INTELLIGENT SUPERVISION OF PETROLEUM PROCESSES BASED ON MULTI-AGENT SYSTEMS

Miguel Angel Ramírez Canelón
Jacinto Dávila
Eliezer Colina Morles
Universidad de los Andes: Facultad de Ingeniería
Complejo La Hechicera, Mérida, Venezuela
E-mails: Ramirezmb@pdvsa.com, jacinto@ula.ve , ecolina@ula.ve

Abstract: This work presents a methodology for the design of an intelligent supervisory system that combines the principles of fuzzy logic, the Internal Model Control (IMC) architecture and the paradigm of Multi-Agent Systems (MAS). The methodology has been conceived to be applied in an intelligent supervisory system, specifically for two kinds of complex petroleum industrial processes: the gas-oil separation process and the oil-heating process. The supervision proposal takes into account the fact of using standard local supervisors schemes connected between themselves and to a global supervisor so that local objectives in each process can be met, thereby letting the global or social objective be obtained through the application of basic mechanism of communication, cooperation and coordination; where these objectives have been previously defined and structured in a hierarchical manner. The paper includes some computational simulations performed under MATLAB / SIMULINK and the results obtained show a good overall system performance.

Keywords: Multi-Agent System, event detectors, intelligent supervisor system, internal model control.

1 Introduction

Intelligent supervision of industrial processes is an area of intense study and many approaches have been used based on artificial intelligence techniques; where most of the resultant methods are based on hybrid systems. Supervisory control is typically based on a global assessment of the current operation point [1], that leads to system parameters characterization with the intention of providing diagnosis and anticipating possible changes that affect its current state of operation through the adjustment of scalar parameters of the controller or changes in the set point.

Among the broad variety of intelligent supervisory system schemes proposed in many publications, [2][3][4][29], there exist three basic elements: the event detector, the model of the process and the decision system. The event detector may be implemented in many different ways; in particular, some common approaches are based on fuzzy logic and neural network; which are used to detect possible operation states in a particular process. There are many alternatives for the determination of the model of the process such as qualitative models, broadly recommended in supervisory scheme and non linear models based on local models such as the Takagi-Sugeno fuzzy model [6]. In addition, multiresolucional modeling is another alternative that integrates a mathematical model, a knowledge based model and a qualitative model [1]. Finally, the decision system guided by a set of objectives and priorities of the process makes some decisions that result in changes of the reference point of the process or changes in the scalar parameters of the controller. It is important to highlight that in a general perspective, the supervision process implies the following connotations: i) detection of the plant´s parameters, deviations and instability, ii) guidance of controller tuning, adaptation and synthesis; iii) identification and diagnosis of process faults [5].

This paper attempts to define an intelligent supervisory scheme to be applied in two petroleum processes connected in cascade: the gas-oil separation process and the oil-heating process. These particular processes have been studied from the point of view of modeling and control in [6][7][8]. The proposal consists of using standard local supervisory schemes interconnected between themselves and to a global supervisor, thus communication and cooperation mechanisms are needed. This is, therefore, a clear case for the use of a supervisory scheme based on multi-agent systems. The rest of the paper is organized as follows. In section 2, the fundamentals of Multi-agent systems are described. Section 3 presents the supervisory scheme formal proposal based on Multi-agent systems. Section 4 describes the case of study showing the results of the computational implementations in MATLAB/SIMULINK. Conclusions and future work are given in section 5.

2 Fundamentals of Multi-Agent Systems.

In order to design an intelligent supervisory control system under the conception of multi-agent system, it is necessary to define the following aspects:

- The definition of a distributed platform, normally determined by the system to be supervised.
- The type of intelligence and knowledge presentation strategy that will be used.
- The development of a communication, coordination and cooperation protocol between the different intelligent modules.

Multi-agents systems technology addresses some of the before mentioned difficulties. In particular, the process control area is a natural one for the application of intelligent agents because controllers may be regarded by themselves as autonomous reactive systems [9].

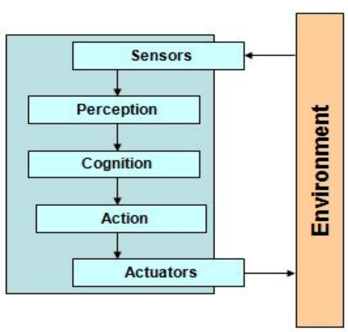

Fig.1 Generic scheme of an agent according to Russel's definition.

A significant number of definitions of agents can be found in the literature and one of the simplest and most accepted is the one presented by Russel [10] that considers an agent as an entity that acts and perceives in a particular environment. Nevertheless, a formal definition of agent was established in [11], where an agent is considered as a computational system situated in an environment and is capable of an autonomous and flexible actions in that environment, in order to meet its design objectives.
Communication, cooperation and coordination are three key issues in multi-agent systems. Coordination is the process by which an agent reasons about its local actions and the anticipated actions of others to try to ensure the community acts in a coherent manner. As Weiss explains in [27]: "The purpose of coordination is to achieve or avoid states of affairs that are considered as desirable or undesirable by one or several agents. To coordinate their goals and tasks, agents have to explicitly take dependencies among their activities into consideration. Two basic, contrasting patterns of coordination are cooperation and competition. In the case of cooperation, several agents work together and draw on the broad collection of their knowledge and capabilities to achieve a common goal. Against that, in the case of competition, several agents work against each other because their goals are conflicting." We add that agents also perform a third type of coordinating behavior which is neither cooperation, nor competition. It is collaboration: agents assist others even though they do not have common goals. This is possible by means of negotiation. Coordination by means of cooperation or negotiation seems to require some form of communication.

In agent based systems, communication can occur at different levels of sophistication ranging from primitive to high-level. Software agents can be developed on different platforms and using different programming languages. They usually interact by exchanging complex symbolic information and possibly have to agree on complex interaction protocols. Several proposals have been designed to develop agent-communication languages such as the KQML (Knowledge Query Manipulation Language) [13] and FIPA-ACL (Foundation for Intelligent Physical Agents - Agent Communication Language) [14].

On the other hand, it is imperative, that agents use a common vocabulary to communicate where the symbols used in messages have the same, predefined meaning for all of the agents. This fact reflects the importance that agents share certain ontology. A computational ontology is an informational model that describes concepts and relations in some specific domain [15].

The capacity of communication between agents can be interpreted as an additional element of perception (receiving messages), of cognition (interpretation and actions to be executed) and actions (sending of messages). These aspects allow agents to conform societies where global objectives may exist apart from the individual objectives of each agent [16]. In this context, cooperation becomes relevant for those societies. A cooperative situation is validated if either adding a new agent could result in an increase in the performance level of a determined group or agent actions serve to avoid actual conflicts [11].

Finally, the area of agents-architecture considers the aspects related to the design of computational systems that satisfies the features of the agents theories [17][18]. It specifies how an agent can be decomposed into the construction of a set of component modules. There exist three kinds of architectures: i) deliberative architectures (cognitive agents) where the agent is designed according to the paradigm of the symbolic artificial intelligence (for instance GRATE and IRMA [19][20]); ii) reactive architectures where the agents do not include any kind of central symbolic world model and do not use complex symbolic reasoning (A classical example is the so called subsumption architecture of Rodney Brooks [21][22]) ; iii) hybrid architecture where the agent is the combination of two subsystem. One subsystem is the deliberative, which contains a symbolic model of the world while the other is the reactive. Classical examples are the Touring Machines, COSY and INTERRAP [23][24][25]. Another example of hybrid architecture is the multi-agent simulation platform GALATEA which provides for the representation of agents that can behave reactively and proactively, according to the circumstances of (simulated) execution [26].

3 Supervisory Proposal Based On Multi-Agent System.

The supervisory proposal based on multi-agent system is shown in figure 2. Local supervisors are grouped depending on the class of process and are connected each other to the global supervisor.

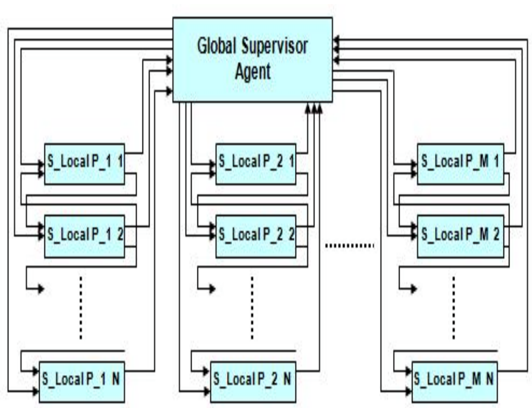

Fig.2 Supervisory Scheme based on Multi-Agent System for Industrial Processes.

It can be observed from figure 2, that there are *M* classes of processes identified as P_1, P_2, …, P_M where *M* is the number of processes classes and N is the amount of processes for each *M* class, so there will be N supervisor local agent for each class of process. For instance, supervisor local identified by S_Local P_1 1 y S_Local P_1 2 belong to the same class of process 1 and are connected each other and to the global supervisor agent. Every process is governed by its respective local supervisor and at the same time is connected to the global supervisor. Figure 3 shows the particular diagram of the processes that will become the case of study.

As it can be observed from figure 3, the output of oil of the two production separator is connected to both fired-heaters. However, depending on the physical dimensions and the specifications of functioning of a fire-heater, the output of one production separator could be connected only to one fire-heater.

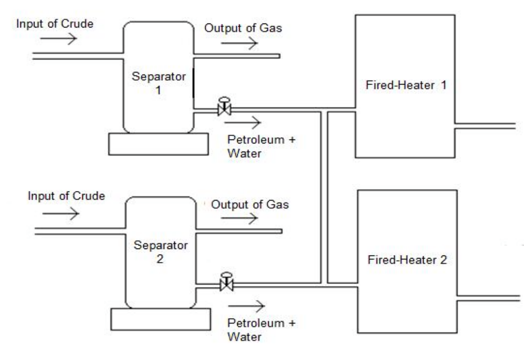

Fig.3 Schematic diagram of the separation process and oil heating process of oil.

3.1 Design of the local supervisor agent.

Taking into account the considerations of the architecture ICM (Internal model control) established in [7] [8] and based on the criteria in [6] for the determination of fuzzy models, the schematic depicted in figure 4 is proposed as the local supervisor agent. It may be observed from figure 4 that the regulatory level is composed by the internal model control architecture, which has the following sub-elements: i) the singleton fuzzy model of the plant determined from input-output data; ii) the inverse of the fuzzy model which is used as the controller; iii) The filter F that acts as an integrator and adds robustness to the feedback control system against the presence of disturbances; iv) The plant to be controlled which, for simulation purposes, consists of a linearized mathematical first order model around an operating point.

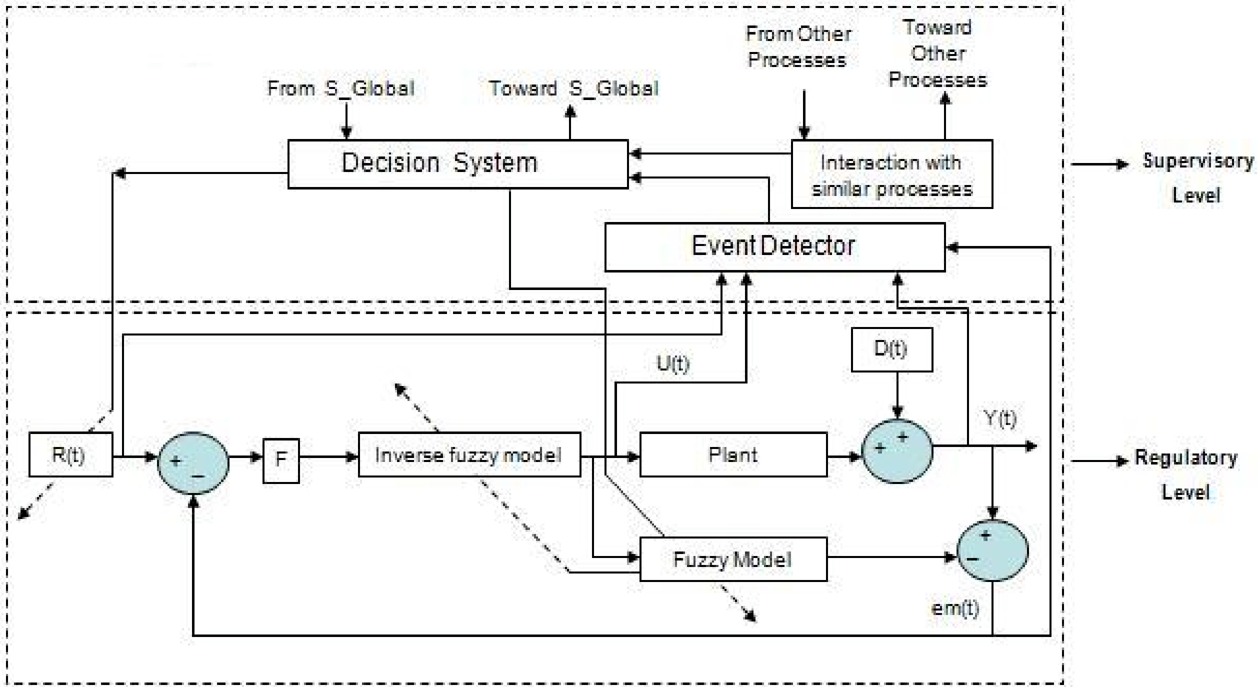

Fig. 4 Supervision local scheme for each process.

The internal model control structure (IMC) is introduced as an alternative to the feedback structure. Its main advantage is that closed-loop stability is assured simply by choosing a stable IMC controller [28]. Also, closed-loop performance characteristics, for instance settling time, are related directed to the controller parameters, which make on-line tuning of the IMC controller very convenient.

In the supervisory level, the first element is the event detector. Once this block detects some change in the state of the process, the information is sent to the decision system.

The decision system converts the information received by the event detector in an index performance and based on the states of other similar processes and a generic knowledge base; it can either adjust the consequent of the fuzzy model (thereby adjusting the controller) or change the reference of the process. Furthermore, if an unacceptable change in the operation point occurred then the system decision would declare a fail in the process.

Finally, the element called "Interaction with other processes" notifies the state of the local process to other similar processes and at the same time receives the states of similar processes. The decision system, based on the state of the local process and the states of the remote similar processes must decide if a local action is taken or if the action has cooperative characteristic.

3.1.1 Design of the event detector

The event detector is based on the linguistic Mamdani model for MISO system (Multiple Input Single Output). A general rule of this kind of model is given by:

$$R^{(k)} : If \;\; x_1 \text{ is } A_1^k \text{ and...and } x_n \text{ is } A_n^k \text{ Then } \mathrm{y} \text{ is } G^k \quad (1)$$

Where $R^{(k)}$ denotes the k-th rule, Ai^k y G^k are fuzzy sets en $U_i \subset \Re$ y $V \subset \Re$, respectively, "y" is the output; meanwhile $x = (x_1,\ldots x_n)^T$ is the vector of input variables. The event detector has two inputs: the modeling error and the derivative of the output (dy(t)/dt). The combination of these inputs is considered as the more adequate combination of inputs for characterizing the events of the process both for the oil separation process and the heating oil process. Indeed, it will be taken a sample of values of the modeling error and the derivative of the output y and the calculation of the average of the samples is done in each time window r_1, r_2,…, r_n. Figure 5 illustrates this idea. Therefore, the input variables are the average modeling error (em_p) and the average output derivative for each time window.

The possible events according to the variation of the input vector are the following: normal condition, little increment of the operation point, significant increment of the operation point, decrement of the operation point, significant decrement of the operation point, fail↑ (this can be caused by a strong increment of the operation point) and fail↓ (this can be caused by a null operation point).

These events can be conceived as operations regions of a determined process and each event is represented as a fuzzy set, so the transition of one event to another is done of a gradual manner. However, it is important to know that the fail state can be caused by malfunctioning of a sensor or an internal fault in the valve control.

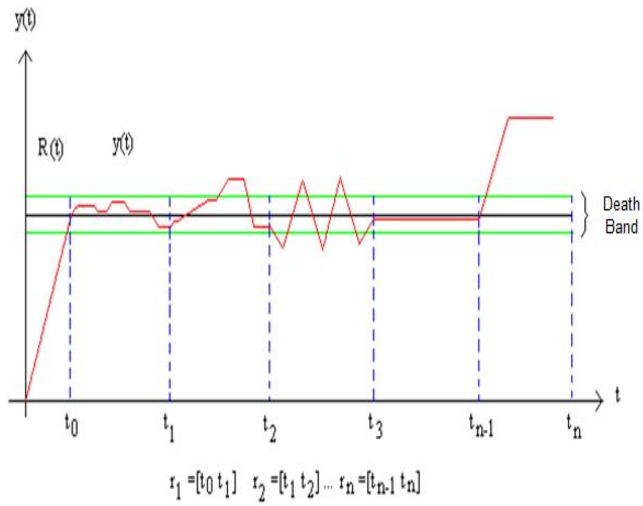

Fig. 5 Time Windows of the Event Detector

Three fuzzy sets were defined for each input variable and output variable. Figure 6 shows the fuzzy sets. The resultant base rule (with its respective weight) is the following:

R1 (w_1 = 1): If dy_p is *Null* and em_p is *Null* Then the process is in *normal condition*.
R2 (w_2 = 0.5): If dy_p is *Null* and em_p is *High* Then the process *has a little increment in the operation point*.
R3 (w_3 = 0.5): If dy_p is *Null* and em_p is *low* Then the process *has a little increment in the operation point*.
R4 (w_4 = 1): If dy_p is *High* and em_p is *Null* Then the process *has a little increment in the operation point*.
R5 (w_5 = 1): If dy_p is *High* and em_p is *High* Then the process *has a significant increment in the operation point*.
R6 (w_6 = 1): If dy_p is *High* and em_p is *Low* Then the process *is in fail* ↑.
R7 (w_7 = 1): If dy_p is *Low* and em_p is *Null* Then the process *has a significant decrement of the operation point*.
R8 (w_8 = 1): If dy_p is *Low* and em_p is *Low* Then the process *has a little decrement of the operation point*.
R9 (w_9 = 1): If dy_p is *Low* and em_p is *High* Then the process *is in fail* ↓.

3.1.2 Design of the decision system

The decision system agent is capable of taking appropriate decision based on the output of the event detector and the operation state of other similar processes. The following expert system with its respective generic rules is proposed:

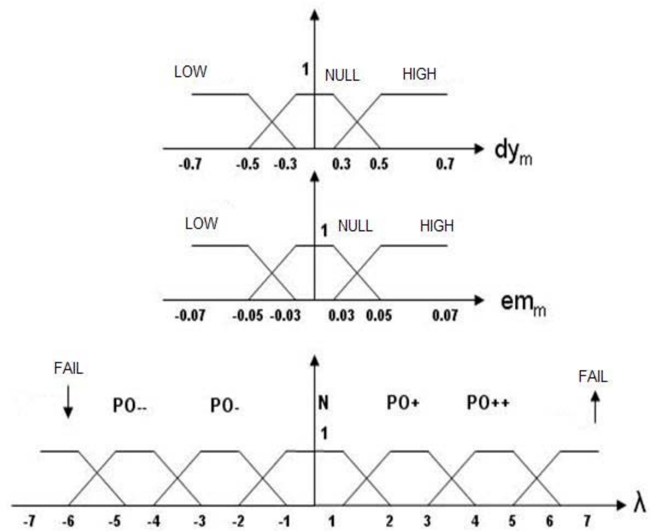

Fig. 6 Fuzzy sets of the input and output variables.

R1: If the local process is in *normal condition* and some remote process has a change in its operating point Then update the status of availability for cooperating and inform to the global supervisor agent.
R2: If the local process has some change in its operation point and at least there is at least one similar remote process in *normal condition* Then take its own corrective action, ask for cooperation and inform to the global supervisor agent.
R3: If the local process is in *normal condition* and all the remote processes are in normal condition Then inform to the global supervisor agent.
R4: If the local process is in fail Then inform the global supervisor agent.

It is important to highlight that in R1, the definition of the cooperation actions will be established by the global supervisor. Essentially, the cooperation deals with the change in the setpoint of a similar process if it is considered in normal condition. For instance, if the separation process 1 is in normal condition and the separation process 2 has some change in its operating point or has some perturbation, the local supervisor agent of the process separation 1 can change its reference in order to try to guarantee that the total flow of oil of the two separators does not have a significant variation. Nevertheless, if for instance, the two separators have some problem then it is no possible to change the reference for this kind of process. In this case, a change in the setpoint in one of the fire-heaters should be done.

With respect to rule 2, R2, the following actions were defined: adjusts the fuzzy model, thereby adjusting the fuzzy controller (the purpose is to decrease the steady state error), changes the set point and declares a process fault. A fault can be caused by drastic increment of the operation point that overcomes the limits of the functioning of the production separator. The reason for changing the set point is to guarantee the fulfillment of a global objective (A cooperation action).

3.2 Design of the global supervisor agent

The function of the global supervisor agent is to establish some mechanisms in order to guarantee the achievement of the global objective of the whole process. For complex industrial processes, we have established that the global objective should be the following: To obtain a better quality product with measured specifications inside a definite range. In this particular case, the temperature of the oil at the output of a fired-heater should have a value between 180 and 190 degree Fahrenheit.

According to the construction specifications of the two separators, the total flow of oil at the output of the two separators should have a value between 45 Barrels/Hour and 51 Barrels/Hour of oil in order to guarantee the optimum range of temperature mentioned.

We have used the principle of energy conservation (Whenever it is permitted), for changing the setpoint and guaranteeing all processes outputs of a specific class do not overcome the maximum limit of the input variable of the next process. For instance, in our particular case of study, we have to guarantee that the flow of oil at the output of the production separator do not exceed the maximum limit of flow of oil permitted at the input of the fired-heater.

In order to achieve this, we have defined the algorithm depicted in figure 7, where Δpot is the total variation of the operation point of all the N processes associated to a specific class of process, j is the amount of processes whose condition is normal, ΔS is the sum of all the changes in the setpoint of each specific process and ΔSp1, ΔSp2, …, ΔSpj are the individual changes of each setpoint.

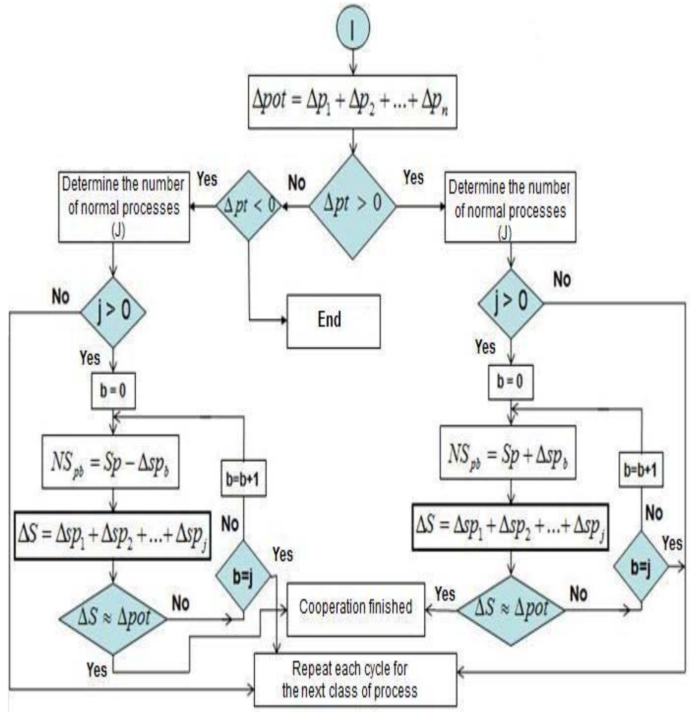

Fig. 7: Algorithm for the global supervisor agent.

4 Case of Study: implementation of the supervision proposal based in SMA.

The implementation of the supervisory proposal was developed in MATLAB / SIMULINK using the S-Functions blocks either in C or m language. These blocks permit the incorporation of customized algorithms inside SIMULINK environment leading the creation of simulation schemes of high complexity. In addition, a user graphical interface was developed using GUIDE in MATLAB.

In order to test the intelligent supervisory scheme two scenarios were established. The first scenario consisted in introducing a drastic change from 24 barrels/hour to 29 barrels/hour in t = 180 seconds as it can be observed in figure 8a. The perturbation lasted 20 seconds.

Another sudden change was introduced in the process separation 2 in t = 40 s (a change in the flow input of oil from 24 barrels/hour to 32 barrels/hour) and the time of the perturbation lasted 20 seconds. Figure 9a shows the output y2(t) of the second separator. Figure 8b and figure 8c shows the behavior of the valve position and the error modelling in the process separation 1.

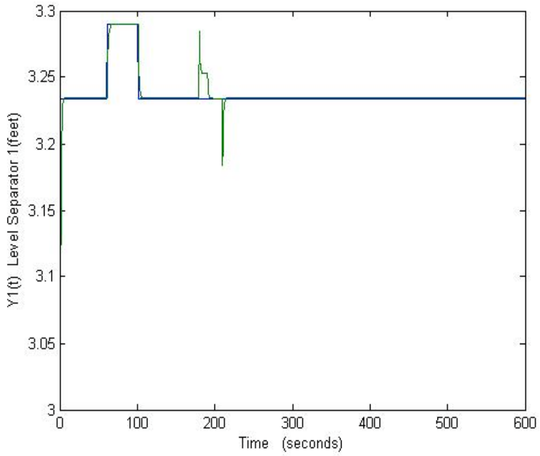

Fig. 8a Output of separator 1

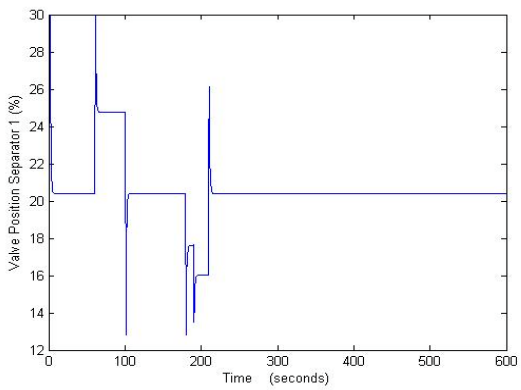

Fig. 8b Valve position of separator 1

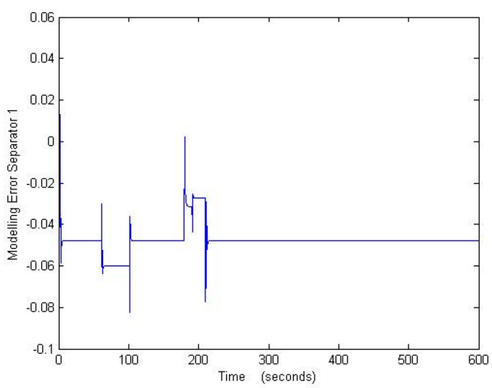

Fig. 8c Error modelling of the first separator

In figure 8a, the output y1(t) of separator 1 has a change in its reference in t = 45 seconds, five seconds after the perturbation occurred in the process separation 2. This change in the reference of separator 1 means a cooperation action as it compensates the increasing of total flow of oil of the two separators because separator 1 is reducing its output flow of oil while separator 2 is increasing its output flow of oil because of the presence of the perturbation. A similar behavior is presented in t = 185 seconds in the output of process separation 2. In this case, as it may be observed in figure 9a, the local supervisor of the process separation 2 does a cooperation action because of its change in its reference in t = 185 seconds.

Figure 9b and figure 9c shows the behavior of the valve position and error modelling of separator 2. As it may be observed in these figures, both behaviors are very similar to the one presented in separator 1 but in different ranges of time.

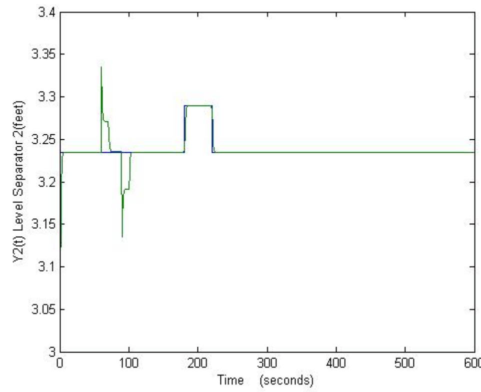

Fig 9a Output of separator 2

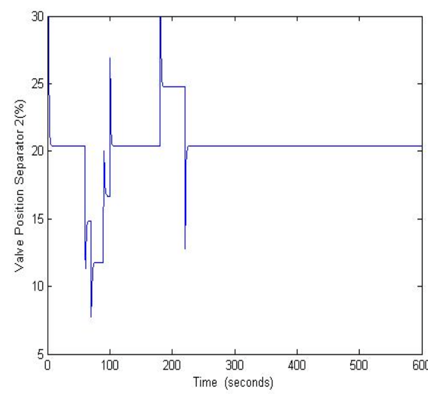

Fig. 9b Valve position of separator 2

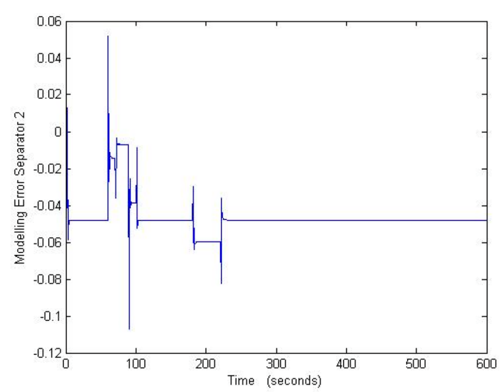

Fig 9c Error modelling of separator 2

In the second scenario, various changes of the current operation point were simulated in both separators in the range of time defined between t = 100 s and t =250 s. Figure 10 shows three changes introduced to the process separation 1. A similar behavior is shown in figure 11 for the process separation 2.

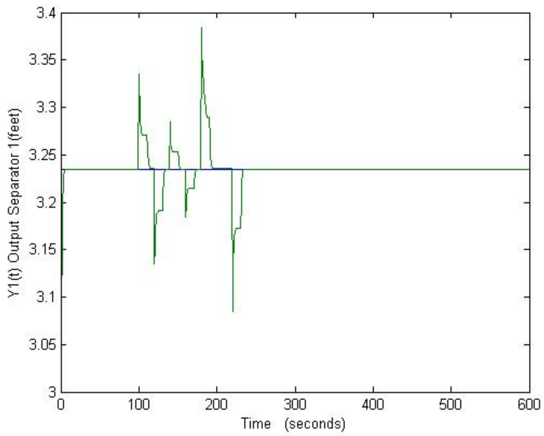

Fig 10 various changes in the current operation point of separator 1

In this particular case, the global supervisor sends a message to both local supervisor agents of the heating processes so that they can change their respective reference.

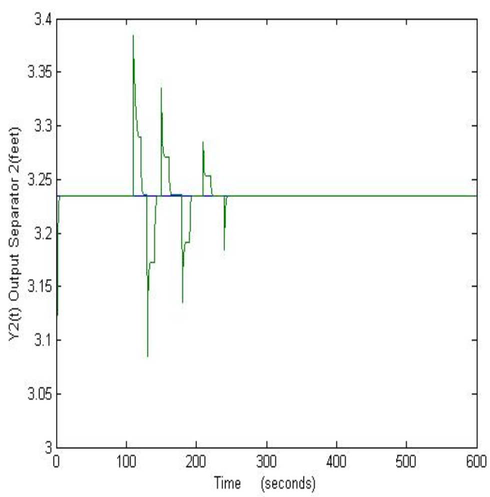

Fig 11 various changes in the current operation point of separator 2

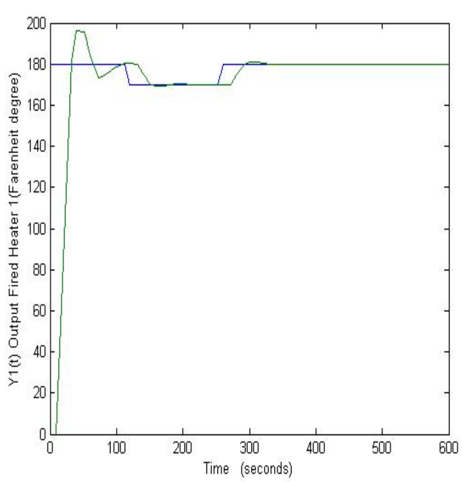

Fig 12 Output temperature of fired-heater 1

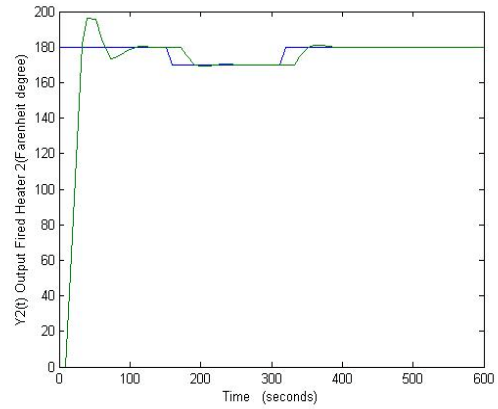

Fig 13 Output temperature of fired-heater 2

As a result, both references in the fired-heaters were changed while changing in the operation point happened in the separation processes. Figure 12 and figure 13 shows the behavoiur of the temperature of oil at the output of each fired-heater.

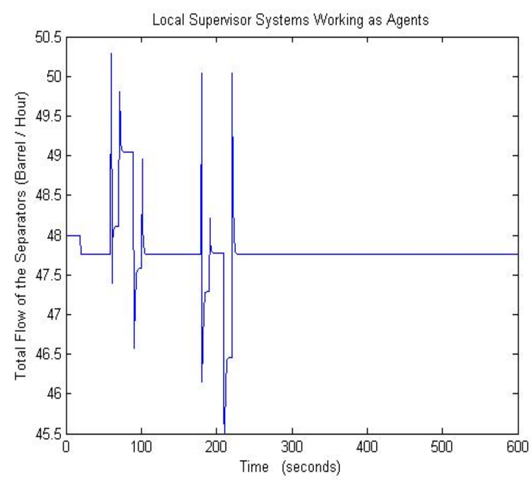

Fig 14 Total Flow of Oil at the Output of the two Separators (Proposed Methodology)

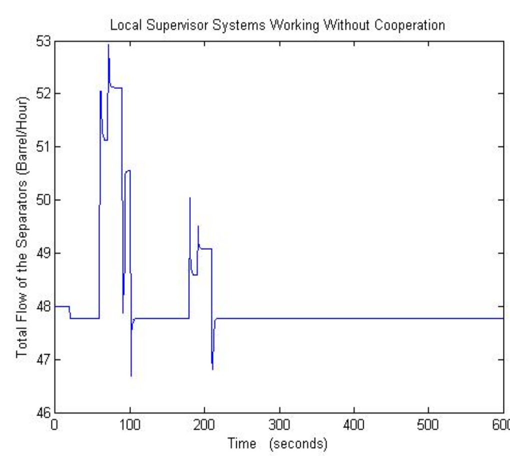

Fig 15 Total Flow of Oil at the Output of the two Separators (Local Supervisor Systems working without connection)

The importance of our proposal can be demonstrated if we analyze the behavior of total flow of oil at the output of the two separators using the methodology described before. As it may be observed, the range of variation of the total flow of oil is between 46 barrels/hour to 50 barrels/hour. On the other hand, figure 15 shows the total flow of oil at the output of the two separators with the local supervisors working without any connection. In this case, the range of variation of the total flow is between 47 barrels/hour to 52.5 barrels/hour of oil. Although the difference seems insignificant, however, when a drastic change in the current operation point lasts a considerable period of time, then the range of variation of the total output may affect considerably the next processes if there is no any cooperation between the local supervisors systems.

5 Conclusions and Future Works

In this paper, a supervisory scheme based on multi-agent systems was proposed as a methodology for the intelligent supervision of industrial processes with multiple feedback controls. It is important to highlight that this methodology can be applied to other sort of processes connected in cascade.

The use of the ICM structure in the regulatory level has been an interesting alternative for supervision tasks. The error modelling is a key information for detecting some deviation in the model of the process.

The computational simulations were done in MATLAB SIMULINK using S-functions for the emulation of the agents: local supervisor agent and global supervisor agent. This is a significant contribution of this paper because there exist many platforms for developing agents and few efforts have been done in Matlab software. The results of the simulations demonstrate that the cooperation between local supervisor agents may permit that the global objective of the whole process can be reached.

Future works contemplate the inclusion inside the local supervisor scheme of a module that can predict a change of event, so the mechanism of cooperation can be enhanced notably.

References:

[1] Lunze, J. “Process supervision by means of qualitative models”. Technische Universitat Hamburg-Harburg, Germany, December 2000.

[2]Altamiranda E., Colina E., y Chacon E. “Intelligent supervisory systems for industrial process control”. Instituto de investigación y desarrollo de tecnologías aplicadas IIDTA, Universidad de Pamplona, Colombia, pp. 1-4, octubre 2002.

[3] Ortega C., Colina E., y Chacon E. “Intelligent supervisory control designs framework for fault exposed processes”. WSEAS, International Journal of Circuits, Systems and Signal Processing, Cambridge University, USA, pp. 1-4, April, 2007.

[4] Rodriguez J., Romagnoli J., and Goodwin G. “Supervisory multiple regime control”. Centre for process systems engineering department of chemical engineering, Sydney University, Australia, March, 2003.

[5] Quek C. “Fuzzy Integrated Process Supervision”. Intelligent systems laboratory, School of applied science, Nanyang Technological University, Singapore, April 1996.

[6] Ramírez, M. and Colina E. “Fuzzy clustering based models applied to petroleum processes”. WSEAS Transactions on Systems and Control, Issue 3, Volume 3, March 2008.

[7] Ramírez, M. and Colina E. “Fuzzy model based control: Application to an oil production separator”, IEEE on Hybrid Intelligent System, Barcelona, Spain, September 2008.

[8] Ramírez, M. y Colina E. “Control basado en modelos Difusos aplicados a procesos industriales complejos”. XIII Congreso Latinoamericano de Control Automático, Mérida, Venezuela, Noviembre 2008.

[9] Jennings, N. “Using GRATE to build cooperating agents for industrial control”. Department Electronic Engineering, Queen Mary and Westfield College, England, February 1999.

[10]Russell, S. and Norvig, P. “Artificial Intelligence: a modern approach”. Prentice Hall International Editions, 1995.

[11] Woolridge, M. and Jennings, N. “Intelligent agents: Theory and practice”. The knowledge engineering review, Issue 10, Volume 2: pp. 115-152, 1995.

[12] Fenster, M., Kraus, S. and Rosenschein, J.S. “Coordination without communication: Experimental Validation of Focal Point Techniques”. In proceedings of the first international conference on multi-agents systems, San Francisco, CA, AAAI Pres/The MIT Press, pp. 102-108, 1995.

[13]Finin, T., Fritzson, R., McKay, D., and McEntire, R. “KQML as an agent communication language”. In the third proceedings On Information and Knowledge Management, 1994.

[14]Agent ACL message structure specification. Technical Report XC00061E, http://www.fipa.org.

[15]FIPA SL content language specification. Technical Report XC0008G, http://www.fipa.org.

[16]Doran, J., Franklin, F., Jennings, N., and Norman, T. “On cooperation in Multiagent Systems”. Knowledge engineering review, 12(3): 309-314, 1998.

[17]Iglesias, C. “Desarrollo de una metodología para el desarrollo de sistemas multiagentes”. Tesis Doctoral, Departamento de Ingeniería de Sistemas Telemáticos, Universidad Politécnica de Madrid, Enero 1998.

[18]Maes, P. “The agent network architecture (ANA)”. SIGARD Bulletin, pp. 115-120, April 1991.

[19]Bratman, M., Israel, D., and Pollack, M.. “Toward an architecture for resource-bounded agents”. Technical report CSLI-87-104, Center for the study of language and informations, SRI and Stanford University, 1997.

[20]Jennings, N. “Cooperation in industrial Multi-Agent Systems”, Department of Electronic Engineering Queen Mary and Westfield College, University of London, 1994.

[21]Brooks, R. “A robust layered control system for a mobile robot”. MIT AI Lab Memo No. 864, Massachusetts, 1985.

[22]Brooks, R. “Intelligence without reason”. Proceedings of 12th Int. Joint Conf. on Artificial Intelligence”. Sydney, Australia, pp. 569-595, August, 1991.

[23]Ferguson, I. “TouringMachines: An architecture for dynamic, Rational, Mobile Agents”. PhD thesis, Clare Hall, University of Cambridge, 1992.

[24]Burmeister, B. and Sundermeyer, K. “Cooperative problem solving guided by intentions and perceptions” Decentralized AI. Elsevier Science Publisher B.V., 1992.

[25] Muller, J. “A conceptual model for agent interaction” In proceedings of the second International Working Conference on Cooperating Knowledge Base System, University of Keel Deen, pp 213-234, 1994.

[26] Davila, J., Gomez, E., Laffaille, K. and Uzcategui, M. “Multi-Agent Distributed Simulations with GALATEA”. DS-RT Proceedings. Ninth IEEE Symposium on Distributed Simulation and Real Time Application, pp. 165-170, Montreal, Canada, 2005.

[27] Weiss, G. “Multiagent Systems. A Modern Approach to Distributed Artificial Intelligence”. MIT Press, 1999.

[28]Morari, M., Zafiriou, E. “Robust Process Control”. Englewood, Cliffs, New Jersey, Prentice Hall, 1989.

[29]Altamiranda E., Colina E., Chacon E. “Intelligent Supervisory Systems For Industrial Process Control”. WSEAS Transactions On System, Issue 7, Volume 4, July 2005.