International Journal of Innovative Computing, Information and Control
Volume **9**, Number **8**, August **2013**

ICIC International ©2013 ISSN 1349-4198
pp. 3185–3196

FAULT DIAGNOSIS BASED ON NEURAL NETWORKS AND DECISION TREES: APPLICATION TO DAMADICS

Yahia Kourd[1], Dimitri Lefebvre[2] and Noureddine Guersi[3]

[1]Department of Electrical Engineering
University of Biskra
BP 145 RP Biskra, 07000, Algeria
ykourd@yahoo.fr

[2]GREAH
University of Havre
25 rue Philippe Lebon, Le Havre 76058, France
dimitri.lefebvre@univ-lehavre.fr

[3]Department of Electronic
University of Badji Mokhtar
Annaba 23000, Algeria
guersi54@yahoo.fr

Received June 2012; revised December 2012

Abstract. *This paper presents a new technique of fault detection and isolation (FDI). This technique is based on Neural Networks fault-free and Faulty behaviors Models (NNF Ms). NNFMs are used for residual generation, while decision trees are introduced for residual selection and evaluation. Each part of the tree corresponds to specific residuals and with the decision tree it becomes possible to take the appropriate decision regarding the actual process behavior by evaluating few residuals. In comparison to the systematic evaluation of all residuals, the proposed technique requires less computational effort and is suitable for on line diagnosis implementation. An application to DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems) Actuator system is presented to illustrate and confirm the effectiveness and the accuracy of the proposed approach.*

Keywords: Fault detection, Fault diagnosis, Model of fault-free behaviors, Faulty behaviors model, Neural networks, Residual evaluation, Decision trees

1. **Introduction.** Fault diagnosis is an important issue because of the increased use of automation in industry. In particular, early fault detection and diagnosis methods are suitable to perform prevention actions. These methods should respect the following requirements: (1) improvement of standards and quality; (2) understanding of the causality between failures and diagnosis; (3) development of new services and technologies with economic benefit.

Major fault diagnosis methods found in the literature are based on linear methodology or exact models. The basic idea of this scheme is to generate signals that reflect inconsistencies between the measured operating system conditions and the expected ones [2-5]. Such signals, called residuals, are usually calculated by using analytical methods such as parity equations [4], observers [5] and parameter estimators [6]. Unfortunately, the common drawback of these approaches is the requirement of an accurate mathematical model of the diagnosed plant. For complex and non linear systems, the task of modeling is often tedious and analytical models cannot be computed or give unsatisfactory results. In these cases, data-based models can be considered. In particular, many researchers have perceived artificial neural networks as an alternative way to represent knowledge about faults [1-3]. Neural networks can filter out noise, disturbances and provide stable, highly sensitive and economic faults diagnosis without the use of mathematical models.

This paper presents an FDI method that generates residuals according to fault-free reference models and also to models of faulty behaviors. A large number of residuals are obtained as a consequence. In order to prevent any combinatory explosion and to provide an FDI method suitable for practical implementation with real time requirements, the residuals are filtered and selected with decision trees. Such trees decide on line which residuals must be evaluated or not. This technique is validated with the DAMADICS benchmark process, an European project under which several FDI methods have been developed and compared [25].

The paper is organized as follows. Section 2 introduces artificial neural networks to compute models of fault-free and faulty behaviors. In Section 3, residuals and decision trees are designed. In Section 4, the method is applied to the DAMADICS actuator benchmark. Finally, in the last section a conclusion about the effectiveness of this approach and future research directions are presented.

2. Neural Networks Models of Faulty and Fault-Free Behaviors.

2.1. **Related literature.** Last two decades, a great deal of attention has been paid to the application of artificial neural networks (ANNs) for fault diagnosis issues [1,3,4] because artificial neural networks provide an excellent mathematical tool for dealing with non-linear problems. ANNs have an important property according to which any continuous non-linear relationship can be approximated with arbitrary accuracy using a neural network with a suitable architecture and weight parameters [10]. Fault diagnosis can be performed by means of ANNs and decision trees were also presented in numerous works [11-15]. The main motivation for this research is to explore the potential of soft computing (SC) approaches to design models of faulty behaviors and to generate residuals for nonlinear systems [2,16-18]. Some methods based on neural networks have been developed [19,33-35]. For example, energy-efficient reprogramming in wireless sensor networks using constructive neural networks; design of clinical decision support with support vector machine; evaluation of intelligent system for diabetes monitoring; multi component fault diagnosis of rotational mechanical system based on decision tree and support vector machine; vibration based fault diagnosis of mono-block centrifugal pump using decision tree. ANNs approaches have many advantages including learning, noise prevention, and parallel data processing. They are considered as multivariable nonlinear analytical tools capable of recognizing patterns from noisy complex data [20]. Neural networks and decision tree have been widely applied in designing decision support systems; some studies find that neural networks are better than decision tree [33], while others have opposite outcomes. The proposed approach aims to combine the advantages of both methods.

2.2. **Model of fault-free behaviors.** In this paper, unknown nonlinear systems are considered with input vector $U(t) = (u_i(t))$, $i = 1, \dots, q$ and output vector $Y(t) = (y_k(t))$, $k = 1, \dots, n$. The state variables are not measurable. ANNs are introduced to generate accurate models of the system in normal operating conditions [28-30]. The comparison between the output of the system and the output $Y'_0(t) = (y'_{k0}(t))$, $k = 1, \dots, n$, of the NN model gives the error vector $E(t) = (e_k(t))$, $k = 1, \dots, n$, with:

$$e_k(t) = y_k(t) - y'_{k0}(t) \tag{1}$$

The ANN model is trained with data collected from the fault-free system utilizing Levenberg-Marquardt algorithm with early stopping that uses three data sets (training, testing and validation) to avoid overfitting. Moreover, this algorithm is known in its fast convergence. The obtained model is then tested and validated again with other sets of data. In order to get the best model, several configurations are tested according to a trial error processing that uses pruning methods to eliminate the useless nodes.

2.3. **Models of faulty behaviors.** When multiple faults are considered, the isolation of the faults is no longer trivial and early diagnosis becomes a difficult task. One can multiply the measurements and use some analysis tools (residuals analysis) in order to isolate them. In particular, a history of collected data can be used to improve the knowledge about the faulty behaviors. This knowledge is then used to design models of faulty behaviors. Such models will be used to provide estimations for each fault candidate. The decision results from the comparison between estimations and measurements collected during system operations. The design of models for faulty behaviors is similar to the method described in Section 2.2. Each model is built for a specific fault candidate f_j considered as an additional input [30].

3. **Residual Evaluation Based on Decision Trees.**

3.1. **Fault detection method.** During monitoring, the direct comparison of the system outputs $Y(t)$ with the outputs $Y_0'(t)$ of the fault-free model leads to residuals $R_0(t) = (r_{k0}(t))$ and $k = 1, \ldots, n$ with:

$$r_{k0}(t) = y_k(t) - y_k'(t), \quad k = 1, \ldots, n. \tag{2}$$

The residual $R_0(t)$ provides information about faults for further processing. Fault detection is based on the evaluation of residuals magnitude. It is assumed that each residual $r_{k0}(t)$, $k = 1, \ldots, n$ should normally be close to zero in the fault-free case, and it should be far from zero in the case of a fault. Thus, faults are detected according to a threshold S_{k0} with Equation (3):

$$\begin{aligned} |r_{k0}(t)| \leq S_{k0} &: \text{ No fault is detected at time } t \\ |r_{k0}(t)| > S_{k0} &: \text{ A fault is detected at time } t \end{aligned} \tag{3}$$

The analysis of residuals $r_{k0}(t)$ also provides an estimation τ_k of the time of occurrence t_f. This estimation will be used for diagnosis issue. When several residuals are used, the estimation τ of the time of occurrence of faults is given by:

$$\tau = \min\{\tau_k, k = 1, \ldots, n\} \tag{4}$$

The main difficulty with this evaluation is that $y_k(t)$ is usually corrupted by disturbances (for example, measurement noise). Thus, it is necessary to assign large thresholds S_{k0} in order to avoid false alarms. Such thresholds usually imply a reduction of the fault detection sensitivity and can lead to non detections. In order to prevent non detections, one can run also the models of faulty behaviors from $t = 0$ and use the method described below. The idea is to evaluate the probability of the fault candidates at each instant. A fault is detected when the probability of one model of faulty behaviors $NNFM(j)$, $j = 1, \ldots, p$ becomes larger than the probability of the fault-free model $NNFM(0)$ [29].

3.2. **Fault diagnosis based on 3-valued residuals.** The proposed approach is based on the analysis of the residuals $R_j(t)$, $j = 0, \ldots, p$ obtained with the parallel computation of fault-free and faulty ANNs models (Figure 1). Detection and diagnosis are achieved according to a decision block.

The faulty models run simultaneously from time $t = \tau$ where τ is the time when the fault is detected. Each model will behave according to a single fault candidate and the resulting behaviors will be compared with the collected data to provide rapid diagnosis.

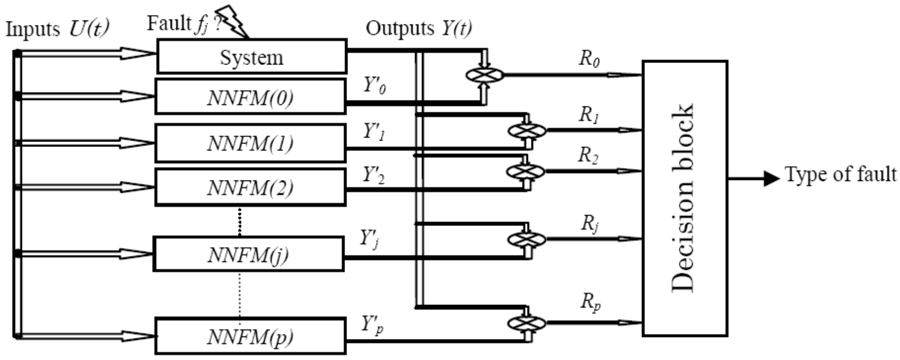

FIGURE 1. FDI design with models of fault-free and faulty behaviors

In case of numerous fault candidates f_j, $j = 1, \ldots, p$, the output $Y'_j(t) = (y'_k(t, f_j, \tau))$ of the model $NNFM(j)$ is compared with the measured vector $Y(t)$ to compute additive residual $R_j(t) = (r_{kj}(t, \tau))$, $k = 1, \ldots, n$. The fault is isolated according to residuals $r_{kj}(t, \tau)$, $k = 1, \ldots, n$, $j = 1, \ldots, p$ resulting from the n outputs and p models of faults:

$$r_{kj}(t, \tau) = y_k(t) - y'_k(t, f_j, \tau) \tag{5}$$

The diagnosis results either from the usual thresholding technique with positive and negative thresholds [5] or from the on-line determination of fault probabilities and confidence factors as explained in next section.

3.3. **Fault probability estimation.** The introduction of probabilities to evaluate the significance of each residual and the reliability of the decision is another contribution of our approach [30]. To evaluate the probability of each fault candidate let us define $R_{kj}(t, T, \tau)$ as the cumulative residuals over the sliding time interval $[\max(0, t - T), t]$ of maximal size T:

$$R_{kj}(t, T, \tau) = \sqrt{\int_{\max(0,t-T)}^{t} (r_{kj}(u, \tau))^2 . du} \tag{6}$$

Then, $D_j(t, T, \tau)$ is the Euclidean norm of the vector $R_j(t, T, \tau) = (R_{kj}(t, T, \tau))$ of dimension n that is used to decide the most probable fault.

$$D_j(t, T, \tau) = \sqrt{\sum\nolimits_{k=1}^{k=n} (R_{kj}(t, T, \tau))^2} \tag{7}$$

The most probable fault candidate is determined with (8):

$$j^*(t, T) = \text{argmin}_j \{D_j(t, T, \tau), j = 1, \ldots, p\} \tag{8}$$

A confidence factor $CF_j(t, T, \tau)$ that the current fault is f_j is also provided by (9):

$$CF_j(t, T, \tau) = \frac{\sum\nolimits_{k=1, k \neq j}^{k=p} (D_k(t, T, \tau))}{\sum\nolimits_{k=1}^{k=p} (D_k(t, T, \tau))} \tag{9}$$

$CF_j(t, T, \tau)$ is near 1 when $D_j(t, T, \tau)$ is near 0 and $CF_j(t, T, \tau)$ is far from 1 when $D_j(t, T, \tau)$ is far from 0. The proposed method uses the time window $[\max(0, t - T), t]$ that can be sized in order to satisfy real time requirements for rapid diagnosis. Multi-steps diagnosis at time t is obtained with a large window (i.e., $T = t$) and includes a diagnosis delay but will lead to a decision with a high confidence index. On the contrary single step diagnosis at time t leads to immediate diagnosis (i.e., $T = 0$) but with a lower confidence index.

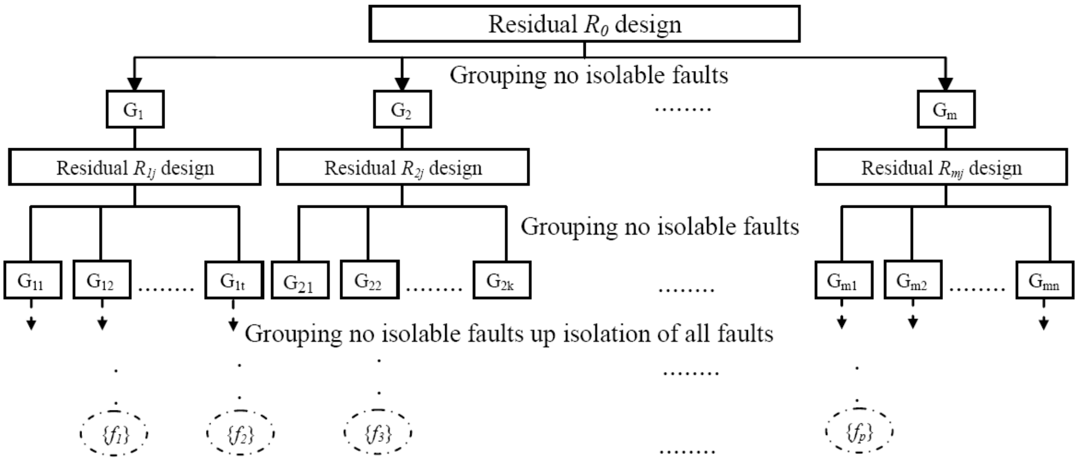

Figure 2. FDI decision tree

3.4. **Decision trees for residual evaluation.** The aim of this section is to propose a hierarchical structure to simplify the diagnosis of automation systems when numerous residuals are computed. The idea is to organize the residuals in a decision tree presented in Figure 2. This tree is used to compute only a selection of residuals that are the most significant for the current signal. The tree starts with the evaluation of the residual that corresponds to the fault free model (step 1). The value of the residual is used to classify the fault candidates in several subgroups (Figure 2, G_1 to G_m). Each subgroup limits the number of fault candidates. The algorithm continues by evaluating another residual for each subgroup resulting from the first step (step 2). Another time the fault candidates are separated into subgroups (Figure 2, G_{11} to G_{1t} for example). According to that evaluation, the algorithm continues until a subset of faults with a single candidate is isolated. Finally, the faults are isolated by the computation of selected residuals. Thus the computational effort is reduced in comparison with the systematic evaluation of all residuals. In practice, the use of hierarchical architecture is feasible on-line or offline depending on the complexity of the system.

4. **Application to an Industrial System.** The DAMADICS benchmark is an engineering research case-study that can be used to evaluate FDI methods. The benchmark is an electro-pneumatic valve actuator in the Lublin sugar factory in Poland [21]. In [22], passive robustness fault detection method using intervals observers is presented. In [23], authors introduce signal model based fault detection using squared coherency functions. An actuator fault distinguishability study is presented in [24]. A data-driven method in FDI is presented in [25], where a novel classifier based on particle swarm optimization was developed. Group methods of data handling (GMDH) neural networks have been used in [26] for robust fault detection. A computer-assisted FDI scheme based on a fuzzy qualitative simulation, where the fault isolation is performed by a hierarchical structure of the neuro-fuzzy networks is presented in [27]. A neuro-fuzzy modeling for FDI, involving a hybrid combination of neuro-fuzzy identification and unknown input observers in the neuro-fuzzy and decoupling fault diagnosis scheme, has been proposed in [20].

4.1. **DAMADICS description.** The actuator consists of a control valve, a pneumatic servomotor and a positioner (Figure 3). In the actuator, faults can appear in: control valve, servo-motor, electro-pneumatic transducer, piston rod travel transducer, pressure transmitter or microprocessor control unit. A total number of 19 different faults is considered ($p = 19$) [23,30,32]. The faults are emulated under carefully monitored conditions,

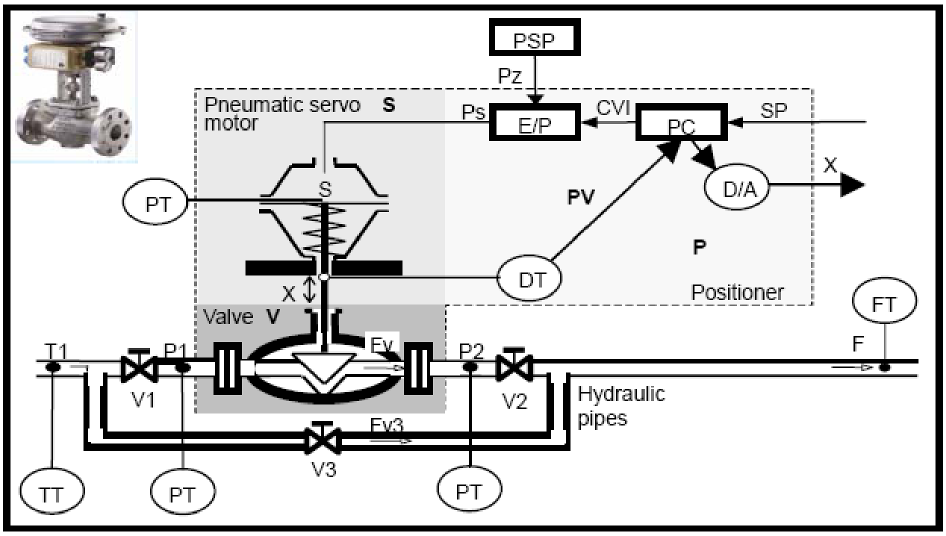

Figure 3. DAMADICS actuator

keeping the process operation within acceptable limits. Five available measurements and one control value signal have been considered for benchmarking purposes: process control external signal (CV), liquid pressures on the valve inlet (P_1) and outlet (P_2), liquid flow rate (F), liquid temperature (T_1) and servomotor rod displacement (X). Within the DAMADICS project the actuator simulator was developed under MATLAB Simulink. This tool makes it possible to generate data for the normal and 19 faulty operating modes [32].

4.2. **Models design with ANNs.**

4.2.1. *Model of fault-free behaviors for DAMADICS.* Two Multi Layer Perceptron ANNs are designed to model the outputs $y_1(t) = X(t)$ and $y_2(t) = F(t)$ of the DAMADICS system in case of fault-free behaviors. $y'_{10}(t) = X'(t)$ and $y'_{20}(t) = F'(t)$ are the estimated values of $X(t)$ and $F(t)$ processed by ANNs:

$$(X', F') = NNFM(0)(CV, P_1, P_2, T_1, X, F) \tag{10}$$

where $NNFM(0)$ stands for the double MLP structures. To select the structure of $NNFM(0)$, several tests are carried out to obtain the best architecture to model the operation of the actuator. Table 1 provides some results obtained during this stage. The training and testing data are simulated using Matlab Simulink actuator model. Validation is done by the measured data provided by 'Lublin Sugar Factory'. From Table 1, the structure $NNFM(0)$ with 6 nodes in the first hidden layer, 3 nodes in the second hidden layer and 2 output neurons is selected in order to avoid the phenomenon of over-learning. Adding more nodes in hidden layers does not improve the performance of $NNFM(0)$.

Table 1. Structure selection for $NNFM(0)$

NNFM	Hidden layer 1	Hidden layer 2	Output layer	MS
(6, 3, 2)	6	3	2	$3.3 * 10^{-4}$
(10, 8, 2)	10	8	2	$1.49 * 10^{-4}$
(21, 12, 2)	21	12	2	$3.91 * 10^{-4}$
(26, 26, 2)	26	26	2	$4.84 * 10^{-6}$

4.2.2. *Models of faulty behaviors for DAMADICS.* The preceding method is applied to build ANNs models corresponding to the 19 fault candidates that are considered with DAMADICS benchmark. For that purpose, it is necessary to create a data base that contains samples for all faults [30] exposed to the DAMADICS system. For example, the network $NNFM(3)$ learns the mapping from $q = 6$ inputs to $n = 2$ outputs when fault f_3 is assumed to affect the system from time $t = 0$. Equation (11) holds:

$$(X'_3, F'_3) = NNFM(3)(CV, P_1, P_2, T_1, X_3, F_3) \tag{11}$$

To select the structure of $NNFM(3)$, numerous tests are carried out to obtain the best architectures. The training and test data are generated using Matlab-Simulink DABLIB models [32]. The best structure is an ANN with 6 nodes in the first hidden layer, 3 nodes in the second hidden layer and 2 output neurons. Validation is done with the measured data provided by the Lublin Sugar Factory in 2001 [32].

4.3. **3-valued residuals for DAMADICS actuator.** The conditions for testing and validating FDI algorithms on DAMADICS benchmark are given in [23,32]. The residual vector $R_0(t) = (r_{k0}(t))$, $k = 1, 2$ for DAMADICS actuator is first considered for fault detection:

$$\begin{cases} r_{10}(t) = X(t) - X'(t) \\ r_{20}(t) = F(t) - F'(t) \end{cases} \tag{12}$$

where X' and F' are the outputs of the NN model of fault-free behaviors. In this application, the detection is obtained by comparing residuals with appropriate thresholds but fault probability can also be computed as developed in Section 3.3. The threshold selection is closely linked to the behavior of residuals and also to constraints that may be imposed such as security margin tolerance [31]. 3-valued signals are obtained (positive, negative and zero). The thresholds are worked out according to the standard deviation of the residual for fault-free case [30]. Let us notice that the choice of constant or adaptive thresholds strongly influences the performance of the FDI system. The thresholds must be carefully selected. For the continuation of our work, the thresholds $S_{10} = 5{*}\sigma_1 = 2.7{*}10^{-3}$ and $S_{20} = 5 * \sigma_2 = 4 * 10^{-3}$ are selected where σ_1 and σ_2 are the standard deviations obtained from the learning process. Table 2 sums up the signatures for the 19 types of faults according to the sign of the residual vector R_0. In Table 2, "+1" means that $r_{k0>} S_{k0}$; "–1" means that $r_{k0<} - S_{k0}$; and "0" means that $-S_{k0} > r_{k0} > S_{k0}$).

From Table 2, six groups of faults with similar signatures can be separated:

- Group 1 : $G_1 = \{f_3\, f_6\, f_9\, f_{18}\}$ with signature $\binom{0}{-1}$
- Group 2 : $G_2 = \{f_1\, f_7\, f_{10}\, f_{16}\, f_{17}\}$ with signature $\binom{+1}{-1}$
- Group 3 : $G_3 = \{f_5\, f_{19}\}$ with signature $\binom{0}{+1}$
- Group 4 : $G_4 = \{f_2\, f_{11}\, f_{12}\, f_{15}\}$ with signature $\binom{-1}{+1}$
- Group 5 : $G_5 = \{f_4\, f_{13}\}$ with signature $\binom{-1}{-1}$
- Group 6 : $G_6 = \{f_0\, f_8\, f_{14}\}$ with signature $\binom{0}{0}$

The faults in groups G_1 to G_5 are detected but not isolated because the signatures over r_{10} and r_{20} are similar within the group. One can also notice that the faults in group G_6 have the same signature as the fault-free behaviors. Thus faults in group G_6

TABLE 2. Signature matrix for DAMADICS Actuator with residual R_0

G_0	f_0	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}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}
r_{10}	0	+1	–1	0	–1	0	0	+1	0	0	+1	–1	–1	–1	0	–1	+1	+1	0	0
r_{20}	0	–1	+1	–1	–1	+1	–1	–1	0	–1	–1	+1	+1	–1	0	+1	–1	–1	–1	+1

Table 3. Signature matrix for group G_1

G_1	f_3	f_6	f_9	f_{18}
r_{13}	0	0	0	0
r_{23}	0	–1	–1	–1
r_{16}	0	0	0	0
r_{26}	–1	0	–1	–1
r_{19}	+1	+1	0	+1
r_{29}	–1	+1	0	+1
r_{118}	0	0	0	0
r_{218}	–1	–1	–1	0

Table 4. Signature matrix for group G_2

G_2	f_1	f_7	f_{10}	f_{16}	f_{17}
r_{11}	0	+1	+1	+1	+1
r_{21}	0	0	–1	–1	–1
r_{17}	–1	0	+1	+1	+1
r_{27}	0	0	–1	–1	–1
r_{110}	–1	–1	0	+1	+1
r_{210}	+1	+1	0	–1	–1
r_{116}	–1	–1	–1	0	+1
r_{216}	+1	+1	+1	0	–1
r_{117}	–1	–1	–1	–1	0
r_{217}	+1	+1	+1	+1	0

Table 5. Signature matrix for group G_3

G_3	f_5	f_{19}
r_{15}	0	0
r_{25}	0	–1
r_{119}	0	0
r_{219}	1	0

Table 6. Signature matrix for group G_4

G_4	f_2	f_{11}	f_{12}	f_{15}
r_{11}	0	+1	+1	+1
r_{22}	0	+1	+1	–1
r_{111}	–1	0	+1	+1
r_{211}	–1	0	–1	–1
r_{112}	–1	–1	0	+1
r_{212}	–1	+1	0	–1
r_{115}	–1	–1	–1	0
r_{215}	+1	+1	+1	0

Table 7. Signature matrix for group G_5

G_5	f_4	f_{13}
r_{14}	0	0
r_{24}	0	+1
r_{113}	–1	0
r_{213}	–1	0

Table 8. Signature matrix for group G_6

G_6	f_0	f_8	f_{14}
r_{10}	0	0	0
r_{20}	0	0	0
r_{18}	0	0	0
r_{28}	0	0	0
r_{114}	0	0	0
r_{214}	0	0	0

cannot be directly detected with residuals r_{10} and r_{20}. For this reason additional residuals generated by adequate faulty behavior neural models $NNFM(j)$ are used for each group. The signatures matrices in Tables 3 to 8 are obtained for the 19 types of faults according to the sign of the residual vector $R_j(t)$ generated by $NNFM(j)$ with $j = 1, \ldots, 19$.

4.4. **Decision trees for the diagnosis of DAMADICS actuator.** In order to prevent the combinatory explosion with the number of residuals, we introduce decision trees to select the more significant residuals. Depending on the order used to evaluate the residuals, numerous decision trees with different sizes are obtained. For example, the decision trees in Figures 4 and 5 are both suitable for the proposed application.

According to Figure 4, the faults f_6 and f_9 are isolated with the residuals generated by the faulty models $NNFM(6)$, $NNFM(3)$ and $NNFM(0)$ (i.e., 6 residuals are required

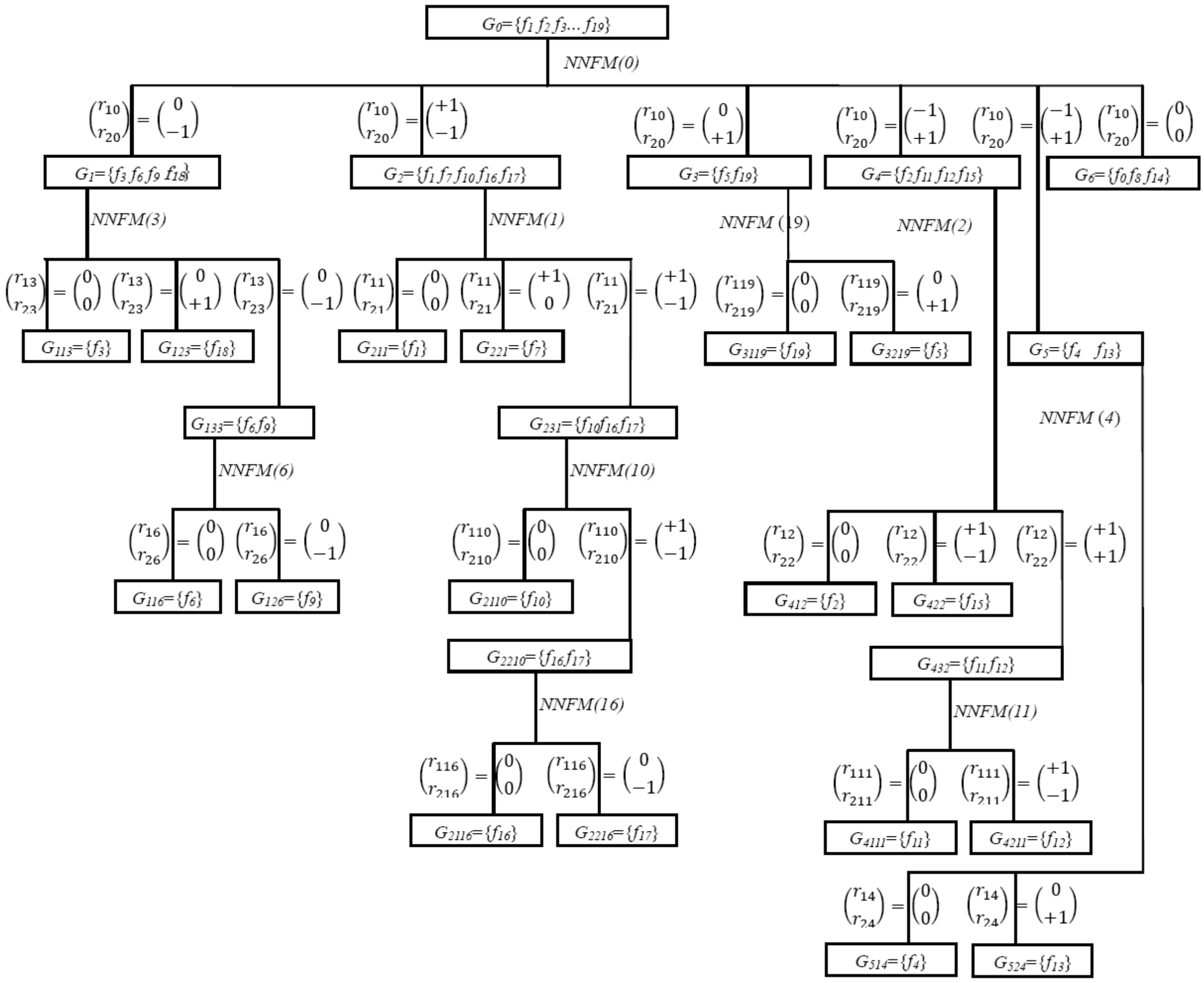

FIGURE 4. An example of decision tree for the diagnosis of DAMADICS (tree 1)

to isolate these two faults). Similarly, to isolate the faults f_{16} and f_{17}, the residuals generated by the faulty models $NNFM(16)$, $NNFM(10)$, $NNFM(1)$ and $NNFM(0)$ are required (i.e., 8 residuals are needed to isolate these two faults) and so on. Figure 5 presents another decision tree that lead to different sets of residuals suitable to isolate the faults.

4.5. **Discussion.** The works that are the more related to the present contribution are [8,24]. In [8] binary-valued evaluation of the fault symptoms is explored and the authors focus on the optimization of the neural network architecture according to Akaike Information Criteria and Final Prediction Error (FPE). Both criteria include the learning error and also a term that depends on the complexity (size of the network in number of nodes) and on the dimension of the learning set in order to optimize the ratio complexity/performance. The authors provide interesting performances with small networks for detection but some faults are not isolable. In comparison, our approach requires a larger number of ANNs with more nodes but the majority of faults are detected and isolated. In [24], multiple-valued evaluation of the fault symptoms is introduced to improve the isolation of faults. Such a method requires a heuristic knowledge about influence of faults on residuals. In comparison, our approach uses 3-valued evaluation of the residuals for fault-free behaviors and binary-valued evaluation of the residuals for faulty behaviors.

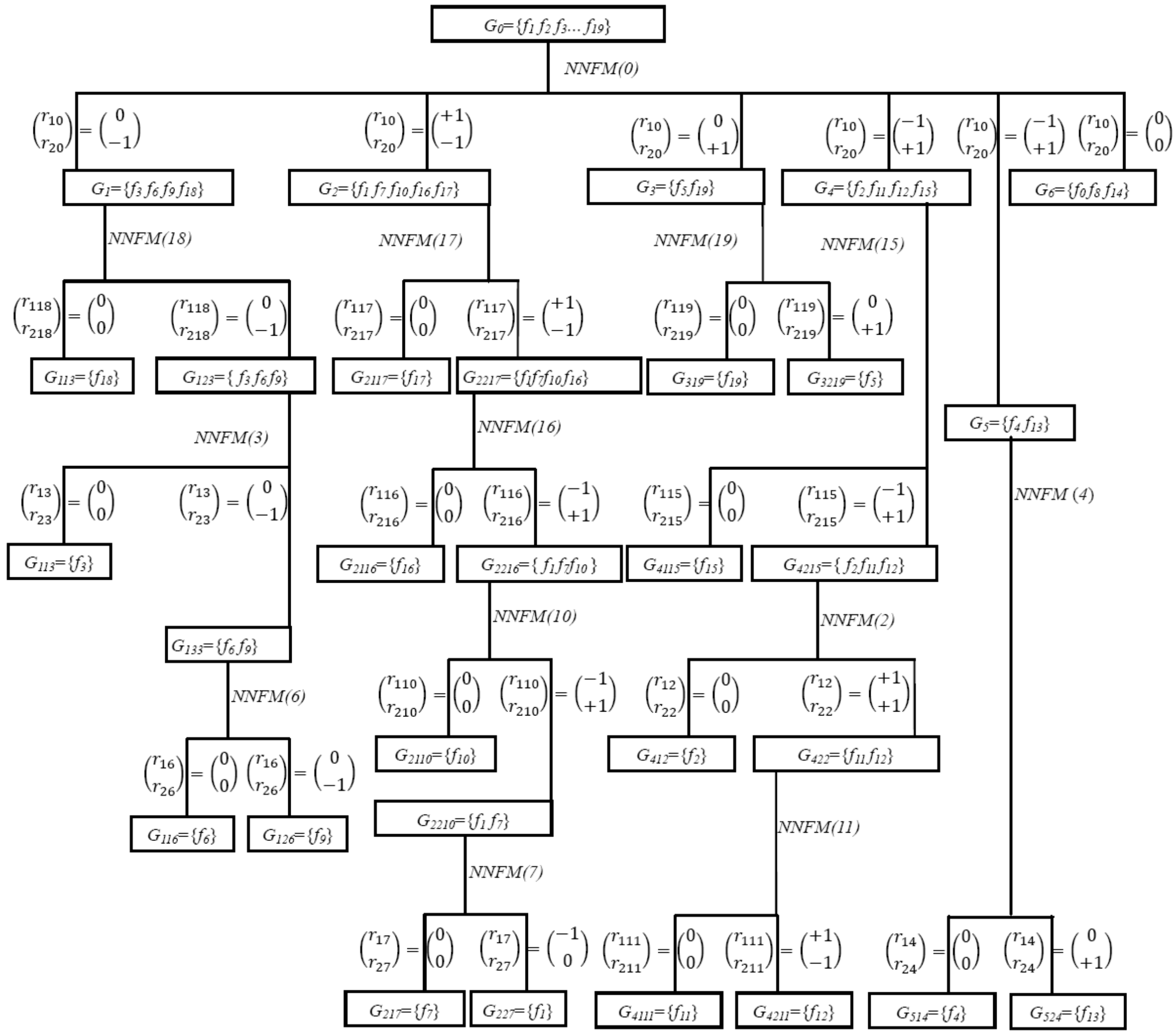

Figure 5. Another decision tree for the diagnosis of DAMADICS (tree 2)

Table 9. Numbers of residuals to isolate each fault according to trees 1 and 2

G_0	f_0	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}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}
Tree 1	2	4	4	4	4	4	6	4	ND	6	6	6	6	4	ND	4	8	8	4	4
Tree 2	2	10	6	6	4	4	8	10	ND	8	8	8	8	4	ND	4	8	4	4	4

Table 10. Comparison between the trees 1 and 2

	Tree 1	Tree 2
Maximal number of residuals	8	10
Average number of residual	4.4	5.4

In order to sum up the performances and complexity of the proposed method, the number of models required to isolate each fault is reported in Table 9 and the maximal and average depth of the trees are reported in Table 10. One can notice that only fault f_8 and f_{14} are not detectable.

Another conclusion is that the average number of residuals is about 4 with tree 1 and this measurement is an indicator of the computation effort that will be required to apply the method on-line. In comparison with the total number of fault-free and faulty models (20), the effort is divided by 10. From Table 10, one can conclude that the decision tree 1 in Figure 4 is more compact than the tree 2 in Figure 5: tree 1 will be preferred for on-line applications.

5. **Conclusion.** In this paper, a multiple-model FDI scheme is presented. ANNs technique is applied for residual generation as an alternative to the usual model-based approach. The proposed ANNs are used to diagnose the faults in the DAMADICS actuator. A probabilistic interpretation of residuals is also proposed. Then decision trees are introduced for residual selection. An iterative technique is used to isolate the fault candidates into several groups with same signatures. This study shows that the developed approach can produce good diagnosis results for a complex system exposed to numerous faults. The use of decision trees divides the computational effort by 10 and can be implemented on line.

Implementing decision trees with residuals generators for the online diagnosis will be the key issue of our interests in future research. We will also apply this scheme on other types of operating systems.

Acknowledgment. The work was supported in part by Ministry of Higher Education and Scientific Research in Algeria and by the Regional Council of Haute Normandy (Project DDSMRI 2012).

REFERENCES

[1] P. M. Frank and B. Koppen-Seliger, New developments using AI in fault diagnosis, *Engineering Applications of Artificial Intelligence*, vol.10, pp.3-14, 1997.
[2] R. J. Patton and J. Korbicz, Advances in computational intelligence, *International Journal of Applied Mathematics and Computer Science*, vol.9, 1999.
[3] J. Calado, J. Korbicz, K. Patan, R. J. Patton and J. S. da Costa, Soft computing approaches to fault diagnosis for dynamic systems, *European Journal of Control*, vol.7, pp.248-286, 2001.
[4] J. Korbicz, J. Koscielny, Z. Kowalczuk and W. Cholewa, *Fault Diagnosis: Models, Artificial Intelligence, Applications*, Springer-Verlag, Berlin, 2004.
[5] J. Chen and R. J. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Kluwer Academic Publishers, Berlin, 1999.
[6] R. J. Patton, P. M. Frank and R. Clark, *Issues of Fault Diagnosis for Dynamic Systems*, Springer-Verlag, Berlin, 2000.
[7] R. Isermann, *Fault Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance*, Springer-Verlag, New York, 2006.
[8] K. Patan and T. Parisini, Identification of neural dynamic models for fault detection and isolation: The case of a real sugar evaporation process, *Journal of Process Control*, vol.15, pp.67-79, 2005.
[9] A. Janczak, *Identification of Nonlinear Systems Using Neural Networks and Polynomial Models: A Block-Oriented Approach (Lecture Notes in Control and Information Sciences)*, Springer-Verlag, Berlin, 2005.
[10] S. Osowski, *Neural Networks in Algorithmitic Expression*, WNT, Warsaw, 1996 (in Polish).
[11] D. Pomorski and P. B. Perche, Inductive learning of decision trees: Application to fault isolation of an induction motor, *Engineering Applications of Artificial Intelligence*, vol.14, pp.155-166, 2001.
[12] A. Browne, B. D. Hudson, D. C. Whitley, M. G. Ford and P. Picton, Biological data mining with neural networks: Implementation and application of a flexible decision tree extraction algorithm to genomic problem domains, *Neurocomputing*, vol.57, pp.275-293, 2004.
[13] Y. Chen, A. Abrahama and B. Yanga, Feature selection and classification using flexible neural tree, *Neurocomputing*, vol.70, pp.305-313, 2006.
[14] V. Sugumaran, V. Muralidharan and K. I. Ramachandran, Feature selection using decision tree and classification through proximal support vector machine for fault diagnosis of roller bearing, *Mechanical Systems and Signal Processing*, vol.21, pp.930-942, 2007.
[15] P. Maji, Efficient design of neural network tree using a new splitting criterion, *Neurocomputing*, vol.71, pp.787-800, 2008.

[16] M. Brown and C. J. Harris, *Neuro-Fuzzy Adaptive Modelling and Control*, Englewood Cliffs, Prentice-Hall, NJ, 1994.
[17] J. S. R. Jang, C. T. Sun and E. Mizutani, *Neuro-Fuzzy and Soft Computing*, Englewood Cliffs, Prentice-Hall, NJ, 1997.
[18] M. Witczak and J. Korbicz, Genetic programming in fault diagnosis and identification of non-linear dynamic systems, in *Diagnostics of Processes. Models, Methods of Artifical Intelligence, Applications*, Cholewa et al. (eds.), Warsaw, Scientific Engineering Press, 2002.
[19] K. Narendra, J. Balakrishnan and M. Kermal, Adaptation and learning using multiple models, switching and tuning, *IEEE Control Systems Magazine*, pp.37-51, 1995.
[20] F. J. Uppal, R. J. Patton and M. Witczak, A neuro-fuzzy multiple-model observer approach to robust fault diagnosis based on the DAMADICS benchmark problem, *Control Engineering Practice*, vol.14, pp.699-717, 2006.
[21] M. Bartys, R. Patton, M. Syfert, S. Heras and J. Quevedo, Introduction to the DAMADICS FDI benchmark actuator study, *Control Engineering Practice*, vol.14, pp.577-596, 2006.
[22] V. Puig, A. Stancu, T. Escobet, F. Nejjari, J. Quevedo and R. J. Patton, Passive robust fault detection using internal observers: Application to the DAMADICS benchmark problem, *Control Engineering Practice*, vol.14, no.6, pp.621-633, 2006.
[23] F. Previdi and T. Parisini, Model-free actuator fault detection using a spectral estimation approach: The case of the DAMADICS benchmark problem, *Control Engineering Practice*, vol.14, pp.635-644, 2006.
[24] J. M. Koscielny, M. Bartys, P. Rzepiejewski and J. S. da Costa, Actuator fault distinguishability study for the DAMADICS benchmark problem, *Control Engineering Practice*, vol.14, pp.645-652, 2006.
[25] C. D. Bocaniala and J. Sà da Costa, Application of a novel fuzzy classifier to fault detection and isolation of DAMADICS benchmark problem, *Control Engineering Practice*, vol.14, no.6, pp.653-669, 2006.
[26] M. Witczak, J. Korbicz, M. Mrugalski and R. J. Patton, A GMDH neural network-based approach to robust fault diagnosis: Application to the DAMADICS benchmark problem, *Control Engineering Practice*, vol.14, pp.671-683, 2006.
[27] J. Calado and J. S. da Costa, Fuzzy neural networks applied to fault diagnosis, *Computational Intelligence in Fault Diagnosis*, pp.305-334, 2006.
[28] Y. Kourd, N. Guersi and D. Lefebvre, A two stages diagnosis method with Neural networks, *Proc. of ICEETD*, Hammamet Tunisie, 2008.
[29] Y. Kourd, N. Guersi and D. Lefebvre, Neuro-fuzzy approach for fault diagnosis: Application to the DAMADICS, *Proc. of ICINCO 2010*, Funchal, Portugal, 2010.
[30] Y. Kourd, D. Lefebvre and N. Guersi, Early FDI based on residuals design according to the analysis of models of faults: Application to DAMADICS, *Advances in Artificial Neural Systems*, 2011.
[31] D. Lefebvre, H. Chafouk and M. Lebbal, *Modélisation et Diagnostic des Systèmes. Une Approche Hybride*, Editions Universitaires Européennes, 2010.
[32] DAMADICS, *Website of the RTN DAMADICS*, http://diag.mchtr.pw.edu.pl/damadics, 2002.
[33] I. Kurt, M. Ture and A. T. Kurum, Comparing performances of logistic regression, classification and regression tree, and neural networks for predicting coronary artery disease, *Expert Systems with Applications*, vol.34, no.1, pp.366-374, 2008.
[34] M. Saimurugan, K. I. Ramachandran, V. Sugumaran and N. R. Sakthivel, Multi component fault diagnosis of rotational mechanical system based on decision tree and support vector machine, *Expert Systems with Applications*, vol.38, pp.3819-3826, 2011.
[35] N. R. Sakthivel, V. Sugumaran and S. Babudevasenapati, Vibration based fault diagnosis of monoblock centrifugal pump using decision tree, *Expert Systems with Applications*, vol.37, pp.4040-4049, 2011.