Preprints of the 2012 20th Mediterranean Conference on Control & Automation (MED), Barcelona, Spain, July 3-6, 2012

New Technique for Online Faults Diagnosis Based on Faulty Models Design: Application to DAMADICS Actuator

Kourd Y. Lefebvre D and Guersi N.

Abstract — **The increased complexity of plants and the development of sophisticated control system have necessitated the parallel development of efficient online fault detection and isolation system. The detection and isolation of faults in industrial system has lately become of great significance. This paper proposes a new technique for online fault detection and diagnosis in dynamic system with multi inputs multi outputs. Numerous diagnosis schemes and architectures have been developed and applied to the benchmark DAMADICS. One of the key issues in designing a fault diagnosis system is the system modeling. Neural networks combined with other methods have been widely investigated for that purpose. The main contribution of this paper is to develop a new method for online fault detection and diagnosis schema with a bank of fault free and faulty reference models designed according to neural networks. Fault detection is obtained according to the comparison of measured signals with the behavior of fault free reference model. Then, calculation of Euclidean norms of the output error signals resulting from the faulty models leads to fault isolation. The effectiveness of this approach is illustrated with simulations on DAMADICS benchmark.**

I. INTRODUCTION

The main objective of online fault detection and isolation is to provide early warnings to operators, such that appropriate action can be taken to prevent the breakdown of the system after the occurrence of faults. The fault tolerant control and reliability issues for industrial systems and technological processes require the development of advanced fault detection and isolation (FDI) approaches. The early detection of faults may help to avoid system breakdowns and material damages. During the last decades many investigations have been made using analytical approaches, based on mathematical models. Among the model-based fault diagnosis methods, parameter estimation, parity relation and observers design are the most often applied techniques [4][5][6][7][8]. Unfortunately these approaches require the state space description of the considered systems. Such description is not so often available in engineering practice. In order to solve this problem, system identification strategies can be applied [9][10]. One of the most popular nonlinear systems identification approaches is based on the application of artificial neural networks (ANNs) [2][11] [12][13][14][15][16].

ANNs are computational models with particular properties such as ability to learn, simplicity of implementation, generalization and good approximation properties. The aim of fault detection is to deliver alarms when fault occur and also to estimate the time of fault occurrence. The aim of fault diagnosis is to determine the type, magnitude and location of faults. Detection and diagnosis procedures are based on the observed symptoms and on the knowledge about the process. So, the inputs of a knowledge-based fault diagnosis system are observed measurement and fault-relevant knowledge about the process [1]. Then, on-line diagnosis of actuators, sensors and system components can be achieved either via a remote and supervisory diagnosis system or using local intelligence and self-validation methods [17][18][19][20]. Such self-validation techniques and condition monitoring are popular for numerous applications in various domains. The potential of ANNs for fault detection and diagnosis in non-linear dynamic systems have been demonstrated in recent years. The neural network approach is especially applicable to systems for which mathematical models are difficult to work out. Adaptive ANNs are used to differentiate various faults from the normal condition, and from one another, according to different fault patterns represented in the measured input-output system data, either by off-line training or by on-line learning of the fault patterns. In addition ANNs are also helpful to model the non-linear dynamics according to non-linear autoregressive structure with exogenous inputs (NARX). [21]

This paper focuses on the problem of robust fault detection and diagnosis using non-linear models that are designed with neural networks. The proposed approach takes into account the parameters uncertainty within the neural model by generating an adaptive threshold for detection. Faults isolation results from the analysis of a bank of residuals obtained from the faults candidate database and from estimation time of occurrence of the faults. The effectiveness of this approach is illustrated according to simulation results obtained with DAMADICS benchmark.

II. FDI METHOD PROPOSED

In this work, fault detection and diagnosis is based on the design of models that represent the fault free behaviors and also the faulty ones. One way to obtain such models is to apply identification techniques.

KOURD Y. is with Mohamed Khider Biskra University, Department of Control Engineering, Algeria (ykourd@ yahoo.fr).

Lefebvre D. GREAH – University of Havre 25 rue Philippe Lebon – 76058 Le Havre – France (dimitri.lefebvre@univ-lehavre.fr).

Guersi N. University of Badji Mokhtar Annaba, Department of Electronic, Algeria (guersi54@yahoo.fr).

978-1-4673-2529-5/12/$31.00 ©2012 IEEE

A. Reference models design

In the following we consider dynamic systems with q inputs $u_i(t)$ and n outputs $y_k(t)$ and it is assumed that the state variables are no measurable. Such systems exhibit often complex dynamics, with strong non – linearities. As a consequence, knowledge – based models are not easy to obtain. Another approach lies in the data – based models. Artificial neural networks are often used for that purpose [22][23][26]. The goal is to design fault free and faulty models that will be used for the design of residuals (figure 1).

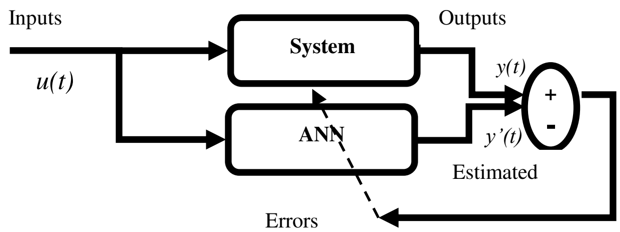

Figure 1: Data-based model design.

In order to get the best ANN architecture, several configurations are tested according to a trial – error processing that uses pruning to eliminate the useless nodes. The learning of the ANN is obtained according to the Levenberg-Marquardt algorithm with early stopping. This algorithm is known for its rapid convergence. During learning stage, the ANN is trained with data collected during the normal functioning (fault free model) and according to faulty behaviors stored in database. Then the ANN reference models are validated with another set of data.

B. Fault Detection

During the monitoring stage, the comparison of the system output $y(t)$ and fault-free model output $y'(t)$ leads to residual $r(t)$:

$$r(t)= y(t)-y'(t) \tag{1}$$

The residual $r(t)$ is a source of information about faults for further processing. Fault detection is based on the evaluation of residual magnitude. It is assumed that residual $r(t)$ 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 by setting of threshold S_y on the residual signal (figure 2). In the same time, the analysis of residual $r(t)$ provide an estimation τ of the time of occurrence t_f that will be used for diagnosis issue. In case several residuals $r_k(t)$, $k = 1,\ldots, n$ are used, the estimation τ of the time of occurrence of fault is given by :

$$\tau = \min \{ \tau_k, k = 1,\ldots, n\} \tag{2}$$

Where τ_k is the estimation given by residual $r_k(t)$.

The fault is alerted when the absolute value of the residual $|r(t)|$ will be larger than the threshold s_y:

$|r(t)| \leq s_y$: no fault is detected at time t

$$|r(t)| > s_y \text{ : a fault is detected at time t} \tag{3}$$

The main difficulty with this evaluation is that the measurement of the system output $y(t)$ is usually corrupted

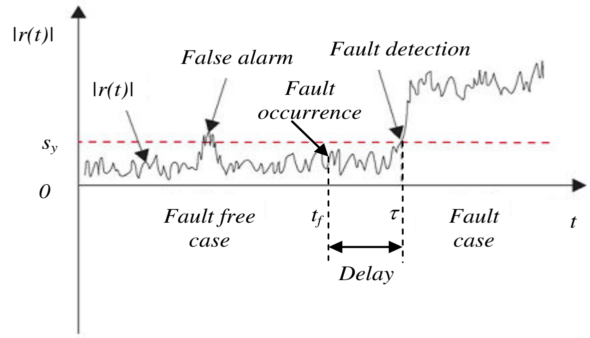

Figure 2: Residual-based fault detection.

by disturbances (for example measurement noise). In practice, due to the modeling uncertainties and disturbances, it is necessary to assign large thresholds S_y in order to avoid false alarms. Such thresholds usually imply a reduction of the fault detection sensitivity and can lead to non detections.

C. Fault diagnosis

When multiple faults are considered, diagnosis is a difficult task. One can multiply the measurements and use some analysis tools (residuals analysis) in order to isolate the faults. But the number of sensors limits the use of such approach. Another approach is to use a history of collected data to improve the knowledge about the faulty behaviors and then to use this knowledge to design faulty models and additional residuals. Such models will be used to provide estimations for each fault candidate and comparison with measurements will provide the most probable fault according to the Euclidean distance between estimated and measured signals. Such approach is developed in the following: p faults candidate are considered. For each fault candidate a faulty model $FM(j)$ $j = 1,\ldots, p$ based on neural networks is designed (figure 3).

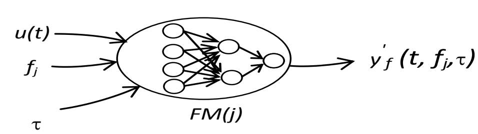

Figure 3: Faulty model design.

The inputs of network $FM(j)$ are the input signals $u(t)$, the fault candidate f_j and the estimation τ of the time of occurrence provided by the detection stage. The output is an estimated faulty behavior $y'_j(t, f_j, \tau)$ worked out assuming that fault f_j disturbs the system from time τ. In case of numerous fault candidates f_j, $j = 1,\ldots,p$, the outputs $y'_f(t, f_j, \tau)$ of the faulty models $FM(j)$ are compared with the measurement $y(t)$ to compute the residuals $r_f(t, f_j, \tau)$:

$$r_f(t, f_j, \tau) = y(t) - y'_f(t, f_j, \tau) \tag{4}$$

Then the most probable fault candidate f_{j*} is worked out according to the comparison of the cumulative residuals over a considered time interval $[0, T]$:

$$j^* = \text{argmin}_j \left\{ \int_0^T \left\| r_f(t, f_j, \tau) \right\| . dt \, , j = 1, \ldots, p \right\} \tag{5}$$

In case of multiple outputs $y_k(t)$, $k = 1,\ldots, n$, the Euclidean distance in R^n space is considered and equation (5) is replaced by (6):

$$j^* = \operatorname{argmin}_j \left\{ \sqrt{\sum_{k=1}^{k=n} \left(\int_0^T \left\| r_{fk}(t, f_j, \tau) \right\| . dt\right)^2}, j = 1, \dots, p \right\} \quad (6)$$

III. Application to The DAMADICS Benchmark

The proposed method is applied on signals obtained with the DAMADICS simulator.

A. *DAMADICS description*

The DAMADICS benchmark is an engineering research case-study that can be used to evaluate detection and isolation methods. The benchmark is an electro-pneumatic valve actuator in the Lublin sugar factory in Poland [25]. The actuator consists of a control valve, a pneumatic servomotor and a positioner (figure 4).

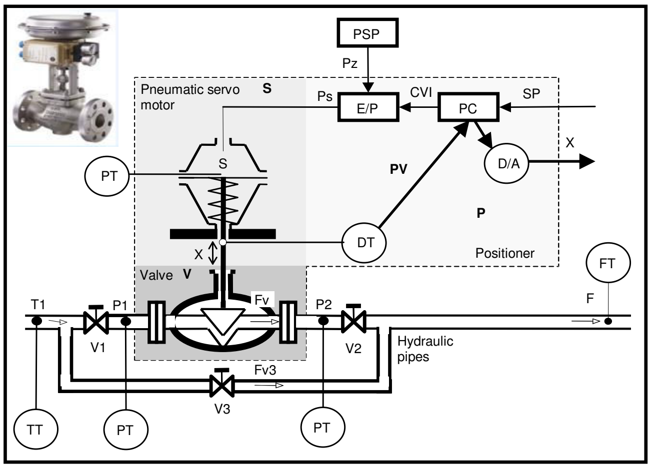

Figure 5: Structure of DAMADICS actuator system.

D/A – data acquisition unit
PC – Positioner Central processing unit
E/P – electro-pneumatic transducer
V1, V2, V3 – bypass valves
DT – displacement
PT – pressure
FT – value flow transducer
TT – Temperature
- The Valve – Quantitative Model:

In the actuator, faults can appear in: control valve, servo-motor, electro-pneumatic transducer, piston rod travel transducer, pressure transmitter or microprocessor control unit. 19 types of faults are considered (p = 19). The faults are emulated under carefully monitored conditions, keeping the process operation within acceptable quality limits.

TABLE I. List of DAMADICS Actuator Faults.

Fault	Description
f1	valve clogging
f2	valve or valve seat sedimentation
f3	valve or valve seat erosion
f4	increasing of valve or bushing friction
f5	external leakage
f6	internal leakage (valve tightness)
f7	medium cavity or critical flow
f8	twisted servo-motor's rod
f9	servo-motor's housing or terminals tightness
f10	servo-motor's diaphragm perforation
f11	servo-motor's spring fault
f12	electro-pneumatic transducer fault
f13	rod displacement sensor fault
f14	pressure sensor fault
f15	positioner spring fault
f16	positioner lever fault
f17	positioner supply pressure drop
f18	unexpected change of pressure difference
f19	fully or partly opened bypass valves

Five available measurements and 1 control value signal have been considered for benchmarking purposes: process control external signal *CV*, values of liquid pressure on the valve inlet *P1* and outlet *P2*, liquid flow rate *F*, liquid temperature *T1* and stem displacement *X* (table 2).

TABLE II. Input and output variables.

Input	Range	Unit	Description
CV	[0,1]	-	control signal from external PI controller
P1	[2000, 4e+6]	Pa	Inlet liquid pressure
P2	[2000, 4e+6]	Pa	Outlet liquid pressure
T1	[30, 110]	C °	Liquid temperature
Output	Range	Unit	Description
X	[0,1]	-	Position of the rod
F	[0,1]	-	Average flow

B. *Residuals design*

The actuator is modeled with two multilayer ANNs: *netX* and *netF* (figure 5) that represent the interaction between q = 4 inputs and the n = 2 outputs according to the equations (7). Model of the positioner and the pneumatic motor and Model of the control valve:

X'=f (CV, P1, P2, T)
F'=f (X, P1, P2, T) (7)

To select the structure of the neural networks *netX* and *netF*, numerous tests have been carried out to obtain the best architectures (i.e. number of hidden layers and number of neurons by layer) in order to model the operation of the actuator. The training and test data were generated by the simulation of the Matlab-Simulink actuator model [24].

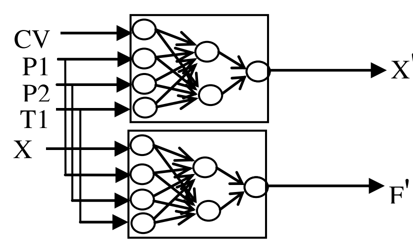

Figure 5: ANN fault-free model

The best structure for our system is an ANN with 6 nodes in layer 1, 3 nodes in layer 2 and a single output neuron. When the training is over, *netF* provides estimated outputs that are close from the actual ones. Validation is done with the measured data provided by the 'Lublin Sugar Factory in 2001 [25]. Validation is illustrated on figure 6.

Two residual are designed according to equation (8):

$r_1(t)=X(t) - X'(t)$

$r_2(t)= F(t)- F'(t)$ (8)

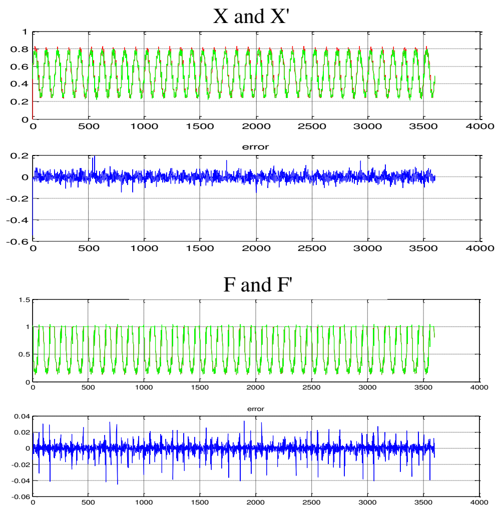

Figure 6: Actual output *X* and estimated output *X'* (a); Residual *r1* (b); actual output *F* and estimated output *F'* (c); Residual *r2* (d);

C. Faults detection

The fault detection is obtained according to the comparison of the current residual value with some thresholds. Such thresholds are worked out according to the standard deviation of the residual for fault free case. For output *X*, σ_1 = 0.0259 and S_{y1}= 5*σ_1= 0.1295. For output *F*, σ_2 = 0.0040 and S_{y2}= 5*σ_2= 0.0199. During normal functioning the residuals remain near zero. In figure 7, the residuals are worked out when the fault f_1 is simulated during interval [20s 80s] time units (times units are in seconds). This fault is alerted by r_1 and r_2.

The table 3 sums up the detection performances for the 19 types of faults (*p* = 19). r_i+ means that residual r_i is large positive (according to the previous thresholds) and r_i- means that residual r_i is large negative. To conclude, all faults are alerted according to residual r_1 and r_2.

Residual analysis is an essential step. Choice of one or more threshold, fixed or adaptive, strongly influences the quality and performance of the diagnosis. The problem of choosing the threshold is closely related to the behavioral residual overlooked faults and constraints that may be imposed such as security margins tolerated [27].

TABLE III. FAULTS TO BE DETECTED AND ISOLATED

Faults \ Residuals	r_1+	r_1-	r_2+	r_2-
Fault free	0	0	0	0
f_1	1	1	1	1
f_2	0	0	1	0
f_3	0	0	0	1
f_4	0	0	1	1
f_5	0	0	0	0
f_6	0	0	0	1
f_7	1	1	1	1
f_8	0	0	0	0
f_9	0	0	0	1
f_{10}	1	1	1	1
f_{11}	1	1	1	0
f_{12}	0	0	0	1
f_{13}	0	1	0	1
f_{14}	0	0	0	0
f_{15}	1	1	1	1
f_{16}	1	0	0	1
f_{17}	1	1	1	1
f_{18}	0	0	0	1
f_{19}	0	0	0	1

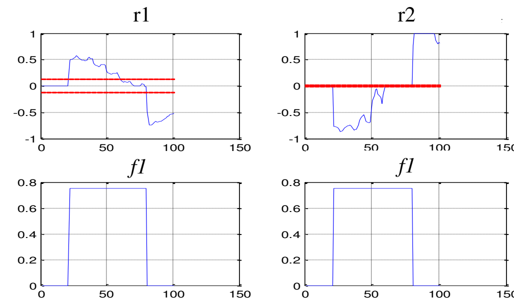

Figure 7: Residual *r1, r2* when f_1 is simulated within interval [20s 80s]

D. Fault diagnosis

According to table 3, three groups of faults with similar symptoms can be separated:

- group 1 = {$f_3\, f_6\, f_9\, f_{12}\, f_{18}\, f_{19}$}
- group 2 = {$f_1\, f_7\, f_{10}\, f_{15}\, f_{17}$}
- group 3 = { *Fault-free* $f_5\, f_8\, f_{14}$}

Within each group, faults are not isolable. For this reason we propose to use the method described in section II.C to improve the isolability of faults. For this purpose 19 faulty models *FM(j) j* = 1,…,19 are designed according to the history of data available with DAMADICS benchmark. Each faulty model *FM(j)* computes two estimated outputs $X'_f(t, fj, \tau)$ and $F'_f(t, fj, \tau)$ and comparisons with measured data lead to the residuals $r_{1f}(t, f_j, \tau)$ and $r_{2f}(t, f_j, \tau)$:

$$r_{1f}(t, f_j, \tau) = X(t) - X'_f(t, f_j, \tau)$$
$$r_{2f}(t, f_j, \tau) = F(t) - F'_f(t, f_j, \tau) \quad (9)$$

Then $S(f_j, 1, T)$, $S(f_j, 2, T)$ and $D(f_j; T)$ are computed:

$$S(f_j, 1, T) = \int_0^T \|r_{1f}(t, f_j, \tau)\|.\,dt$$

$$S(f_j, 2, T) = \int_0^T \|r_{2f}(t, f_j, \tau)\|.\,dt$$

$$D(f_j, T) = \sqrt{S^2(j, 1, T) + S^2(j, 2, T)} \quad (10)$$

$$S(f_j, 1, T) = \int_0^T \|r_{1f}(t, f_j, \tau)\|.\,dt\; S(f_j, 2, T) = \int_0^T \|r_{2f}(t, f_j, \tau)\|dt.$$

The diagnosis is obtained with equation (11):

$$j^*(T) = \text{argmin}_j\{D(j, T), j = 1, \ldots, n\} \quad (11)$$

For example, f_{15} is considered within time interval [444s 1000s] (figure 8). A fault is detected at time τ = 451s, and the group 2 = {$f_1\, f_7\, f_{10}\, f_{15}\, f_{17}$} is isolated.

The application of diagnosis stage leads to the results in table 4. Each fault candidate is simulated from estimated occurrence time τ = 451s and within a time interval of size *T* = 1000s. The figure (9) plots the relative location of each faulty model *FM(j)* in plan ($S(f_j, 1, 1000)$, $S(f_j, 2, 1000)$) (the position (0, 0) corresponds to the measured outputs). The model *FM(15)* corresponding to the fault candidate f_{15}

provides the estimated outputs with the smallest Euclidean distance from the measured outputs. To conclude f_{15} is the most probable fault. Similar conclusions have been obtained for numerous other simulations.

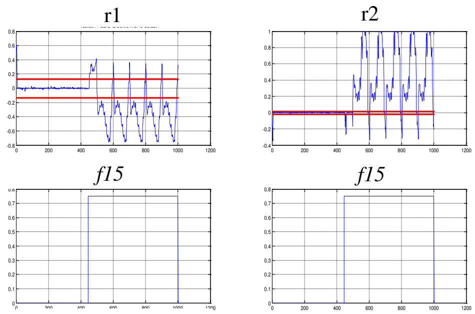

Figure 8: Residuals r_1, r_2 when f_{15} is simulated within interval [444s 1000s].

We studied the variation of the delay τ according to the detection threshold; the results are summarized in the table below.

TABLE IV. DIAGNOSIS ACCORDING TO THE BANK OF FAULTY MODELS

Faulty model *FM(j)*			
Candidate fault f_j	S(j,1,1000)	S(j,2,1000)	D(j,1000)
f_1	14.96	20.02	25.00
f_2	9.94	11.23	15.00
f_3	10.03	14.79	17.88
f_4	9.99	14.29	17.44
f_5	10.01	13.47	16.78
f_6	10.01	18.13	20.71
f_7	14.15	20.02	24.52
f_8	10.01	13.57	16.86
f_9	10.08	13.65	16.97
f_{10}	11.33	16.65	20.14
f_{11}	9.93	13.59	16.83
f_{12}	8.12	11.01	13.68
f_{13}	10.09	20.01	22.41
f_{14}	10.01	13.57	16.86
f_{15}	0.53	0.24	0.58
f_{16}	11.30	16.42	19.93
f_{17}	10.19	15.29	18.37
f_{18}	10.01	20.02	22.38
f_{19}	10.01	8.65	13.23
Fault free	10.01	13.57	16.86

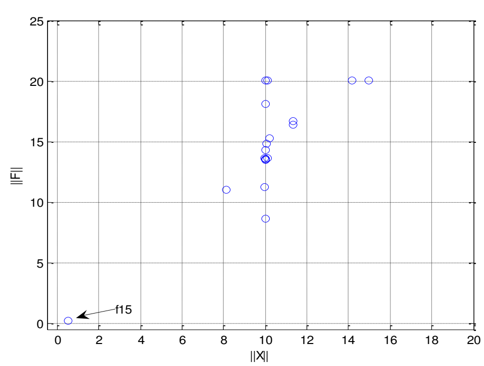

Figure 9: Location of the faulty model *FM(j)* in plan (*S(j,1,1000), S(j,2,1000)*)

Table 5 shows that the delay of detection increases as the threshold increases. The threshold must be thoroughly selected. For the continuation of our work we chose a threshold $S_y = 5 * \sigma$. For the considered system, this value provides the best compromise between the detection rate and the false alarm rate.

TABLE V. VARIATION OF THE DELAY ACCORDING TO THE THRESHOLD

Threshold / Delay	σ standard deviation	2* σ	3* σ	4* σ	5* σ	6* σ	7* σ
f_{12}: dτ=(t-τ), t=500s	2s	3s	12s	39s	40s	60s	ND
	2s	2s	2s	2s	2s	2s	2s
f_{18}: dτ=(t-τ), t=231s	False Al	ND	ND	ND	ND	ND	ND
	1	2s	2s	2s	2s	2s	2s
f_{11}, dτ=(t-τ), t=485s	2+FA	2s	2s	2s	2s	3s	ND
	2+FA	2s	2s	2s	2s	2s	2s
f_{15}, dτ=(t-τ), t=444s	7s	7s	8s	8s	9s	17s	51s
	1s+ FA	1s+ FA	7s	7s	7s	8s	8s
f_3, dτ=(t-τ), t=405s	False Al	ND	ND	ND	ND	ND	ND
	4s+ FA	16+ FA	29s	29s	35s	45s	47s

Depending on the size of the time horizon T, the decision provided by equation (11) can be obtained off-line (large value of T) or on-line (small value of T). The charts in figures 10 and 11 provide both algorithms (off-line and on- line methods).

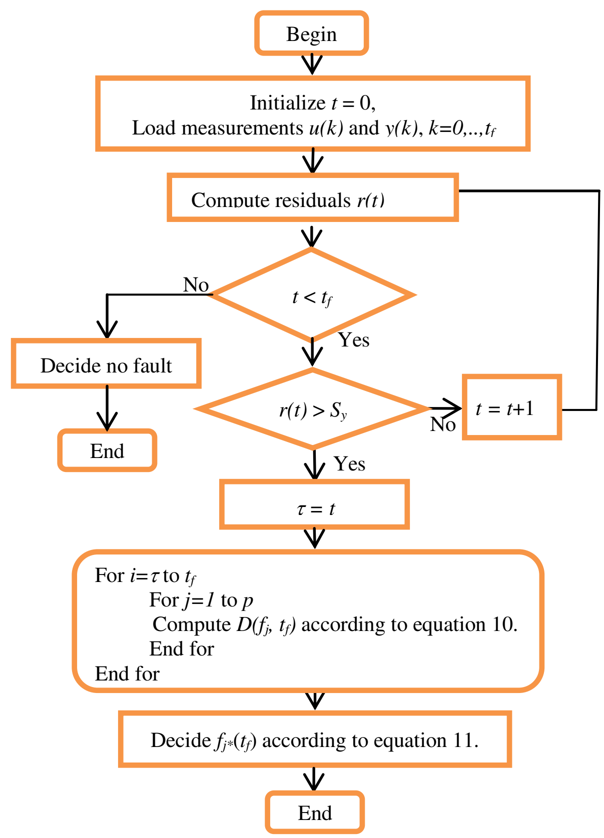

Figure 10: Diagnosis with off-line method.

If the fault f_{12} occurs, we obtain the results presented in figure 12. This figure show the variations of models *FM (j)* for each fault j versus time in a plan (*S (j, 1, t), S (j, 2, t)*). The model *FM(12)* (i.e. fault f_{12}) is the closer to the origin (0,0) for all value of time in interval [300, 1000]. So, f_{12} is the most probable fault for this example.

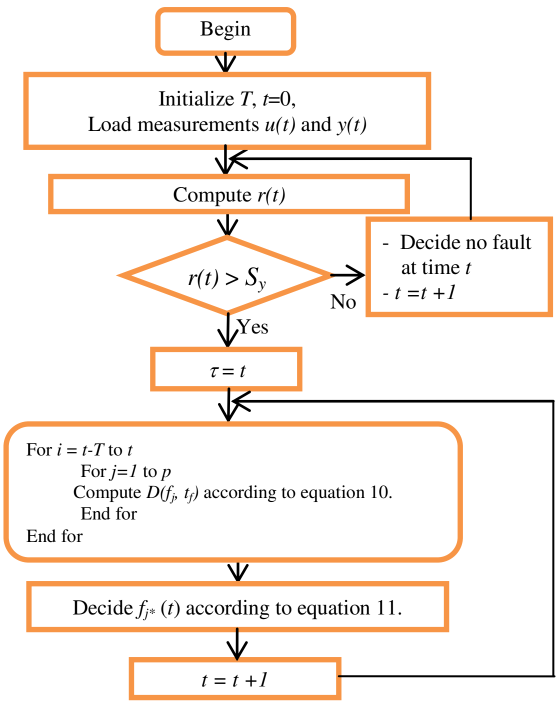

Figure 11: Diagnosis with on-line method.

We represent the fault localization for 19 different faults f_1...f_{19} in figure 12.

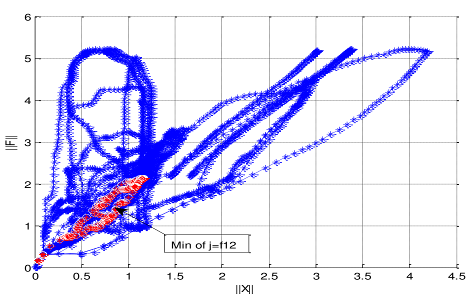

Figure 12: Location of the faulty model *FM(j)* in plan (*S(j,1,700), S(j,2,700)*) represented by red color.

IV. CONCLUSION

The new technique developed and used in this paper for the on line fault detection and diagnosis, combines the computing power and the robustness of the neural networks with the use of a threshold and the Euclidean distance (Section II.C) to strengthen the isolability faults and distinguish the most probably fault. The results obtained show the effectiveness of this technique, if one makes a good choice of the detection threshold.

Our future works are also to validate this technique by applying it on other systems with various operating condition and various faults.

REFERENCES

[1] R. Isermann 2005. Model-based fault-detection and diagnosis – status and applications. *Annual Reviews in Control*, 29, 71-85.

[2] M. Mrugalski M. Witczak, J.Korbicz, 2008. Confidence estimation of the multi-layer perceptron and its application in fault detection systems. ELSEVIER. *Engineering Applications of Artificial Intelligence*, 21, 895–906

[3] R. Isermann , 2006. *Fault-Diagnosis Systems. An Introduction from Fault Detection to Fault Tolerance*. Springer, Berlin.

[4] J. Korbicz, J. M. Koscielny, Z. Kowalczuk, & W. Cholewa, 2004. *Fault Diagnosis. Models, Artificial Intelligence, Applications*. Springer, Berlin.

[5] R.J. Patton, J. Chen, 2000. On eigenstructure assignment for robust fault diagnosis. *International Journal of Robust and Nonlinear Control*, 10, 1193–1208.

[6] H.R. Scola, R. Nikoukah, F. Delebecque, 2003. Test signal design for failure detection: A linear programming approach. *International Journal of Applied Mathematics and Computer Science*, 13 (4), 515–526.

[7] M. Witczak, 2007. *Modeling and Estimation Strategies for Fault Diagnosis of Non-linear Systems. From Analytical to Soft Computing Approaches*. Springer, Berlin.

[8] E. Delaleau, Louis, J.P., Ortega, R., 2001. Modeling and control of induction motors. *International Journal of Applied Mathematics and Computer Science,* 11 (1), 105–129.

[9] T. Soderstrom, et al, 1989. *System Identification*. Prentice-Hall International, Hemel Hempstead.

[10] T. Bouthiba, 2004. Fault location in ehv transmission lines using artificial neural networks. *International Journal of Applied Mathematics and Computer Science*, 14 (1), 69–78.

[11] M.M. Gupta, 2003. *Static and Dynamic Neural Networks*. Wiley, Hoboken, New Jersey.

[12] O. Nelles, 2001. *Non-linear Systems Identification. From Classical Approaches to Neural Networks and Fuzzy Models*. Springer, Berlin.

[13] S.C. Tan, et al, 2007. A hybrid neural network model for rule generation and its application to process fault detection and diagnosis. *Engineering Applications of Artificial Intelligence*, 20 (2), 203–213.

[14] V. Uraikul, et al, 2007. Artificial intelligence for monitoring and supervisory control of process systems. *Engineering Applications of Artificial Intelligence* 20 (2), 115–131.

[15] J. Vieiraa, et al, 2004. Artificial neural networks and neuro-fuzzy systems for modelling and controlling real systems: a comparative study. *Engineering Applications of Artificial Intelligence*, 17 (3), 265–273.

[16] B. Michal, et al, 2006. Introduction to the DAMADICS actuator FDI benchmark study, *Control Engineering Practice,* 14 (2006) 577–596

[17] J. Yang, et al, 1999. The self-validating actuator. *Control Engineering Practice*, 7, 249–260.

[18] M. P. Henry, 2000. Plant assessment management via intelligent sensors: Digital, distributed and for free. *Computing & Control Engineering Journal*, 11(5), 211–213.

[19] M. Tombs, 2002. Intelligent and self-validating sensors and actuators (special feature editorial). *Computing & Control Engineering Journal*, 13(5), 218–220.

[20] J.B. Gomm, 1999. Sensor fault diagnosis in a chemical process via RBF neural networks. *Control Engineering Practice*, 7. 49-55.

[21] Sjöberg J., Zhang Q., Ljung L., Benveniste A., Delyon B., P.Y., Hjalmarsson, H.& Juditsky, A., 1995. Non-linear black-box modeling in system identification: a unified overview. *Automatica*, 31 (12), 1691–1724.

[22] Kourd Y., Guersi N., Lefebvre D. , 2008;A two stages diagnosis method with neuronal networks, Proceeding ICEETD 2008 Hammamet Tunisie.

[23] Kourd Y., Guersi N., Lefebvre D., 2010, A two stages diagnosis method with Neuro-fuzzy approach, Proceeding CIFA 2010 Nancy, France.

[24] Kourd Y., Guersi N., Lefebvre D., 2010, Neuro-fuzzy approach for fault Diagnosis: application to the DAMADICS, 7th international Conference on Informatics in Control, Automation and Robotics. Proceeding ICINCO 2010 Funchal, Madeira-Portugal.

[25] DAMADICS, 2002. Website of the RTN DAMADICS. (http://diag.mchtr.pw.edu.pl/damadics).

[26] Juditsky A., Hjalmärsson H., Benveniste A., Delyon B., Ljung L., Sjöberg J., Zhang Q., 1995, Nonlinear black-box models in system identification: mathematical foundations. *Automatica,* 31, 1725-1750.

[27] Zhao. J., Huang J., Sun W., 2008. On line early fault detection and diagnosis of municipal solid waste incinerators. *Waste Management*. 28, 2406-2414.