Faults Detection and Isolation Based On Neural Networks Applied to a Levels Control System

Raphaela Galhardo Fernandes, Diego Rodrigo Cabral Silva,
Luiz Affonso H. Guedes de Oliveira, Adrião Duarte Dória Neto

***Abstract*— Each time more grows the necessity of guaranteeing itself security and trustworthiness of the equipment during the execution of the industrials processes. Then, it is very important that faults in the processes can be detected and isolated. This paper presents an approach to process fault detection and isolation (FDI) system applied to a levels control system connected with an industrial network Foundation Fieldbus. The FDI system was developed using artificial neural networks (ANN) and tested in real environment.**

I. Introduction

In a dynamic system, the faults can cause alterations in parameters critical of the system, or even changes in its dynamics. It is very important that the undesired or not permited states can be perceptible. Thus, avoding damages situations or any accident.

This way, it is ideal that the faults can be detected, isolated and elimated, providing more confidence in the system. Detect a fault in the system is to be able to say if it exists or not. On the other hand, isolate it is according about what kind of fault is present, e.g., indicate in which component of the system the fault occurred. The fault tolerance is associated with the magnitude of the fault supported by the system in order to keep the system in adequate operation condition. The whole process is defined as Fault Detection and Diagnostic (FDD).

To detect and diagnostic faults in appropriate way it is very important to have a previous knowledge about the system behavior. Thus, automation engineers can associate the faults with the signal patterns. So, they can project more efficient FDD systems.

There are many methods already consolidated of detection and isolation of faults. Some of them are based on the physical redundancy, in other words, in the duplication of hardware components, like sensors, actuators and controllers [1]. In these cases, the outlets of the equivalent pairs of components are the initial point for a possible abnormal detection. The subtraction of these two values is called, in the technical jargon, residue, which in turn, is one of the basic pairs in the FDI systems [1]. Basically, if the outputs values of the pairs are close, or, the residue shows to be very close to zero, it means that there is not a fault. There is a divergence between them and also if a considerable residue then, there is a fault. The principle disadvantage of these methods is the use of extra equipment that can increase the maintenance at a high rate, besides this it needs more physical space for its accommodation.

In the seventies arose other techniques based on the paradigm of analytic redundancy. With this procedure, one can use the system model to generate redundancy signals which are compared with signals measured from the physical sensors [2]. Similarly to the physical redundancy the outputs of the sensors are compared to the ones calculated through the mathematical model with the object of generating the residues, and consequentially, if there is a significant difference between them, it characterizes the occurrence of a fault. One of the difficulties with this procedure is the necessity of having a mathematical model very close to reality.

In recent years, researches made in the context of Faults Detection and Isolation system (FDI) presented procedures that uses the principles of Computer Intelligence, such as, Fuzzy Logic system [3] and Artificial Neural Networks (ANNs) [4].

This article describes the project and implementation of a fault detection and isolation system applied to a dynamic system in real time, for instance, a plant composed of two tanks (plant of levels), in which one of the tanks has its level controlled. Basically, the FDI system was divided in two parts: the first corresponds to neural identification of the plant model and the second, to the detection and isolation of faults in process.

In this paper, we first turned the system parameters for a simulated environment, then we used there parameters to detect and to isolate faults in a real system. It is showed that the FDI system was able to accuse the faults at the moment they are ocurring, surpassing several difficulties found only in real environment, such as: noises and unmodeling dynamic.

II. System Architecture

The FDI system development was realized in two phases. The first one was the levels system neural identification. The second one was the fault detection and isolation process. The tests were done as much in simulated environment as in real environment. It was mounted in the LAMP (Petroleum Measurement and Evaluation Laboratory) at UFRN.

The general scheme of the functioning system is shown in Figure 1. In this case, while the level system is in execution, an elaborated system from the ANNs tries to find its identification using its inputs ($x(k)$). Each time, the output level system ($y(k)$) is compared to the output identification system ($y'(k)$), generate a residue value ($r(k) = y(k) - y'(k)$). The

Raphaela Galhardo Fernandes, Diego Rodrigo Cabral Silva, Luiz Affonso H. Guedes de Oliveira, Adrião Duarte Dória Neto are with the Department of Automation and Computation Engineering, UFRN, Brazil; emails: raphaela@dca.ufrn.br, diego@dca.ufrn.br, affonso@dca.ufrn.br, adriao@dca.ufrn.br).

residue, in the other phase, will be used in the fault isolation system. Then one analyzes the residue values and indicates the occurrence or not of faults and, in case that it occurs, indicates what type it is.

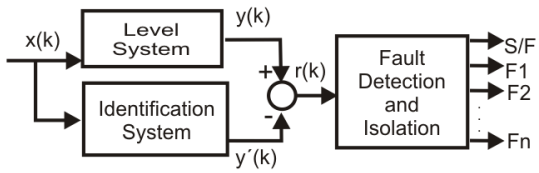

Fig. 1. General scheme of FDI system

Figure 2 shows the level plant composed of two tanks in cascade, representing a second order model with an input. A Proportional Integrative Derivative (PID) [6] control strategy is applied in this plant to control the level of tank 2.

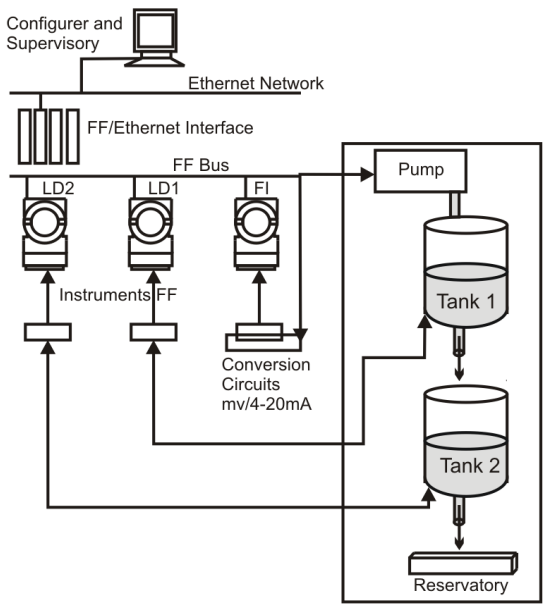

Fig. 2. Architecture from the real environment.

The water that flows out from the small hole of tank 1 falls into tank 2. This in turn, also has a small hole through which the water falls directly to the reservoir. A pump is responsible for taking the water from the reservoir to the tank 1. In each tank there is a Foundation Fieldbus (FF) pressure sensor [7], used to measure the corresponding levels connected to the Fieldbus network. Besides the pressure sensors, a FF/ loop of current from 4 to 20 mA converter is used to inject signals to the water pump.

The industrial network Foundation Fieldbus is connected to a supervisory computer through Ethernet network interfaces. From this computer is carried out all the devices configuration process and later supervised.

The development of the FDI system needed to apply signals to the pump input so that it made possible the water injection in the tank 1. Also, it was needed extract data from the levels system (tank levels, control signal, etc.) through industrial network. To do this, some industrial automation technologies had been used.

Among thsese technologies we can detach: SCADA [FALTA REFERENCIA] and OPC (OLE for Process Control) [8]. The first one is the acronym for Supervisory Control And Data Acquisition. It acts in the area of control and supervision of remote devices. It is used to show, in centered way, the functioning of all industrial plant to the user. Including still, the possibility of interacting with the plant, changing configurations and acting in devices through graphical screens that represent the real plant.

The second one, OPC, consists of an open standard communication developed or the industrial automation. It offers a standardized way so that softwares, that run in computers of the supervision network, can access information that passes through the field network.

Another software, MATLAB, uses OPC technology to access information at the control network in real-time.

The first phase of the FDI system (neural identification) was implemented using recurrent neural networks, whose was trained from input/output data applied/ collected, respectively, from the real system. To do so, OPC (OLE for process control) [8] client was used to generate the input signals to the plant and to capture its respective outputs. The software used to generate and insert the signals to the FF network was the Elipse SCADA. The software used to capture the signals was the MATLAB.

The second phase (faults detection and isolation) of the system was developed in MATLAB and it supervises the plant in real-time through the OPC protocol, evaluating the presence of faults and classifying it, in positive case.

The FDI system was applied to two environments: simulated and real. To simulate the dynamic of the plant, a non-linear mathematical model was used as described in the equations 1 and 2 [9]. The equantion 1 calculates the tank 1 level and the equation 2 calculates the tank 2 level.

$$\dot{L}_1 = -\frac{a_1}{A_1}\sqrt{2gL_1} + \frac{K_m}{A_1}V_p \tag{1}$$

$$\dot{L}_2 = -\frac{a_2}{A_2}\sqrt{2gL_2} + \frac{a1}{A2}\sqrt{2gL_1} \tag{2}$$

Table I presents the descriptions and values of the parameters shown in equations 1 and 2.

TABLE I
PLANT PARAMETERS

Name	Description	Value
K_m	Pump Constant	$4.6(cm^3/s)/V$
V_p	Tension applied to pump	$-22 < V_p < 22$
a_1	Tank 1 output diameter	$0.178139cm$
a_2	Tank 2 output diameter	$0.178139cm$
A_1	Tank 1 area	$15.5179cm^2$
A_2	Tank 2 area	$15.5179cm^2$
g	Gravity acelerator	$980cm/s^2$

Finally, the level plant model together with the controller PID is illustrated in Figure 3. As it is possible to see, the reference signal is the level desired for tank 2. The desired signal ($d(k)$) is compared to the actual level from the tank 2 ($L_2(k)$), generating an error ($e(k)$) which is used to the PID controller to generate a control signal (V_p) to the pump that injects water to tank 1.

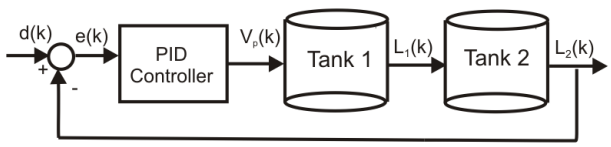

Fig. 3. Level plant with PID controller

In the following section, it will be presented the developing of the neural networks which will make identification, detection and isolation of faults in the level system.

III. Level System Neural Identification

The neural identification system was defined as identification in two steps, which means the existence of an ANN to evaluate the level of tank 1 and another to evaluate the level of tank 2. The training of the ANNs can be illustrated by Figure 4.

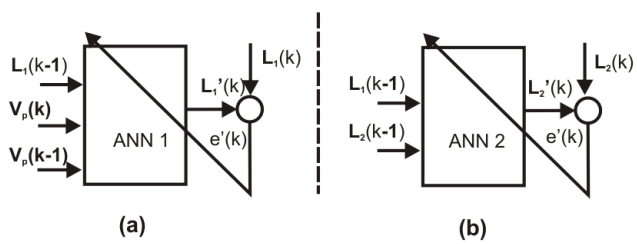

Fig. 4. Determination of tank 1 and tank 2 levels. (a) Step 1. (b) Step 2.

The output of the ANN1 results in the representation of the estimated tank 1 level, evaluating in the actual instant k, $L_1^{'}(k)$. Your architecture was:

- Three nodes in the input layer, where the inputs are: $V_p(k)$, $V_p(k-1)$ and $L_1(k-1)$;
- A hidden layer with 3 neurons with the sigmoid activation function;
- A neuron on the output layer (linear activation function), representing the level of tank 1 evaluating in the actual instant k, $L_1^{'}(k)$.

The ANN 2 architecture, used to evaluate the estimated tank 2 level $L_2^{'}(k)$, was:

- Two nodes in the input layer which are $L_1(k-1)$ and $L_2(k-1)$;
- A hidden layer with 3 neurons with the sigmoid activation function.
- One neuron in the output layer (linear activation function), representing the level of tank 2 evaluating in the actual instant k, $L_2^{'}(k)$.

To implement the strategy of neural level system identification, it used a Multiple Layers Perceptron (MLP) architecture [10]. The number of neurons from the hidden layers was chosen from initial tests and the neurons are completely connected.

In the next section it will be presented the FDI system development in the level plants, which is based on the neural identification in the two steps showed before.

IV. Fault Detection and Isolation

In accordance with the identification neural project considered, it was verified therefore that it would be useful for the fault isolation/classification stage, if the output of ANN 1 diverges from the real output of tank 1, for example, it means that a fault could not have occurred on the side and sensors from the process that evolves tank 2, and vice-versa, that is, reducing the possibilities at the moment of classifying the fault.

With identification in two steps, it is possible to get two residues, R_1 and R_2, where $R_1 = L_1 - L_1^{'}$ and $R_2 = L_2 - L_2^{'}$. The fault isolation strategy can be seen in Figure 5. In this case, a ANN, denominated ANN 3, is trained receiving as input data the values from R_1 and R_2. The networks output corresponds to a vector of $N+1$ numbers, where N is a quantity of faults that the network will be able to classify. For example, considering $N = 3$, the possible outputs for the network shown on table II would be:

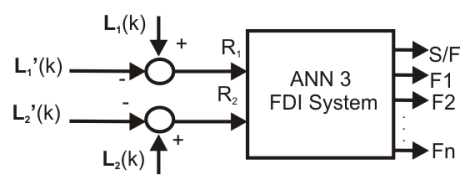

Fig. 5. General project for fault detection and isolation

TABLE II
Possible outputs for the classification of 3 faults.

Normal Situation	1	0	0	0
Fault 1	0	1	0	0
Fault 2	0	0	1	0
Fault 3	0	0	0	1

Considering the two residues to detect and isolate the faults, the FDI system could detect in maximum 8 different faults, in which the situation $R_1 = 0$ and $R_2 = 0$ would be a normal behavior of the system. With the intention of testing, we managed to foresee only six types of faults with joint distinct residues. Table III shows the six types of faults selected for the result analysis, where $+$ represents the positive residue, $-$ the negative residue and 0 the equal residue or very close to zero.

As the neural classifier works based on positive values, negative and null values for the residues, it could have been

TABLE III

DISPOSITION OF THE 6 FAULTS RESIDUES

Fault	Description	R_1	R_2
	Absence of fault	0	0
1	New hole in tank 1. No fall of water in tank 2	-	0
2	Decrease of hole in tank 1	+	-
3	Decrease of hole in tank 2	-	+
4	Increase of hole in tank 2	0	-
5	Extra flow of water in tank 1	+	0
6	Extra flow of water in tank 2	0	+

translated to a set of rules with conditional sentences. For example, if R_1 was positive and R_2 negative, it would be sufficient to make a conditional test to indicate that fault 2 is occurring. On the other hand, if another dynamic system could be modeled for N residues, where N is a very high value, it would be a little laborious to codify several conditional judgments to inform the fault. Besides this, the neural can classify faults with similar residues (two different faults with the same R_1 and R_2, for example, $R_1 = +$ and $R_2 = 0$), however with different amplitudes, without the necessity of a new conditional test verifying the threshold of the fault.

V. NEURAL IDENTIFICATION RESULTS

The data captured for training had been generated from the normal functioning of the plant levels, that is, it applied reference signals for the level of tank 2 and the control signal was generated from PID controller. The reference signals were always unit degree or sinusoidal type, with amplitude varying through the time.

Figure 6 shows one of the data sets used for training. In the first chart, there is the value of tank 1 throughout the time. The second chart shows the level of tank 2 in blue and the reference used by the PID controller in red. The third chart shows that the controller signal is being applied to the pump, generated by the controller.

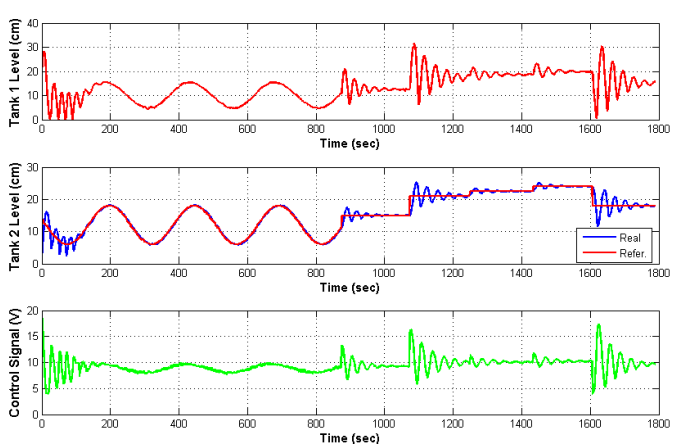

Fig. 6. Example of data set for training

After the training of the two networks used for the neural identification, the medium square errors found for the ANN 1 and the ANN 2 were approximately, 10^{-4} cm^2 and 10^{-5} cm^2, respectively. Examples of neural identification for data not presented in the training of the networks can be seen in Figure 7 and Figure 8. On the first chart are the levels of tank 1 real (blue) and the one generated by ANN 1 (red) throughout the time. On the second chart it is easy to see the level of tank 2 real (blue), generated by ANN 2 (red) and the reference to be followed by the level of this same tank (green). Finally, in the last chart are presented the residue values R_1 (blue) and R_2 (red).

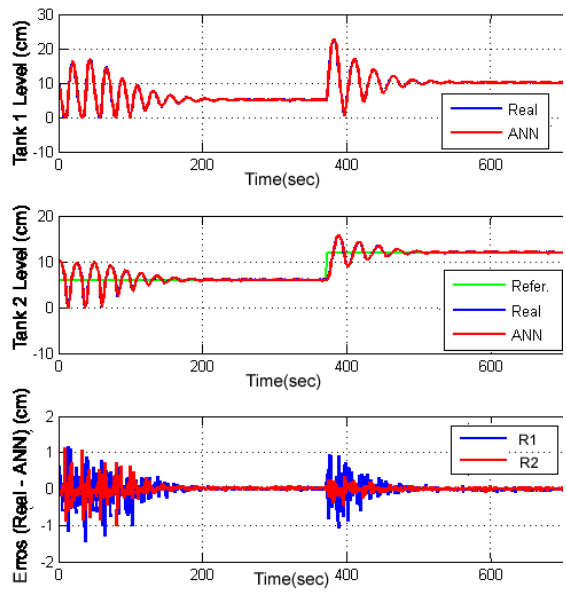

Fig. 7. Example 1 Neural Identification

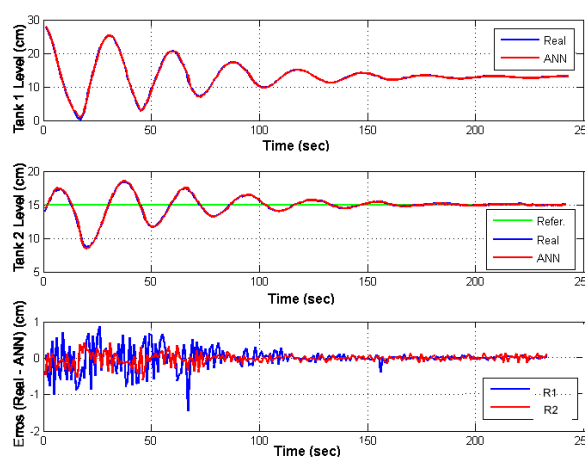

Fig. 8. Example 2 Neural Identification

Based on the results of the neural identification, it can be inferred that, for both situations, the errors between the level of the tanks and the generated ones in the identification are bigger during the transitory period of the tank 2 level. This

is due to the result of a lot of noises being present during the test and the pump has components with non-linear dynamic, making difficult the learning of neural networks.

In the next section results from the detection and isolation system will be presented as well as the fault level system isolation.

VI. Fault Detection and Isolation Results

The neural network, ANN 3, used to detect and isolate faults was trained only by using the neural identification results using the mathematical model of system [5]. With the mathematical model, faults were simulated and the residues captured, allowing the data generation suggesting the normal function of the system or the occurrence of fault. This approach has been chosen due to the fact that in real situations, where one does not have a data base with the information of occurring faults all the time, it could be complicated to generate faults in the system in order to be able to develop an application that could detect them.

Figure 9 shows a output of the supervisor program developed to monitor the system operation. It is possible to observe the occurrence or the non-occurrence of the system's faults according to the strategy described in section IV.

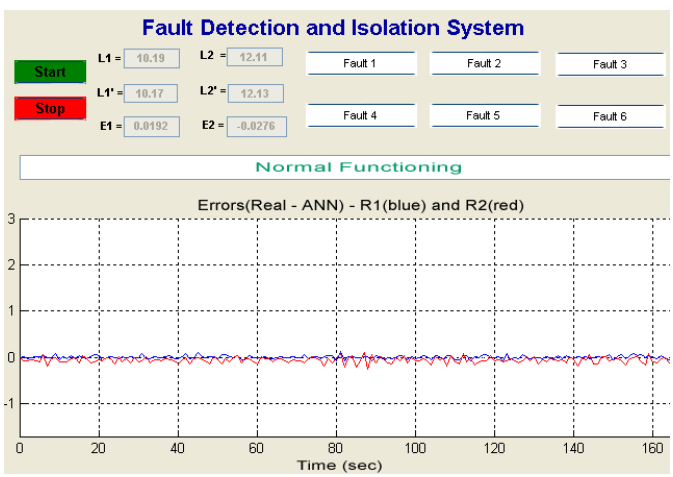

Fig. 9. Supervisory Program

In real time, instant values are exhibited from the real tanks level, as well as the estimated levels by the neural networks and the residue values. The supervisor program also shows a graph with the evaluation of residue values throughout time. In the case that faults occur, a corresponding alarm with the fault will be highlighted in red and a message informing the fault is displayed.

The four graphs of Figures 10, 11, 12 and 13 illustrate the moment when the faults of types 2, 3, 5 and 6 occur (see table II). At this moment, the residues diverge the values close to zero and the ANN 3 is able to identify and classify the fault. At this point the fault corresponding button changes to color red and the messages "Water increase in tank 1 and water decrease to tank 2" , "Water increase in tank 2 and water decrease in tank 1", "Water increase in tank 1" and "Water increase in tank 2" are shown as faults 2, 3, 5 and 6, respectively.

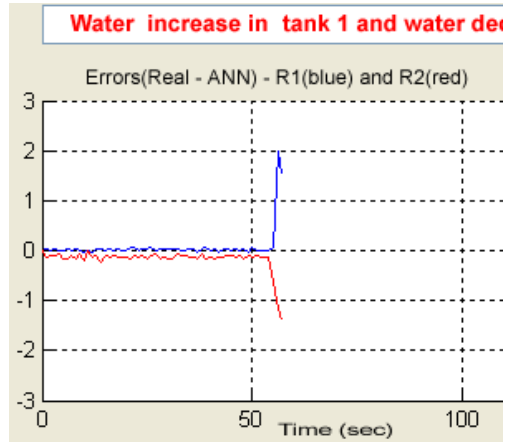

Fig. 10. Fault 2 - Decrease of small hole in Tank 1

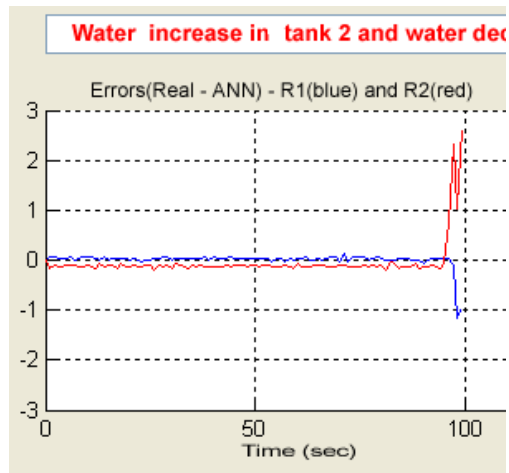

Fig. 11. Fault 3 - Decrease of small hole in Tank 2

As already mentioned, the strategy of the FDI system was implemented as much in the simulated environment as was in the real one. Many difficulties have been discovered while working in the real system, as for example: tests should have been made considering that the water in tank 1 would not have overflowed: the necessity of using extra equipment (power supply, module amplifier, etc.); the presence of leaks between hoses and pressure measures; problems with the neural identification at transitory moments due to the non-linearity of the pump (mathematically modeled to an ideal form), among others.

VII. Final Considerations and Expectations

This paper showed the development of a fault detection and isolation system. It was applied to a real system which was connected through an industrial network. This system was divided into two parts: neural identification and classification/isolation of the faults. It is executed in the course of real time and is able to accuse the faults the moment that

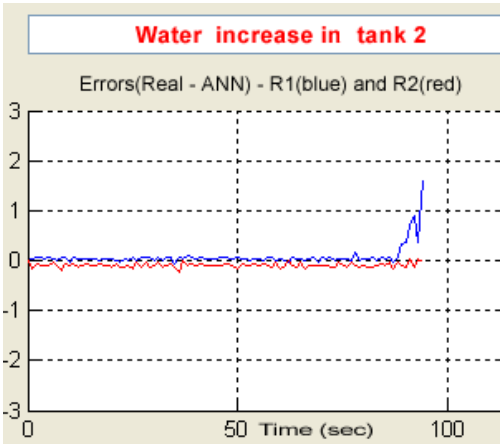

Fig. 12. Fault 5 - Extra water low in Tank 1

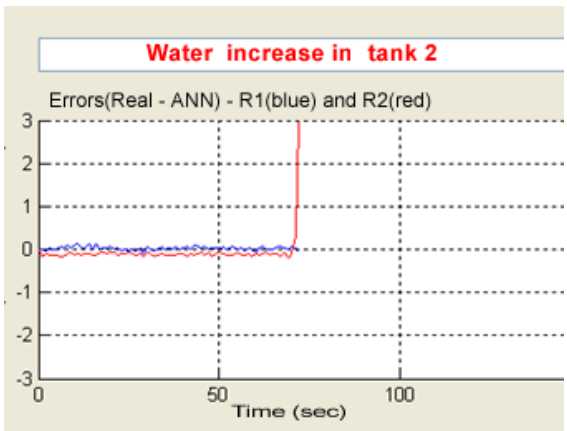

Fig. 13. Fault 6 - Extra water low in Tank 2

the abnormalities occur, surpassing all difficulties not found in the simulated environment.

Perspectively, it intends to improve the neural identification for situations in which the PID controller still has not established the level of tank 2 (transitory). Besides this, improve the neural classifier so that it can be able to detect and isolate faults that have similar behaviors to the residues R_1 and R_2.

REFERENCES

[1] L. Weber, "A script for explanation of the basic concepts of tolerance of faults", 2002.

[2] S. Persin, B. Tovornik, N. Muskinja and D. Valh, "Increasing process safety using analytical redundancy," Electrotechnical Review, 2002, pp. 240-246.

[3] 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, 2005.

[4] H.J. Shin, D. Eom and S. Kim "One-class support vector machines - an application in machine fault detection and classification," *Computer and Industrial Engineering* , 2005, pp. 395-408.

[5] R. Fernandes, D. Silva, L.A. Guedes, and A. Neto, "Neural Identification of the Systems of Levels in the Foundation Fieldbus environment," *Latin American Congress of automatics*, Salvador-BA-Brazil, 2006.

[6] Katsuhiko Ogata, *Modern Control Engineering,* LTC, 1998.

[7] F. Lima, L. Guedes, A. Ortiz and A. Maitelli "Hybrid Enviroment for Tests and Training in Fieldbuses," *VI Induscon* , 2004.

[8] Li Zheng and H. Nakagawa, "OPC (OLE for process control) specification and its developments," in *41st SICE Annual Conference*, vol. II, 2002, pp. 917-920.

[9] Coupled Water Tank Experiments. Quanser Innovate Educate.

[10] Simon Haykin, *Neural Networks: Principles and Practice,* Bookman, 2001.