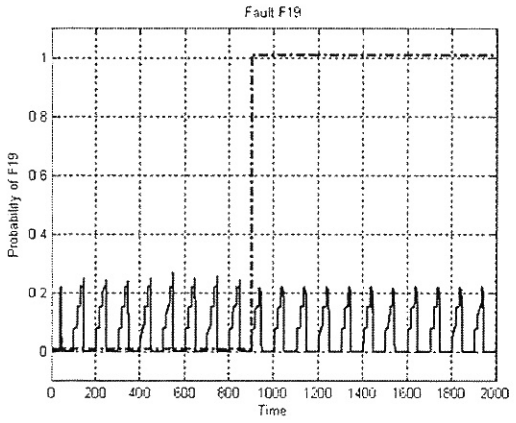

Fig. 4f. Output of BN used to detect F19, in terms of probability of fault occurrence (continuous line). The fault occurs at second 900 (dotted line).

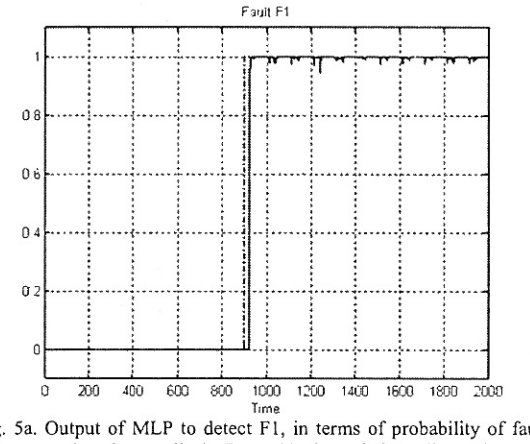

Fig. 5a. Output of MLP to detect F1, in terms of probability of fault occurrence (continuous line). For MLP the unit in ordinate does not represents a probability in strict sense. The fault occurs at second 900 (dotted line).

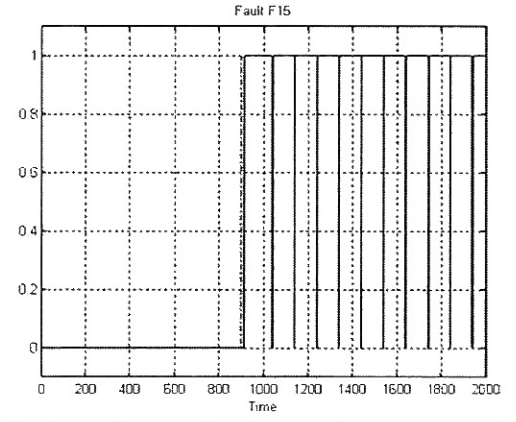

Fig. 5b. Output of MLP used to detect F15, in terms of probability of fault occurrence (continuous line). The fault occurs at second 900 (dotted line).