Consideration on real time implementation of fault/leak detection system

Ciprian Lupu
Faculty of Automatic Control and Computers
University Politehnica of Bucharest
Bucharest, Romania
ciprian.lupu@acse.pub.ro

Serban Iftime
Faculty of Power Engineering
University Politehnica of Bucharest
Bucharest, Romania
serban.iftime@elsaco.ro

Roxana Miclaus
Faculty of Medicine
University Transilvania of Brasov
Brasov, Romania
roxicum@unitbv.ro

***Abstract*— Fault or leak detection in the utilities distribution and energy fluids networks represents an objective with significant implications, where, the pollution and safety of life are priorities included. Reality and scientific literature contain dedicated solutions, proposed by theory and validated in time, and industry. Starting from a few of these, this paper proposes and analyzes some applicative solutions based on the models identification, (mathematical) modeling and supervision of normal and fault operation in the transportation and distribution networks for (liquid/fluid) utilities. Proposed supervisory structure is based on sensors/data acquisition and control architecture. The main purpose is to detect the fault situation as fast as possible and to indicate more accurately the affected area/point. The methods have been real time hardware and software implemented and validated with experimental laboratory facility.**

Keywords—Fault detection; real time; SCADA; mass transfer; model identification;

I. INTRODUCTION

The classic manner in which we can test if an element or subsystem does not work correctly in the local/global system, is to compare the evolution of the correct operation model with the effective, real time trends (Fig. 1) [1]. To avoid malfunctioning operation of the control system, a continuous monitoring of many measured variables is necessary and also checking their tolerance limits [2].

From the safety perspective, a diagnosis system must be designed and implemented so as to detect all unnatural changes in the real system evolution and to suggest, as fast and precise as it is possible, the human operator of possible fault and remedial solutions, or to start the procedures for automatic countermeasures. Starting from industrial practical experience [3], there are different methods that offer different advantages, and, corresponding drawbacks [4], [5], [6].

In many literature approaches [1], [2], [4], [7] few main detection methods can be identified:

- • Hardware based methods: using acoustic sensors, gas detectors, negative pressure detectors, and/or infrared thermography;
- Software based methods: varying complexity and reliability are used. Examples include flow/pressure change detection and mass/volume balance, model based systems and pressure point analysis;
- Biological methods: the dogs walk along the pipes and look for obvious damage, smell and sound.

Proposed paper use methods based on software applications (implemented on PLC, SCADA) mixed with Model Based Systems (MBS) solutions. Some interesting elements about this:

The common denominator for all model based systems is that the pipe flow is described mathematically. Leaks are detected when discrepancies between calculated and measured values differ. Equations used to model fluid flow in pipes are:

- Conservation of mass;
- Conservation of momentum;
- Conservation of energy;
- Equations of state for the fluid.

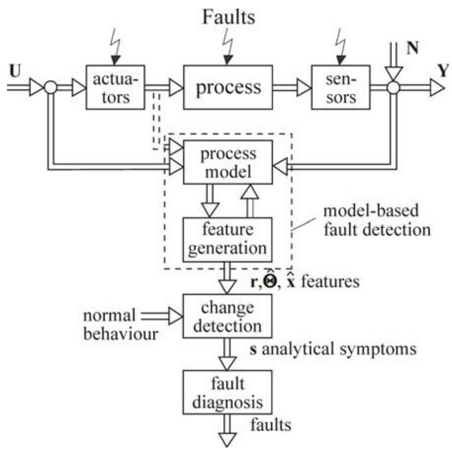


Fig. 1. The system proposed by R. Isermann [1]

II. PROPOSED SOLUTION

Some classical approaches (e.g. “change flow” or “model based systems”) use a nonlinear observer [8], neural networks [9], [10]; or finite difference [11]. The presented solution is based on real time simulation of some transfer functions, combined with parameters evolution supervision. Computed transfer functions characterize both normal and fault operation system.

Obviously, the real time fail detection is based on a complex acquisition system, provided by the oil; water etc. network’s associated SCADA structure. One of the main components of the proposed solution comes as a (software) module, included into supervisory level.

The other elements that define the solution are a series of flow and/or pressure sensors, disposed along the network, on every individual/important section, as presented in Fig. 2 (S1, S2, S3, …). Functionally, the (fault) detection, is based on the fact that a fault occurred on a section will be “visible” in the measures acquired from (flow and pressure) adjacent sensors, section respectively. Because the transport system can operate at different nominal values (different set points), for fail/leak detection, some transfer functions will be used - from the actuators (compressor, pump etc.) to those associated network sensors.

In Figure 2, the most important elements are:

- DP1– Distribution Pipeline;
- F11, F12, F2, F3, F4 – Fail/Defect/Leak ;
- S0, S1, S2, S3, S4 – Sensor (S0 – pressure; S1, 2, 3, 4 - flow);
- H11, H12, H2, H3, H4 – Transfer functions

The calculated command (in real time) for the real process is applied, in parallel and identified models. Normal function is characterized by a low measurement error from process exits face identified models (for normal evolution). In this case, the minimum error will be provided by the corresponding defect models respectively. Figure 4 shows the structure of the system. Between two sensors area on can "detect" multiple faults. (e.g.: in Fig. 2 in the first segment, marked by S1 and S2, are two faults - F11 and F12).

The usual steps for the configuration and implementation of the detection system are the following:

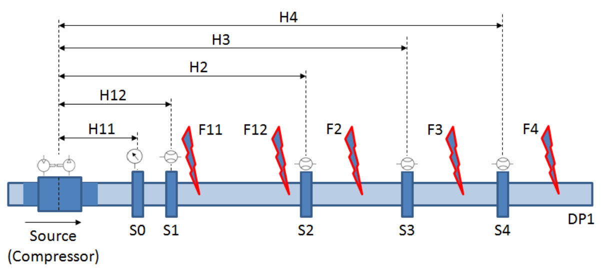


Fig. 2. Proposed system - general scheme

- data acquisition;
- identification of the transfer function / models;
- the design of the control algorithm;
- the implementation of the detection structure.

A. Data acquisition

Data acquisition - first of all, it consists in the application of some test signals (PRBS - Pseudo Random Binary Sequence [13]) for a normal function, in an admissible domain (field), and in a fault condition. There will be registered, with the same sampling period, the order values (pressure/flow, applied command) and all the values of the implicated sensors. The application of the (additional) PRBS signal is made for a system found in a normal state of function, and, for every type of defined fault. In this manner, there would be more datasets resulted, one for every type of defined fault (as presented in [17] and [18] - Figure 3).

B. Models identification

The identified model structure can be ARX type [12] (1). The identification is made with the help of (some) recursive least squares method (RLSM) [13] (2). The models are further validated with a procedure based on the whitening of the residue [13].

$$\begin{aligned} y(k) &= \frac{q^{-d}B(q^{-1})}{A(q^{-1})}u(k) \\ A(q^{-1}) &= 1 + a_1 q^{-1} + \ldots + a_{nA} q^{-nA} \ , \\ R(q^{-1}) &= b_0 + b_1 q^{-1} + \ldots + b_{nB} q^{-nB} \end{aligned} \tag{1}$$

$$\hat{\theta}(k+1) = \hat{\theta}(k) + F(k+1)\phi(k)\varepsilon^0(k+1), \forall k \in N$$

$$F(k+1) = F(k) - \frac{F(k)\phi(k)\phi^T(k)F(k)}{1+\phi^T(k)F(k)\phi(k)}, \forall k \in N \tag{2}$$

$$\varepsilon^0(k+1) = y(k+1) - \hat{\theta}^T(k)\phi(t), \forall k \in N$$

with the following initial conditions:

$$F(0) = \frac{1}{\delta}I = (GI)I, 0 < \delta < 1$$

C. Control algorithm

Control algorithm design is made on one of the identified models, depending on the controlled parameter, flow or pressure. Since models may have high order (> 2) the control algorithm is an RST type (3), with two degrees of freedom [12] (Fig.3), based on the poles placement design procedure.

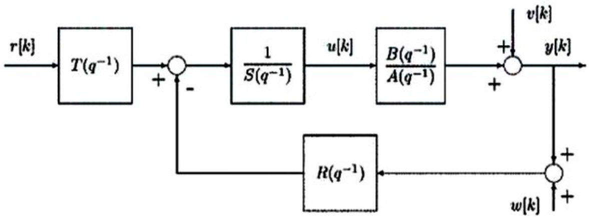


Fig. 3. RST control algorithm structure [12]

The design of the control algorithm is made on the basis of one of the identified models.

$$
\begin{aligned}
R(q^{-1}) &= r_0 + r_1 q^{-1} + \ldots + r_{nR} q^{-nR} \\
S(q^{-1}) &= s_0 + s_1 q^{-1} + \ldots + s_{nS} q^{-nS} \\
T(q^{-1}) &= t_0 + t_1 q^{-1} + \ldots + t_{nT} q^{-nT}
\end{aligned} \tag{3}
$$

D. Fault detection structure

As presented before, one of the special elements of this approach consists in using the identified models for control as well as for diagnosing the defects. So, the calculated command made by the control algorithm is applied in real time in process, as well as the models for normal functioning and the ones that describe the functioning affected by a defect. The structure is presented in figure 4. Here, the notations are the following:

- P - process - transport network;
- C – algorithm for the control of the main parameter;
- r - set point;
- y - controlled parameter;
- u - command calculated by the controller;
- yS0, yS1, ...,ySk - data delivered by the S0, 1, ..., k sensors (or concentrated Y(S0, S1, ..., Sk));
- MnS0, MnS1, ...,MnSk - normal operating models calculated towards the 0, 1, ...k sensors;
- MfiS0, MfiS1, ...,MnfSk - operating models with the i defect calculated towards the 0, 1, ..., k sensors ;
- y_MnS0, y_MnS0, ...,y_MnSk - estimated outputs from the processing models with normal operating calculate towards the 0, 1, ..., k sensors;
- y_MfiS0, y_MfiS1, ..., y-MfiSk – estimated outputs from processing models with the i defect calculated towards the 0, 1, ..., k sensors;
- e_Mn_S0, e_Mn_S1, ...,e_Mn_Sk - estimated output errors from the processing models with normal operating calculated towards the 0, 1, ..., k sensors;

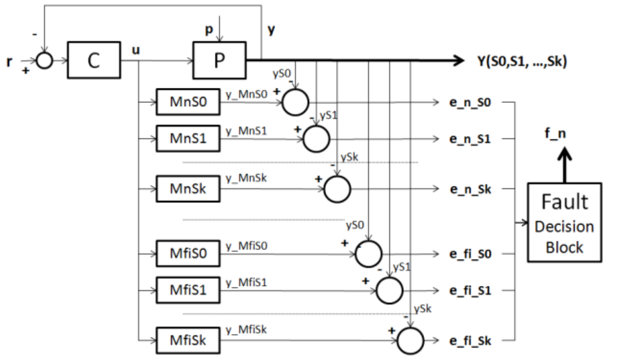


Fig. 4. Proposed fault detection system - general scheme [18]

- e_Mfi_S0, e_Mfi_S1, ...,e_Mfi_Sk - estimated output errors, from the processing models with the ***i*** defect calculated towards the 0,1, ..., k sensors;
- f_n - identified defect
- Fault Decision Block - the block that has the role of identifying the defect.

As presented in the introduction, there are different methods offered different advantages, and corresponding drawbacks [5], [14]. Some of them are based on of the sums of related errors to each defect (4). The sum that has the minimum value, compared to all others, is an indication related to the occurred fault. The expressions (4), (5) describe these solutions. The weighting factors w_{ji} define the influence of the "*i*" fail in the "*j*" sensor data. (It is possible that a fail may not affect the measurements of all sensor measurements).

$$
S\_fi = \sum_{j=1}^{k} w_{ji} * e\_Mfi\_Sj \quad \text{where i} = 1,\ldots,\text{k}
$$
$$
w_{ji} \text{ - weighting factor} \tag{4}
$$

$$
f\_n = \min\left(S\_f0, S\_f1, \ldots, S\_fk\right) \tag{5}
$$

E. Fault detection procedure

Detection can be implemented based on [18] proposed structures (real time flow variation and models response), but these show only the segment where the fault occurred, without specifying precisely the fail position.

In the same time, the problem arises in the multitude of sensors, but especially for normal/fail models to be made - not simply to achieve those conditions.

The fault cannot be determined only from models response - they are (also) affected by the defect. The precise fault location (necessary to) includes measurement data and possibly, the static characteristics of the sensor.

In this context, proposed fail detection procedure may include next steps:

- Verify models response: if N (N - normal) or, F11 or, respectively, F12 situation (F - fail), identified by the (minimal) corresponding model's error of the respective defects, a pre- identified fail was detected;
- If N error is high and simultaneously, F11 and F12 errors are small but, none is minimal, there is identified an *intermediary* fail *(with unknown position)*;
- Intermediary fail position (distance) is calculated based on real time values (computed control value (u) and S1 and S2 sensors data) and S1, S2 static characteristics value (calculated for corresponding actual control value). Distance ratio between F11 and F12 is calculated for each S1, S2 sensor and, the finally value, represents the mean of them. There is obtained a x% ratio between S1 and S2;

(An example of S1 and S2 calculated value based on corresponding static characteristics is presented in results section – Table III).

III. Experimental Results

Some real time software applications and (real) experimental laboratory stand have been developed in order to implement and prove proposed solutions.

A. Hardware and software structure

The goal is to offer a versatile platform for testing the fault detection structures in fluid distribution pipelines. The main component parts are axial fans, pressure and flow sensors, drivers and signal adapters (for sensors and fans) and different pipe profiles (Fig. 6). For testing the fault events, some special elements were included into distribution pipelines structure. This element allows fluid leak (by removing some caps). The connection and the control of a stand are made by some data acquisition systems (e.g. NI USB 6008) or by means of PLC. Software applications can be developed in specific or general languages, from CScape (Horner TM), RS Logix (Rockwell Automation) to Lab View and LabWindows CVI (National Instruments) [15] or Matlab/Simulink (MathWorks) [16].

The used structure for the current experiment contains one pressure (P), two flow sensors (S) and two axial fans arranged in series [17] (as source - compressor) (Fig. 5).

The fluid source has variable speed. The acquisition and the control are made by an acquisition system NI USB 6008. The software application that implements the control, fault detection and acquisition algorithms were developed in the

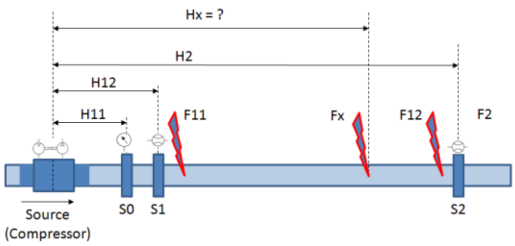


Fig. 5. Tested system - general scheme

Fig. 6. Experimental instalation (with real sensors and actuators). Fails are provoked by remuving pipe caps – right part.

Lab Windows/CVI (ANSI C) environment (National Instruments).

In Figure 5, the main elements are similar with Fig. 2, with difference H1x – unknown position fail.

As we specified, the effects of a fail can be visible in the data acquisition (from sensors). It is not mandatory for any fault to be "visible" by any sensor. For the laboratory considered structure, every fail is visible in every sensor's data, but with different importance. Causality defect - affected data, is presented in Table 1.

Generally, the pressure decreases to the appearance of these fails. As the position of a fail is far away, the effect is less visible. However, if a control algorithm exists, this one will maintain it (pressure) constant, as presented in Table 1.

TABLE I. THE EFFECTS OF FAILS (S0 – IN CLOSE LOOP)

Fail/ Sensor	**F11 effect/(vary)**	**F1x effect/(vary)**	**F12 effect/(vary)**
S0 pressure	constant	constant	constant
S1 flow	up / (high)	up / (medium)	up / (medium)
S2 flow	down / (medium)	down / (medium)	down / (medium)

The F11 and F12 fails lead to a decrease evolution (even to the value zero), recorded by S1 and S2 flows.

The experimental installation is presented in Fig. 6.

B. Real time evolutions

The fails' effect, from the point of view of the operating points, can be exemplified in Table 2, for a imposed set point of S0=49.00% (with computed command for the first compressor) and u2=20% for the second compressor [17].

For models identification a PRBS signal [13] was generated with n=10 (register length) and 20% amplitude variation.

TABLE II. THE EFFECTS OF FAILS FOR NOMINAL POINT

Fail/ Sensor	**Normal %**	**F11 %**	**F1x %**	**F12 %**
u1 (for S0=49.00%)	47.00 %	66.00 %	62.00 %	60.00 %
S1 (flow)	44.00 %	80.00 %	54.00 %	56.00 %
S2 (flow)	38.00 %	34.12 %	34.73 %	35.45 %

Corresponding models for normal (no fault) functioning is (6-8):

$$\mathrm{M}_{\mathrm{S0n}}(q^{-1}) = \frac{0.01977}{1 - 0.98297q^{-1}} \tag{6}$$

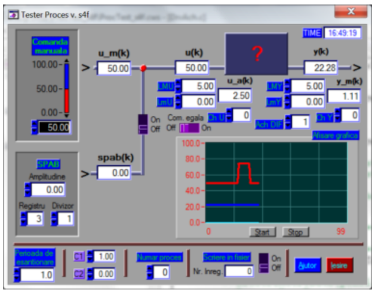


Fig. 7. Software application for real time data acquisition

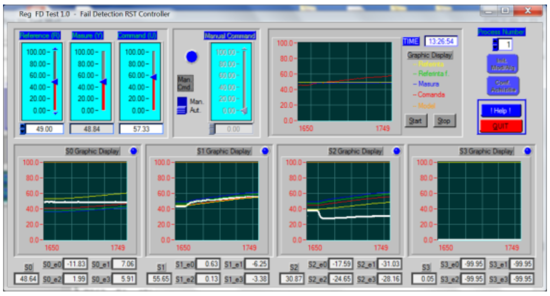


Fig. 8. Software application for real time control and fault detection

$$M_{S1n}(q^{-1}) = \frac{0.00985 + 0.04696q^{-1}}{1 - 1.20647q^{-1} + 0.26457q^{-2}} \quad (7)$$

$$M_{S2n}(q^{-1}) = \frac{0.00915 + 0.03357q^{-1}}{1 - 1.16366q^{-1} + 0.21295q^{-2}} \quad (8)$$

and, (e.g.) the models for F11, F12 fault functioning for S0 sensor are:

$$M_{S0f11}(q^{-1}) = \frac{0.01612}{1 - 0.97954q^{-1}} \quad (9)$$

$$M_{S0f12}(q^{-1}) = \frac{0.01752}{1 - 0.98023q^{-1}} \quad (10)$$

The control algorithm for pressure (using S0 sensor) is:

$$\begin{aligned} R(q^{-1}) &= 29.116591 - 23.155438q^{-1} \\ S(q^{-1}) &= 1.0 - 1.0q^{-1} \\ T(q^{-1}) &= 50.581689 - 71.185382q^{-1} + 26.564846q^{-2} \end{aligned} \quad (11)$$

The software component contains two main elements: first one, the application for the data acquisition (Fig. 7) and, the second one, the debit/pressure control and fault detection application (Fig. 8). For the model identification and control algorithm design the Advantech's WinPim and WinReg [13] software applications were used.

The software application for data acquisition has the following functionalities: real time data acquisition through an acquisition device (e.g.: NI USB 6008), choosing the sample period, choosing the acquisition and command canals, setting the limits for data variation, setting the filter constants, generating and applying the PRBS signal.

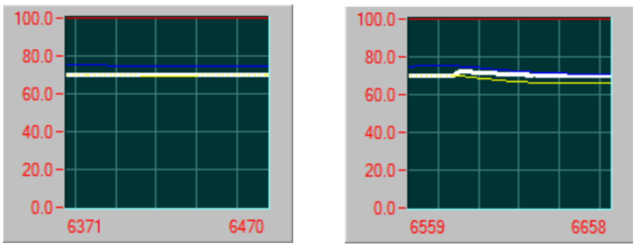


Fig. 9. Normal functioning (left) vs. fault detection (right)

The second application, for pressure control and fault detection, has the following functionalities: setting the control algorithm (RST), implementing the fault detection algorithm, real time data acquisition, choosing the sample period, choosing the acquisition and command canals, setting the limits for data variation, loading the fault/normal functioning model from the files.

Some real time tests are made for normal and fault (F11, F12, F1x) state. Figure 9, for example, presents a normal evolution vs. fault occurred. Here, with white (thick line) is the response process (S1 - flow) and yellow and blue (thin lines) are the model's responses normal functioning (yellow), respective fault (blue) (caused byF11 fault).

From the previous figure one can notice that in the moment of the fault, the answer of the fault model approaches from the real evolution. The evolution of the model of the normal functioning gets away from the data acquired in real time.

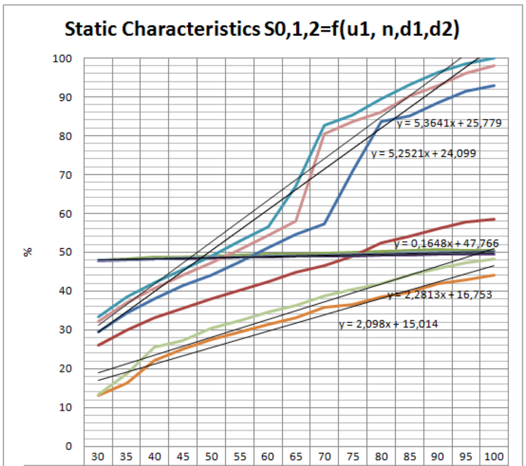
This trend occurs around the corresponding functioning point, the identified models. If the real process is linear previously evolution is similar to respect other points of operation. If the process is nonlinear new identifications procedure are necessary for each different functioning point. It reaches such a "multiple models" structure [13].

Static characteristic for S0, S1 and S2 sensors are experimentally determined for each 10 percents and presented in Fig. 10, and are visible that for some of sensors is nonlinear.

For testing proposed procedure detection accurately, an "unknown" position fail (60% close to F11 and respectively, 40% close to F12) was caused. The corresponding control algorithm output is 63% and for S1 = 54% and S2 = 34.73% values ware read. From static characteristics (SC), corresponding distance relative to F11 and F12 ware (graphically) determinate. Table III represent this situation:

TABLE III. FAIL POSITION DETECTION (65%)

Fail/ Sensor	Real time	F11 SC Pos [%]	F12 SC Pos [%]	Delta F11	Delta F12	Dist. To S1 [%]
S1 flow	54%	67%	58.00%	13%	4%	76%
S2 flow	34.73%	33.10%	36.36%	1.65%	1.65%	50%
Mean val.						63.2%



	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100
S0n	48	48,2	48,6	48,7	49,1	49,2	49,4	49,5	49,7	50	50,2	50,4	50,6	50,5	50,5
S1n	29,4	34,4	37,9	41,4	44	47,4	51,1	54,5	57,2	71	83,5	85	88,5	91,5	93
S2n	26	30,1	33	35,7	37,9	40,2	42,4	44,7	46,4	49	52,4	54	56	57,7	58,5
S0d1	47,8	48	48,1	48,3	48,5	48,6	48,7	48,9	49	49	49,2	49,3	49,4	49,5	49,6
S1d1	33,3	38,5	42,2	45,5	49	52,8	56,5	67	82,7	85,3	89,5	93,2	96,4	98,6	100
S2d1	13,1	16,2	22,1	25	27,4	29,4	31,4	33,1	35,8	36,5	38,4	39,6	41,9	43	44,2
S0d2	47,9	48	48,2	48,5	48,6	48,7	48,9	49	49,2	49,5	49,6	49,9	50	50,1	50,2
S1d2	32,1	36,9	40,6	44,1	47,3	50,7	54,4	58	80,6	83,7	86	90,3	93	96	98
S2d2	13,3	18,8	25,5	27,2	30,5	32,5	34,6	36,4	38,7	40,4	42	44	45,8	47,2	48,3

Fig. 10. Static characteristics (SC) for S0, S1 and S2 sensors

Finally the estimated position for unknown fail is 63.2%, which represent a 3.2% estimation error. In other situations this value was higher (> 10%). Some real time implementation observations can be made here:

Fail detection procedure based on pre-identified fail/normal functioning models provide good results for reduced neighborhood of corresponding (model) identifications points. (E.g. 0.5% modified set point value leads to normal functioning but even F11 and F12 situation were not correctly detected.

For these reasons, supplementary model set identification, (in other functioning points/values) are necessary, to extend the (same) fail detection procedure (multiple model structure similarity).

Fail detection precision is not high (10%) and, to improve it, some supplementary flow/pressure real time data acquisition values could be added;

IV. Conclusions

In this paper, a practical solution for fail/leak detection in distribution pipeline, is presented. Their applicability in actual gas / liquid utility systems is obvious. In order to validate the proposed solution an experimental stand and two real time software applications have been developed. The detection algorithm is based on normal / fail process transfer functions (models), specific to real structure.

The experimental results validate the proposed solution and lead to interesting conclusions. Meanwhile, some applications in the medical field can be glimpsed.

The proposed solution has some software and real hardware components. Additive flow sensors need to be installed on each important section, and, of course, a corresponding communication network, too. This imposes supplementary costs, but actual noninvasive measurement solutions and modern communication have acceptable prices compared with the costs of the damages caused by defects.

Identified models, as base of software detection structure, may be obtained using installed SCADA system. The test signals (PRBS), imposed for correct identification may be applied during normal functioning without special regimes and devices.

Acknowledgment

This work was co financed by the University Politehnica of Bucharest - A.C.P.C. Research Center, and by the Automatic Control and Computer Faculty projects.

References

[1] R. Isermann, “Process fault detection based on modeling and estimation methods. A survey”, Automatica,vol. 20, iss. 4, July 1984, pp. 387–404.

[2] N. C. Turner, “Hardware and software techniques for pipeline integrity and leak detection monitoring. Offshore Europe”, 1991.

[3] Chemical Process Safety (Third Edition), edited by Roy E. Sanders, Gulf Professional Publishing, Burlington, 2005.

[4] I.G. Geiger, “Principles of leak detection. Fundamentals of leak detection”, KROHNE oil and gas, 2005.

[5] S. Datta, and S. Sarkar, A review on different pipeline fault detection methods, Journal of Loss Prevention in the Process Industries, Volume 41, May 2016, pp. 97-106.

[6] R. Isermann, “Fault-Diagnosis Systems: An Introduction from Fault Detection to Fault Tolerance”, Springer-Verlag Berlin Heidelberg.

[7] O. Sindre, “Leak Detection in Pipelines by the use of State and Parameter Estimation”, Master Thesis, Norwegian University of Science and Technology Department of Engineering Cybernetics, January, 2014.

[8] S. A. Ashton, D. N. Shields, and S. Daley, "Fault detection in pipelines using nonlinear observers," Control '98. UKACC International Conference on (Conf. Publ. No. 455), Swansea, 1998, pp. 135-140.

[9] M. Araujo, J. Aguilar, and H. Aponte, "Fault detection system in gas lift well based on artificial immune system," Neural Networks, 2003. Proceedings of the International Joint Conference on, 2003, vol. 3, pp. 1673-1677.

[10] Á M. Avelino et al., "Real time leak detection system applied to oil pipelines using sonic technology and neural networks," Industrial Electronics, 2009. IECON '09. 35th Annual Conference of IEEE, Porto, 2009, pp. 2109-2114.

[11] G. Besancon, M. Guillén, J-F. Dulhoste, R. Santos, and D. Georges, “Finite-difference modeling improvement for fault detection in pipelines”, IFAC Proceedings vol. 45, Iss. 20, January 2012, Pages 928-933.

[12] I. D. Landau, R. Lozano, and M. M'Saad, “Adaptive Control”, Springer Verlag, London, 1996.

[13] D. Popescu, D. Stefanoiu, C. Lupu, C. Petrescu, B. Ciubotaru, and C. Dimon, “Industrial Automation”, Editura AGIR, Bucuresti, 2006.

[14] S. L. Scott, and M. A. Barrufet, “Worldwide assessment of industry leak detection capabilities for single & multiphase pipelines”, Department of Petroleum Engineering Texas A&M University,Agust, 6, 2003.

[15] *** National Instruments, NI USB 6008, http://www.ni.com

[16] MATLAB, www.mathworks.com

[17] D. Chiriţă and C. Lupu, “Multiple model control solution for series fans processes (with variable load)”, U.P.B. Sci. Bull., Series C, Vol. 78, Iss. 2, pp. 65-78.

[18] C. Lupu, D. Chiriţă, and S. Iftime, “Consideration on leak/fault detection system in mass transfer networks”, accepted for EENVIRO 2016, 26-28 October 2016, Bucharest, Romania