LEAK DETECTION IN GAS PIPELINES - A KNOWLEDGE BASED APPROACH

Rodrigo Rizzi Starr
Laboratory of Hydraulic and Pneumatic Systems, UFSC – Federal University of Santa Catarina
rrs@emc.ufsc.br

Jonny Carlos da Silva
Laboratory of Hydraulic and Pneumatic Systems, UFSC – Federal University of Santa Catarina
jonny@emc.ufsc.br

Abstract.
Leak detection systems are a fundamental part of the management system associated with any long distance pipeline. Such systems are necessary to avoid or reduce environmental and economical losses in case a leak develops in the pipeline.
There are several solutions, with different efficiency and cost-effectiveness, for this problem. One of the most appealing is the use of software based leak detection systems (SBLDS). These systems use already available data from the Supervisory Control and Data Acquisition System to detect and locate leakages in real-time. This solution is appealing due to its cost-effectiveness and real-time operation. Nevertheless, this technique relies heavily on the sensing equipment already installed on the pipeline. Noise, drift, low sampling rates, low sensor resolution, and other aspects have a negative influence on system performance.
Most of the schemata devised for SBLDS uses a lot of statistical analysis and filtering to cope with these problems. But a lot of useful information is not taken into account in the development of these systems. This information is available as operational knowledge, pipe reliability and risk analysis information. Such background knowledge can be used to improve a SBLDS performance. This article presents a software based leak detection method and proposes an implementation. The method is an extension of dynamic model based leak detection with operational and risk analysis knowledge, using expert system techniques.

Keywords: *Expert systems, leak detection systems, dynamic simulation, multi-agent systems*

1. Introduction

Gas pipelines are an important mean of transporting natural gas over long distances, being both economical and efficient. As natural gas is expensive to store, pipelines must run continuously, and an interruption of service can have costly consequences. For example, the sudden stop of a power plant during a period of high demand may overload the electrical distribution grid, and even cause a blackout. Mainly due to this reason, gas transportation companies are bound by contract to deliver gas at specified conditions of pressure, temperature and composition, subject to heavy fines if these specifications are not met.

Therefore, continuous operation is strictly required from the gas pipeline management company. It is important that any critical failure be detected early, and corrective actions be applied accordingly.

Failure detection involves monitoring and analysing hundreds of inter-related parameters that reflect the pipeline state. Modern pipelines are usually operated from a central control room, with few pipeline operators responsible for controlling the whole pipeline dynamics such that the specified delivery conditions are met. Also, as these controllers are usually the only ones who are checking the sensors measurements all the time, they are responsible for detecting failures, and warning the maintenance team, which has to act timely to solve the problem.

All these responsibilities put a lot of strain on the operators and maintenance team. To support pipeline operation and failure identification, a combination of expert systems and simulation software has usually been employed (Sampath and Yee, 2000; Wallooppillai *et al.*, 2000). This combination is fruitful because the expert system can simulate (to some extent) the cognitive process of the pipeline operator, analysing a lot of inputs, and producing output that is both more informative and more accurate for the operator. The simulation software makes it possible for the system to evaluate future states, and validate the current state of the pipeline as informed by the sensors. This combination provides better ability to find failures and misreadings.

One such a system has been developed in a partnership between the Laboratory of Hydraulic and Pneumatic Systems (LASHIP) at UFSC – Federal University of Santa Catarina and two companies, PETROBRAS and TBG (the Bolívia-Brasil gas pipeline transportation company). This system was first aimed as a supporting tool to identify failures and potentially dangerous operating conditions at delivery stations. This system is called SEGRED (which is a portuguese acronym for Expert System for Management of Natural Gas Pipelines).

As the work progressed, the companies showed interest in expanding the system to the pipeline operation as a whole.

So, two systems are now being developed: SEGRED Stations, with a knowledge base and interface that focuses the delivery stations, and is aimed at the maintenance team, and SEGRED Network (Silva *et al.*, 2004), which focuses on the pipeline as a whole (without getting deep in the details of delivery stations, for example) and is aimed at the pipeline operators.

These systems are being developed using an incremental process (Giarratano and Riley, 1998; Sommerville, 1995). Although this process has some drawbacks, as stated by Ferber (1999, p. 134), this is the preferred process for expert systems development, because it allows the tackling of a huge problem of knowledge representation in manageable steps. The incremental process is based on the implicit modularity of rule-based systems and object-oriented modeling.

As the work progresses, TBG requests new features to be added to the system. This work is about one of these features, a leak detection system. A leak is the main functional failure that can happen to the pipe itself, and its fast detection is of extreme importance. Besides commodity loss, a leak imposes risks both to the integrity of the pipeline itself and to any nearby community.

According to experts from TBG, the problem of leak detection has a few different facets. It can be broken down in these aspects:

- Fast detection of large leaks (larger than 10 % of mass flow) and line ruptures;
- Detection of small leaks (smaller than 10 % of mass flow);
- Detection of incorrect closing of shutting down valves (SDVs).

SDVs are automatic closing valves that are distributed along the pipeline (in the case of Bolívia-Brasil pipeline, they are distributed at an average distance of 30 km from each other). Their function is to close if pressure reaches a value below some set point, or if there is a strong descending pressure gradient. They are used to isolate a section of the pipeline in critical cases. The problem is that they are sometimes also activated by transients caused by start-up of compressor stations, or other operational conditions. This situation must also be detected, so the valve can be reopened before interruption of service.

The advantage of embedding a leak detector in SEGRED is that it can also use its expert system structure, improving over other leak detection software that is already present in the market, as it will be presented in section 4.

This paper is structured as follows: in the next section, a brief overview of SEGRED is given, together with its architecture. Then an overview of leak detection and leak detection techniques, with emphasis on software-based leak detection techniques is made. Finally, the leak detector to be implemented in SEGRED is discussed, together with its improvements over standard leak detection systems. Finally, a brief review of the system proposed is made on the conclusion.

2. An overview of SEGRED

SEGRED is composed of two programs: SEGRED Stations (Castelani *et al.*, 2002; Silva Jr. and Silva, 2002) and SEGRED Network (Silva *et al.*, 2004; Silva and Porciúncula, 2003). These programs deal with distinct aspects of the network, and from a user's perspective, are two modules of SEGRED. These programs use expert systems concepts, as knowledge base clustering, object oriented programming, and in the case of SEGRED Network, the concept of agents. As this work is mainly based on SEGRED Network, only this module will be described.

2.1. Architecture

SEGRED is based on the integration of an expert system with a dynamic simulation software. This combination has been used before in support software for pipeline management, as can be seen in Johnson (1999), Uraikul *et al.* (2000) and Wallooppillai *et al.* (2000). The simulation package can be seen as the knowledge about pipeline dynamics, which is, in this case, not heuristic at all.

This integration allows for analysis of operational scenarios, validation of incoming measurements, and automatic extrapolation of actual operational conditions for prognosis. This emulates a process that is performed by pipeline operators, of studying the behaviour of some actions in a simulator before taking a real action. But, as it takes time to do it, and to analyse the result, this is usually done only for a few situations. Because the expert system automatizes this process, it can be done for more situations than before, improving the decisions made by the operator (Sampath and Yee, 2000).

On the expert system side, the knowledge base is clustered in a few groups of rules, each one with a well defined function, and a few objects that represent the pipeline components. The actions of each one of these clusters are coordinated by a control module, which is, in this implementation, built as a series of control rules.

The objects represent knowledge about topology and structure of the pipeline, providing information about each component state, and their organization. Their class diagram can be seen in Fig. 1. In this diagram, SDV stands for shutting down valve, DS for delivery station, CS for compressor station, HS for heating subsystem and PCS for pressure control subsystem. AREA is the class that holds the pipeline section of interest to the expert system. This structure is updated with data input by the user, or fetched from the supervisory control and data acquisition (SCADA) system.

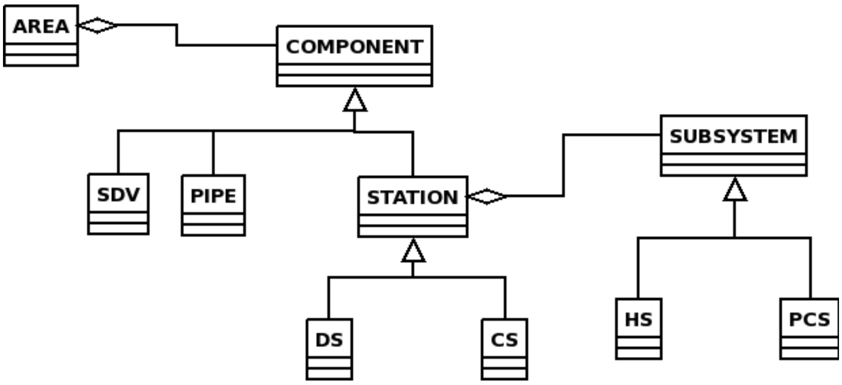

Figure 1. Class diagram for the knowledge base of SEGRED.

The clusters carry the following actions:

- Validation of gathered data;
- Analysis of the actual state of the pipeline, with a focus on detecting any failures;
- Prognosis of system behaviour;

Validation During the validation cycle, all data points read from the SCADA system or input by the operator are checked for consistency. If they are not consistent, an alarm is issued and the possible causes are evaluated. If desired, the points which are problematic can automatically be corrected, so the expert system can carry on with further analysis.

Analysis The actual state of each pipeline component is compared with its operational limits. This comparison yields a set of symbols, indicating semantically the component state. These symbols may indicate, for example, that the pressure upstream of a delivery station is too low, and the gas temperature of delivery is low. This information is then matched with a series of possible fault conditions, and the fault condition that is better described by that state is reported. These fault conditions are taken from the RCM (Reliability Centered Maintenance) report for that component.

Prognosis During the prognosis phase, the simulator is used to predict the pipeline state a few hours ahead (usually 6 or 8 hours). Then, the state of each component is evaluated, with a time step of 15 min, and messages relative to operational conditions are issued. Each message indicates the component, time, and a description of the situation. Also, each message has a criticality tag, that indicates a warning level (this tag is translated to the end user as different background colors for the message).

To implement the communication with the simulator, the agent concept has been used. The concept of agent is, according to Jennings (2000):

> [...] an encapsulated computer system that is situated in some environment and that is capable of flexible, autonomous action in that environment in order to meet its design objectives.

Based on this concept, and as a first study on these techniques, an agent was implemented, responsible for interfacing with the simulator. This agent has a very simple behaviour. Its main function is managing data that is input to the simulator, converting its output back to the expert system, and acting as a helper to the control structure and enabling the diagnosis cluster of the knowledge base. One of the advantages of its use is that it makes the expert system independent of the simulation software.

Nevertheless, the creation of an agent framework opens the doors for a more advanced use of these components. As such, the leak detection system proposed in section 4 is intended to be implemented as another agent within the system. An overall view of the architecture of the system can be seen in the Fig. 2.

3. Leak detection and leak detection systems

There are a lot of different methods for detecting leaks in gas pipelines. Each has its advantages and disadvantages. From an user point of view, the following metrics can be used to evaluate each option. They take into account both technical as well as economical criteria (Energy Solutions, 2004; Zhang, 1997).

Sensitivity The smallest leakage possible to detect, under ideal conditions, i.e. with SCADA System working correctly and without any transients on the pipeline. Sensitivity should be expressed as a flow rate.

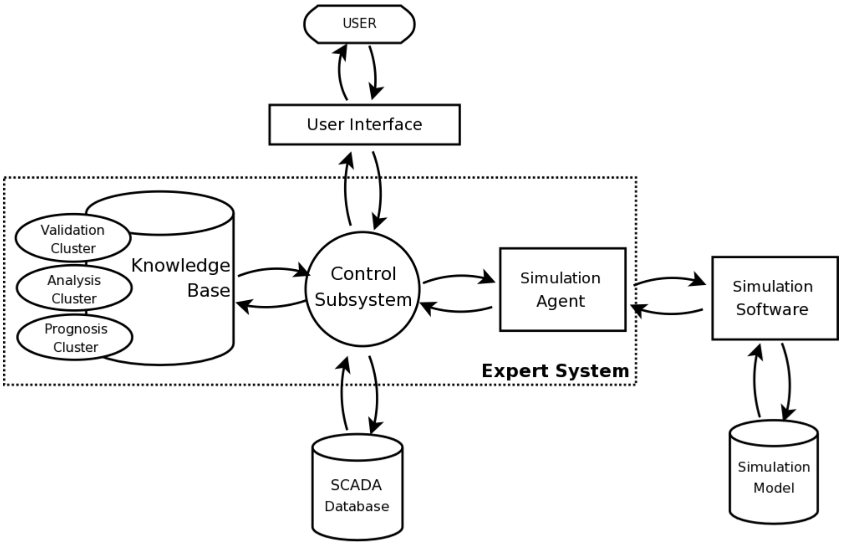

Figure 2. Overview of SEGRED architecture.

Robustness It represents the percentage of time that the system is able to detect the smallest possible leakage to detect, provided that instrumentation is working as expected. In certain situations, for example, during transients or maintenance operations, the LDS must be desensitized (that is, to have its sensitivity decreased), to avoid false alarms. Also, there are some LDS that do not operate continuously (e.g. the biological ones).

Reliability A measure of the maximum number of false alarms that the system is going to produce, provided that the SCADA System is working accordingly.

Capital expenditure (CAPEX) The amount of money spent to acquire all the equipment and software necessary to the LDS, plus costs of training, installing and commissioning the system.

On going operating costs (OPEX) Everything that is spent with maintenance, training, personal and other resources necessary to properly operate the system.

Usually, LDS are classified in three groups: biological, hardware-based and software-based. Following is a brief description of each category.

3.1. Biological

These methods usually require a technician to be at the leakage point to detect a leak. Normally, this means that the whole pipeline strip must be walked through frequently. These techniques usually use changes in the vegetation that grows above the strip. It is one of the most sensitive methods available, sometimes the only one able to detect very small leaks. In spite of that, it is not very robust, because it is available only during pipeline inspections, which are very rare (sometimes 6 months apart, or more).

3.2. Hardware based methods

Hardware-based LDS require installation of new measurement facilities along the pipeline. There are a lot of different methods under this category. For example, using a series of acoustic transducers to detect the noise produced by the leaking gas.

These methods usually present good sensitivity and reliability, and good robustness, with the exception of a few methods that work intermittently. On the other hand, their CAPEX is usually much higher than the other two types, and sometimes the OPEX is high as well.

3.3. Software based methods

Software based LD (SBLD) are characterized as a compromise between cost and sensitivity. The point is to offer a relatively cheap system, with a high robustness and reasonable sensitivity. These systems make use of sensors already installed in the pipeline for other purposes, usually in compressor and delivery stations, to search for signal irregularities that hint for a leak.

There are a series of techniques to carry out this analysis. These techniques are usually based on statistics and first principles (fluid-dynamics equations). The overall architecture of these systems can be seen as a feature extractor followed by a classifier. The classifier usually uses statistical analysis over the results of the feature extractor, which can be based both on first principles as well as on other statistical analysis. The output of the classifier is usually a warning, usually with two (no leak, leak) to four levels (no leak, warning, suspected leak, confirmed leak).

Following, a few techniques are going to be exposed and explained. The categorization used is common in the literature, but is by no means perfect, and, as usual with this kind of division, some methods could easily be included in more than one category (Energy Solutions, 2004; Matko *et al.*, 2003; Nicholas *et al.*, 1992; Zhang, 1997).

3.3.1. Pressure Point Analysis (PPA)

The PPA is one of the simplest (and also less sensitive) of the SBLD methods. It is based on the principle that a leak causes a pressure drop in the pipeline. This pressure drop can be detected using statistical filtering.

This method is very unreliable, and can be used only when the pipeline is in steady state. During transients any system based on this method must be desensitized, to avoid false alarms. One improved variant is to combine this method with a dynamic model, and calculate the PPA over the difference between SCADA measurements and the model predictions.

3.3.2. Mass balance

This method is based on the principle of mass conservation. In a pipeline, mass is conserved if there is no leak. The principle is simple:

$$\dot{M}_i(t) - \dot{M}_o(t) = \frac{dM_s}{dt} \qquad (1)$$

where $\dot{M}_i$ is mass flow at the inlet, $\dot{M}_o$ is mass flow at the outlet, and M_s is the mass stored in the pipeline, or line pack.

The greatest problem with this technique is the evaluation of the line pack, $\frac{dM_s}{dt}$. Usually, steady state models are used to calculate the line pack, and the system is then desensitized when there are transients. But, because of the compressibility of gas, steady state models are unreliable. Another option is to use a dynamic model. In general, when this is the case, this technique is called Model-Based LD method (MB-LD), and is covered in the next session.

Classification is usually done based on the line imbalance, R, which is a feature (on the feature extractor/classifier framework) calculated by:

$$R(t) = \dot{M}_i(t) - \dot{M}_o(t) - \frac{dM_s}{dt} \qquad (2)$$

And, in case $R(t) > A$, where A is the sensitivity, the condition is classified as a leak.

A common development line is to improve the classifier with some statistical analysis over the values of $R(t)$, in order to make the system more robust.

3.3.3. Model based leak detection systems

Some LD methods are based on dynamic models of the pipeline. These models can be used in a variety of ways. Two of these ways have already been presented: as extensions of the PPA method and the mass balance method.

Other ways can be found in the literature, but these methods yet have not found their way to commercial products. Among these methods, it is worth to highlight the work of Loparo *et al.* (1991), that uses a multiple model approach, some of them representing failure states, other representing non-failure states. The classification consists in identifying which model better represents the pipeline dynamics. Other approaches use model identification techniques to detect a leak, for example, in (Vostry, 2002), (Billman and Isermann, 1987) and (Benkherouf and Allidina, 1988). In a different approach, Belsito *et al.* (1998) uses a dynamic model as part of the input for an artificial neural network, which acts as a leak detector.

Despite of the increase in sensitivity that a dynamic model can offer, this technique has a few drawbacks. The first one is that, among all the SBLD methods, this is the most expensive one, due to the costs of creating and maintaining the model, and to the costs of the simulation software, which is usually expensive. Then, some parameters of the model are difficult to obtain, namely: the roughness of the pipeline and sensors set points. According to some authors, useful models can only be obtained if used together with adaptive models on these parameters (Nicholas *et al.*, 1992).

4. Proposal for a leak detection module for SEGRED

As described in the previous section, there are a lot of software based leak detection techniques. All of them are based in more or less sound statistical and dynamic systems theory. Nevertheless, they all fall short of the expectations when it comes to practical results (Vostry, 2002; Westhoff, 1999). That is not their fault, as they try to tackle a tough problem: to identify the state of a system naturally fraught with uncertainty, using a measurement structure that is neither very precise, nor correctly positioned for the task (Energy Solutions, 2004). The performance of a LDS is very closely tied to the measurement infrastructure. When this infrastructure can be more precise and more correctly positioned, as is the case with some hardware based LDS, e.g. those based on ultrasound flowmeters, the performance becomes very high.

Although some systems claim to be able to detect relatively small leaks, as 1 % or even 0.5 % of mass flow (Zhang, 1997), the smaller the leak, the longer it takes to be detected. On the other hand, 1 % of mass flow in a medium sized pipeline can be as much gas as is used by a small delivery station, and the noise produced by the gas escaping can be heard from far away. So, the preferred method for detecting small leaks by most companies is to put signs throughout the pipeline, indicating a phone number to be called in case of accident. The response time of the phone call is usually far shorter than that of the LDS. Thus LDS becomes in fact effective for small leaks only in desert and remote places.

Nevertheless, leak detection systems are still important. They are usually required by the country's legislation, and can provide useful information for pipeline operators. For large leaks and ruptures, usually LDS can provide faster response than the phone. A LDS can also inform about the incorrect closing of a SDV.

In reality, pipeline operators adapt themselves to the information provided by the LDS. So, instead of only alarms and warnings, the interface usually provides some information on the internal state of its parameters. So an operator can identify if that state is really associated with a leak, or with the closing of a SDV or instrumentation misreadings. Moreover, sometimes a risk analysis report for the pipeline is available, and in case of older pipelines some reliability figures as well. And there is information that is gathered from the regular pigging of the pipeline, that may inform about places of the pipeline where the risk of leakage is higher.

All these aspects point to the possibility of improving existent LDSs using an expert system to analyse the results of the system, and then present more meaningful information to the user. Thus, from a feature extractor/classifier point of view, the system can be improved both with new features (the reliability figures), and with an improved classifier. This improvement can be seen as a knowledge based approach, in a sense that, along with standard techniques for leak detection, heuristic knowledge from pipeline operators and maintenance team is also used, to have the most information available to help in correct identification of leaks.

4.1. Proposed system architecture

The system that is being proposed can be described by Fig. 3. The new knowledge based system is stacked upon the existing SBLDS, using a data access abstraction layer (DAAL). Most commercial SBLDS allow for some access to its results, usually for integration with other pipeline management software. So, the DAAL connects with this interface, acquire the data and translates it to a standard format that is used by the expert system.

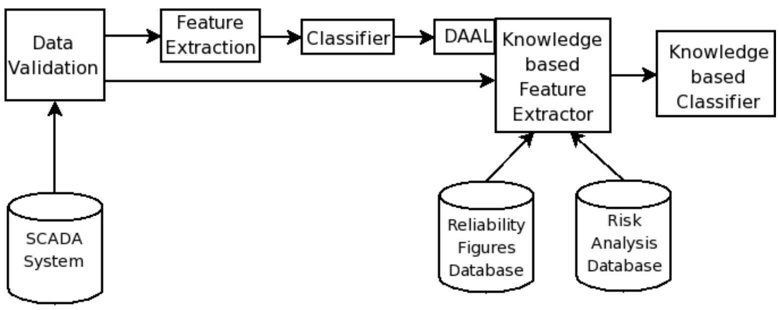

Figure 3. Proposed knowledge-based leak detection system.

The core components of the system are the two knowledge related layers, the knowledge-based feature extractor (KbFE) and the knowledge-based classifier (KbCl). Accessory to these two parts are two databases, one of pipeline reliability figures (RFDb) and one of risk analisys (RADb). The KbFE uses information from these databases, together with that provided by the already installed SBLDS, to produce high-level features, that are used by the KbCl to conclude for a leak, or maybe instrumentation error, or the incorrect closing of a SDV.

From an implementation point of view, this structure is going to be programmed in SEGRED as a group of agents. Using this kind of modularization, the system becomes more maintainable, and open for a future distribution, if necessary. Also, it becomes easier if in the future integration among different systems is desired, as shown by Jennings *et al.* (1993). Moreover, it permits the study of multi-agent systems on a real application.

The agent architecture that is going to be used is a simple one. The agents interact in a world implemented as the fact base of the expert system, which is implemented using the CLIPS language (Giarratano and Riley, 1998). The agents can sense (gather information of facts and objects in the fact base) and actuate (delete and assert facts, and change object parameters) in this world. They can also communicate with one another, passing messages. The message protocol is based on (Ferber, 1999), and allows for multicasting and broadcasting messages.

Internally, each agent has its own knowledge base. In the actual development, the agents work in *simple collaboration* (Ferber, 1999), because each agent has enough resources, compatible goals, but insufficient skills to solve the problem. The following agents are going to be used:

Data Access Agent (DA) Implements the DAAL. It is also responsible for pooling the LDS in a regular basis for new information. To some extent, all other functions are subordinated to this agent. This agent works as a proxy to the leak detection system that is being used by TBG;

Knowledge based leak detection agent (KBLD) Implements KbCl and KbFE. It was chosen to combine these two layers in only one agent because they are very closely interrelated;

Data base management agent (DBM) Responsible for retrieving and updating information from the RFDb and RADb;

Data validation agent (DV) Implements the data validation layer. Tight communication between this agent and the KBLD permits more meaningful information to be presented to the user, as the KBLD can inform this agent if it detects that a leak alarm was actually a misreading.

In this way the layered structure shown in Fig. 3 can be broken down in a few interacting agents, making a more flexible and maintainable design.

The agents are going to be implemented in this order: first the KBLD, them the DA, them the DV and finally the DBM. As of writing, the KBLD is being implemented.

5. Conclusion

Based on a study of leak detection techniques, it was identified that their weakness was in the lack of ability to analyse the result. Based on this, a improvement is proposed, that uses an expert system to carry on a further analysis over the results of the LDS. This analysis emulate the analysis made by a pipeline operator, checking if the alarm provided by the LDS was a real one or a false alarm, and identify other operational conditions that could have caused the alarm. An implementation of this system was then proposed, based on multi-agent techniques, to provide modularity, reusability and flexibility.

6. Acknowledgements

The authors acknowledge the financial support of CNPq, Conselho Nacional de Desenvolvimento Cientifíco e Tecnológico, PETROBRAS/CENPES and TBG.

7. References

Belsito, S., Lombardi, P., Andreussi, P. and Banerjee, S., 1998, "Leak detection in liquefied gas pipelines by artificial neural networks", AIChE Journal, Vol. 44, No. 12, pp. 2675 – 2688.

Benkherouf, A. and Allidina, A. Y., 1988, "Leak detection and location in gas pipelines", IEEE Proceedings in Control Theory and Applications, Vol. 135, No. 2, pp. 142–148.

Billman, L. and Isermann, R., 1987, "Leak detection methods for pipelines", Automatica, Vol. 23, No. 3, pp. 381 – 385.

Castelani, M. R., Galaz, L. A. and Silva, J. C., 2002, "Sistema especialista para gerenciamento operacional de redes de distribuição de gás natural", COCIM.

Energy Solutions, 2004, "Best practices in leak detection", White Paper.

Ferber, J., 1999, "Multi-Agent Systems - An introduction to distributed artificial intelligence", Addison-Wesley, 1st ed.

Giarratano, J. and Riley, G., 1998, "Sistemas expertos – Principios y programación", International Thomson Editores, 1st ed.

Jennings, N. R., 2000, "On agent-based software engineering", Artificial Intelligence, Vol. 117, No. 2, pp. 277–296.

Jennings, N. R., Varga, L. Z., Aarnts, R. P., Fuchs, J. and Skarek, P., 1993, "Transforming standalone expert systems into a community of cooperating agents", International Journal of Engineering Applications of Artificial Intelligence, Vol. 6, No. 4, pp. 317–331.

Johnson, A. T., 1999, "A practical approach to the application of an expert system to gas pipeline operation and data integrity", PSIG – Pipeline Simulation Interest Group.

Loparo, K. A., Buchner, M. R. and Vasudeva, K. S., 1991, "Leak detection in an experimental heat exchanger process: a multiple model approach", IEEE Transactions on Automatic Control, Vol. 36, No. 2, pp. 167–177.

Matko, D., Geiger, G. and Werner, T., 2003, "Leak detection and localisation - a survey", PSIG – Pipeline Simulation Interest Group.

Nicholas, R. E., Reet, J. V. and Whaley, D. R. S., 1992, "A tutorial on computer based leak detection methods", PSIG – Pipeline Simulation Interest Group.

Sampath, S. and Yee, S., 2000, "Automating the predictor to answer routine operational questions", PSIG – Pipeline Simulation Interest Group.

Silva, J. C., Hirano, E. W., Moura, N. R. and Freire, L. G. M., 2004, "Sistema especialista para gerenciamento de redes de gás natural – SEGRED", Rio Oil & Gas 2004.

Silva, J. C. and Porciúncula, G. S., 2003, "Sistema especialista para gerenciamento de redes de transporte de gás natural", Rio Pipeline Conference & Exposition.

Silva Jr., A. C. and Silva, J. C., 2002, "Integração entre sistemas especialistas e simulação para o monitoramento de redes de transporte de gás natural", Congresso Nacional de Engenharia Mecânica, CONEM.

Sommerville, I., 1995, "Software engineering", Addison Wesley Longman Limited.

Uraikul, V., Chan, C. W. and Tontiwachwuthikul, P., 2000, "Development of an expert system for optimizing natural gas pipeline operations", Expert Systems with Applications, Vol. 18, pp. 271–282.

Vostry, Z., 2002, "New leak detection and localization method", PSIG – Pipeline Simulation Interest Group.

Wallooppillai, R. K., Marquart, B. D., Istre, M. L. and Johnson, A. T., 2000, "Integrating an expert system and pipeline simulator to enhance gas pipeline operation, profitability and safety", PSIG – Pipeline Simulation Interest Group.

Westhoff, M. A., 1999, "Using operating data at natural gas pipelines", International Symposium on Transportation Recorders. National Transportation Safety Board.

Zhang, J., 1997, "Designing a cost effective and reliable pipeline leak detection system", Pipes & Pipelines International, pp. 20–26.

8. Responsibility notice

The authors are the only responsible for the printed material included in this paper