

Chapter 7

LabVIEW-Based SCADA System for Sequential Leaks' Diagnosis in Pipelines

Rolando Carrera¹ and Cristina Verde¹

Most of the leak diagnosis contributions have only been tested off-line by the academic community, and there are few studies on the implementation of algorithms as an integral part of a SCADA (Supervisory Control And Data Acquisition) system with diagnostic capability. This paper aims to fill the lack of valuable knowledge from a practical point of view about the implementation of diagnosis systems. Specifically, we discuss one that introduces the design of LabVIEW software implemented on a CompactRIO controller, tasked with localizing sequential leaks according to an algorithm previously developed at the Instituto de Ingeniería, UNAM. This program was designed as part of the SCADA system for a hydraulic pilot plant. Some recommendations and remarks are addressed based on the authors' experience.

7.1 Introduction

Fault detection and identification (FDI) is a relative new research area in the automatic control community. FDI solutions have been developed almost for every engineering field. One field of interest is faults in pipelines because they may cause heavy financial losses or worst: environment damages or human casualties (Fang et al., 2019). Thus, for this purpose different pipeline leakage detection methods have been performed. For instance: exterior methods that are based on sensors that detect the presence of the conducted material in the environment along the pipeline; interior or computational methods based on dynamic behavior of the fluid and; visual or biological methods based on human or artificial vision or acute smell capacity of dogs (Fang et al., 2019).

Diverse technologies for leak diagnosis through software have been developed based on the model of the dynamics of a liquid element in a pipe, starting with the works of Billman and Isermann (1987) by assuming only available data of pressure and flow rate at the ends of the pipeline. Recently, the state of the art of the leak diagnosis methods has been published in Moubayed et al. (2021), Zhang et al. (2015) and Liu et al. (2019). The detectors by software use data driven models such as neural networks, analytical models derived from the dynamic model, and also combinations of both models. Thus, one can find simple static relationships easy to implement (Carrera and Verde, 2001) up to complex observers for hyperbolic equations by boundary injection (Aamo, O. M and Salvesen, J and Foss, 2006).

¹Instituto de Ingeniería, UNAM, Mexico, e-mail: rcarrera@unam.mx and verdeunam.mx

In contrast to the diverse options that isolate a single leak, few methods have been validated by considering a multiple leak scenario. This fact could be motivated by the remark about the weak detectability of two simultaneous leaks published in Verde et al. (2003) and the practical condition of avoiding permanent manoeuvres in a line for inflammable fluids. These issues imply that some leaks cannot be isolated, once the transient response produced by the leak has disappeared. On the other hand, when the useful lifetime of a pipeline is exceeded, sequential leaks could occur. This fact motivated the search for possible solutions for the multiple sequential leaks (*MSL*) problem in which the time interval required by the application to isolate the leaks determines how close in time the leaks can appear.

One research line in the Coordinación de Eléctrica y Computación at the Instituto de Ingeniería, UNAM, is fault detection and identification of complex systems, specifically early fault detection in pipelines with this goal in mind, a pipeline prototype (Carrera, 2019) was built in the Hydromechanics Laboratory, as seen in Fig. 7.1. Data from experiments can be downloaded in order to perform FDI off line. More importantly, however, with this facility one can test FDI methods in real-time.



Fig. 7.1: Partial view of the facility. Pipeline in blue to the right and hydraulic pumps, instruments and water reservoir to the left

The prototype is instrumented with flow rate and pressure sensors and is also complemented with a data acquisition system and a personal computer for software issues. This prototype has been used to test several fault detection methods throughout its existence (the first prototype was built in 1999), currently we are testing wave pressure and multiple leaks detection methods. The last method is the focus of this chapter.

In this chapter we present a new supervisory control and data acquisition (SCADA) system called SCADA-II (Instituto de Ingeniería). The goal of this system is to apply in real-time a multiple sequential leaks method in our prototype. In the following sections, the whole process of achieving this goal is explained.

7.2 Experimental Prototype Description

The prototype is a pipe without lateral branches with coil geometry, with pressure and flow sensors at the ends, and several valves and servo valves for emulation of leaks along the pipe. Pressure sensors can be attached at the ends of each level of the coil. Therefore, this prototype is suitable for the application of the method described in Section 7.3.

7.2.1 Prototype configuration

The fault detection prototype was built with a 7.62 mm diameter iron pipe. Table 7.1 shows its physical parameters, and the layout of the pipe is shown in Fig. 7.2. The main parts of the prototype are the following: a) 10 m³ water reservoir; b) two hydraulic pumps of 7.5 and 10 HP, which can be combined or used alone; c) Coil shape pipe on a vertical plane that ends in the drinking water reservoir; d) ball intake valves for flow regulation; e) pressure and flow measure points; and f) valves for leaks emulation. The downstream pressure transducer is located 4.5 m from the end of the pipe, which is open to the atmospheric pressure. The prototype is monitored and regulated by a personal computer (PC) attached to a data acquisition system, which monitors the variables of flow rate and pressure, actuates the servo valves and modifies the hydraulic pump velocity. The personal computer, the data acquisition system and the program in the PC compose a SCADA system.

Table 7.1: Pilot pipeline parameters

Parameters	Values	Description
D	0.076 m	Inner diameter
L	163.715 m	Length
g	9.81 m/s ²	Gravity acceleration
a	1330 m/s	Pressure wave velocity
ν	10 ⁻⁶ [m ² /s] at 20° C	Fluid kinematic viscosity
ϵ	0.2 mm	Rugosity
Material		Schedule 40 galvanized iron

The mathematical model that describes the flow behavior in a pipe was developed from the analysis of a straight pipe. Because of space limitations, however, the prototype is an array of pipes in a coil shape on a vertical plane. This array is not a drawback since from this array one can have an equivalent in the shape of a straight horizontal pipe by using the height relative to a reference level (Section 7.2.2).

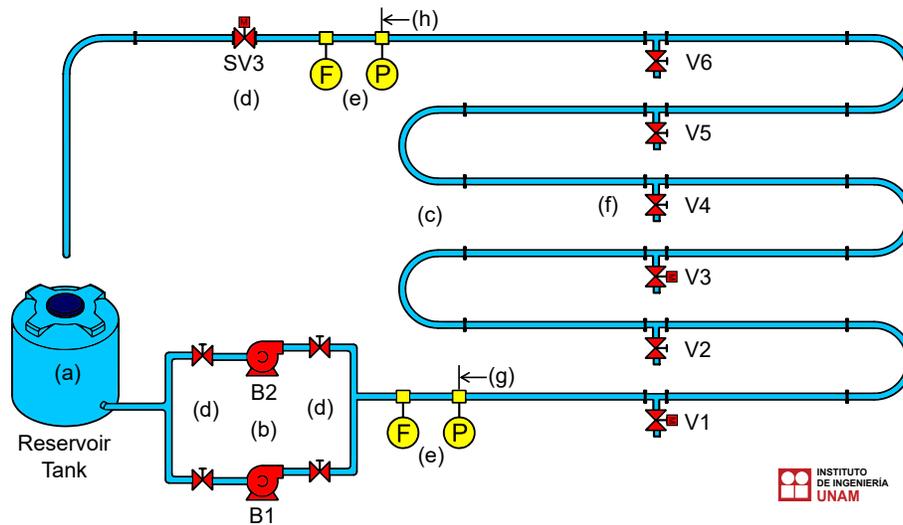


Fig. 7.2: Schematic of the prototype: a) water reservoir, b) hydraulic pumps, c) pipe, d) input and output valves, e) flow and pressure sensors, f) leak valves, g) beginning of the pipe and h) end point

The part of the pipe useful for fault detection is the coil between measurement instruments. These two points, (g) and (h) (see Fig. 7.2), are the marks of the beginning and ending of the monitored pipe. There are also six points where leaks can be created; these points are composed of 2.54 cm diameter iron pipes welded to the main pipe and ball valves that regulate the leak flow (valves V1 to V6, Fig. 7.2). Two of these valves are servo valves (V1 and V3) that are manipulated in the SCADA-II in order to synchronize time and set the size of the leak.

7.2.2 Topographic profile and pressure modification

The prototype has the layout shown in Fig. 7.2. As the area and conduit material are constant along the pipe, then the pressure head (Bansal, 2005) is

$$\tilde{H}(z) = \frac{P(z)}{\rho g}, \quad (7.1)$$

where $P(z)$ is the manometric pressure at point z , ρ is the fluid density and g the gravity constant. Furthermore, (7.1) can be written in terms of piezometric head $H(z) = \tilde{H}(z) + h(z)$ where $h(z)$ is the pipe height over any level reference, and z is the spatial coordinate along the pipe.

To handle this pilot plant as it were a horizontal plane conduit, the whole length L is calculated by adding the length of the straight and u-shape sections, and by taking the pressure as the water column height that is achieved by transforming the pressure head $\tilde{H}(z)$ according to the reference level height $h(z)$ with $h(0) = 0$ m, at the inlet pipe (Sieber and Isermann, 1977). The last assumption is the same as adding gravity pressure (caused by the height difference to the reference level) to the pressure head at any point z of the line. Experimental studies have shown that for the specific parameters of the pilot plant the model error produced by the above assumptions is neglected (Carrera and Verde, 2001).

7.2.3 Sensors and equipment

In this subsection, the main features of the sensors are described, as well as the equipment used in the facility: hydraulic pumps, electronic power inverters that drive the pumps, servo valves, data acquisition system (DAQ) and personal computer (PC).

Flow sensors: Promass 83F from Endress + Hauser (E+H, 2018b), sensor based on the Coriolis effect, $\pm 0.05\%$ maximum error

Pressure sensors: Cerabar PMO71 from Endress + Hauser (E+H, 2018a), piezoresistive sensor, $\pm 0.05\%$ maximum error

Inverters: Frequency variators F800 and E740 from Mitsubishi (Mitsubishi, 2018)

Centrifugal pumps: Siemens 10 HP and 7.5 HP (Barmesa, 2018)

Data interface: compactRIO (NI, 2018) with A/D and D/A converters

Servo valves: Georg Fischer type 133 (GF, 2018), and Remote Control RCE 15, (RC, 2018)

Personal computer: Panasonic Toughbook CF-53 (Panasonic, 2013), 64-bit processor, 2.5 GHz clock, 446 GB HD and 4 GB RAM

All sensors, servo valves and inverters are connected to the data interface by means of 4-20 mA current loops. The PC communicates with the data interface via an USB connector. More details are given in Subsection 7.4.2.

7.3 Diagnosis Model for Two Sequential Leaks

The transient flow model in pressurized pipes is based on the conservation laws of momentum and mass that govern transient flows. The partial differential equations:

$$\begin{aligned} \frac{1}{a_1} \frac{\partial Q(z,t)}{\partial t} + \frac{\partial H(z,t)}{\partial z} + J(Q(z,t), \theta_1) &= 0, \\ \frac{\partial H(z,t)}{\partial t} + a_2 \frac{\partial Q(z,t)}{\partial z} &= 0 \end{aligned} \quad (7.2)$$

describe a fluid in a one dimensional straight line of one-dimension in the absence of leaks and are taken from Chaudhry (2014), where $H(z,t)$ and $Q(z,t)$, respectively, denote the piezometric pressure head and the flow rate at the center of the pipe for a pipe of length L . Furthermore, $z \in [0, L]$ is the spatial variable, and $t \geq 0$ is the time variable. The parameters are $a_2 = a^2/a_1$ with a the pressure wave speed, $a_1 = gA_r$ with A_r the constant cross-sectional area and g is the gravitational acceleration. The hydraulic gradient function per unit length, $J(Q(z,t), \theta_1) = \theta_1 Q^2$, depends on specific friction approximations, and here a quadratic function is used, where θ_1 is the parameter associated with the pressure gradient. Model (7.2) is developed by assuming elastic pipes and weakly compressible fluids with a flow rate significantly lower than the pressure wave speed. This model is the base of diverse leak diagnosis methods by using mass balance and the hydraulic gradient of the fluid. One can consult the reference Verde and Torres (2017), where diverse methods are introduced.

A leak at point L_e on the pipe is characterized by:

$$\begin{aligned} Q(L_e - \varepsilon, t) &= Q(L_e + \varepsilon, t) + \mu(t - \tau)Q_e(t), \\ H(L_e - \varepsilon, t) &\approx H(L_e + \varepsilon, t) \end{aligned} \quad (7.3)$$

with ε an infinitesimal value close to 0, and the time τ defines the initialization of the leak with μ as the step function. The outflow rate induced by the piezometric head at L_e is given by:

$$Q_e(t) = \lambda_e \sqrt{H_e(L_e, t)}, \quad (7.4)$$

where the leak coefficient λ_e depends on the discharge coefficient, the leak cross section area s_a and the gravitational acceleration g . Thus, the leak conditions (7.3) and (7.4) must be considered at each point where a leak is present in the pipeline.

To evaluate the SCADA-II system, which is the subject of this chapter, a MSL (Multiple Sequential Leaks) method developed at the Instituto de Ingeniería has been integrated to the data acquisition system. From an implementation point of view, the main advantages of the method are

- the high velocity of the location of the leak since the dimension of the estimation algorithms is of minimum order and independent of the number of leaks.
- the continuous identification of the gradient parameter simultaneously with the leaks estimation during the monitoring task.

The reader interested in the detail of the algorithm derivation can consult Rojas (2021). Here, only the main details are given. A key result to simplify the MSL problem is the set of equations obtained from Eqs. 7.2, 7.3 and 7.4, which characterizes the models equivalence, in steady state, when there are two leaks and one leak, with the friction as a function of the flow rate by sections (Verde and Rojas, 2017). Figure 7.3 describes variables and parameters of the pipelines used to characterize the equivalence relationships between one and two leaks using an input-output model.

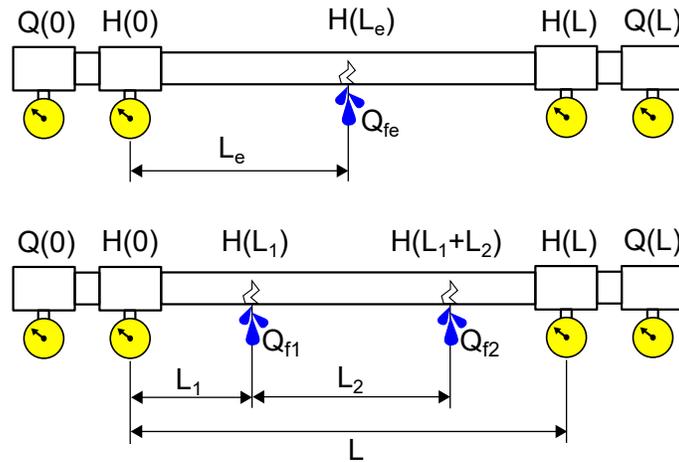


Fig. 7.3: Variables description of the equivalent models in steady state for the pipeline with one leak in $z_e = L_e$ and with two leaks ($z_1 = L_1$ and $z_2 = L_1 + L_2$)

Fact 1: Let the two leaks scenarios be in steady state as shown in Fig. 7.3 where both pipelines have the same physical characteristics (length, diameter, fluid and roughness). Assume that the outflow Q_e is

Table 7.2: Fluid Models by Sections with one and two Leaks

Leak L_e	Leaks L_1 y L_2
Upstream Section	
$z_{1e} \in [0, L_e]$	$z_1 \in [0, L_1]$
$\frac{1}{a_1} \frac{\partial Q(z_{1e}, t)}{\partial t} + \frac{\partial H(z_{1e}, t)}{\partial z_{1e}} + J(Q(z_{1e}, t), \theta_1) = 0$ $\frac{\partial H(z_{1e}, t)}{\partial t} + a_2 \frac{\partial Q(z_{1e}, t)}{\partial z_{1e}} = 0$	$\frac{1}{a_1} \frac{\partial Q(z_1, t)}{\partial t} + \frac{\partial H(z_1, t)}{\partial z_1} + J(Q(z_1, t), \theta_1) = 0$ $\frac{\partial H(z_1, t)}{\partial t} + a_2 \frac{\partial Q(z_1, t)}{\partial z_1} = 0$
Boundary Condition	Boundary Condition
$H(0, t)$ $Q(L_e - \varepsilon, t) = Q(L_e + \varepsilon, t) + Q_e(t)$ $H(L_e - \varepsilon, t) = H_L(L_e + \varepsilon, t)$	$H(0, t)$ $Q(L_1 - \varepsilon, t) = Q(L_1 + \varepsilon, t) + Q_1(t)$ $H(L_1 - \varepsilon, t) = H(L_1 + \varepsilon, t)$
Middle Section	
-	$z_2 \in [L_1, L_1 + L_2]$
-	$\frac{1}{a_1} \frac{\partial Q(z_2, t)}{\partial t} + \frac{\partial H(z_2, t)}{\partial z_2} + J(Q(z_2, t), \theta_1) = 0$ $\frac{\partial H(z_2, t)}{\partial t} + a_2 \frac{\partial Q(z_2, t)}{\partial z_2} = 0$
-	Boundary Condition
-	$Q(L_1 + L_2 - \varepsilon, t) = Q(L_1 + L_2 + \varepsilon, t) + Q_2(t)$ $H(L_1 + L_2 - \varepsilon, t) = H(L_1 + L_2 + \varepsilon, t)$
-	
Down Stream Section	
$z_{2e} \in [L_e, L]$	$z_3 \in [L_1 + L_2, L]$
$\frac{1}{a_1} \frac{\partial Q(z_{2e}, t)}{\partial t} + \frac{\partial H(z_{2e}, t)}{\partial z_{2e}} + J(Q(z_{2e}, t), \theta_1) = 0$ $\frac{\partial H(z_{2e}, t)}{\partial t} + a_2 \frac{\partial Q(z_{2e}, t)}{\partial z_{2e}} = 0$	$\frac{1}{a_1} \frac{\partial Q(z_3, t)}{\partial t} + \frac{\partial H(z_3, t)}{\partial z_3} + J(Q(z_3, t), \theta_1) = 0$ $\frac{\partial H(z_3, t)}{\partial t} + a_2 \frac{\partial Q(z_3, t)}{\partial z_3} = 0$
Boundary Condition	Boundary Condition
$H(L, t)$	$H(L, t)$

related to the outflow rates of line 2 by

$$Q_e = Q_{f1} + Q_{f2} \quad (7.5)$$

with Q_{f1} and Q_{f2} as the outflows of pipeline 2, respectively. Thus, the differences between both lines are described next:

Pipeline 1. A leak with flow rate Q_e at L_e in line 1, and

Pipeline 2. Two leaks with flow rates Q_{f1} and Q_{f2} at coordinates L_1 y $L_1 + L_2$, respectively

Considering the segments between leaks of Fig. 7.3, each fluid can be described by equations of the form (7.2) and its respective boundary conditions. Thus, the models shown in Table 7.2 were obtained for each section.

If both flows are in equilibrium conditions, from the models of Table 7.2, the relation between the leaks positions and the hydraulic gradients of the three sections is given by

$$\left(L_e - L_1\right)\left(J(Q(0), \theta_1) - J(Q(L), \theta_1)\right) = L_2\left(J(Q(0) - Q_{f1}, \theta_1) - J(Q(L), \theta_1)\right) \quad (7.6)$$

for the two pipelines are obtained. This equation is an input-output equivalent model for two and one leak and can be interpreted as pressure indistinguishable profiles for both lines at their ends, respectively. These indistinguishability regions justify the necessity of a transitory regime to locate simultaneous leaks. In terms of the parameter λ_1 and the piezometric pressure head at the ends of the pipeline, the distance between the first and second leak can be written as

$$L_2 = \frac{H(0) - H(L) - L_1 J(Q(0), \theta_1) - (L - L_1) J(Q(L), \theta_1)}{\theta_1 \left(Q(0) - \lambda_1 \left[H(0) - L_1 J(Q(0), \theta_1) \right]^{1/2} \right)^2 - J(Q(L), \theta_1)}. \quad (7.7)$$

Moreover, by considering (7.5), the parameters associated with the second leak can be expressed as

$$\lambda_2 = \frac{Q(0) - Q(L) - \lambda_1 \left(H(0) - L_1 J(Q(0), \theta_1) \right)^{1/2}}{\left(H(L) + (L - L_1 - L_2) J(Q(L), \theta_1) \right)^{1/2}}. \quad (7.8)$$

Finally, from the hydraulic gradient along both lines the inequality

$$z_1 = L_1 < L_e < z_2 = L_1 + L_2 \quad (7.9)$$

can be validated. In other words, the positions of the leaks in a two leaks scenario are upstream and downstream of the equivalent leak position L_e .

Thus, if the pair (L_1, λ_1) together with the equivalent leak position L_e are assumed to be previously known, one can estimate the rest of the unknown variables of the two leaks pipeline from (7.6), (7.7), (7.8) and the condition imposed by the inequality (7.9). This means L_2 , λ_2 and Q_{f2} can be estimated.

Therefore, by considering the above relationships and that only one leak occurs at the same time, an iterative algorithm can be implemented as part of the monitoring system. The algorithm will be activated each time a new equivalent appears and the equivalent leak (L_e, λ_e) is estimated. The values of this estimation together with the historical leakage data of the line given by L_i and λ_i then allow the estimation of the values associated with the leak $i + 1$. The following subsection addresses a proposed solution for the estimation of sequential leaks with small occurrence times between them.

7.3.1 Sequential leak location algorithm

By assuming that the physical parameters of the duct are known and the data of the flow rates and piezometric pressures at the ends of the line $(Q(0), Q(L), H(0), H(L))$ are available, the algorithm for the location of two sequential leaks on-line is described here. Since the friction parameter uncertainty has been recognized in experimental studies (Billman and Isermann, 1987; Rojas et al., 2021), the diagnostic algorithm is robustified by including an adaptive observer for the identification on-line of the gradient

$J(Q, \theta_1)$. Moreover, to reduce the number of false leaks, an innovative redundant approach for symptom generation is also incorporated. The equivalent model

$$\begin{aligned}\dot{Q}(0) &= \frac{a_1}{L_e} \left(H(0) - H(L_e) \right) - a_1 J(Q(0), \theta_1), \\ \dot{H}(L_e) &= \frac{a_2}{L_e} \left(Q(0) - Q(L) - Q_e(t) \right), \\ \dot{Q}(L) &= \frac{a_1}{L - L_e} \left(H(L_e) - H(L) \right) - a_1 J(Q(L), \theta_1)\end{aligned}\quad (7.10)$$

is used for the identification of the pair (Q_e, L_e) with $L > L_e > 0$, or of the parameter θ_1 , according to the assigned task.

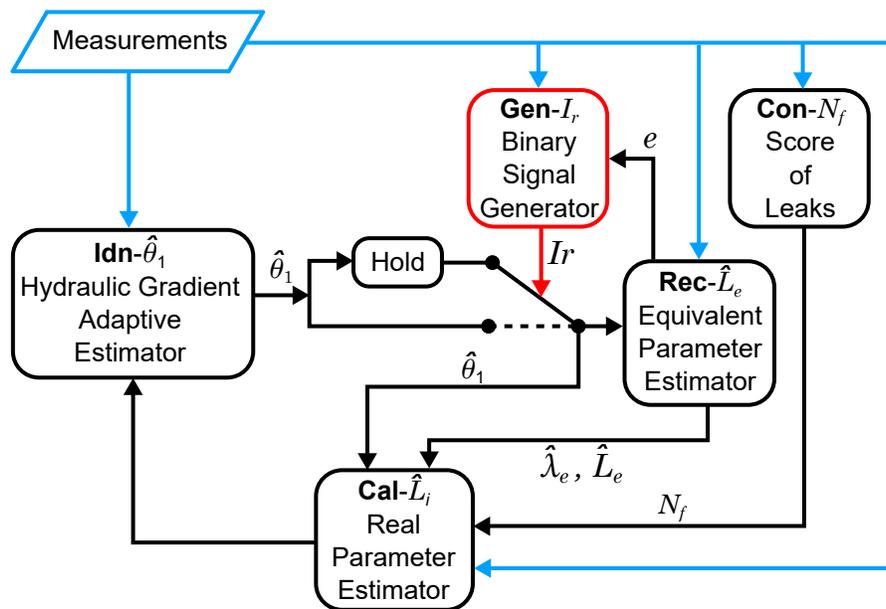


Fig. 7.4: Diagnosis system scheme with main tasks **Gen** – I_r , **Rec** – \hat{L}_e , **Idn** – θ_1 , **Cal** – \hat{L}_i , **Con** – N_f , and the holder and the switch

The algorithm consists mainly of coupled estimation tasks driven by the symptom or binary signal

$$I_r = \begin{cases} 0 & \text{if the system is awaiting the next leak} \\ 1 & \text{if the leak parameters are being estimated,} \end{cases}$$

as a function of the data set and residual state. Figure 7.4 shows the interconnection of the estimators where the prescribed tasks are detailed below.

- **Gen** – I_r . Generation of the binary signal I_r that governs the state of the algorithm according to the following parallel conditions of the flow rates
 - Condition 1 is active, if the convergence error given by $S_{EKF} = |e_1| + |e_2|$ of the EKF (Extended Kalman Filter) with $e_1 = Q(0) - \hat{Q}(0)$ and $e_2 = Q(L) - \hat{Q}(L)$ is greater than the positive threshold T_{EKF} .

- Condition 2 is active, if the difference of the flow rates at the ends $r = Q(0) - Q(L)$ is greater than the positive threshold th .

Thus, if both conditions are not satisfied, I_r is off and $\hat{\theta}_1$ is calculated over and over in (7.10). In addition, the *EKF* simultaneously generates e_1 and e_2 . If both conditions are active, $I_r = 1$, and the value of $\hat{\theta}_1$ is held. Furthermore, the *EKF* identifies the new equivalent parameters L_e and λ_e . Both thresholds of the above conditions are adjusted according to Isermann (2006).

- **Idn** – $\hat{\theta}_1$. Identification of $\hat{\theta}_1$ with known L_e . This task is feasible according to Besançon (2007) by assuming unknown signals $H(L_e)$ and θ_1 . Defining the output $y = (Q(0) \ Q(L))^T$, the input $u = (H(0) \ H(L))^T$, and the unknown constant vector $\Theta = (\theta_1 \ H(L_e))^T$ the adaptive observer assumes the model (7.10) transformed to the canonical observer form

$$\dot{y} = \alpha(y, u, t) + \beta(y, t)\Theta \quad (7.11)$$

with $\alpha(y, u, t) = \begin{pmatrix} \frac{a_1}{L_e} u_1 \\ -\frac{a_1}{L-L_e} u_2 \end{pmatrix}$ and $\beta(y, t) = \begin{pmatrix} -a_1 y_1^2 & -\frac{a_1}{L_e} \\ -a_1 y_2^2 & \frac{a_1}{L-L_e} \end{pmatrix}$. Thus, according to Besançon (2007), the adaptive system

$$\begin{aligned} \dot{\hat{y}} &= \alpha(\hat{y}, u, t) + \beta(\hat{y}, u, t)\hat{\Theta} - k_y(\hat{y} - y); \text{ for any scalar } k_y > 0 \\ \dot{\hat{\Theta}} &= -k_\theta \beta(\hat{y}, u, t)^T (\hat{y} - y); \text{ for any scalar } k_\theta > 0, \end{aligned} \quad (7.12)$$

which includes Θ as a new state, is an observer of (7.11). Furthermore, if β is persistently exciting (Narendra and Annaswamy, 1987) and $\dot{\beta}$ is bounded, the parameter error also goes to 0. Thus, the observer (7.12) is implemented for the estimation of θ_1 .

- **Rec** – \hat{L}_e . Reconstruction of the equivalent parameters L_e and λ_e for the equivalent system (7.10) with a known constant value of $\hat{\theta}_1$ by applying an *EKF*. The *EKF* is designed by considering that the dynamics of $H(L_e)$ is faster than the dynamics of $Q(0)$ and $Q(L)$. This assumption allows the replacement of the second state in (7.10) by $\dot{H}(L_e) = 0$ with the constraint

$$0 = \hat{Q}(0) - \hat{Q}(L) - \hat{\lambda}_e \sqrt{\hat{H}(\hat{L}_e)}. \quad (7.13)$$

Therefore, the identification task can be solved with the augmented model

$$\begin{aligned} \dot{x} &= f(x, u, \hat{\theta}_1), \\ y &= Cx \end{aligned} \quad (7.14)$$

with

$$x = (Q(0) \ Q(L) \ H(L_e) \ L_e)^T, \quad C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \text{ and}$$

$$f(x, u, \hat{\theta}_1) = \begin{pmatrix} \frac{a_1}{x_4}(u_1 - x_3) - a_1 J_s(x_1, \hat{\theta}_1) \\ \frac{a_1}{L-x_4}(x_3 - u_2) - a_1 J_s(x_2, \hat{\theta}_1) \\ 0 \\ 0 \end{pmatrix},$$

where only the unknown constant parameter L_e is added to (7.10). Therefore, according to Reif et al. (1998), the EKF algorithm for the continuous nonlinear system (7.14) is reduced to

$$\begin{aligned}\dot{\hat{x}} &= f(\hat{x}, u, \hat{\theta}_1) + K(t)(y - \hat{y}), \\ \hat{y} &= C\hat{x},\end{aligned}\tag{7.15}$$

where $K(t) = P(t)C^T R^{-1}$ and $P(t)$ is the solution of the Riccati equation:

$$\dot{P}(t) = (A(t) + I\eta)P(t) + P(t)(A(t) + I\eta)^T + Q_R - P(t)C^T R^{-1}CP(t),\tag{7.16}$$

with the time-varying matrix $A(t) = \frac{\partial f(x,u,t)}{\partial x}$ evaluated at \hat{x} , Q_R and R are the noise covariance matrices. When the error

$$e = y - \hat{y}$$

converges to 0, the estimated value \hat{L}_e and $\hat{\lambda}_e$ are used for the last leak evaluation.

- **Con** – **N_f**. Leak counter. The score N_f is initialized to 0 and is incremented by 1 each time a new leak occurs. The task is achieved by searching for a positive rising edge of $(Q(0) - Q(L))$.
- **Cal** – **L_i**. Evaluation of the leak parameter L_i , after the filter convergence, by the static relations (7.6), (7.7) and (7.8) and considering the score N_f .

Since the static relations of **Cal** – **L_i** and that the switch driven by the symptom I_r separate the signals coming from the functions **Idn** – **θ₁** and **Rec** – **L_e**, respectively, the estimators of each task can be independently designed. Therefore, by assuming that the algorithm starts under healthy conditions, the MSL procedure is accomplished as follows.

The identifier **Idn** – **θ₁** transfers the parameter $\hat{\theta}_1$ to the estimator **Rec** – **L_e**, which estimates the system state (7.14) in normal condition. This condition remains as long as the symptom I_r is off. At time t_f , when I_r switches on, the *EKF* estimates the equivalent leak L_e and λ_e with the fixed parameter $\hat{\theta}_1$ estimated at t_f , and the score increases by 1. Moreover, the estimated equivalent leak parameters are supplied to the function **Cal** – **L_i**, which calculates the real parameter of the leak. Once the error S_{EKF} is less than T_{EKF} , the **Idn** – **θ₁** hold function is off, and the actual $\hat{\theta}_1$ is used by the *EKF*. In addition the algorithm is again ready to catch another leak.

From an operation point of view, the algorithm works in three states.

- **Detection State**. By using (7.12) with L_e and λ_e known, $\hat{\theta}_1$ is estimated, **Gen** – **I_r** simultaneously generates I_r and the *EKF* error remains close to 0. This stage remains so long as the symptom is off. When I_r turns on, $\hat{\theta}_1$ is held, and the algorithm is switched to the Location State.
- **Location State**. By using (7.15) with constant $\hat{\theta}_1$ and adding L_e and λ_e as states, the new equivalent leak is estimated. When the parameters converge, the residual I_r turns off, and the update stage starts.
- **Update State**. By using the equivalent parameters of the location stage, L_e and λ_e , the physical parameters of the last leak are calculated by (7.7), (7.8) and (7.6). Moreover, the leak score is increased by 1, and the leak parameters are stored. Thus, the algorithm returns to the detection stage, and it is ready to detect the next leak.

7.4 SCADA-II System for Sequential Leaks' Location

Let us begin from the whole SCADA-II system and finish with the main details. First of all, we have two main parts in this system (in Fig. 7.5); the pipe prototype and the SCADA (supervisory control and data acquisition) composed of the data acquisition interface and the PC with the FDI program in it.

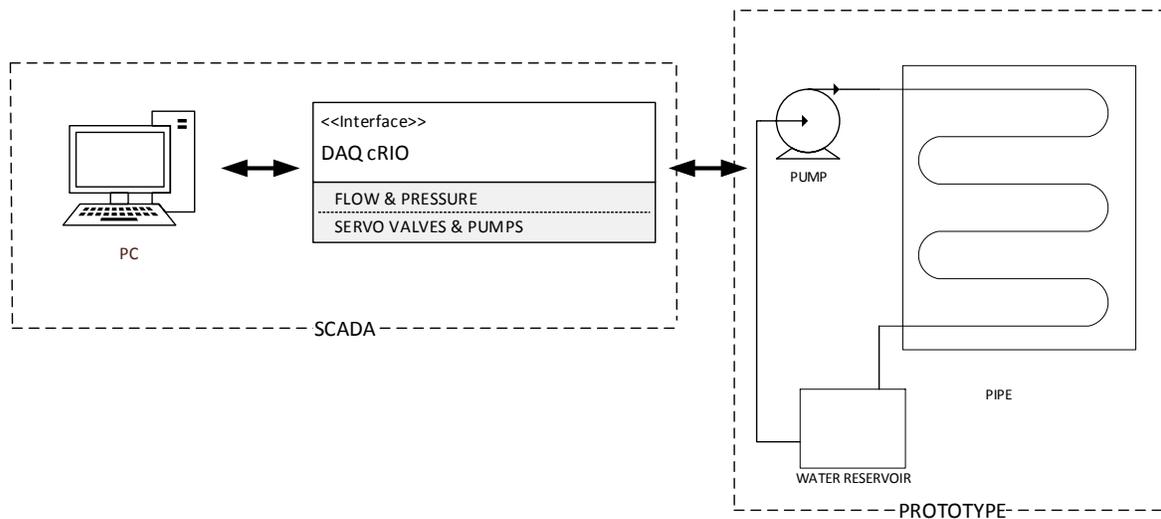


Fig. 7.5: Layout of the SCADA-II system. Arrows mean data interchange. The PROTOTYPE dashed block corresponds to the scheme of Fig. 7.2

As presented in Section 7.2, the prototype permits establishing the experiment conditions such as set point flow rate in the pipe and which valve should be opened, as well as when and how much. Those are the experiment conditions, but simultaneously there are flow rate and pressure sensors at the beginning and end of the pipe. In addition, these sensors feed the SCADA with the actual flow and pressure data, for FDI goals.

Analog data coming from the sensors need to be equal to the data processed in the computer. In order to do that, an interface is needed that converts electrical signals (4-20 mA) to binary data and vice versa. This interface is the DAQ cRIO shown in Fig. 7.5. The interface or data acquisition system (DAQ) is integrated by a CompactRIO (cRIO) controller and two I/O (input/output) modules. Additionally, by means of this interface the SCADA commands are sent to servo valves and pumps.

The SCADA-II, Fig. 7.5, is a LabVIEW program on a personal computer. This program handles signals from and to the pipe system and runs the algorithm that detects the leaks and localizes them.

7.4.1 LabVIEW programming as a tool for SCADA-II implementation

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a design platform that uses a visual programming language. This visual programming, called graphical language G, was developed by National Instruments (National-Instruments, 2021), and it is mainly used for data acquisition, instrument

control and industrial automation. One of its characteristics is the data flow programming: when all data is available all the subroutines (called sub VI, where VI is the acronym for virtual instrument) or functions are executed. The graphical programming consists of wires that connect nodes creating a program called virtual instrument (VI); a node can be another sub VI or a function. LabVIEW has a whole set of functions available to the programmer. A VI has three components: a front panel, block diagram and a connector pane. The front panel is the user interface and has controls and indicators (graphs or displays). The block diagram or back panel is the graphical program associated with the source code. The connector pane has the input-output signals or data that may be connected to other VIs, controls or indicators. To emulate real-time with Labview, a VI must be executed with very small time intervals.

7.4.2 Data acquisition system

Figure 7.6 shows the data flow of the proposed SCADA-II. The kernel of the data acquisition system is the cRIO system that receives signals from sensors, sends signals to the inverter and the servo valves and communicates with the HMI (human machine interface) in the host computer (PC).

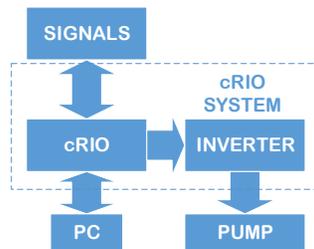


Fig. 7.6: cRIO data and signals' layout of the SCADA-II

The data acquisition system is composed of the modules NI9203 and NI9265 connected to the cRIO-9067 controller. The NI9203 module is an 8-channel current to binary data converter, and the NI9265 module is an 8-channel binary to current converter. The current loops range from 4 to 20 mA. Communication with the PC can be achieved via USB or Ethernet connector.

Figure 7.7 shows the cabinet with the inverter, the cRIO controller, the IO modules, the terminal block connector and voltage source. IO modules are connected to the controller via an internal bus and are also connected to the external devices via the terminal block connector.

7.4.3 The SCADA-II program

The main part of the SCADA-II is the software: the main program. The SCADA-II system is composed of a physical phenomenon -flow in a pipe- with sensors that retrieve the phenomenon behavior and actuators that change setpoints. All these parts, however, are nothing without a program that organizes the entire operation of the system. In the next section, we explain what the user can see and do by means of this program.

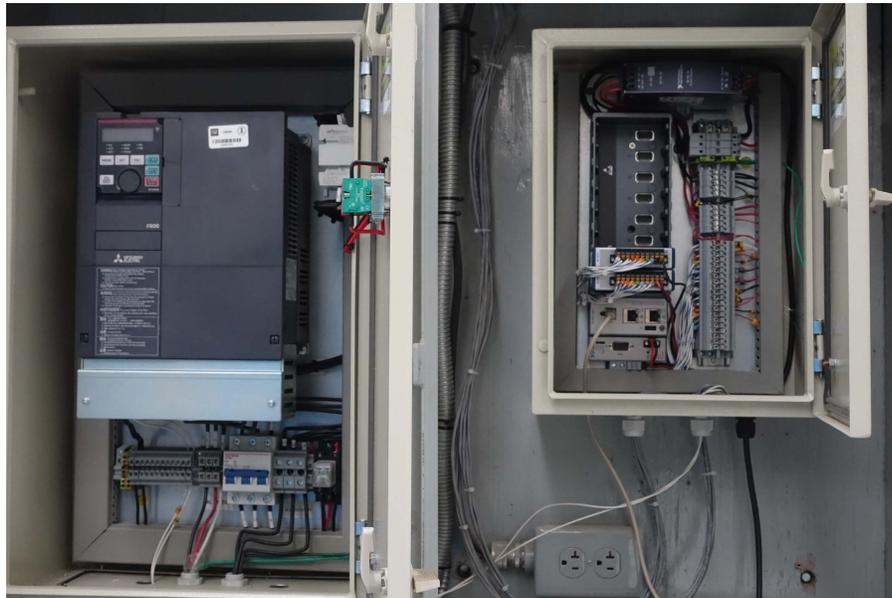


Fig. 7.7: Hardware: Inverter on the left and cRIO controller, modules and terminal block connector on the right

7.4.3.1 SCADA-II user interface

Communication between the user and the main program takes place on the user interface or HMI shown in Fig. 7.8. The screen is divided into three windows: Process **Commands**, Data **Monitoring** and **Graphs** of the estimation. The program runs with a sampling period of 100 ms.

Commands are as follows: sampling period, pump frequency, opening of the servo valve *SV1*, opening of the servo valve *SV2* and the stop button. The sampling period in milliseconds is established prior to the system operation, by considering the time response of the system that is approximately 5 s. The aperture of servo valves can be changed any time from 0 to 100%. Pushing the stop button ends the operation, and the historic data sheet is then saved in the cRIO controller HD prior to stop main program.

The number of leaks, the time elapsed and the leak alarm are the monitored variables. Each time a new leak is detected the variable **Leak No** is incremented. The elapsed time corresponds to the time recorded from the beginning of the operation until the operation ends. The leak button is a luminous alarm that turns red when a leak appears and remains red as long as a leak exists.

In the graphs window of the screen, there are two columns. In the left column pressures, flow rates at the ends and servo valves apertures are shown. In the right column, estimated and calculated data are shown: the θ parameter associated with the hydraulic gradient, the parameters relative to the leak size and of position λ_e and L_e , respectively, and the position of the leaks z_1 and z_2 . To have a steady screen, the user interface (HMI) is actualized each 5 periods, that is each 0.5 s.

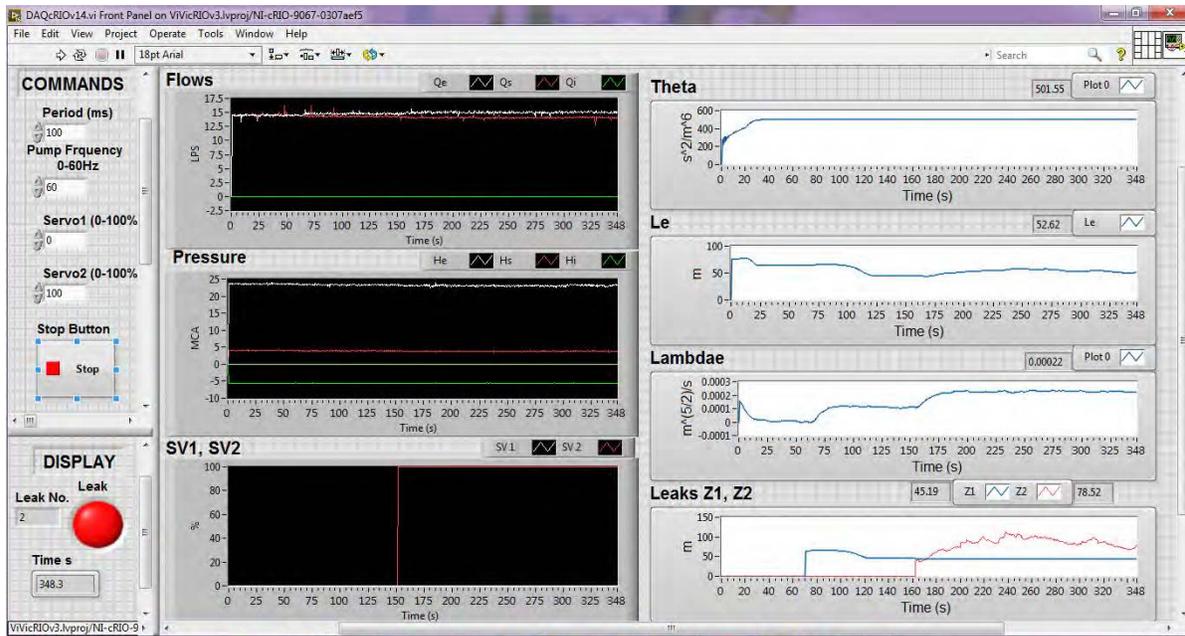


Fig. 7.8: PC screen shot of the SCADA-II HMI with leaks estimated at z_1 and z_2

7.4.4 Main program

Behind the user interface, there is a main program written in LabVIEW language. Figure 7.9 shows an overall view of the flowchart for the SCADA-II program. This describes the set of activities that take place during the process and the sequence in which they are accomplished. Furthermore, it can be divided in two main activities: 1) those that occur at the beginning of and during the operation and 2) those that correspond to parameter identification and leaks' location. In the following, the states of the diagram will be thoroughly explained.

7.4.4.1 Operation setup

The program begins with setting initial values of variables and constants, and the user establishes the sample time period. This time period is related to the time of the loop that contains the main program. The block *if* of the flowchart shown in Fig. 7.10 (a) indicates when the loop execution starts. Figure 7.10(b) shows the flowchart of the program that interacts with the user. The program can be ended any time when the user presses the stop button. Right after pressing the button, all data recorded are downloaded to a data sheet that will be saved on the hard disk in the cRIO controller. If the program continues, the hydraulic pump velocity can be changed any time in the interval from 30 to 60 Hz. The pipeline prototype has two servo valves that can be actuated from the program in the range from 0 to 100% of aperture.

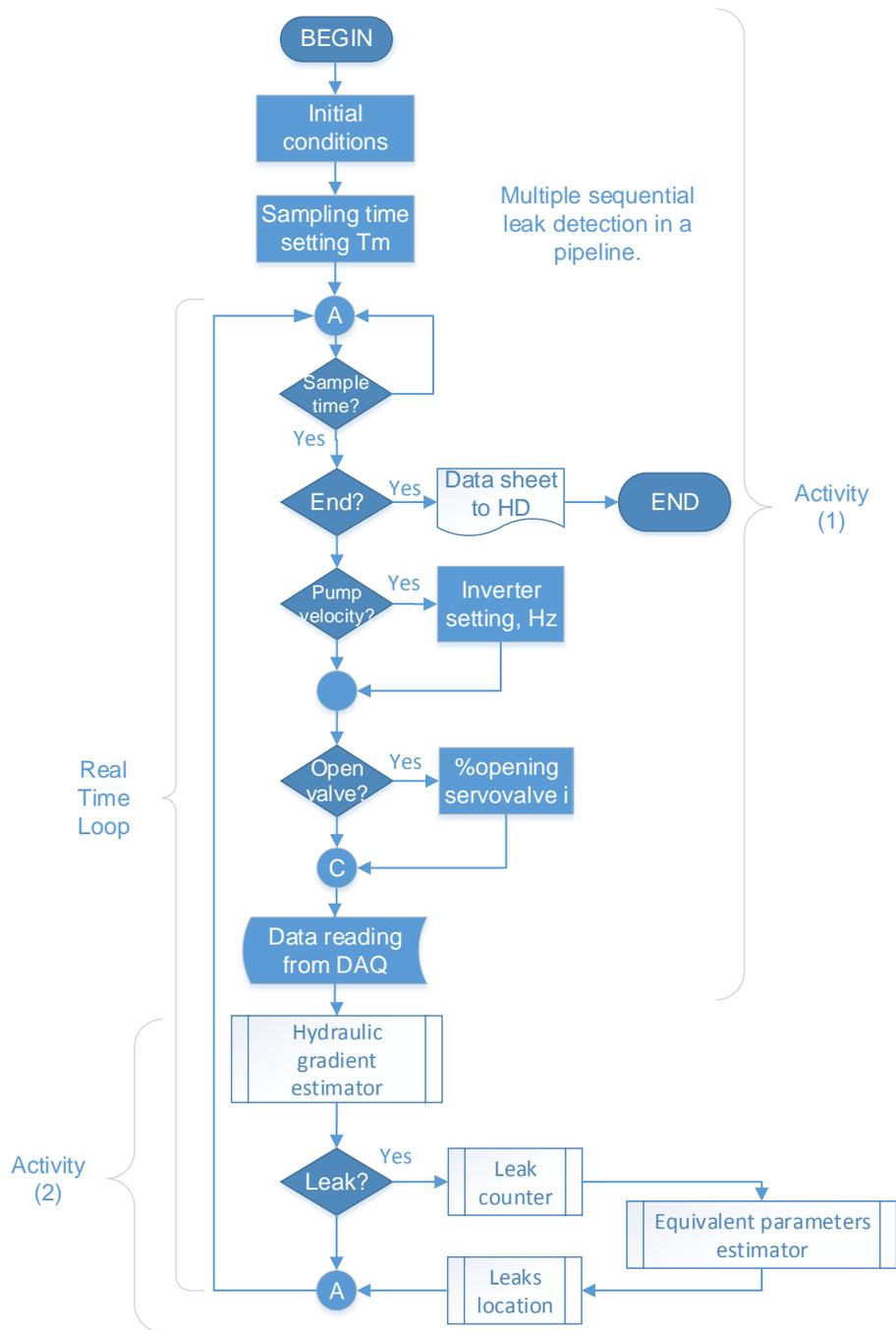
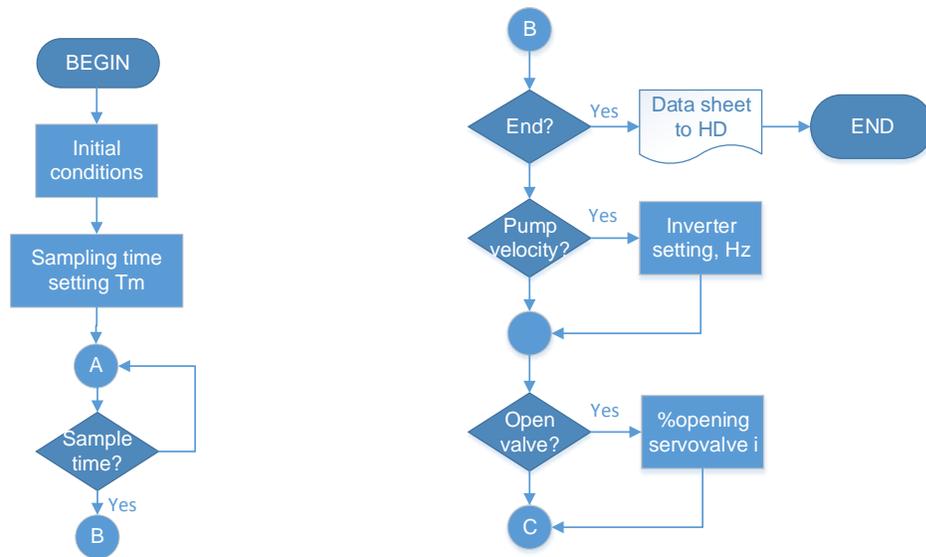


Fig. 7.9: Flowchart of the SCADA program

7.4.4.2 Parameter identification and leaks location

Figure 7.11 shows the flowchart of the main program for leak diagnosis. It begins with the data reading from the data acquisition system (DAQ). The reading process is shown in Algorithm 4. This is a subrou-



(a) Setting parameters and initial conditions (b) Operation point, pump and valves settings

Fig. 7.10: Flowcharts of subprograms: setting operating data and user interaction

tine that runs in the FPGA (field programmable gate array) area of the cRIO controller, where reading and writing data are available in fact. AO_i are the analog outputs, and AI_i are the analog inputs.

Algorithm 4 Target program in FPGA: sensors reading and commands to inverter and servo valves

Input: $T_s, AO_i, i = 1, \dots, 4$

Output: $A_i, i = 1, \dots, 6$

Set rank (0-20 mA) in AIs

for k do

$[AI_1, \dots, AI_6](k) \rightarrow FIFO$

$AO_0(k) \leftarrow F_{pump}$

$AO_1(k) \leftarrow Servo \ 1$

$AO_2(k) \leftarrow Servo \ 2$

$AO_3(k) \leftarrow Servo \ OUT$

end for

The second block of the flowchart shown in Fig. 7.11 is the *hydraulic gradient estimator*, that is characterized by $(J(Q, \theta_1) = \theta_1 Q^2)$ described in Section 7.3. The sub VI that performs this task is shown in Fig. 7.12, and Algorithm 5 shows the parameter estimation implementation.

Returning to the flowchart shown in Fig. 7.11, the next block is the *Leak* condition; if there is no leak, the flow returns to node A and waits for a new sample period. If there is a leak, a leak counter and location loop follows. The system identifies a leak when residual $r(t)$ is greater than its threshold th .

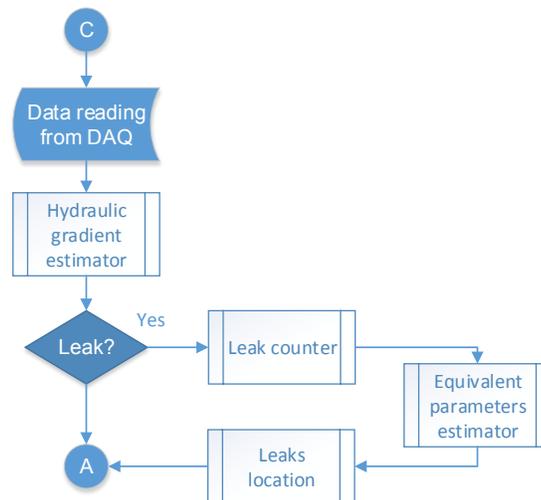


Fig. 7.11: Flowchart of parameter estimation and leaks location

Algorithm 5 Pressure gradient parameter estimation $\hat{\theta}$ (7.12)

Input: $H_0, H_L, Q_0, Q_L, I_r, L_e$
Output: $\hat{\theta}_1$

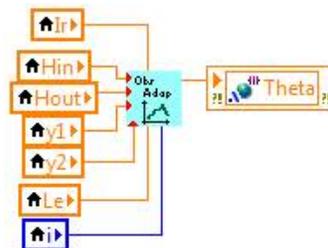
 Set k_y, k_θ
if $I_r = 0$ **then**

$$\hat{y} = \alpha(\hat{y}, u, t) + \beta(\hat{y}, u, t)\hat{\theta} - k_y(\hat{y} - y)$$

$$\hat{\theta} = -k_\theta \beta(\hat{y}, u, t)^T (\hat{y} - y)$$

else [$I_r = 1$]

 Hold $\hat{\theta}$
end if

Fig. 7.12: sub VI for θ identification

To avoid false leak alarms, an adaptive threshold for $r(t)$ is implemented. This adaptive threshold is an improvement of proposal $th(t) = a_{t1} + a_{t2}s(t) + a_{t3}|\dot{s}(t)|$, given by Isermann (2011), where $s(t)$ is a reference signal taken from the system, Q_0 is used in this implementation. The improvement here consists of removing the absolute value of the derivative. The reason for the change is because if one considers the flow rate $Q_0(t)$ as the reference signal, $\dot{Q}_0(t)$ could be negative if noise, uncertainties or operation point changes exist. This condition then produces that the threshold still increases the creation

of false alarms. Thus, the novel adaptive threshold of the SCADA-II is given by

$$th(t) = a_{t1} + a_{t2}Q_0(t) + a_{t3}\dot{Q}_0(t), \quad (7.17)$$

with a_{t1} , a_{t2} and a_{t3} empirical values set by a trial and error process. As a consequence, when $r(t) > th(t)$ a binary flag alarm is activated to avoid the increment of the leak counter when no leak occurs.

The loop of estimation of Fig. 7.11 begins with the *Leak counter* block where the residual $r(t)$ is analyzed, and each time a rising edge appears, a new leak has occurred, and the counter L_c , is incremented.

The next block is the *Equivalent parameters estimator* where the EKF is implemented to find the equivalent parameters and the binary signal. The LabVIEW sub VI that performs this task is shown in Fig. 7.13. Algorithm 6 shows the sequence of operations that occur in this VI. It begins by solving a Riccati equation, then finds the equivalent parameters L_e and λ_e and sets the binary signal I_r .

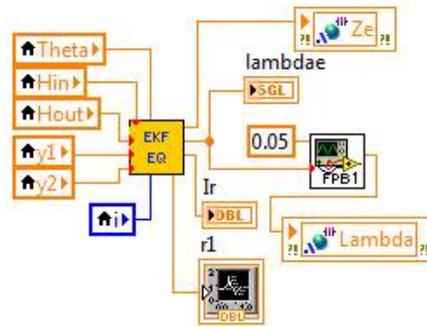


Fig. 7.13: sub VI for equivalent parameter identification by *EKF*

The last block of Fig. 7.11 which is the *Leaks location*, is a process that depends on leak counter value and estimated equivalent parameters and starts assuming the first leak $z_1 = L_1$ as \hat{L}_e . Figure 7.14 shows the sub VI that generates the distance L_2 of the second leak by assuming that first leak is already known and calculates the position of the second leak $z_2 = L_1 + L_2$. The commands that perform the location are shown in Algorithm 7.

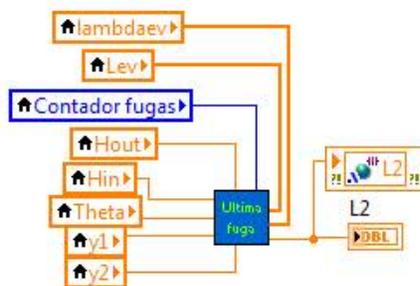


Fig. 7.14: sub VI for leak location

Algorithm 6 L_e and λ_e estimation and I_r setting**Input:** $H_0, H_L, Q_0, Q_L, \theta_1, th$ **Output:** $\hat{L}_e, \hat{\lambda}_e, I_r$ Set Q_R, R, T_{EKF} , $r = Q_0 - Q_L, e_1 = Q_0 - \hat{Q}_0, e_2 = Q_L - \hat{Q}_L, S_{EKF} = |e_1| + |e_2|$ **if** $r > th$ **then** **if** $S_{EKF} > T_{EKF}$ **then** Solves Riccati equation for \dot{P} , Eq. (7.16) $K(t) = P(t)C^T R^{-1}$ Solves \hat{x}, \hat{y} , Eq. (7.15) Solves $\hat{\lambda}$, Eq. (7.13) **else** Hold $\hat{\lambda}, \hat{L}_e$ **end if****else** Hold $\hat{\lambda}, \hat{L}_e$ **end if****if** $r > th$ **then** **if** $S_{EKF} > T_{EKF}$ **then** $I_r = 1$ **else** $I_r = 0$ **end if****else** $I_r = 0$ **end if****Algorithm 7** Leaks location $z_1 = L_1$ and $z_2 = L_1 + L_2$ **Input:** $\hat{L}_e, \hat{\lambda}, L_c, Q_0, H_L, Q_0, H_L$ **Output:** z_1, z_2 **if** $L_c = 0$ **then** $z_1 = 0, z_2 = 0$ **else** **if** $L_c = 1$ **then** $z_1 = \hat{L}_e$ **else** **if** $L_c = 2$ **then** Solves Eq. (7.6) for L_2 $z_2 = z_1 + L_2$ **end if** **end if****end if**

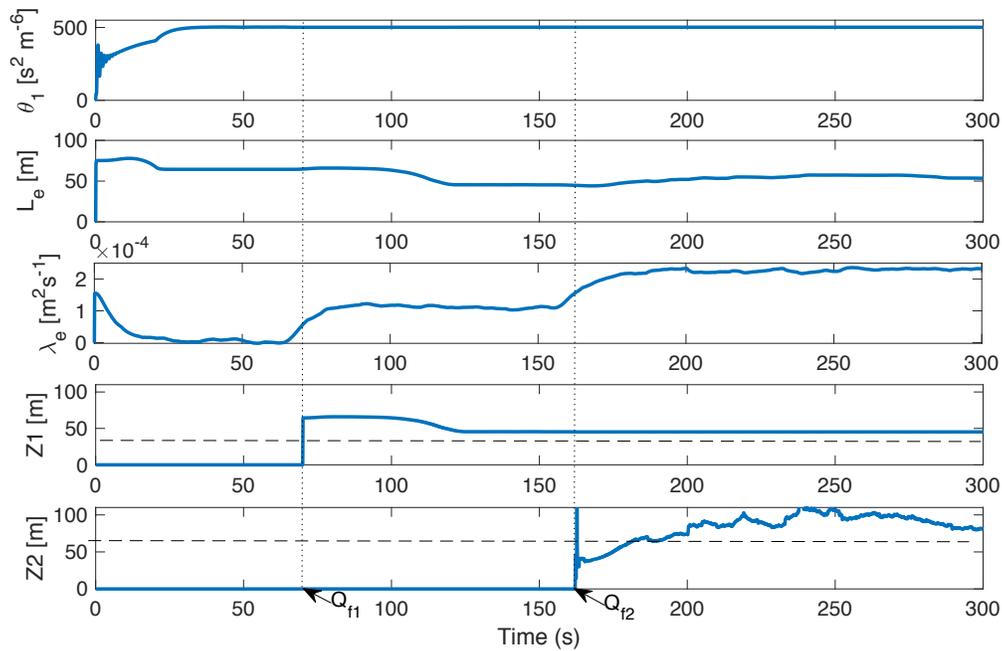


Fig. 7.15: Estimated parameters and leaks location. Arrows indicate the occurrence time of the leaks. Dashed lines are the real position of leaks

7.5 Conclusions

A remarkable finding is the success of the method for detection and localization of two consecutive leaks implemented in this SCADA-II, which operates in real-time. Even a Riccati equation is solved at each step.

Figure 7.15 shows the time evolution of the estimation plotted with the help of MATLAB from on-line data. The experiment consists of two sequential leaks caused with the valves V2 and V3. Upper graphs show the estimated parameters associated with the equivalent leak, and the lower two graphs show the physical position of the leaks.

Some characteristics are to remark of the hydraulic system. First of all is, even though the system is working in steady state: signals are not completely steady, they are changing around a mean value. This is because pumps, discontinuities inside the pipeline and continuous recycling of the leakage flow causes disturbances in the system.

Another remark is that the parameter estimation requires around 30 s to reach the final value. As a consequence, this time interval constrains how close in time the leaks can occur.

With respect to the performance of the novel adaptive threshold, diverse experiments have shown that the alarm signal is activated with an outflow of the leaks less than 1% with respect to the nominal flow rate. Figure 7.16 shows the front panel of the corresponding sub VI. One can see the behavior of the residual versus the adaptive threshold when a leak and a change in the operating point (OP) are present. The leak appears at 58 s and the change in OP at 115 s. The adaptability of the threshold, as a function of $Q(0)$, can be seen when the operating point is changed by the pump frequency from 60 Hz to 55 Hz, for a while residual goes down so the threshold.

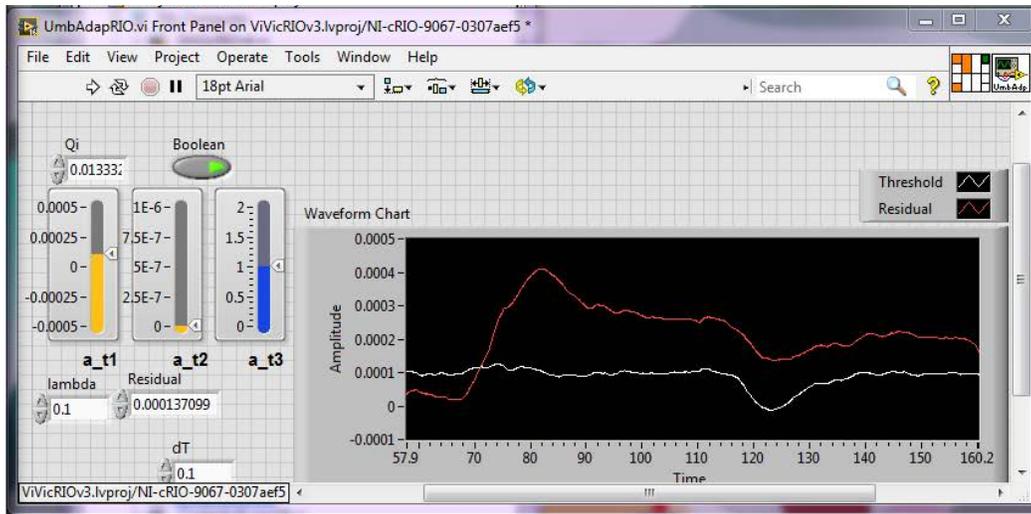


Fig. 7.16: Adaptive threshold front panel

As a conclusion, the goal of designing a SCADA at UNAM and its validation for sequential leaks detection and localization was fully achieved. Nevertheless, work remains to be done for future versions. The settling time of the estimators and location error must be reduced.

References

- Aamo, O. M and Salvesen, J and Foss, B. (2006). Observer Design Using Boundary Injection for Pipeline Monitoring and Leak Detection. In *ADCHEM 2006, IFAC*, pages 53–58.
- Bansal, R. (2005). *Fluid mechanics and hydraulic machines*. Laxmi Publications (P) LTD.
- Barmesa (2018). *Centrífugas mediana presión*. Barmes pumps.
- Besaçon, G. (2007). *Nonlinear Observers and Applications*, chapter Parameter/Fault Estimation in Nonlinear Systems and Adaptive Observers, pages 211–221. Springer.
- Billman, L. and Isermann, R. (1987). Leak detection methods for pipelines. *Automatica*, 23(3):381–385.
- Carrera, R. (2019). *PROTOTIPO PARA DETECCIÓN DE FALLAS EN TUBERÍAS: MANUAL DE USO*. Instituto de Ingeniería, UNAM.
- Carrera, R. and Verde, C. (2001). Localizador Automático de Fugas en un Ducto. *Ingeniería Hidráulica en México*, XVI:139–151.
- Chaudhry, M. C. (2014). *Applied hydraulic transients*. Springer: New York.
- E+H (2018a). *Cerabar s pmp71*. Endress + Hauser.
- E+H (2018b). *Proline promass 80F, 83F*. Endress + Hauser.
- Fang, W., Adegboye, M., and Karnik, A. (2019). Recent advances in pipeline monitoring and oil leakage detection technologies: Principles and approaches. *sensors*, 19(2548).
- GF (2018). *Type 130/133 electrically actuated ball valve*. Georg Fischer Piping Systems.
- Isermann, R. (2006). *Fault-Diagnosis Systems*. Springer, Berlin.
- Isermann, R. (2011). *Fault Diagnosis Applications*. Springer Verlag.
- Liu, C., Li, Y., and Xu, M. (2019). An integrated detection and location model for leakages in liquid pipelines. *Journal of Petroleum Science and Engineering*, 175:852–867.
- Mitsubishi (2018). *Inversores freqrol*. Mitsubishi Electric.
- Moubayed, A., Sharif, M., Luccini, M., Primak, S., and Shami, A. (2021). Water Leak Detection Survey: Challenges & Research Opportunities Using Data Fusion & Federated Learning. *IEEE Access*.
- Narendra, K. S. and Annaswamy, A. M. (1987). Persistent excitation in adaptive systems. *International Journal of Control*, 45(1):127–160.
- National Instruments (2021). ¿qué es labview? <https://www.ni.com/es-mx/shop/labview.html>.
- NI (2018). *Qué es compactRIO*. National Instruments.
- Panasonic (2013). *CF-53 Reference Manual*. Panasonic Corporation.
- RC (2018). *Remote control electric actuators*. Remote control.

- Reif, K., Sonnenmann, F., and Unbehauen, R. (1998). An EKF-Based Nonlinear Observer with a Prescribed Degree of Stability. *Automatica*, 34(9):1119–1998.
- Rojas, J. (March, 2021). *DIAGNÓSTICO DE FUGAS SECUENCIALES*. PhD thesis, Posgrado en Ingeniería UNAM.
- Rojas, J., Verde, C., and Torres, L. (June 2021). Estimation of hydraulic gradient for a transport pipeline. *ASME. J. Pressure Vessel Technol*, 143(3):031801.
- Sieber, H. and Isermann, R. (1977). Leckerkennung und lokalisierung bei pipelines durch on-line-korrelation mit einem prozessrechner. *Regelungstechnik*, 25:69–74.
- Verde, C., Bornard, G., and Gentil, S. (2003). Isolability of Multi-leaks in a Pipeline. In *Proceedings 4th MATHMOD, Vienna*, Vienna, Austria. ARGESIM Verlag, ISBN-3-901608-24-9.
- Verde, C. and Rojas, J. (2017). *Modeling and Monitoring of Pipelines and Networks*, chapter Recursive Scheme for Sequential Leaks' Identifications, page 134. Springer.
- Verde, C. and Torres, L. (2017). *Modeling and Monitoring of Pipelines and Networks*. Springer.
- Zhang, T., Tan, Y., Zhang, X., and Zhao, J. (2015). A novel hybrid technique for leak detection and location in straight pipelines. *Journal of Loss Prevention in the Process Industries*, 35:157–168.