Design Framework for Intelligent Supervision of Industrial Processes

CARLOS PARRA ORTEGA[1,2]
ELIEZER COLINA MORLES[2]
EDGAR CHACÒN RAMÍREZ[2]
[1]Departamento EEST
Universidad de Pamplona
Ciudadela Universitaria, Pamplona
COLOMBIA
[2]Escuela de Ingeniería de Sistemas
Facultad de Ingeniería
Universidad de Los Andes
Sector la Hechicera, Mérida
VENEZUELA
cparra@ula.ve
ecolina@ula.ve
echacon@ula.ve

Abstract: - Timely faults detection in an industrial process is a key aspect for designing a framework for intelligent supervisory control. A way to perform supervisory control is by providing intelligence to the supervision mechanism in continuous processes exposed to faults in order to cope with the identification of a diversity of faults, classify them and to be able to anticipate the consequences derived by their occurrence. In this article we propose an extension of the multiresolutional models approach to construct a fuzzy logic-agent technology-event detection approaches-based supervisory framework. Also the suggested framework is validated by means of a discrete-event simulation program.

Key-Words: - Intelligent Supervisory Control, Multiresolutional models, Fault detection, Fuzzy Events Detection, Discrete-event simulation.

1 Introduction

There is a set of abnormal conditions in the operation of a complex industrial process that may influence its performance and cause specification changes of final products or a misused of raw material. Many fault conditions might justify a plant halt, in special those caused by malfunctioning of equipment or devices. Some times, a timely detected fault means less repairing costs and the possibility of a non stopping plant operation. It is under such cumbersome circumstances that automation is better justified.

Supervisory Control Systems (SCS) are designed to provide high-level control capabilities for processes under normal operational conditions. The objective of this work is to present a framework for supervising a continuous production process, which is exposed to faults, so that it is possible to avoid halting the operation, despite the presence of minor faults. Also, the fault type will be diagnosed in order to put forward a set of actions that allow solving the fault condition. At the same time, the system itself will be able to postulate this set of actions in an automatic manner.

A multi-agent system to deploy the supervisor design framework combined with fuzzy logic event detectors and multiresolutional reasoning models within a distributed environment is contemplated in order to achieve the before mentioned specifications.

The validation of the framework will be performed by using discrete-event systems techniques applied to a hydro-pneumatic system.

Some previous works related to the proposed approach are the following: Cerrada *et al* [1] proposed single rules-based agent system design for faults detection in industrial processes, with specific roles for each agent type. Guillén *et al* [2] proposed a method for detecting faults using the wavelet transform in chemical processes, Altamiranda, *et al* [3] proposed an architecture that supports decisions taking in a control context, using multiresolutional models with fuzzy logic, applied to hybrid control systems. Chacón *et al* [4] presented an approach to build supervisory systems for complex continuous production processes.

This paper is structured as follows: section 2 presents the statement of the problem. Section 3 contains the suggested design framework to solve the proposed problem. Section 4 shows a study case. The validation mechanism is explained in section 5 while section 6 shows some experimental results. Finally, the work includes some conclusions.

2 Supervisory control highlights

An ideal condition for continuous process control is to get one hundred percent reliability. Unfortunately, fault conditions which may be caused by damage or malfunction of industrial equipments or the presence of disturbances in measurement process instruments are part of the set of situations which must be considered in the operation of the process. In continuous production processes these situations are critical and might imply to halt the process with the consequent losses in time and resources.

It is in this context where measurements of process variables take place and should be used to detect the occurrence of events. Sometimes these measurements contain disturbances, which arises from the nature of variable to be measured, for example: the waves in tanks with input and output flows, unforeseen inputs of raw material (as rain in tanks) or the accuracy of measuring devices, among other cases. In production systems, the state variables that describe the processes are of discrete nature and therefore it is important to apply correctly a mapping function to transform the continuous variables to discrete states.

For systems with the disturbance features mentioned before, the use simple direct rules (for example, if-then-else) in order to find changes in operational regions is not recommended, particularly because of the following facts: it may be that because of the sampling period of the measurement devices, during a time period there are many commutations between successive regions, causing the execution of control actions before the system becomes stable, which is not recommended in critical mission systems. It may also happen that some system elements might present faults, which in some way would be reflected in the measured variables, and whose occurrence is difficult to detect in a direct manner.

3 Proposed design framework

The focus of this design framework is to obtain a intelligent supervision system based on combining several automation and computational methods, such as the ones described in [5, 6, 7, 8, 9, 10, 11].

3.1 Reasoning model design

An extension of Sanz's [6] multiresolutional model is depicted in figure 1, where the starting point is a numeric model, (N Model), obtained using physical principles and measurements of interest variables. A Local Qualitative Model (Q_L Model) describes the process logic and operation under nominal conditions. A Global Qualitative Model (Q_G Model) describes the general state of the process, whereas a Reasoning Model (R Model) represents the knowledge about the process by means of logical rules.

Taking process decisions in a supervisory control system requires using adequate models, capable of describing the dynamical behavior of the process to be controlled. Such models allow specify control actions, facing the occurrence of random disturbances or changes in the operational regions of the process in order to assure its stability and reliability.

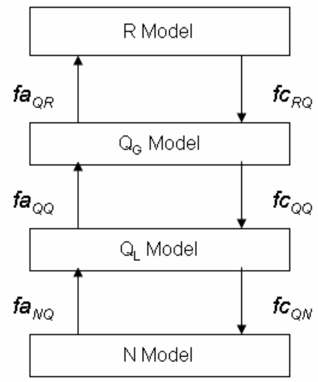

Figure 1. The proposed multiresolutional model.

Multiresolutional models in supervision and control environments allow simultaneously:

- To describe dynamical behavior of the process facing disturbances or control actions.
- To provide relevant and updated information about the process's status.
- To make easy the communication between users and processes.

These models are coupled by means of abstraction functions (f_a) and concretion functions (f_c), used to specify values of high-level variables starting from values of variables in shop-floor level.

A merge of the above mentioned approaches is depicted in fig 2. The behavior of the industrial process located at shop-floor is described in terms of an N model using physical principles, such as mass balance and energy conservation. Here, measurements taken from interest variables and knowledge of the operational condition of the process are assumed to be available. The industrial process is exposed to disturbances and its measured variables are sampled and sent to an Event Detector that applies an abstraction function in order to map these values with the occurrence of events, which are received by the Supervisor. This Supervisor contains the qualitative and reasoning models of the process. The event received by the supervisor is applied to a Q_L model, which contains the normal operation conditions. Some of the detected events will correspond to global events, which are detected by the fa_{CC} function and denote the faults occurrence. The Supervisor contains a plan whose activities can be expressed as a sequence of events, which are generated by rules modeled in the R model.

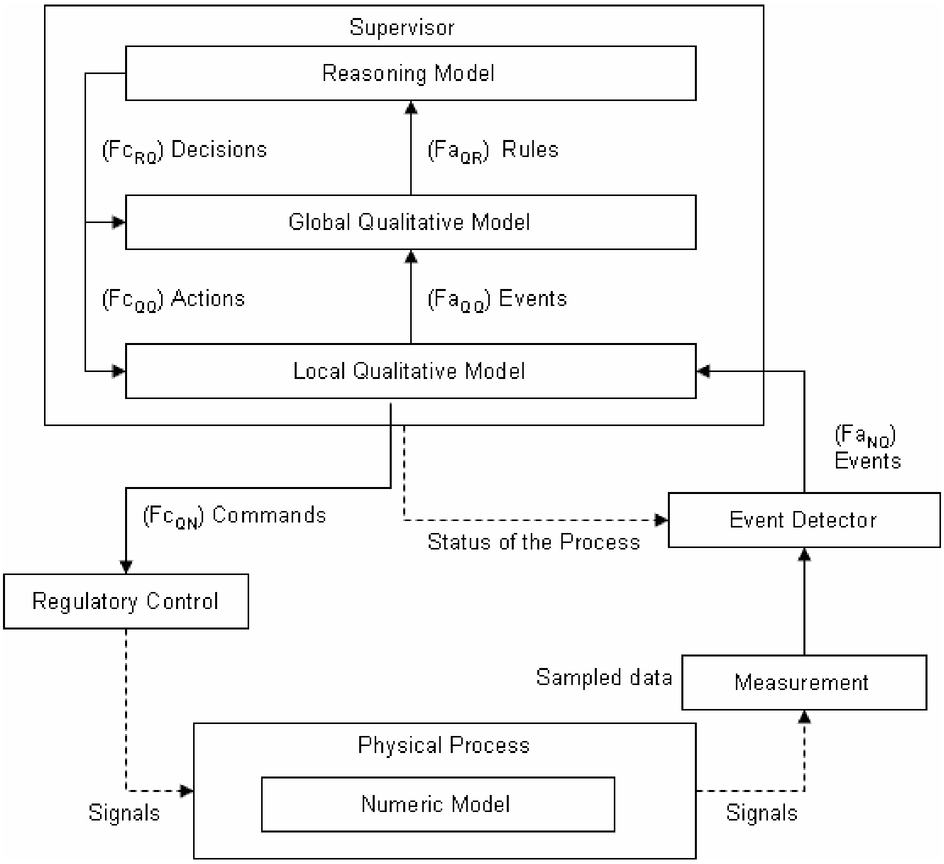

Figure 2. Supervisory framework

On the other hand, depending upon the fc_{CC} actions provided by the reasoning model and the detected events, the local qualitative model generates fc_{CN} rules of regulatory control which are translated into references to the regulatory controllers, in order to produce changes in the physical properties of the process, which has influence over the numeric model. For example, executing a repairing action to reinitialize a process or changing the configuration of the plant when is being operated under normal conditions.

3.2 Computational implementation issues

A way of implementing a Supervisory Control System with the design specifications mentioned above is by using an agent based approach that facilitates the emulation of a distributed environment similar to an industrial plant, with computers and electronic devices connected by a field net. One of the agents implements an Event Detector, whose mission is to receive the samples of the physical process sensors, which have a failure probability. According to the trend of measured data, the event detector will apply a set of fuzzy rules in order to determine a) the operational region of the system, b) a change in operational region, and c) the presence of some devices fault types. The agent that implements the supervisor with reasoning rules loaded from a master rules file is highly flexible, since it can switch the reasoning rules as a consequence of changes in the conditions of the system. This rules are specified in terms of First Order Logic principles, according to Kowalski model [12] and there are two types of rules: proactive rules, that generate plans of activities (aimed to produce event sequences), and reactive ones, which make that agent responds to observed events. The premises of reactive rules corresponds to the evolution of discrete-event systems (DES), expressed as finite-state machine, which are similar to the logical description of local and global qualitative models. Other rules that are no related with DES correspond to communication between agents. There are complementary rules to indicate how to select an action among several actions (selection), or how to inhibit the occurrence of a predictable future event. Fig 3 depicts the implementation of the supervisor and event detector.

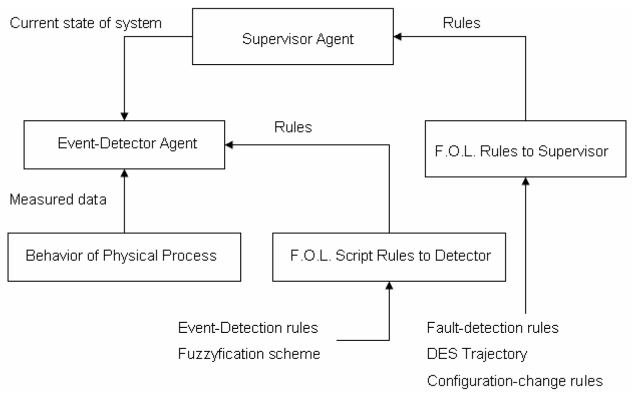

Figure 3. Implementation diagram.

4 A Case of Study

Next, a supervisory control scheme design for a fault-exposed hydro-pneumatic system is shown. The system to be supervised is composed of the following components: a horizontal cylinder shaped tank that contains water and air, an external tank, an electrical pump and a compressor. The global objective of the system consists in providing a good water supply, while keeping the water level and air pressure within adequate range of values. The air pressure determines the water supply quality for the users; and therefore, it is a variable that must be controlled. If the level of water descends below certain value it is necessary to pump water, which causes an incoming flow into the tank, while if the level grows further a maximal value, the pumping is stopped. The measurement of water level is influenced by some perturbations, such as surge effect and turbulence. On the other side, the compressor, the electrical pump, level sensors and pressure sensors may be subjected to failure at any time. These fault conditions are expressed in some measured values that do not match with the expected behavior of the system, as established by the Supervisory Control. The values of the tank level must vary within 0.65m and 1.22m, while the system pressure must fluctuate between 1.43 kg-f/cm2 and 5.17 kg-f/cm2. Figure 4 illustrates the system.

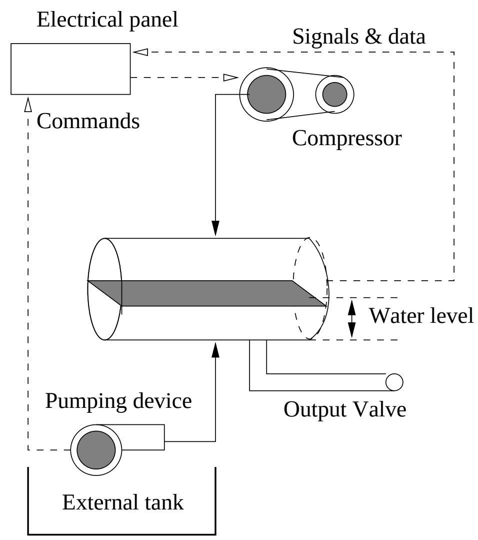

Figure 4. Hydro-pneumatic system scheme

4.1 Numeric Model

The water level and the pressure in the tank will be considered as state variables of the system. Each variable may be represented by the following ordinary differential equations:

$$h'(t) = \frac{\lambda(t) - \mu(t)}{r^2 \cos^{-1}(1 - h/r) - (r - h)\sqrt{2rh - h^2}} \quad (1)$$

$$P'(t) = P_0 \frac{-V'(t)}{V_t} \quad (2)$$

Where $h'(t)$ is the rate of change for water level in the tank, h is the instant height in a given time, r is the radius of the tank, while $\lambda(t)$ and $\mu(t)$ are the incoming flow of water and out flow of water, respectively and $P'(t)$ is the rate change of pressure at time t, P_0 is the initial pressure at time $t=0$, $V'(t)$ is the volumetric rate of change for air at time t, and V_t is the remaining volume in the tank being filled by water at time t.

In order to simulate the measuring of disturbed data two techniques are combined: the *sliding window*, as proposed by Sarrate [13] and the perturbation analysis, by Ho and Cao [14]. The measurements are obtained as it follows: the last simulated reading of a sensor is averaged with the seven prior readings to obtain a *window attribute*, which also includes the standard deviation. Then, a small deviation is added to the window attribute according to a random perturbation factor, which multiplies the standard deviation of the sample. Thus, a disturbed attribute value is obtained which emulates a sensor whose measuring instrument was exposed to electronic noise.

4.2 Local Qualitative Model

This model describes the discrete state under normal operational conditions. For this purpose, nine operational regions which involve level and pressure have been defined, as shown in fig 5.

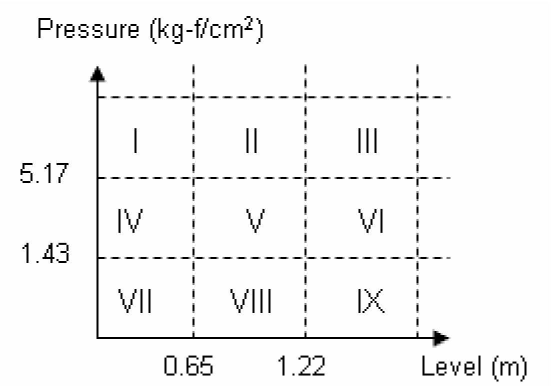

Figure 5. Normal condition operation regions.

These operation regions can be mapped to a finite-state automaton, as shown in fig 6.

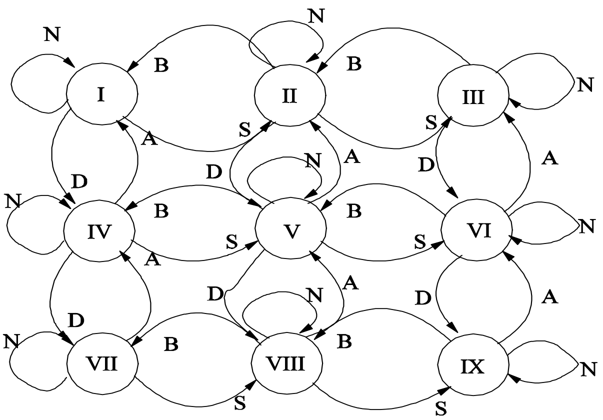

Figure 6. Normal tank operation automaton

Each state corresponds to one operation region, and there are nine statuses. The possible transitions are:
N: Variables changes suggest that none event happen.
S: Variables changes suggest a rise in water level.
B: Variables evolutions suggest a fall in the water level.
A: Variables evolutions suggest a pressure rise.
D: Variables evolutions suggest a pressure decrease, changing the region to another region with less pressure.

If a state change occurs with other transition different to those mentioned above, it will indicate a fail occurrence. In the following subsections this topic will be treated in more detail.

4.2.1. Operation state settling - Fa_{NC} function

In order to avoid unnecessary state changes, due to fluctuations in measurement devices or by the effect of surge within the tank, a fuzzy logic based scheme is required. This scheme allows mapping in a non-ambiguous manner the numeric values taken from variable's measurement to linguistic variables that describe the system's status (operation regions). Fig 7 shows the fuzzyfication scheme for pressure, mapped to three values: low, medium and high.

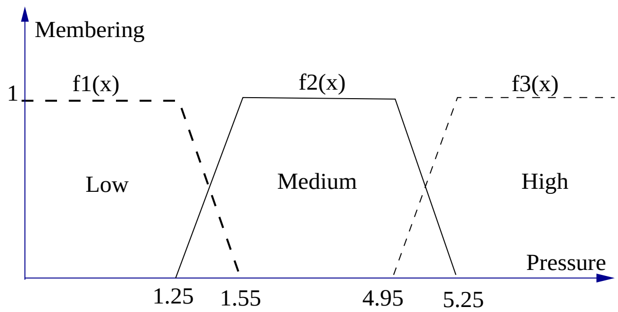

Figure 7. Pressure values fuzzyfication.

Figure 8 summarizes the classification in order to deduce in which operational region the system is evolving, given the qualitative values of water level and pressure. L\P means Level\Pressure.

L\P	Low	Medium	High
Low	VII	VIII	IX
Medium	IV	V	VI
High	I	II	III

Figure 8. Operation region classification.

This classification, expressed in ruled-form, will take the following form, for example:

If level is low and pressure is low, the operation region is VII

If level is medium and pressure is medium, the operation region is V

And so forth, for the other rules.

4.2.2 Events determination rules

They have the form as shown in the example:

If operation region is V and the pressure falls, the region switches to VIII (Event D)

For every operational region it is possible to define these rules, exhaustively. Table 1 shows some of these rules.

Table 1. Normal operation mode rules.

Current Region	Qualitative variable	Event
I	Low level, high pressure	N
I	Medium level, high pressure	S
I	Low level, medium pressure	D
I	Other values	Fault
II	Medium level, high pressure	N
II	Medium level, medium pressure	D
II	Low level, high pressure	B
II	High level, high pressure	S
II	Other values	Fault
⋮	⋮	⋮
IX	High leve, low pressure	N
IX	Médium leve, low pressure	B
IX	High level, medium pressure	A
IX	Other values	Fault

4.2.3. Fault types.

Due to the presence of mechanical, electrical and electronic devices in the system some faults may occur, changing its normal operational condition. Commons faults are the following ones:

- Fault in compressor, or exists an air leak (Despite of be running, the pressure don’t rises, even without water’s consumption)
- Fault in the pump, or exists a water leak in the pipe, or the external tank are exhaust (Despite of the pump is running, the level continues decreasing).
- Fault in the water height’s sensors or pressure sensors (the measurements are disturbed or are irregular)

4.3 Global Qualitative Model

The hydro-pneumatic system has a normal condition, where all operation regions mentioned before are feasible. When a fault happens, it is not valid to consider the operation regions shown in fig 5. The global dynamics of this fault-exposed system is shown in fig 9, using a finite-state machine. The initial state is labeled "normal", when it is allowed to consider the operational regions mentioned before; certain events detected in the normal state do not change this condition, while other events suggest that a fault condition is happening. When an event *F* (fault) occurs, the system changes its state to "Fault", and starting from there two events may happen: if the fault is serious the system evolves to state "Not available", due to the occurrence of event *E*, or the fault may be repaired without dismantle the system, which implies that event *G* would have happened, which leads to "Repairing" state. If this repairing process is successful, the system enters to normal operation again, by means of event *K*. Some repairing processes imply that the system will be reset; therefore it will be guided to a "stopped" condition, starting from which the system starts (event *A*) in order to return to normal operation. For example in the replacement of a defective pump, the tank must be drained out in order to install the new pump, and then it will be required to pump water and air until the values of variables reach normal ranges.

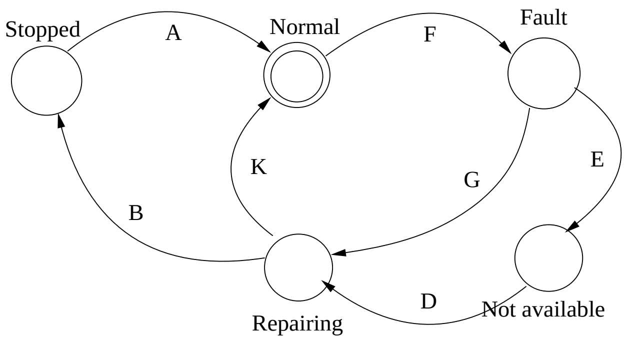

Figure 9. Global behavior automaton

In order to apply the concepts of supervisory control based on language theory, it is required to characterize the behavior of the system to be supervised. This is achieved by defining the marked language of the automaton showed in fig. 8. The generated language given by the automaton is:

$$Lm(G) = (F(G + ED)K + BA)^* \quad (3)$$

Here, events *K, A, D* and *G* are controllable, whereas events *F*, *B* and *E* are uncontrollable. When the system is in normal operation mode, it is valid to consider the finite state machine showed in fig 6.

4.4 Supervision

As seen in fig 8, there is only one desirable state: normal operation condition, but given that event *F* can not be inhibited, then the plant evolves through the Fault – Repairing states, at best case. Likewise, due to event *E* is uncontrollable and can not be inhibited, the plant also evolves through the Fault – Not available – Repairing states, at worst case. Therefore the supervision is achieved by causing events after the fault condition, in other words, whenever event *F* occurs it is necessary to allow and produce an event *G* and *K*, which are controllable. On the other hand, if the sequence of events *F-E* occurs, the sequence *D-K* is allowed and must be generated, and so on for other sequences. Since the hydro-pneumatic system has to work without halting, the designer has to consider some repairing actions be carried out while the system is running.

4.5. Production system setting - (R Model):

The settings of hydro-pneumatic system may be established by changing the pump and compressor states, described by two discrete variables. Their values are 0 (off) and 1 (on). This setting must change according to a plan made by the Supervisor, based on the operational region in which the system is in, and the last detected event. In general, it operates in terms of condition - action rules, as shown in the next examples.

If operation region is V, then do not to manipulate.

If the variable has not been manipulated and region is 1 and compressor is on, then turn off the compressor.

If the variable has not been manipulated and region is 3 and pump is on, then turn off the pump.

If the variable has not been manipulated and region is 7 and pump is off, then turn on the pump.

The following table shows how to manipulate according to the operation region and configuration of the system:

Table 3. Reasoning Model.

Z	B=0	C=0	B=1	C=1
1	-	-	-	$C \rightarrow 0$
2	-	-	-	$C \rightarrow 0$
3	-	-	$B \rightarrow 0$	$C \rightarrow 0$
4	-	-	-	-
5	-	-	-	-
6	-	-	-	-
7	$B \rightarrow 1$	-	-	-
8	-	-	-	-
9	-	$C \rightarrow 1$	-	-

Where the symbol "-" indicates that the system configuration can not be modified, "→0" suggests that the device must to be switched off, and "→1" suggests that the device must to be switched on.

Once this reasoning model is obtained, consignments to the controllers are derived, depending on the desired status for the pumping device and the compressor. Moreover, high-level commands may be deducted in order to indicate faults occurrence, which require repairing actions or even human intervention in order to reinitialize the system.

5 Design validation

The validating schema is depicted in figure 10. A DEVS & DESS-based formalism simulator [15] that implements discrete-event systems represents the hydro-pneumatic system elements. It runs on a simulation platform named *Galatea* [16], developed in the Java programming language. The multi-agent environment JADE [17] was chosen for implementing the Detector and Supervisor agents. Both agents will map their actions over the DEVS simulator by means of an interface program.

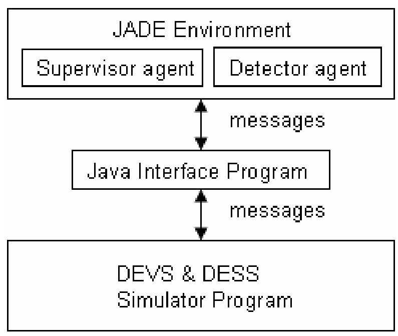

Figure 10. Validation scheme.

Once the test bed with the agents, DEVS Simulator and interface were implemented, some simulations were executed under the following settings:

- Reference height for level: between 0.5 and 1.2m.
- Reference pressure: between 1.47 and 5.17 Kg-f/cm^2.
- Input flow: 0.01 m^3/seg (10 liters/sec).
- Output flow: randomly fluctuates around 0.001 m^3/sec.
- Initial pressure: 2.067 Kg-f/cm^2
- Time between observations of event-detector agent: 2 seconds.
- The event-detector samples eight last observations in order to apply perturbation factor.

6 Obtained results

Some traces that represent the behavior of the relevant variables of the system under different scenarios, normal operation, operation with faults in sensors and operation with faults in the pumping device, were obtained after many simulation runs. The trace of continuous variables shows the states evolution, which qualitatively represents both level and pressure and the global state evolution. Figures 11 and 12 illustrate the water average level and pressure, respectively. It may also be observed on the first hand, the system behavior under normal mode of operation, with the level fluctuating between the minimal and maximal allowed values and secondly, the behavior under a fault condition in the pumping device, which implies to shut the output valve while the replacement or reparation of the pumping device was executed, which may takes some unspecified time. The tank level was stable during the repairing phase because there was not any flow in or out of the tank.

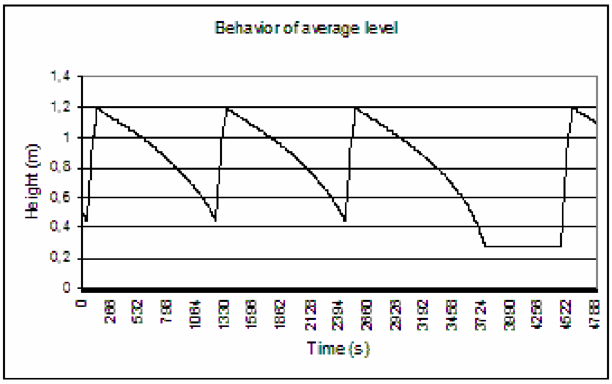

Figure 11. Average water level evolution in the tank

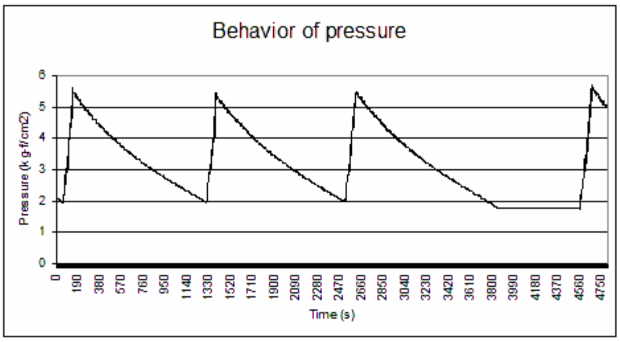

Figure 12. Trace of pressure in the tank.

Another manner of assess the system behavior consists in recording the global status path, just like qualitative regions in which the process is operating at all times. This will allow executing a trace of state's changes, eventually facing with fault conditions and checking the system behavior before and after that fault event for all variables. Figures 13 and 14 illustrate these traces. Notice that the time scale was modified in order to represent more exactly the changes caused by the occurred events. The continuous line shows the evolution of the global system state: value 0 indicates normal operation, whereas value 1 indicates fault condition, and value 2 indicates repairing situation. The dotted line shows the evolution of the qualitative state under normal operation. Each value corresponds to one of the regions shown in fig 5.

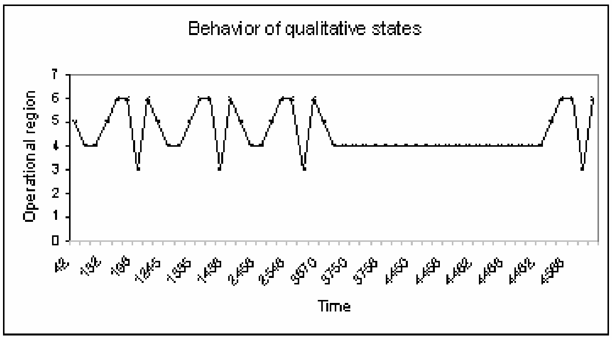

Figure 13. Trace of behavior of qualitative states.

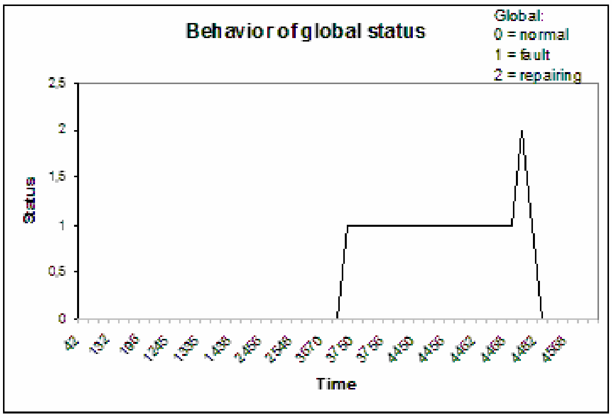

Figure 14. Trace of global status.

The system may be analyzed by tracing its configuration over the time, and confronting with the evolution of variables, both continuous and discrete. A fragment of the configuration trace has the following form:

$$Q_5 \rightarrow Q_4 \xrightarrow{X_1} Q_5 \rightarrow Q_6 \xrightarrow{X_2} Q_3 \rightarrow Q_6 \rightarrow Q_5 \rightarrow Q_4 \cdots$$

Where Q_i indicates *i-th* operational region, whereas X_1 and X_2 indicate the actions of turn on and turn off the pumping device, respectively.

7 Conclusions

The following conclusions are the result of experiences gathered during the design of the mechanism that supervises the considered *hydro-pneumatic system* exposed to faults and the combination and implementation of several approach as well as their validation.

- The combination of the methodology of multiresolutional model with fuzzy logic techniques and supervisory control to obtain a strong mechanism of decision taking was a successful task.
- The discrete-event simulation environment is an appropriate method to validate the design of a supervisory control system, since it allows the measuring of performance of the control mechanism.
- The application of fuzzy logic techniques for events detection in disturbed variables processes allow to determine the process status efficiently.
- The main elements of supervisory control system may be implemented with agent technology, which enforce the intelligence of control and communication capabilities.

References:

[1] Cerrada M, Aguilar J, Cardillo J, Faneite R, Agent-based design for fault management system in industrial processes, Revista Técnica de Ingeniería, Universidad del Zulia, Vol 29, No. 3, 2006, pp. 258-268

[2] Guillén P, Paredes M, Camacho O, A proposal method for fault detection and diagnosis in chemical processes instrumentation using wavelet transform, *Revista Técnica de Ingeniería, Universidad del Zulia*, Vol.28, No.1, 2005, pp. 68-77.

[3] Altamiranda E, Colina E, Chacón E. Intelligent Supervisory Systems for Industrial Process Control. *WSEAS Transactions on Systems*, Vol 4, No. 7, 2005, pp. 945 – 949.

[4] Chacón E, Altamiranda E, Colina E. A Holonic Architecture for Continuous Processes Supervisión and Coordination. *WSEAS Transactions on Systems*, Vol 4, No. 7, pp 941 – 944.

[5] Ramadge J, Wonham M. Supervisory Control of a class of Discrete-Event Process. *SIAM Journal of Control and Optimization*, Vol 25, No. 1, 1987, pp. 206 – 230

[6] Sanz R, *Arquitectura de Control Inteligente de Procesos. PhD Thesis*, Universidad Politécnica de Madrid, 1990.

[7] Parra C, Colina E, Chacón E, Intelligent Supervisory Control Design Framework for Fault Exposed Processes. *NAUN Internacional Journal of Circuits, Systems and Signal Processing*, Vol 1, Issue 3, 2007, pp. 251-258.

[8] Parra C, Colina E, Chacón E, Intelligent Supervisory Control of Continuous Processes exposed to Faults: a practical approach. In *proceedings of WSEAS American Conference on Applied Mathematics Math'08*, 2008, pp. 252 – 257.

[9] Zadeh L, Outline of a new approach to the analysis of complex systems and decision processes, *IEEE Trans. On Systems, Man and Cybernetics*, 3, 28-44, 1973.

[10] Mamdani E, Application of fuzzy algorithms for control of simple dynamic plant, *Proceedings IEEE* 121, 1585-1588, 1974.

[11] Yulong J, Yongsun C, Xiong Y, Myeonggil C, Kwanhee H, Sangmoon S, A Knowledge Acquisition and Management System for Fault Diagnosis and Maintenance of Equipments. *WSEAS Transactions on Systems*, Vol 6. No. 11, 2007, pp. 1330-1336.

[12] Kowalski R, *How to Be artificially intelligent - the logical way*. Draft, revised February 2004,

Available on line, URL http://www-lp.doc.ic.ac.uk/UserPages/staff/rak/rak.html.
[13] Sarrate R. *Supervisió intelligent de processos dinàmics basada en Esdevenivents*. PhD Thesis. Universitat politècnica de Catalunya, 2002.
[14] Ho Y, Cao X. Perturbation Analysis of discrete event dynamic systems. Kluwer international series on discrete event dynamic systems, 1993.
[15] Zeigler B, Praehofer H, Kim T. *Theory of modeling and simulation,* Academic Press, 1999
[16] Uzcátegui, M. *GALATEA, a multi-agent simulation language. Msc thesis*. Universidad de Los Andes, 2002.
[17] JADE. url: http://www.tilab.com