Intelligent Supervisory Control Design Framework for Fault Exposed Processes

CARLOS PARRA ORTEGA
Systems and Telecommunications Department
Universidad de Pamplona
COLOMBIA

ELIÉZER COLINA MORLES
Control Systems Department
Universidad de Pamplona, Mérida
VENEZUELA

EDGAR CHACÓN RAMÍREZ
Computing Deparment
Universidad de Los Andes, Mérida
VENEZUELA

Abstract:- The relevance of timely detecting faults in an industrial process has motive our research work, which consists in designing a framework for intelligent supervisory control for industrial processes. In order to carry out supervisory control in continuous processes exposed to faults it is required to provide intelligence to the supervision mechanism in order to cope with the identification of a diversity of faults starting from data obtained by measuring the process variables, classify the origin of these faults and to be able to anticipate the consequences derived by their occurrence. In this article we propose to extend the idea of using multiresolutional models to construct a framework based upon fuzzy logic, agent technology and event detection approaches. Also in this work the suggested framework is validated by means of a discrete-event simulation program.

Keywords:- Intelligent Supervisory Control, Multiresolutional model, Fault detection, Fuzzy Events Detection, Discrete-event simulation.

1. Introduction

The automation of industrial continuous processes is a complex task, particularly when the process can not be stopped while a variety of situations take place, and these situations do have influence over the configuration of the plant that may cause changes in the specification of final products, or the raw material flowing in. 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.

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 proposal 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. Besides, the fault type can 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.

In order to achieve the specification mentioned above, a practical approach that combines the classical supervisory systems design with fuzzy logic methods, in order to detect events and multiresolutional reasoning models to obtain control actions is proposed. Also, a multi-agent system to deploy the supervisor design and event detectors within a distributed environment is contemplated.

The resulting design is validated in terms of using modeling and simulation of discrete-event systems techniques, applied to a hydro-pneumatic system case.

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.

This document is structured as follows: section 2 presents the statement of the problem. The following section contains a proposal to solve the proposed problem. Section 4 shows a study case. The validation mechanism is exposed in section 5 while section 6 shows some experimental results. Finally, the work includes some conclusions.

2. Supervisory Control under Faults

Within the context of continuous production process, an ideal condition for process control is to get one hundred percent reliability. Unfortunately, fault conditions are part of the set of situations which must be considered and which may be caused by damage or malfunction of industrial equipments, the presence of disturbances in measurement process instruments, misinterpretation of the state in which the process is in; which may cause the execution of inadequate procedures, or a combination of the before mentioned facts. In continuous production processes these situations are critical, since they might imply to halt the process, in order to carry out corrective actions, and then reset the process, with the consequent losses in time and resources.

In the context of these continuous systems it is necessary to make measurements of the variables which pertain to the process to be controlled, in order 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. Since in production systems, the state variables that describe the processes are of discrete nature there is a function that maps from continuous variable to discrete states. For this reason, it is very important to apply this function correctly.

For systems with the disturbance features mentioned earlier, it is not possible to use simple direct rules (for example, if-then-else) in order to find changes in operational regions, because of the following facts: it may be that during a time period there are many commutations between successive regions due to the sampling period of the measurement devices, causing the execution of control actions before the system becomes stable, which is not recommended in critical mission systems. It may also occur 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 Solution

A way to solve the problem of supervising an industrial continuous process exposed to faults is to combine several automation and computational methods like the classic supervisory control approach based on finite-state machines, as proposed by Ramadge and Wonham [4]. The extension of the Sanz's multiresolutional model [5] in the context of intelligent control can be implemented in order to generate control decisions rules starting from the dynamical behavior of the process, described by continuous variables. Reasoning models based on artificial intelligence techniques, such as fuzzy logic [6, 7], in order to handle data exposed to large disturbances or changing trends can also be contemplated. Finally, the use of an intelligent agents system, which may be implemented in a distributed environment such as an industrial plant [8] and provide communication and collaboration easiness is yet another approach. The proposed solution is based upon using multiresolutional models merged with a fuzzy logic and agent based system to perform fault event detection and control.

3.1 Reasoning Model Design

The extension of Sanz's 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.

In order to take process decisions in a supervisory control system it is required to use adequate models, capable of describing the dynamical behavior of the process to be controlled. It is by means of such models that it is possible to specify control actions, facing the occurrence of random disturbances or changes in the operational regions of the process that assures its stability and reliability.

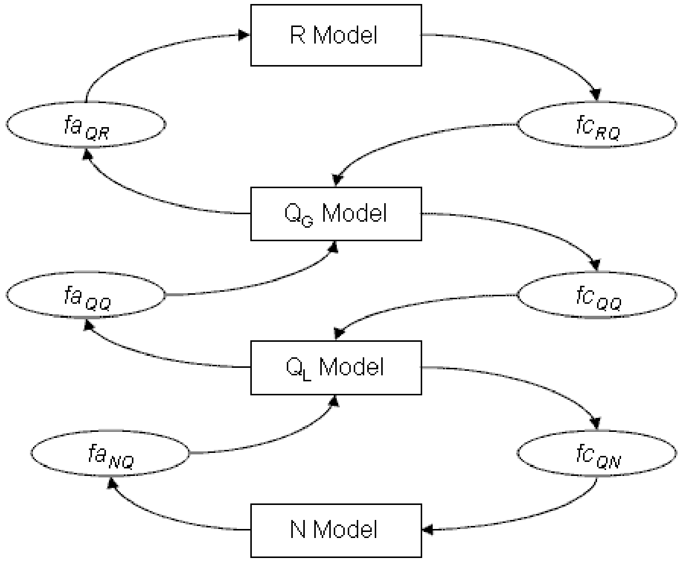

Figure 1. Multiresolutional model that extends the NQR model proposed by Sanz.

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 (fa) and concretion functions (fc), used to specify values of high-level variables starting from values of variables in shop-floor level.

A scheme that shows a merge of the above mentioned approaches is depicted at 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 data 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 QL model, which contains the normal operation conditions. Some of the detected events will correspond to global events, which are detected by the Fa_{QQ} 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.

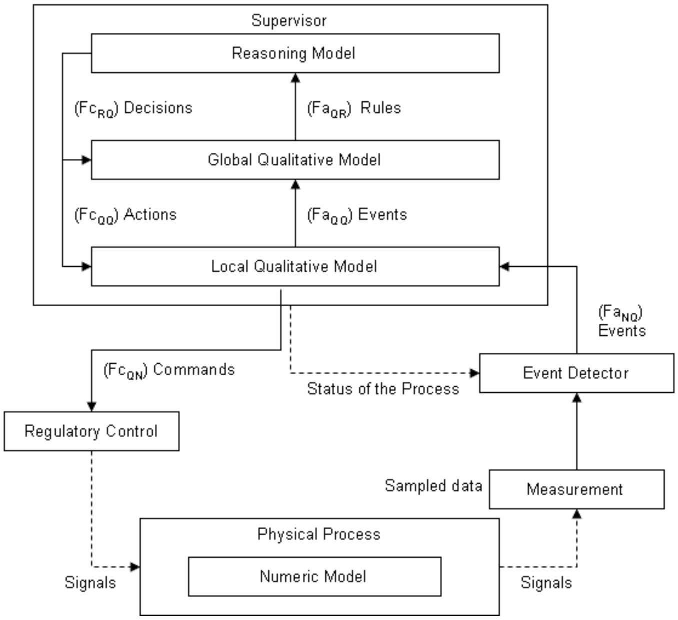

Figure 2. Proposed supervisory scheme.

On the other hand, depending upon the Fc_{QQ} actions provided by the reasoning model and the detected events, the local qualitative model generates Fc_{QN} 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

There are several ways of implementing a Supervisory Control System with the design specifications mentioned above. 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 is used here. 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 [9] 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 that allow indicating how to select an action among several actions (selection), or how to inhibit the occurrence of a predictable future event. Fig 3 shows the way of implementing the supervisor and event detector.

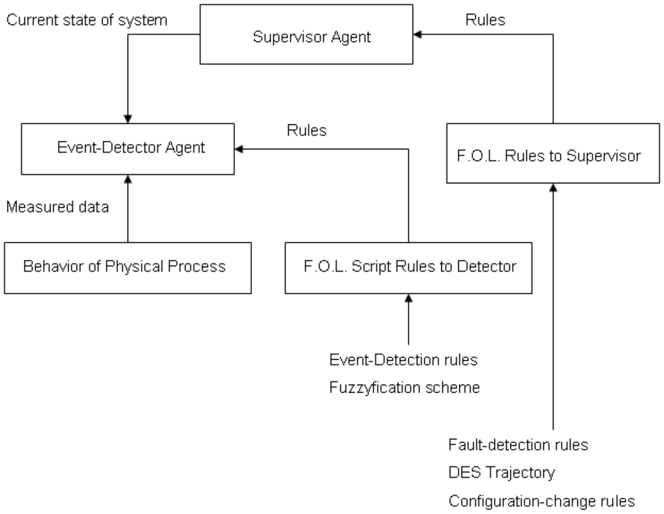

Figure 3. Schema of implementation.

4 A case of study

The design of a supervisory control scheme for a fault-exposed continuous process is shown next for the problem of controlling a hydro-pneumatic system in a building. The system to be supervised is composed of the following components: a tank whose shape is a horizontal cylinder, and 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 don't 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/cm^2 and 5.17 kg-f/cm^2. A scheme of this system is shown in figure 4.

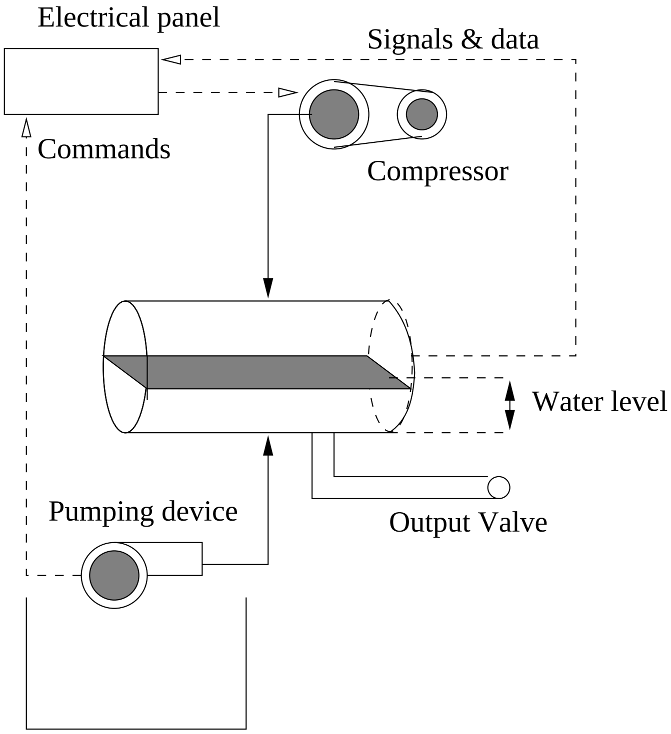

Figure 4. Scheme of hydro-pneumatic system

4.1 Numeric Model

In order to describe the physical behavior of the system, the water level and the pressure in the tank will be considered as state variables. Each variable can be represented by ordinary differential equations, which are shown as follows:

The level of the tank is expressed by the following equation:

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

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.

Assuming that the air in the tank behaviors as ideal gas, the pressure change is described by the expression:

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

Where $P'(t)$ is the changing rate 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 are the volume remaining in the tank being filled by water at time t.

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.

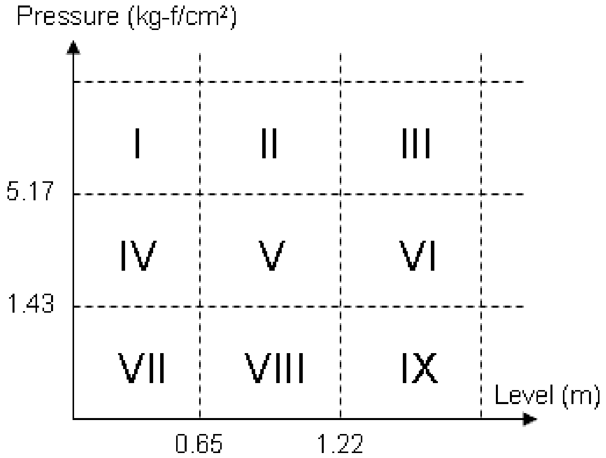

Figure 5. Normal condition operation regions.

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

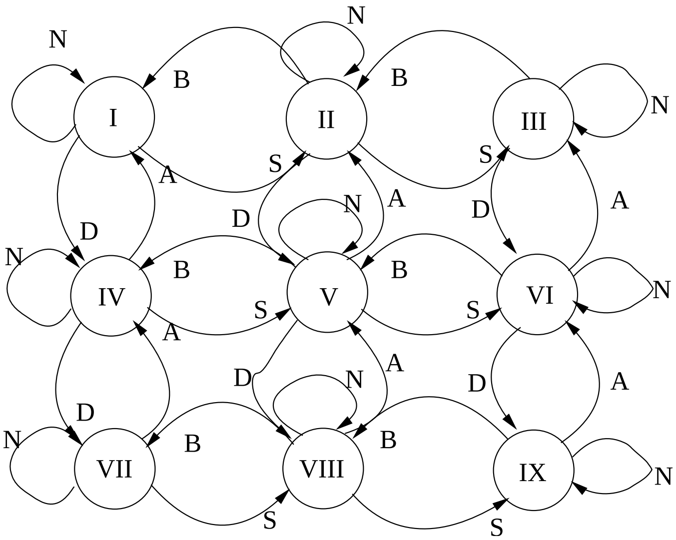

Figure 6. Normal tank operation automaton.

Each state corresponds to one operational 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 fault occurrence. The following subsections this topic will be treated in more detail. In fact, this automaton operates as a high-level event detector.

Settling the operation state of production system – Fa$_{NC}$ function. In order to avoid unnecessary state changes due to fluctuations in measurement devices or due to the effect of surge within the tank, a scheme based on fuzzy logic, that allows mapping in a non-ambiguous manner the numeric values taken from variables' measurement to linguistic variables that describe the system's status (operational regions) is required. Fig 7 shows the fuzzyfication scheme for pressure, mapped into three values: low, medium and high.

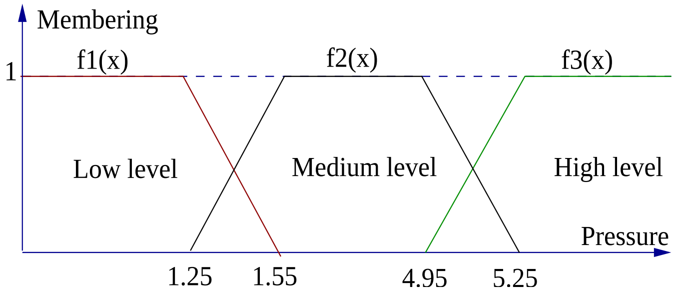

Figure 7. Fuzzy functions used to find the state that corresponds to different pressure values

The table I 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 in the table.

Table I.
Classification made in order to find operation region

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

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

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

And so forth, for the other rules.

Rules that determine the occurrence of events. They have the form:

If operational 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 II shows some of these rules.

Table II.
Rules for detecting events in normal operation mode

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

Fault types. Since the hydro-pneumatic system is composed of mechanical, electrical and electronic devices, 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 sensors of water level or pressure (the measurements are disturbed or are irregular)

Global behavior description of the fault-exposed system – Global Qualitative Model. There is a normal condition for the hydro-pneumatic system where all operational regions mentioned before are feasible. When a fault happens, it is not valid to consider the operational regions shown in fig 5, since the system condition is changed. The global dynamics of this fault-exposed system is shown in fig 8, 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 occurs an event F (fault) the system changes its state to "Fault", and starting from there two events may occur: 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. One example of starting consists 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.

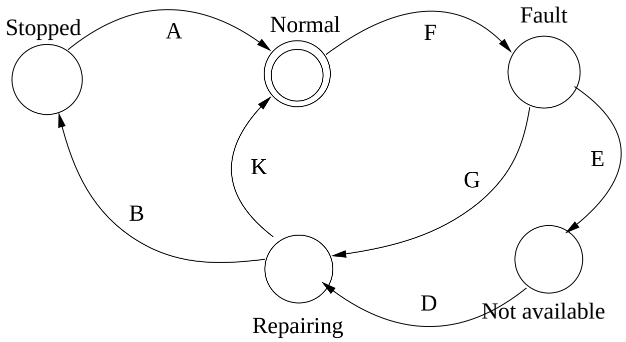

Figure 8. Global behavior automaton of the hydro-pneumatic system

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.

4.3 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 in terms of forcing events after the fault condition; in other words, whenever event F occurs it is necessary to force the occurrence of event types G and K, which are controllable. On the other hand, if the sequence of events F-E occurs, then the sequence D-K is allowed and must be generated, and so on for other sequences. Since the hydro-pneumatic system must operate without halting, its designer has to consider some repairing actions to be carried out while the system is running.

4.1 Changes in the settings of production system (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 example.

If the variable has not been manipulated and operation region is V, then do not to manipulate.

If the variable has not been manipulated and region is other, then do manipulate according to following table:

Table III summarizes this reasoning model.

Table III.
Reasoning Model for the hydro-pneumatic system.

Z	B=0	C=0	B=1	C=1
1	-	-	-	C→0
2	-	-	-	C→0
3	-	-	B→0	C→0
4	-	-	-	-
5	-	-	-	-
6	-	-	-	-
7	B→1	-	-	-
8	-	-	-	-
9	-	C→1	-	-

Where the symbol "–" indicates none modification in system configuration, "→ 0" suggests that the device must be switched off, and "→ 1" suggests that the device must be switched on.

Once this reasoning model is obtained, we derive consignments to the controllers, 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 Validating the Design

The validating schema is depicted in figure 9. A DEVS-based formalism simulator [10] that implements discrete-event systems represents the hydro-pneumatic system elements. It runs on a Galatea platform [11], developed in the Java programming language. The multi-agent environment JADE [12] was chosen for implementing the Detector and Supervisor agents. Both agents will map their actions over the DEVS simulator by means of an interface developed in Java.

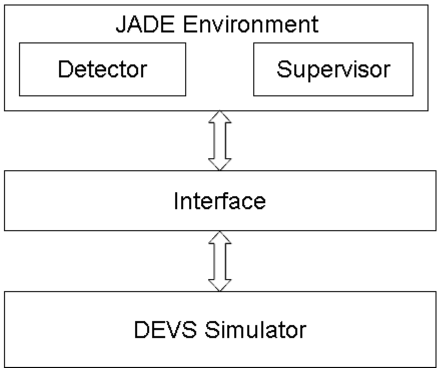

Figure 9. Validation scheme.

Once the test bed with the agents, DEVS Simulator and interface were implemented, the computational 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)
- Variable 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

6 Results Obtained

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. Figure 10 illustrates the water average level. 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.

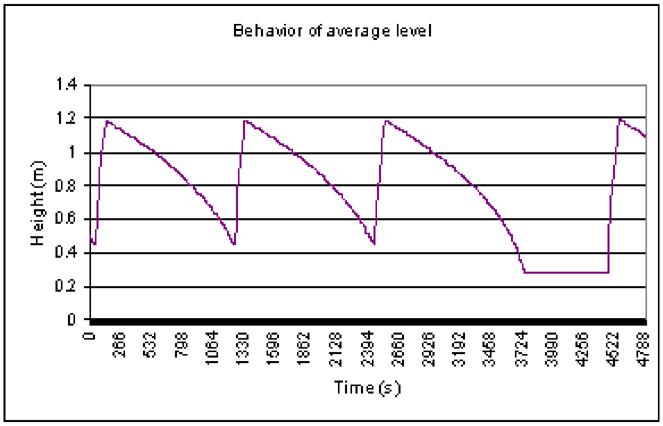

Figure 10. Average water level evolution in the tank

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. Fig 11 illustrates this trace. 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.

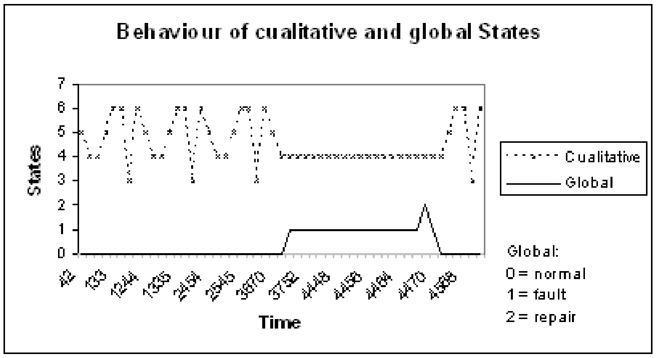

Figure 11. Trace showing the behavior of operations regions and the global state of the system.

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:

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

Where Qi 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 approaches as well as their validation.

- The combination of the methodology of multiresolutional modeling with fuzzy logic techniques and supervisory control to obtain a strong mechanism for 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 can be implemented with agent technology, which enforce the use of artificial intelligence with 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] 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

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

[6] 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.

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

[8] Jennings, N, Cooperation in Industrial Multi-agent system., World Scientific Publishing, Singapore, 1994

[9] 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.

[10] Zeigler B, Praehofer H, Kim T. Theory of modeling and simulation. Academic Press, Singapore, 2000

[11] Uzcátegui, M. GALATEA, a multi-agent simulation language. Msc thesis. Universidad de Los Andes, 2002.

[12] JADE. url: http://www.tilab.com