A SCADA Expansion for Leak Detection in a Pipeline*

Rolando Carrera* **Cristina Verde*** and **Raúl Cayetano****
*Universidad Nacional Autónoma de México, Instituto de Ingeniería
e-mail: rcarrera@unam.mx, verde@unam.mx
** Universidad Nacional Autónoma de México, Posgrado de Ingeniería
email: rcayetanos@ii.unam.mx

Abstract

A solution for expanding an already existing pipeline SCADA for real time leak detection is presented. The work consisted in attaching a FDI scheme to an industrial SCADA that regulates liquid distribution from its source to end user. For isolation of the leak a lateral extraction is proposed instead of the traditional pressure profile of the pipeline. Friction value is a function of pipe physical parameters, but on line friction estimation achieved better results. Aspects that were important in the integration of the FDI scheme into the SCADA were the non synchrony of pipeline variables (flow, pressure) and their accessibility, that leaded to data extrapolation and the use of data base techniques. Vulnerability of the location algorithm due to sensors bandwidth and sensitivity is showed, so the importance of selecting them. The FDI scheme was programmed in LabVIEW and executed in a personal computer.

1 Introduction

Leak detection and isolation in pipelines is an old problem that has attracted the attention of the scientific community since decades. A paradigmatic example is the oil leakage in the Siberian region [1], where the effects on the surrounding nature have been disastrous. In Mexico, a semi desert country, there is the need to transport water to the population on long distances via aqueducts; this requires complex supervision systems that detect leakages in early ways. Also, there exist a complex net of pipelines that transport oil and its by-products; in this net, besides the leakage problem, there exist also the illegal extraction of the product transported in the pipeline; this forces that the distribution system should have a leak detection and location monitoring system.

Since the 1970's years have been issued several works that have been fundamental for the detection and location of leakages as the one of Siebert [2], where on the basis of the steady state pressure profile along the pipeline simple expressions are derived, based on correlations,that detect and locate a leakage. Later Isermann [3] published a survey showing the state of the art on fault detection by using the plant model and parameter identification. Recently, Verde published a book [4] making emphasis on signal processing, pattern recognition and analytical models for failure diagnosis.

But all the later is pure academical, our aim here is to share some of our practical experiences acquired during a re-engineering project that consisted on adding a real time leak detection and location layer to an already existing SCADA. The original objectives of that SCADA were the administration and delivery of some products, through pipelines, from the source to the end user. As it was our first approach to integrating a FDI to an existing SCADA and that we didn't have experience on this subject, we proposed a solution that involves simple algorithms for detecting and locating a leak. In future work we'll use more elaborate algorithms as dedicated observers or detecting two simultaneous leaks.

In order to show how we solved the targets of the project we divided the solution in five major parts (each one included in sections 2 to 6 down here). Some of them are extracted from available theory, as the dynamical model for a flow in a pipe and the expression for leak location, and others are consequence of the experience achieved in our lab facilities, as the calculus of pipe friction and and the choice of sensors, and finally the data acquisition imposed by the nature of the available SCADA.

Delivering a fluid to clients means steady operation, then our solution required a suitable model for that condition, section two describes how to achieve a simple steady state model for a pipeline. Once the model is at hand an appropriate expression for leak location is needed, for that purpose in section three a simple method for locating a leak is presented. From our experience, pipe friction plays a fundamental role in the exact location of the leak and that real time estimated friction is better than a beforehand constant one; an on-line expression for calculating the pipeline friction is showed in section four. In this project we didn't have the option to choose sensors, but we consider appropriate to share here our experience in this matter, a comparative study on how different type of sensors affect the leak location is presented in section five. The data acquisition system of the SCADA is based on a MODBUS system and a database with the information of the pipe variables, we didn't have the right to get into the MODBUS but in the database, section six shows how the indirect measurement of pipe variables issue was solved by using ethernet and data bases, also, the extrapolation of data of non existing data during sample times is presented. Finally, the concluding remarks of this work are presented in section seven.

*Supported by II-UNAM and IT100414-DGAPA-UNAM.

2 Pipeline steady state model

In most applications a dynamical model of the system is required but not here because of the steady operation of the pipeline, then a steady state model is more suitable. Besides, the pipeline lies buried in the field and has an irregular topography, but it is possible to derive a model that handles it like a horizontal one. This model is simpler as will be showed.

In the following we modify the model of a pipeline with topographical profile as showed in Figure 1 into one with a right profile piezometric head, where the pressure variable depends on a reference value h, as is the hight over sea level along the pipeline. Consider the one dimension simplified flow model in a pipeline with n sections [5],

$$\frac{1}{A^i}\frac{\partial Q^i(z^i,t)}{\partial t} + g\frac{\partial H^i(z^i,t)}{\partial z^i} + \frac{fQ^i(z^i,t)|Q^i(z^i,t)|}{2D^i(A^i)^2} + g\sin\alpha^i = 0 \qquad (1)$$

$$\frac{\partial H^i(z^i,t)}{\partial t} + \frac{b^2}{gA^i}\frac{\partial Q^i(z^i,t)}{\partial z^i} = 0 \qquad (2)$$

which assumes that fluid is slightly compressible, pipe walls are slightly deformable and negligible convective changes in velocity. Q is volumetric flow, H is pressure head, A is pipe cross-sectional area, g is gravity, f[1] is the D'Arcy-Weissbach friction [6], b is the velocity of pressure wave, D is pipe diameter, z is distance variable and t the time. Super index $i = 1, 2, ..., n$ indicates pipeline section characterized by its slop with angle α^i, n is the total number of sections.

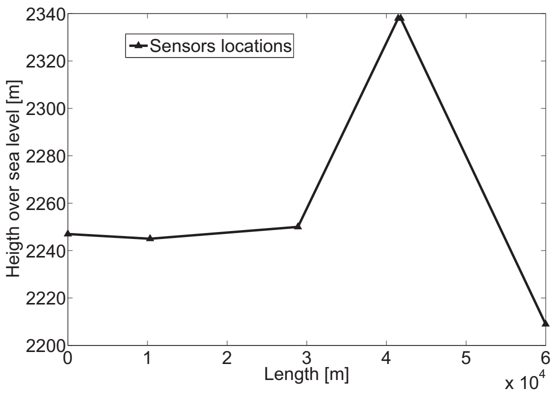

Figure 1: 60 km Pipeline topographical layout

We start with the following hypothesis: the system works in steady state and that the pipeline lay on an horizontal surface. Therefore we need a steady state model that takes into account these conditions.

In order to describe the behaviour of the pressure head $H^i(z^i,t)$ along a section without branches it is assumed steady state flow, so from (2) one gets

$$\frac{\partial Q^i(z^i,t)}{\partial z^i} = 0 \quad \Rightarrow Q^i \text{ constant} \qquad (3)$$

Combining (1) and (2)

$$\frac{dH^i(z^i)}{dz^i} + M^i(Q^i) = 0, \qquad (4)$$

[1] This friction characterizes the shear stress exerted by the conduit walls on the flowing fluid.

with

$$M^i(Q^i) = \mu^i Q^i|Q^i| + \sin(\alpha^i) = m^i(Q^i) + \sin(\alpha^i) \quad (5)$$

that is independent of the spacial coordinate z^i, and $\mu^i := f^i/2D^i(A^i)^2 g$. Then the solution of (4) reduces to

$$H^i(z^i) = -M^i(Q^i)z^i + H^i(0) \quad \text{for} \quad 0 \le z^i \le L^i \quad (6)$$

with $H^i(0)$ the pressure head at the beginning of section i. Defining boundary conditions for section i in terms of pressure at the ends:

$$H^i(z^i = 0) := H^i_{in} \qquad H^i(z^i = L^i) := H^i_{out}. \quad (7)$$

with (7) in (6), we obtain

$$H^i_{in} - H^i_{out} = M^i(Q^i)L^i = m^i(Q^i)L^i + \Delta H_i, \qquad (8)$$

where $\Delta H_i = L^i sin(\alpha^i)$ is the height difference between section ends.

It is reported in [7] and [8] that the pressure head

$$H^i(z^i) = \frac{P^i(z^i)}{\rho g} \qquad (9)$$

can be written in terms of the *piezometric head* $\tilde{H}^i(z^i)$, wich depends on a heigth h that can be related to sea level, i.e.

$$\tilde{H}^i(z^i) = H^i(z^i) + h(z^i), \qquad (10)$$

$h(z^i)$ in m over reference datum or sea level, ρ is fluid density. Then the profile pressure (8) is equivalent to

$$\tilde{H}^i_{in} - \tilde{H}^i_{out} = m^i(Q^i)L^i \qquad (11)$$

for section i and sea level $h(z^i)$ along the section. Finally, considering that boundary conditions are related by

$$\tilde{H}^i_{out} = \tilde{H}^{i+1}_{in}, \qquad (12)$$

from this equation and (11) one gets

$$\tilde{H}^1_{in} - \tilde{H}^n_{out} = \sum_{i=1}^{n} L^i m^i(Q^i) \qquad (13)$$

which is function of the piezometric head for a pipeline with n sections without branches.

The profile of Figure 1 corresponds to the topography of the pipeline under study. The pressure head $H(z)$ and the resulting piezometric head $\tilde{H}(z)$ are shown in Figures 2 and 3, respectively. Take into account the uniformity of $\tilde{H}(z)$ similar to the one of a horizontal pipeline. The reference datum was the height of the first sensors location.

As a consequence, if $\tilde{H}^1_{in} = \tilde{H}_{in}$ and $\tilde{H}^n_{out} = \tilde{H}_{out}$, besides if $m^i(Q^i) = m(Q) = M(Q)$ for all i, then Equation (13) becomes

$$\tilde{H}_{in} - \tilde{H}(z) = LM(Q) \qquad (14)$$

where $L = \sum_{i=1}^{n} L^i$ the total length of the pipeline. Equation (14) is the steady state piezometric model for the pipeline viewed as a horizontal one.

3 Leak location

We consider a leakage as an outlet pipe at the leak location as is shown in Figure 4. A branch or lateral pipe in section i breaks the continuity of variables $Q(z,t)$ and $H(z,t)$, therefore new boundary conditions must be satisfied [9]. In

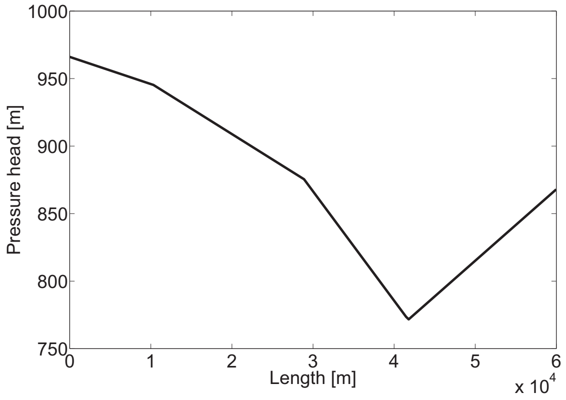

Figure 2: Pipeline pressure head profile $H(z)$

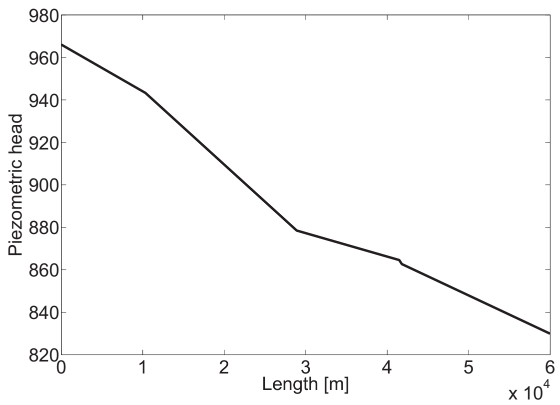

Figure 3: Profile of the piezometric head $\tilde{H}(z)$

particular, the union of three pipes is associated to a geometry shown in Figure 4 and the corresponding conditions that describe the action of separating flow are reduced to

$$H_2 = H_1 + \kappa_{12}(H_2, H_1) \quad (15)$$

$$H_3 = H_1 + \kappa_{13}(H_3, H_1) \quad (16)$$

where H_2 and H_3 are pressures at the beginning of pipes 2 and 3 and the functions $\kappa_{1\eta}(\cdot,\cdot)$ with $\eta = 2, 3$ represent losses caused by friction and change of flow direction. For adjusting the order of magnitude of these functions flow simulations were held with Pipelinestudio [10] with the topology of the study case shown in Figure 1. Simulation reported that terms κ_{12} and κ_{13} were negligible, then

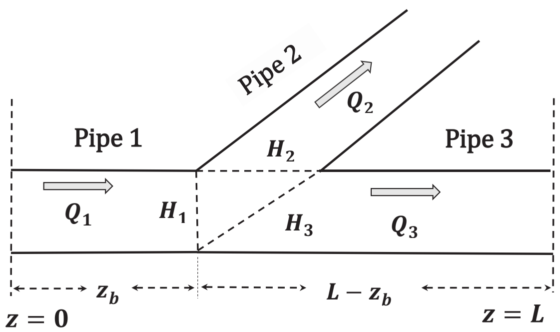

Figure 4: Union of three branches in point z_b of pipeline with transversal section areas A_1, A_2 and A_3

$H_1 = H_2 = H_3$. Thereafter in the study was included only the balance

$$Q_1 - Q_2 - Q_3 = 0, \quad (17)$$

as consequence,

$$Q_1 = Q_{in}, \quad Q_3 = Q_{out} \quad (18)$$

with Q_{in} y Q_{out} flows at the ends of the pipeline. So the differential equation (4) transforms in two equations

$$\begin{aligned} \frac{dH^1(z)}{dz} - M(Q_1) &= 0; \quad \text{for} \quad 0 \le z \le z_b \\ \frac{dH^3(z)}{dz} - M(Q_3) &= 0; \quad \text{for} \quad z_b < z \le L, \end{aligned} \quad (19)$$

describing the pressure head along the section with a branch in point z_b. As the equations (19) have the same form as (4), their solutions also have the same as (6). Therefore, with boundary conditions:

1. $H^1(z = 0) = H_{in}$,
2. $H^3(z = L) = H_{out}$,
3. $Q_{in} = Q_{out} + Q_{z_b}$ and
4. $H_{z_b} - \epsilon = H_{z_b} + \epsilon$ with $\epsilon \to 0$

Assuming that all pipes have same diameters, solutions of (19) evaluated at the ends are reduced to

$$\begin{aligned} \frac{H_{in} - H_{z_b}}{z_b} - M(Q_{in}) = 0 \\ \frac{H_{z_b} - H_{out}}{L - z_b} - M(Q_{out}) = 0. \end{aligned} \quad (20)$$

Obtaining the variable z_b associated to the position of the branch

$$\begin{aligned} z_b &= \frac{M(Q_{out})L^i + H_{out} - H_{in}}{M(Q_{out}) - M(Q_{in})} \\ &= \frac{L \sin\alpha + m(Q_{out})L + H_{out} - H_{in}}{m(Q_{out}) - m(Q_{in})}, \end{aligned} \quad (21)$$

in terms of the piezometric head

$$z_b = \frac{m(Q_{out})L + \tilde{H}_{out} - \tilde{H}_{in}}{m(Q_{out}) - m(Q_{in})}. \quad (22)$$

Equation (22) is the key for leak isolation. In order to see the performance of this leak location method some experiments were held in our pipe prototype [11], which is an iron pipe of 200 m long, 4 inches diameter and six valves attached to it for leak simulations. Table 1 shows the percent deviations of locating the leak position. In each experiment a valve was fully open. Coriolis sensors were used.

4 Pipeline friction

The D'Arcy-Weissbach friction is a function of the pipe parameters, [6] and [12], and operation conditions, as the Reynolds number. For practical purposes the friction f is obtained from tables provided by the pipe manufacturers. But we observed that that value differs from the real one of a working pipeline where, no matter that is working in steady state, the value is influenced by noise -caused by pipe inner surface imperfections and attachments (nipples, elbows, etc.)-, therefore using a previous fixed value of f is of no use in Equation (1).

Table 1: Location error in percentage of total pipe length

Experiment	Δz_b [%]	Valve position [m]
1	1.66	11.54
2	2.93	49.83
3	0.135	80.36
4	0.54	118.37
5	0.375	148.93
6	3.42	186.95
Mean	1.0	

To overcome the problem of not having the friction right value, we proposed a solution that was an on line friction estimation. In the following we show how to calculate this friction. For that, we part from the steady state momentum equation, Equation (4). Turning back the original parameters we get

$$g\frac{dH}{dz} + \frac{f}{2DA^2}Q\left|Q\right| + gsin\alpha = 0 \tag{23}$$

solving the integral, considering that H_0 and H_L are pressures at he beginning and at the end of the pipeline and L the length, results

$$g(H_L - H_0) = -(\frac{f}{2DA^2}Q_\infty^2 + gsin\alpha)L \tag{24}$$

where Q_∞ is volumetric flow in steady state, the absolute term disappears when flow goes in one direction only. Friction has the following expression

$$f = \frac{2DA^2g}{L}\frac{(H_0 - H_L - Lsin\alpha)}{Q_\infty^2} \tag{25}$$

Equation (25) is used to calculate on line the friction value, as is shown in Figure 5, experiment realized in our pipeline prototype. The calculated friction has a considerable amount of noise, but this noise can be attenuated via weighted mean value with forgetting factor (MVFF, continuous line in figure). Actually, we are working on the use of recursive identification procedures for a better friction estimated.

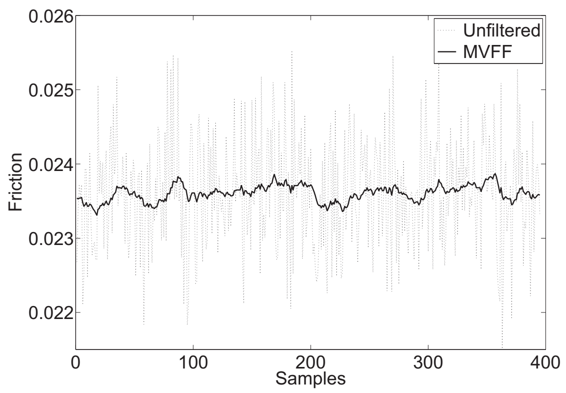

Figure 5: Friction estimated, raw and filtered

5 Influence of sensors on location

Flow measurement in a pipeline is fundamental for leak location, in view that most of the pipeline leak detection methods are based on processing a residual that is a flow difference. Due to our lack of experience, and by suggestion of a supplier, we start our flow measurements with a paddle wheel flow sensor [13]. Later on, as ultrasonic sensors are widely used in the field, we decide to change to them [14], thinking that our measurements would be better. Finally, we reached the conclusion that success on leak detection and location depends strongly on the sensors quality (make and sensing principle), so we acquired sensors based on the Coriolis effect [15].

An experiment that we made in our pipe prototype was to cause a leakage (outflow in a extraction point) and estimate the location with the measurements of the three sensors. Figure 6 shows the deviation of the calculated location depending on the type of sensor. Oscillations are observed around the operating point, which leads to the necessity of signal filtering in the diagnosis process. Table 2 shows the error leak location, Paddle Wheel and Coriolis sensors have similar error, but standard deviation is bigger with the Paddle Wheel. In order to compare performance in the fourth column the accuracy of the instruments are presented; remark that Coriolis error standard deviation is about seventy times bigger than sensor accuracy. The observation here is that the quality of the results depends more on the behavior of the flow than on the accuracy of the instrument used.

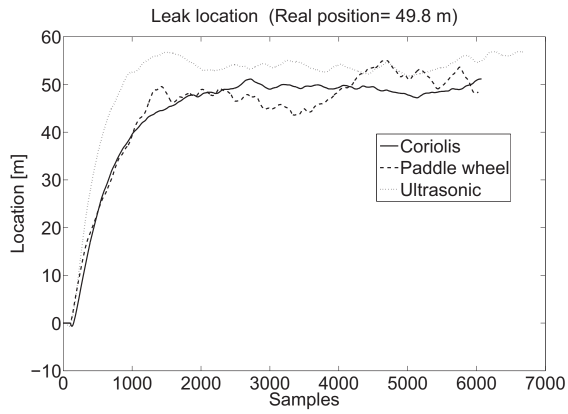

Figure 6: Leak location with the three sensors

Table 2: Leak location errors

Sensor	Error [%]	Error STD [%]	Accuracy [% FS]
Paddle wheel	-0.28	3.36	0.50
Ultrasonic	2.12	1.39	2.00
Coriolis	0.28	0.84	0.05

One of our goals in the SCADA expansion project was to deliver results in real time. For this, sensors experiments were performed to determine which one would have the faster response. An index to take into account is the time response, it can be appreciated in Figure 6 but is practically the same, therefore we measured the settling time from the moment when the leakage valve is opened. In Figures 7, 8 and 9 the flow development is observed, dotted line indicates the time when the leakage valve is opened to 100%. In Table 3 are the measured times, being the ultrasonic sensor which requires more time (this by the number of points used to calculate a mean value).

Table 3: Sensors settling time

Sensor	$t_s[s]$
Paddle wheel	3
Ultrasonic	35
Coriolis	4

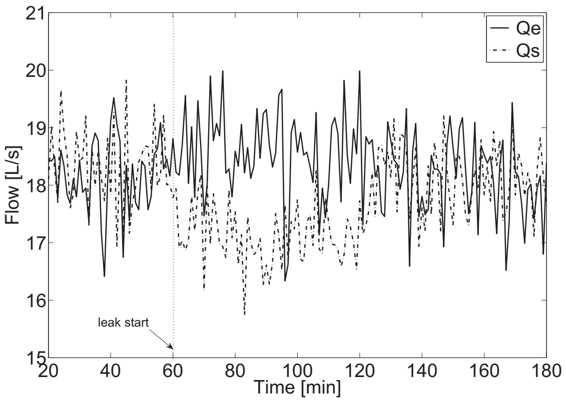

Figure 7: Flow measurement at the pipe ends, paddle wheel sensors

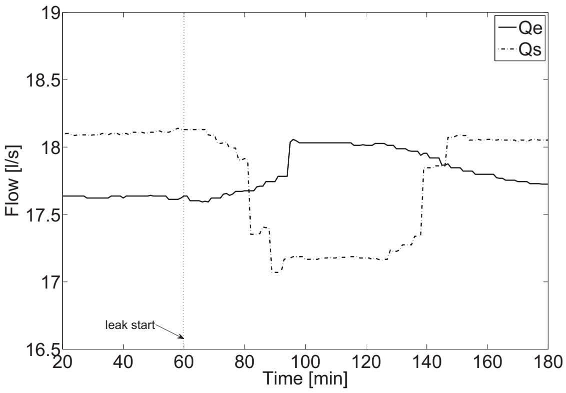

Figure 8: Flow measurement at the pipe ends, ultrasonic sensors

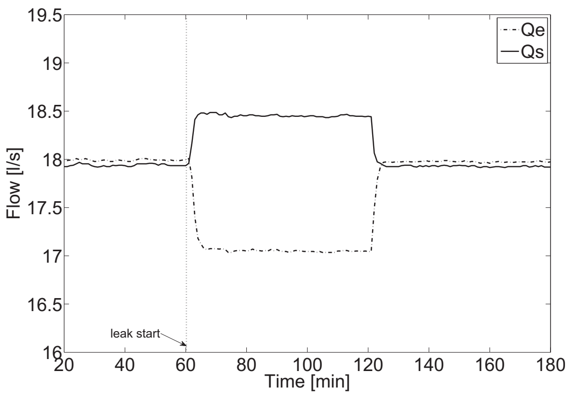

Figure 9: Flow measurement at the pipe ends, Coriolis sensors

Considering the settling time and noise in measurements (taking the STD as the measure for that), Coriolis sensor has the best performance. Experiments showed in this section were made with 1 s sampling period.

6 Asynchronous data and data bases

In the academy, we are used to work with benchmark systems or laboratory facilities with *ad hoc* data acquisition systems, sufficient sensors, controlled environments, etc. But these conditions are not necessarily in the practice, as was the case of the SCADA expansion, where the access to flow and pressure sensors of the pipeline were not available, but through a database. So the solution adopted was as follows:

1. The leak locator is on a dedicated computer, independent of the system that regulates de distribution of the fluid, it connects to the database server, see Figure 10, via intranet or VPN (Virtual Private Network) connection in a LAN (Local Area Network) system.
2. With proper permission a program, task performed with Visual Studio 2010 tool that runs every minute (it is a program without GUI -Graphic User Interfacer- that runs silently), brings system data and creates a database with pipeline flow and pressure information, data required by the locator for proper operation.
3. The locator program (made in the LabVIEW platform, [16]) periodically takes data (through SQL data server of Microsoft), applies the detection algorithm and when detects a leak proceeds to locate it, displays on the screen the location of the leak (Figures 12 and 13), generates a visual warning and creates a file with data leakage.

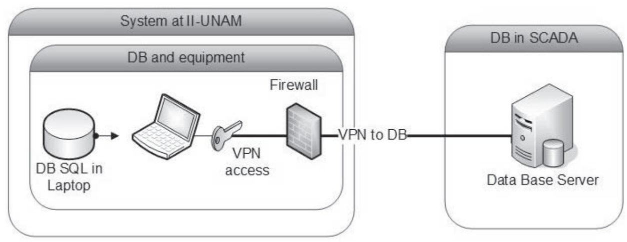

Figure 10: Communication scheme between leak locator and database

But the data acquisition system of SCADA do not meet the condition of sampling the system variables with constant sampling period. The nominal sampling period was 3 min, but in reality this varies from one to several tens of minutes. On the other hand, the locator was assigned a sampling period of 3 min, determined by the condition that nominally SCADA performs a polling of all measuring stations in that time span. To solve the problem of having a value of flow and pressure of each station at all sampling time, it was added to the localizer an algorithm that extrapolates the missing data when it is not available. Two algorithms were tested, one that retains the last data in the following sampling periods and one that generates straight line with the last two values available, that when the value of the variable that is brought from the database is not a new one, then the one determined by straight line is used. In order to compare results with both proposals a simulation with real data with three leaks was carried on, in Figure 11 the real and extrapolated input flow data are shown. It can be seen that at certain intervals the extrapolation by a straight line delivers values that may be beyond the normal range of measurements, this situation is exacerbated in large intervals with empty data as the line grows monotonically delivering data outside the region of validity. In Figure 12 the location of a leak is shown when extrapolated data are used and in Figure 13 when retained data are used. The pipe length is about 20 km, so that retention has outperformed extrapolation, since the latter yields higher values than the length of the pipe. Original leak location was about 10 km.

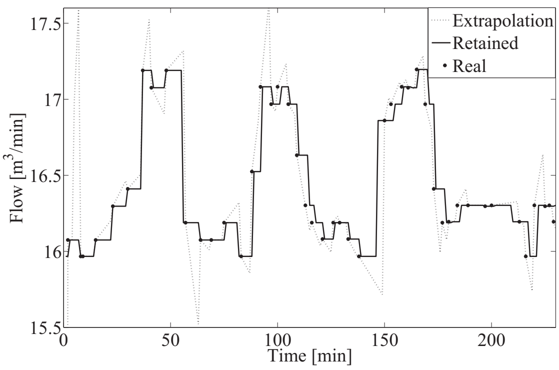

Figure 11: Graphics with original, extrapolated and retained data of input flow with three leaks

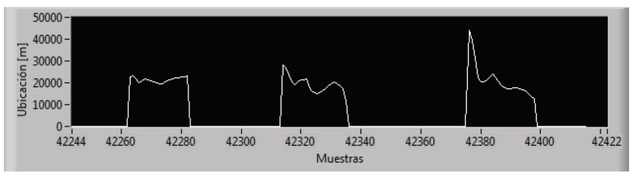

Figure 12: Leaks location with extrapolated data

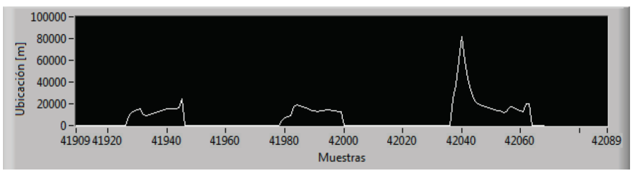

Figure 13: Leaks location with retained data

6.1 Alternate database communication

As part of the project requirements, an alternate way of communication with the SCADA database was experimented. In previous section the communication between leak locator and database was direct trough a LAN system, the alternate way was through a third party via internet and VPN connection. Figure 14 shows the principal elements of this scheme.

The client is the computer with the locator program build in LabVIEW platform that performs basically two activities: leak detection and location, and request and sending data to communications broker using JSON strings. The remote client interface is a Java process that runs locally and handles communication, authentication, data formatting, encryption and security of the communication with data server. It connects to the database in the SCADA through TCP sockets and VPN.

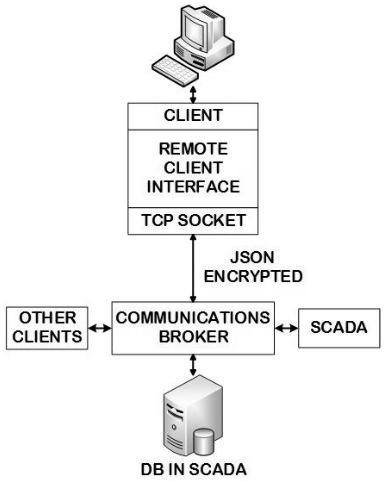

Figure 14: Communications between client and database

For data handling JSON format is used, which is broadly used for information interchange trough internet. JSON (Java Script Object Notation) is a data interchange text format, easy for humans to read and write [17]. JSON is a collection of pairs {variable name : value}, realized as an object, record, structure, dictionary, hash table, keyed list, or associated array, see in Figure 15 an object example.

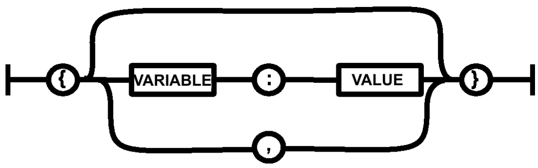

Figure 15: JSON data format for an object

An example of a JSON string for reporting a leak is the following:

```
{"service":"event",
"options": {
  "action":"new",
  "vector": {
    "Module":XXX,
    "EventID":XXX,
    "Quantity":XXX,
    "PipeID":XXX,
    "Location":XXX,
    "TimeEvent":"yyyymmddhhmmss"}
} }
```

Communications broker attends clients requests (leaks locator is not the only one) and also SCADA requests. The database attached to the broker contains not only pipeline data but also data generated by the other clients. At the end, the SCADA has an interface in which information of leakage events is displayed.

Figure 16 shows a test ran with real data but off line. That experience showed that locator not always received answers from the broker. But this communications scheme is still in development.

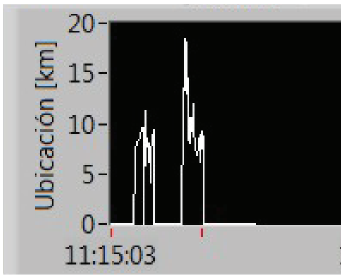

Figure 16: Off line experiment with real data. Detail of the graph, y axis is leak location in km

7 Conclusions

An interesting result is that a pipeline with certain topography may be analyzed as an horizontal pipe in which the piezometric head is a sum of measurements and terrain heights, Equation (10), as seen in section 2.

Compared with traditional methods for locating a leak in a pipe, the method shown here, Equation (22), requires less computational effort and has a simple expression for calculating it.

Another relevant result is the expression for on line calculation of the pipeline friction, Equation (25), as it is enough to measure pressure at the ends and steady state flow. The value of friction was found to be a key parameter for the exact location of the leak. It is to remark that when a leak occurs the pressures change modifying the friction value; in order to avoid wrong location of the leak we keep a delayed value of friction that is frozen when leak alarm occurs.

On the other hand, is to highlight the importance of choosing the appropriate sensor. It is not enough to choose a sensor capable of measuring a certain physical variable, also must be included in the selection process the purpose for which the measurements are needed.

The world of measurements for control targets is not limited to direct measurement of the physical variable, it is possible to achieve the control objectives with indirect measurements, as was the case of reading the variables from the plant via the network to a database. Also, with the partial absence of data we cannot use the plant model to predict data, then the use of extrapolation methods proves to be a powerful tool that helped to achieve the goal of this project; in this paper we use two simple methods, but this is an area that we continue to explore.

The experience with JSON format strings showed that it is easier to work with text characters than with specialized database commands and, no matter the VPN connection and data encryption, the scheme depends strongly on internet conditions. If internet fails leak detection scheme fails, situation that scarcely appears when the locator connects with database through a LAN system.

To the moment this paper was written our FDI system is in the proof stage at the SCADA facilities and we are waiting for in the field results.

8 Acknowledgments

Authors are very thankful to Jonathán Velázquez who helped us by solving the database issues emerged in this project.

References

[1] I. Bazilescu and B. Lyhus. Russia oil spill. http://www1.american.edu/ted/KOMI.HTM, 1994.

[2] H. Siebert and R. Isermann. Leckerkennung und -lokalisierung bei pipelines durch on-line-korrelation mit einem prozessrechner. *Regelugnstechnik*, 25:69–74, 1977.

[3] R. Isermann. Process fault detection based on modeling and estimation methods - a survey. *Automatica*, 20(4):387–404, July 1984.

[4] C. Verde, S. Gentil, and R. Morales. *Monitoreo y diagnóstico automático de fallas en sistemas diámicos*. Trillas, 2013.

[5] M. Hanif Chaudhry. *Applied Hydraulic Transients*. Springer, third edition, 2014.

[6] C. F. Colebrook and C. M. White. Experiments with fluid friction in roughened pipes. *Proceedings of the Royal Society of London*, 1937.

[7] J. Saldarriaga. *Hidráulica de acueductos*. Mc Graw-Hill, 2003.

[8] R. Bansal. *Fluid mechanics and hydraulic machines*. Laxmi Publications (P) LTD, 2005.

[9] Oke A. Mahgerefteh, H. and O. Atti. Modelling outflow following rupture in pipeline networks. *Chemical Engineering Science*, (61):1811–1818, 2006.

[10] PipelineStudio. Software in energy solutions international. http://www.energy-solutions.com/, 2010.

[11] R. Carrera and C. Verde. *Prototype for leak detection in pipelines: Users Manual*. Instituto de Ingeniería, UNAM, Ciudad Universitaria, D.F., Noviembre 2010. In Spanish.

[12] C. Tzimopoulos G. Papaevangelou, C. Evangelides. A new explicit relation for the friction coefficient in the darcy-weisbach equation. In *PRE10: Protection and Restoration of the Environment, Corfu, 05-09 July, 2010*, 2010.

[13] G. Fisher. *Signet 2540 high performance flow sensor*. Georg Fisher Signet LLC, El Monte, CA, 2004.

[14] Panametrics. *Two-Channel TransPort Mod. 2PT868 Portable Flowmeter. User's Manual*. PANAMETRICS, Inc., Waltham, MA, USA, 1997.

[15] E+H. *Proline Promass 83 Operating Instructions*. Endress + Hauser Flowtec, Greenwood, IN, USA, 2010.

[16] National Instruments. *LabVIEW user manual*. National Instruments Corporation, Austin, 2013.

[17] JSON Organization. Ecma-404 the json data interchange standard. http://www.json.org/.